Lab 3: Extended Kalman Filter for Localization

Josephine King, Peter Johnson

***Abstract*—Localization has been a central problem in the field of robotics. This report details the design and implementation of an Extended Kalman Filter (EKF) for the localization of a robot in a coordinate frame using a known landmark as reference. With a lidar and an inertial measurement unit as the sensors, the filter was able to achieve an overall path tracking Root Mean Square Error of 0.2 m.**

I. Introduction

LOCALIZATION is a fundamental problem in the field of autonomous mobile robotics. Mobile robot localization is the process of determining the pose of a robot relative to a given map of the environment. This report details the localization of a robot equipped with a lidar and an inertial measurement unit (IMU) navigating a 10 by 10 meter square path located in the center of an open field. At the center of the square is a pylon, the only object discernible by the lidar. This pylon serves as a landmark at a known location in the global coordinate frame (5.0 m, -5.0 m), allowing for Extended Kalman Filter (EKF) localization. The global frame defines east as the positive-x direction and north as the positive-y direction. The yaw angle is 0 rad along the x-axis, and the positive angle direction is counter clockwise (CCW). The IMU's coordinate frame is centered on the robot, with the y-axis pointing to the robot's left and the x-axis pointing forward. Similarly, the lidar's y-axis points forward on the robot and the x-axis is to the right. Figure 1 shows all three coordinate frames relative to the robot. The robot begins its square navigation at the origin of the global frame, which is the north west side of the square.

The robot's state is $\mathbf{x_t} = [\dot{x}_t, x_t, \dot{y}_t, y_t, \dot{\theta}_t, \theta_t, \theta_{pt}]$. All state variables are defined in the global frame. x_t and y_t are the location, $\dot{x}_t$ and $\dot{y}_t$ are the velocities, $\dot{\theta}_t$, θ_t, and θ_{pt} are the yaw velocity, yaw angle, and previous yaw angle, respectively.

II. Extended Kalman Filter Design

The Extended Kalman Filter is a generalization of the Kalman Filter to nonlinear state transition and measurement models. It uses Jacobian matrices of the nonlinear models to linearize around estimates of the mean and variance. See the Algorithm below.

Algorithm Extended Kalman Filter($\mu_{t-1}, \Sigma_{t-1}, z_t$)

$$\begin{aligned} \bar{\mu}_t &= g(\mu_{t-1}, u_t) \\ \bar{\Sigma}_t &= G_{x,t}\Sigma_{t-1}G_{x,t}^T + G_{u,t}R_tG_{u,t}^T \\ K_t &= \bar{\Sigma}H_t^T(H_t\bar{\Sigma}_tH_t^T + Q_t)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t(z_t - h(\bar{x}_t)) \\ \Sigma_t &= (I - K_tH_t)\bar{\Sigma}_t \end{aligned} \tag{1}$$

return μ_t, Σ_t

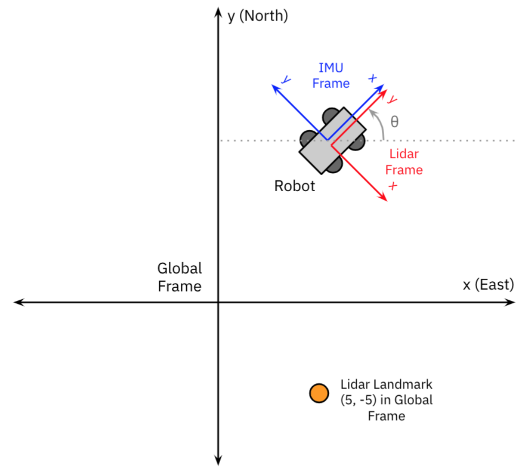

Fig. 1. Top-down view of robot with local frames, global frame, and landmark.

In this algorithm, μ_t is the state estimate at the current time step, $\bar{\mu}_t$ is the predicted estimate, Σ_t is the covariance matrix, $\bar{\Sigma}_t$ is the predicted covariance matrix. R_t is the covariance matrix associated with the control/odometry vector. Q_t is the covariance matrix associated with the measurement vector. $G_{x,t}$, $G_{u,t}$ and H_t are the Jacobians associated with the state transition and measurement models. K_t is the Kalman Gain.

For state transition function $x_t = g(x_{t-1}, u_t)$ (where x_{t-1} is the previous state, and u_t is the current control or odometry) and measurement model $z_t = h(x_t)$ (where z_t is the measurement vector), the Jacobian matrices are:

$$G_{x,t} = \begin{bmatrix} \frac{\partial g_{x1}}{\partial x_1} & \cdots & \frac{\partial g_{x1}}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{xn}}{\partial x_1} & \cdots & \frac{\partial g_{x1}}{\partial x_n} \end{bmatrix} \tag{2}$$

$$G_{u,t} = \begin{bmatrix} \frac{\partial g_{x1}}{\partial u_1} & \cdots & \frac{\partial g_{x1}}{\partial u_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{xn}}{\partial u_1} & \cdots & \frac{\partial g_{xn}}{\partial u_n} \end{bmatrix} \tag{3}$$

$$H_t = \begin{bmatrix} \frac{\partial h_{x1}}{\partial x_1} & \cdots & \frac{\partial h_{x1}}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{xn}}{\partial x_1} & \cdots & \frac{\partial h_{x1}}{\partial x_n} \end{bmatrix} \tag{4}$$

The control/odometry vector is defined to be $u_t = [a_x, a_y, u_\theta]$ where a_x and a_y are the accelerations measured in the IMU frame. u_θ is the yaw angle in the global frame.

The time between samples is dt. The state transition function is denoted to be

$$\begin{bmatrix} \dot{x}_t \\ x_t \\ \dot{y}_t \\ y_t \\ \dot{\theta}_t \\ \theta_t \\ \theta_{pt} \end{bmatrix} = \begin{bmatrix} \dot{x}_{t-1} + (a_x cos(yaw) - a_y sin(yaw))dt \\ x_{t-1} + \dot{x}_{t-1}dt \\ \dot{y}_{t-1} + (a_x sin(yaw) - a_y cos(yaw))dt \\ y_{t-1} + \dot{y}_{t-1}dt \\ (\theta_{t-1} - \theta_{pt-1})/dt \\ yaw \\ \theta_{t-1} \end{bmatrix} \tag{5}$$

The Jacobian with respect to state would be:

$$G_{x,t} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ dt & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & dt & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/dt & -1/dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

The Jacobian with respect to control would be:

$$G_{u,t} = \begin{bmatrix} dtcos(u_\theta) & -dtsin(u_\theta) & (-a_x sin(u_\theta) - a_y cos(u_\theta))dt \\ 0 & 0 & 0 \\ dtsin(u_\theta) & dtcos(u_\theta) & (a_x cos(u_\theta) - a_y sin(u_\theta))dt \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

The measurement vector is defined to be $z_t = [x_p, y_p]$ where x_p and y_p are the location of the pylon in the global coordinate frame. These are obtained by using the current prediction of θ_t to rotate the measurements in the lidar frame: x_{pL} and y_{pL} by the following clockwise rotation:

$$z_t = \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} x_{pL} * cos(\theta_t) + y_{pL} * sin(\theta_t) \\ -x_{pL} * sin(\theta_t) + y_{pL} * cos(\theta_t) \end{bmatrix} \tag{8}$$

The measurement model based on the globalized measurements and the global location of the landmark $X_L = 5$, $Y_L = -5$ and the current estimate of the robot position x_t and y_t is defined as follows:

$$z_t = \begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} X_L - x_t \\ Y_L - y_t \end{bmatrix} \tag{9}$$

The control/odometry covariance matrix is R_t and the measurement covariance is Q_t:

$$R_t = \begin{bmatrix} \sigma_{ax}^2 & 0 & 0 \\ 0 & \sigma_{ay}^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} Q_t = \begin{bmatrix} \sigma_{xpL}^2 & 0 \\ 0 & \sigma_{ypL}^2 \end{bmatrix} \tag{10}$$

These parameters were all determined experimentally by taking multiple measurements of a single true value and then taking the variance for each measurement.

III. Performance

The estimated path of the robot using the EKF matched closely to the expected path, as shown by Figure 2. Figure 3 shows the robot's angle over time, and it turns right at each corner, as expected. According to the estimated path and expected path, the GPS appears be off by about 1m, which makes sense given the accuracy of most GPS systems. The corners of the estimated path veer away from the expected path, which have been caused by the data collection method. The person collecting data may have struggled to turn the cart at a 90° angle, and thus backed up at an angle to make the forward angle less sharp. The estimated path given by the EKF is consistent with this turning method.

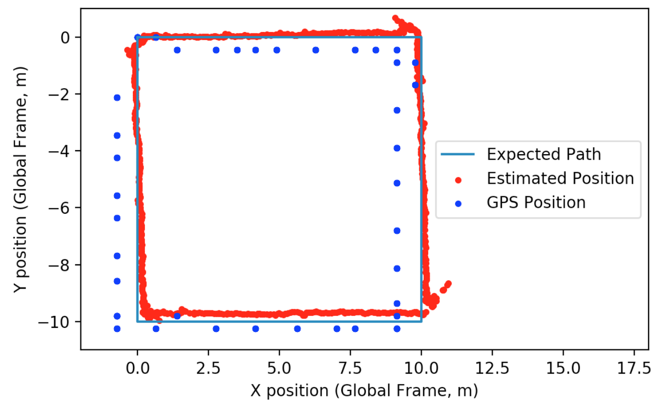

Fig. 2. Top-down view of the robot's expected path, GPS path, and estimated path. The path starts at the north west corner of the square and move clockwise around it.

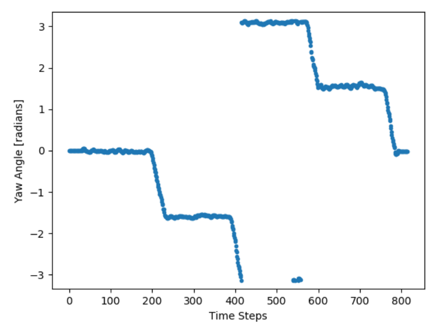

Fig. 3. Yaw angle of the robot over time.

The error of the estimated path to the expected path is shown in Figure 4, along with total RMSE and the RMSE over time. In the case that the estimated position is either in between the x-bounds of the square or the y-bounds of the square, error may be calculated according to the equation

$$error_t = min(|x_{est,t} - 10|, |x_{est,t}|, |y_{est,t} + 10|, |y_{est,t}|) \tag{11}$$

If the position is beyond these bounds, it may be computed as the distance to the nearest corner, using the distance formula. These equations were used to calculate the error at each time step, and the RMSE was calculated according to the equation

$$RMSE_t = \sqrt{\frac{1}{t+1}\sum_{i=0}^{t} error_i^2} \tag{12}$$

where t is an integer representing the time index. As shown by Figure 4, the error is largest at the northeast and southeast corners of the square (at 200 time steps and 400 time steps). These spikes in error align with Figure 2, which shows a large deviation from the expected path at the northeast and southeast corners. There are also spikes in error at the other two corners, but they are less significant. The overall RMSE is about 0.2 m for naviating around the entire square.

The estimated variances of the robot's estimated positions are shown in Figure 5. Because the position variances are very small (on the order of $10^{-5}m$), they are shown 100 times larger in this picture so that they can be visualized. Figures 6 and 7 show estimated variances of state variables plotted against time steps. The blue triangles show the estimated state variances when correction is performed on every prediction. The red circles show the estimates state variances when correction is performed on every second prediction. Thus, the variance grows at every other time step, representing increased uncertainty. When correction is performed, the variance decreases again to match the variance when corrections are performed on every prediction.

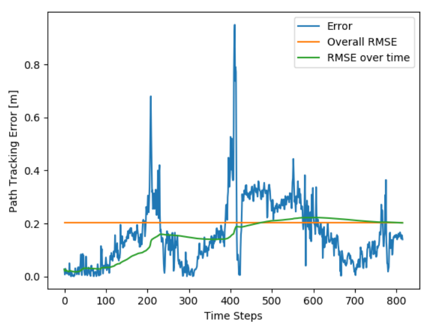

Fig. 4. Plot of the path tracking RMSE over time, as well as RMSE for each individual timestep

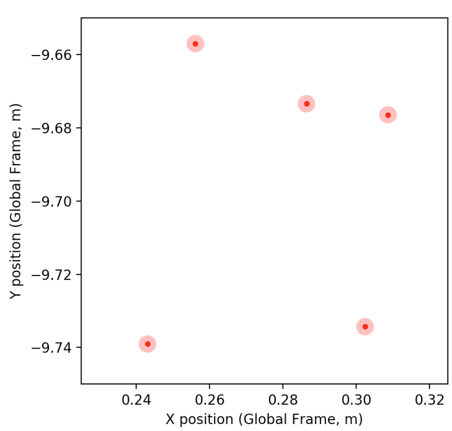

Fig. 5. Robot position estimates at the south west corner of the square, with covariance ellipses around each position estimate. The covariances were too small to visualize (on the order of 10^{-5} m), so they are shown here 100 times bigger than they actually are.

IV. Conclusion

An Extended Kalman Filter was successfully formulated and implemented to localize a robot using a known landmark. Based on the expected path, the RMSE of the estimated path was 0.2 m. The success of this implementation is reliant

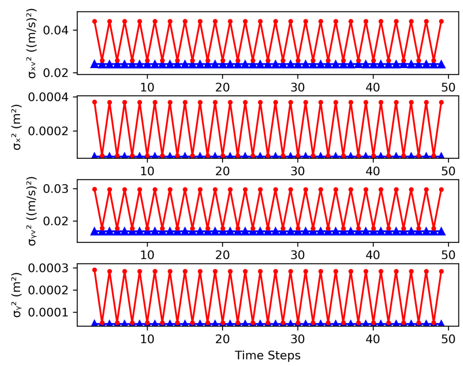

Fig. 6. Estimated variance with correction for every prediction (blue, triangles), plotted against estimated variance with correction for every other prediction (red, circles). Estimated variances for position and velocity are plotted.

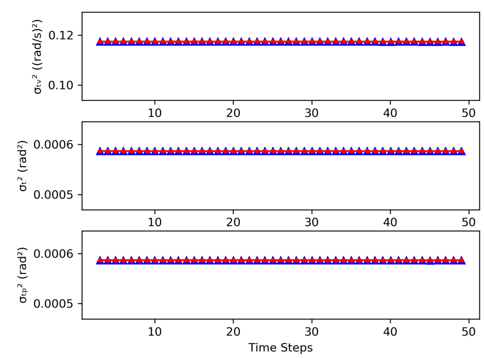

Fig. 7. Estimated variance with correction for every prediction (blue, triangles), plotted against estimated variance with correction for every other prediction (red, circles). Estimated variances for angular velocity, angle, and previous angle.

on the quality of the yaw data produced by the IMU. By using the yaw angle prediction ($\bar{\theta}_t$) to convert the lidar data into the global orientation before the measurement function $z_t = h(x_t)$, the measurement function and its Jacobian H_t were significantly simplified. However, because of this the uncertainty in yaw is not incorporated into the correction step. In fact, yaw is only present in the prediction step. By simply setting θ_t to be the yaw measurement at that time step, we are not filtering that data at all. If our yaw data was not as accurate, and had more variance then this method would not be possible. We would need to predict using $\dot{\theta}_{t-1}$ and then correct using the yaw measurement.