Project 2: Robot Localization

MAE 195 Introduction to Robot Motion Planning and Navigation

Alex Nguyen

Department of Mechanical and Aerospace Engineering
University of California Irvine
Instructor: Solmaz S. Kia
June 11, 2021

Contents

1 Introduction

Our problem considers the unicycle robot trajectory in Figure 1. Here, the robot has its motion restricted to a 2D space where there are 35 uniquely labeled landmarks in the environment. The robot will make relative range and bearing measurements from any landmark within a circular "measurement zone." The considered zones are of radius 5, 10, and 15 meters. Furthermore, we assume the robot has a data association and detection function to be able to uniquely identify each landmark with its respective measurements. The project objective is to conduct a beacon-based and SLAM-based [1], [2], [3] localization with range and bearing measurements.

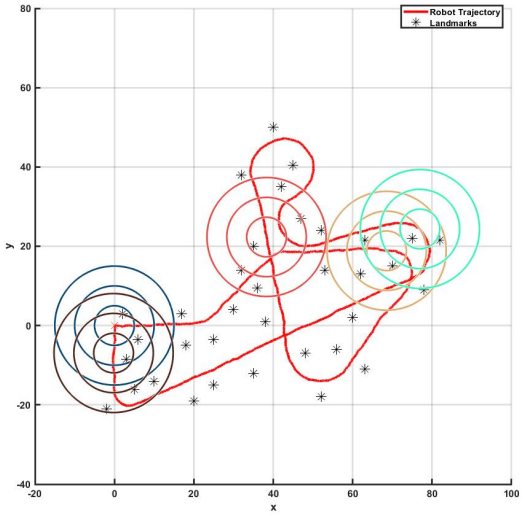

Figure 1: Unicycle robot trajectory.

In practice, we typically do not know the actual robot trajectory so a standard performance metric used to evaluate localization filters is loop-closure. In loop-closure, a robot starts from a known initial condition and returns back to it at the end of the experiment. The loop-closure error is defined as the difference between the known start location and the estimated final return point $k = N$ with $k = 0, 1, 2, \cdots, N$ (i.e., Loop Closure $= \|x(0) - \hat{x}(N)\|_2$).

2 Beacon-Based Localization

The beacon-based localization portion of the project focuses on robot state estimation *only*. The robot assumes to know the location of all landmarks, i.e., when the robot detects the landmark it knows both the unique landmark id and the associated location. The three different "measurement zone" cases (r = 5 m, 10 m, and 15 m) will be looked at in the beacon-based localization framework.

The system (i.e., dynamics) and measurement model for a robot moving on a flat 2D surface performing localization *only* are defined as follows:

$$
\begin{aligned}
x(k+1) &= \begin{bmatrix} x_r(k) + \delta_t u_1(k)\cos(\theta_r(k)) \\ y_r(k) + \delta_t u_1(k)\sin(\theta_r(k)) \\ \theta_r(k) + \delta_t u_2(k) \end{bmatrix} + w(k), \qquad \text{where } w(k) = \begin{bmatrix} w_{x_r}(k) \\ w_{y_r}(k) \\ w_{\theta_r}(k) \end{bmatrix} \\
z_k^i &= \begin{bmatrix} r_i \\ \phi_i \end{bmatrix} + v(k) \\
&= \begin{bmatrix} \sqrt{(x_r(k) - x_{b_i})^2 + (y_r(k) - y_{b_i})^2} \\ \text{atan}_2(y_r(k) - y_{b_i}, x_r(k) - x_{b_i}) - \theta_r(k) + v(k) \end{bmatrix} \quad (x_{b_i}, y_{b_i}), \;\; i \in \{1, ..., 35\}
\end{aligned}
$$

The linearized state transition and observation Jacobian matrices for a robot moving on a flat 2D surface performing localization *only* are defined as follows:

$$
\begin{aligned}
F_k &= \begin{bmatrix} 1 & 0 & -\sin(\hat{\theta}_r^+(k))u_1(k)\delta_r \\ 0 & 1 & \cos(\hat{\theta}_r^+(k))u_1(k)\delta_r \\ 0 & 0 & -1 \end{bmatrix} \\
H_{k+1}^i &= \begin{bmatrix} \frac{\hat{x}_r^-(k+1) - x_{b_i}}{\sqrt{(x_r(k)-x_{b_i})^2 + (y_r(k)-y_{b_i})^2}} & \frac{\hat{y}_r^-(k+1) - y_{b_i}}{\sqrt{(x_r(k)-x_{b_i})^2 + (y_r(k)-y_{b_i})^2}} & 0 \\ \frac{-(\hat{y}_r^-(k+1) - y_{b_i})}{\left[1 + \left(\frac{\hat{y}_r^-(k+1) - y_{b_i}}{\hat{x}_r^-(k+1) - x_{b_i}}\right)^2\right](\hat{x}_r^-(k+1) - x_{b_i})^2} & \frac{1}{\left[1 + \left(\frac{\hat{y}_r^-(k+1) - y_{b_i}}{\hat{x}_r^-(k+1) - x_{b_i}}\right)^2\right](\hat{x}_r^-(k+1) - x_{b_i})^2} & -1 \end{bmatrix}
\end{aligned}
$$

The process noise for the system model (i.e., $w(k)$) was provided in as a data file. Even so, the process noise covariance for the robot and the measurement noise covariance for each landmark are defined as follows:

$$Q_{\text{Robot}} = \begin{bmatrix} (0.4\delta_t)^2 & 0 & 0 \\ 0 & (0.4\delta_t)^2 & 0 \\ 0 & 0 & (0.05\delta_t)^2 \end{bmatrix} \quad R_i = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (\frac{3\pi}{180})^2 \end{bmatrix}, \quad i \in \{1, \cdots, 35\}$$

The Extended Kalman Filter (EKF) for beacon-based localization was developed based on these set parameters. For beacon-based localization, the filter is straightforward since only 3 states are being estimated $\hat{x} = [x_r, y_r, \theta_r]^\top$. The EKF, also, will search for the subset of landmarks within the specified "measurement zones" when constructing the measurement vector (i.e., ranges and bearing angles) and linearized measurement matrix at each time step.

The estimated trajectories for the different "measurement zone" radii $r = 5$ m, 10 m, and 15 m are shown in Figure 2. As expected, the $r = 5$ m case gave the worse estimate trajectory while the $r = 15$ m case gave the best estimate trajectory. Most likely, this is due to the $r = 5$ "measurement zone" containing time periods with no measurements transmitting to the robot. This means the EKF does not have a correction (or update) step since there are no measurements to provide feedback (e.g., innovation) for our propagation (or prediction) step. The $r = 10$ m and $r = 15$ m cases always has at least one measurement within the "measurement zone" at all times which leads to better performance.

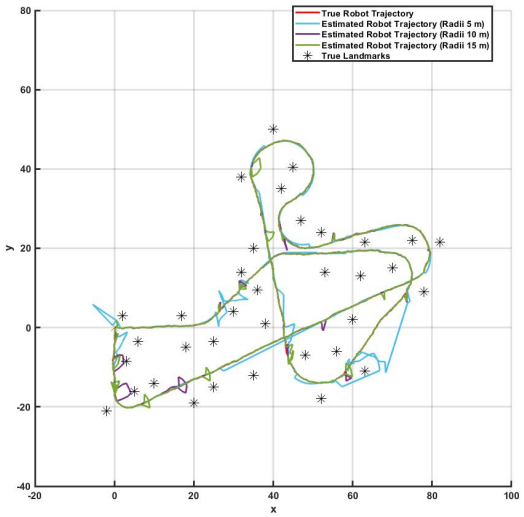

Figure 2: Beacon-based localization estimated robot trajectories for $r \in \{5, 10, 15\}$ m.

The time at which each uniquely identified landmark is detected for each "measurement zone" radii is shown in Figure 3. Notice in the $r = 5$ m case, there are several occasions where no measurements are transmitted to the robot (e.g., $\sim 45 - 55$ seconds). This led our filter to drift at several time intervals during simulation.

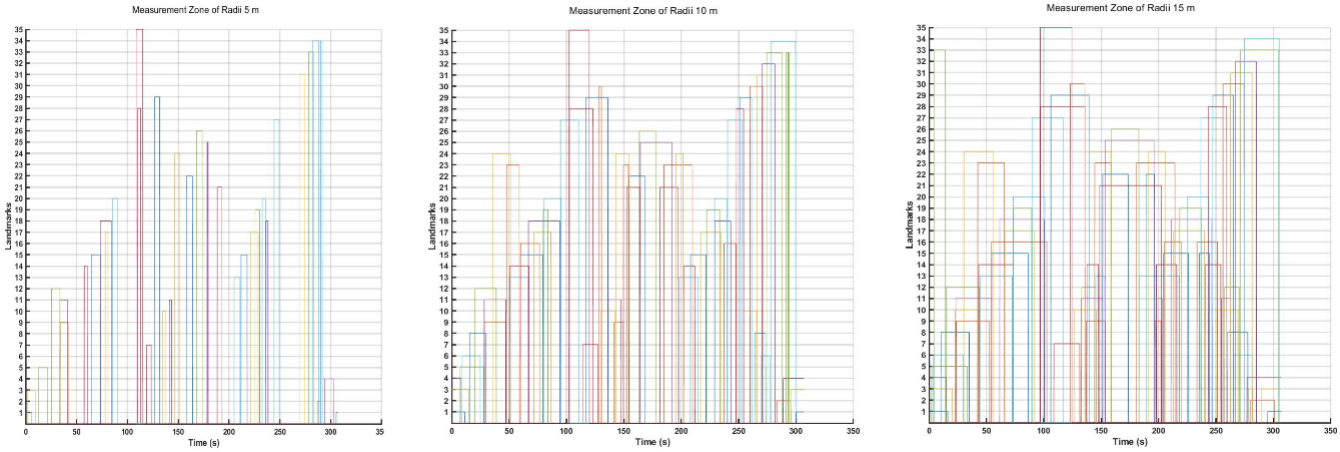

Figure 3: Beacon-based localization landmark detection times.

The loop-closure error for each case study is summarized below in table 1. As we expect, the loop-closure gets better with a larger "measurement zone" radius. Although, the loop-closure for $r = 15$ m is only slightly better than for $r = 10$ m.

Table 1: Beacon-based localization loop-closure.

Zone Radii (m)	**Loop-Closure [x, y] Error (m)**	**Loop-Closure Error (m)**
5	[-0.1750, 0.1033]	0.2032
10	[-0.1393, 0.0681]	0.1550
15	[-0.1393, 0.0680]	0.1550

3 SLAM Localization

The SLAM localization portion of the project focuses on a robot estimating its own states *and* the unknown landmark positions. The robot does not know the location of the landmarks, so when it detects a landmark it only distinguishes its unique label. The same study

will be repeated from above, three different "measurement zone" cases ($r = 5$ m, 10 m, and 15 m) will be looked at in the SLAM localization framework.

The system (i.e., dynamics) and measurement model for a robot moving on a flat 2D surface performing localization *only* are defined as follows:

$$x(k+1) = \begin{bmatrix} x_r(k) + \delta_t u_1(k)\cos(\theta_r(k)) \\ y_r(k) + \delta_t u_1(k)\sin(\theta_r(k)) \\ \theta_r(k) + \delta_t u_2(k) \\ x_{l_1}(k) \\ y_{l_1}(k) \\ \vdots \\ x_{l_{35}}(k) \\ y_{l_{35}}(k) \end{bmatrix} + w(k), \quad \text{where } w(k) = \begin{bmatrix} w_{x_r}(k) \\ w_{y_r}(k) \\ w_{\theta_r}(k) \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{aligned} z_k^i &= \begin{bmatrix} r_i \\ \phi_i \end{bmatrix} + v(k) \\ &= \begin{bmatrix} \sqrt{(x_r(k) - x_{b_i})^2 + (y_r(k) - y_{b_i})^2} \\ \text{atan}_2(y_r(k) - y_{b_i}, x_r(k) - x_{b_i}) - \theta_r(k) + v(k) \end{bmatrix} \quad (x_{b_i}, y_{b_i}), \;\; i \in \{1, ..., 35\} \end{aligned}$$

The linearized state transition and observation Jacobian matrices for a robot moving on a flat 2D surface performing localization *only* are defined as follows:

$$F_{Robot} = \begin{bmatrix} 1 & 0 & -\sin(\hat{\theta}_r^+(k))u_1(k)\delta_r \\ 0 & 1 & \cos(\hat{\theta}_r^+(k))u_1(k)\delta_r \\ 0 & 0 & -1 \end{bmatrix}, \quad F_k = \begin{bmatrix} F_{Robot} & 0 & \cdots & 0 \\ 0 & I_{2\times 2} & & \vdots \\ 0 & & \ddots & 0 \\ 0 & \cdots & & I_{2\times 2} \end{bmatrix}$$

$$H_{r,k+1}^i = \begin{bmatrix} \frac{\hat{x}_r^-(k+1) - x_{b_i}}{\sqrt{(x_r(k)-x_{b_i})^2 + (y_r(k)-y_{b_i})^2}} & \frac{\hat{y}_r^-(k+1) - y_{b_i}}{\sqrt{(x_r(k)-x_{b_i})^2 + (y_r(k)-y_{b_i})^2}} & 0 \\ \frac{-(\hat{y}_r^-(k+1) - y_{b_i})}{\left[1 + \left(\frac{\hat{y}_r^-(k+1) - y_{b_i}}{\hat{x}_r^-(k+1) - x_{b_i}}\right)^2\right](\hat{x}_r^-(k+1) - x_{b_i})^2} & \frac{1}{\left[1 + \left(\frac{\hat{y}_r^-(k+1) - y_{b_i}}{\hat{x}_r^-(k+1) - x_{b_i}}\right)^2\right](\hat{x}_r^-(k+1) - x_{b_i})^2} & -1 \end{bmatrix}$$

$$H_{l,k+1}^i = \begin{bmatrix} \frac{-(\hat{x}_r^-(k+1) - x_{b_i})}{\sqrt{(x_r(k)-x_{b_i})^2 + (y_r(k)-y_{b_i})^2}} & \frac{-(\hat{y}_r^-(k+1) - y_{b_i})}{\sqrt{(x_r(k)-x_{b_i})^2 + (y_r(k)-y_{b_i})^2}} \\ \frac{\hat{y}_r^-(k+1) - y_{b_i}}{\left[1 + \left(\frac{\hat{y}_r^-(k+1) - y_{b_i}}{\hat{x}_r^-(k+1) - x_{b_i}}\right)^2\right](\hat{x}_r^-(k+1) - x_{b_i})^2} & \frac{-1}{\left[1 + \left(\frac{\hat{y}_r^-(k+1) - y_{b_i}}{\hat{x}_r^-(k+1) - x_{b_i}}\right)^2\right](\hat{x}_r^-(k+1) - x_{b_i})^2} \end{bmatrix}$$

The process and measurement noise are similar to before, but now we include process noise for each landmark. But since each landmark is stationary, its process noise is zero ($Q_{l_i} = 0_{2\times 2}$). Hence, the process noise covariance for our states and the measurement noise covariance for each landmark are defined as follows:

$$Q_{\text{Robot}} = \begin{bmatrix} (0.4\delta_t)^2 & 0 & 0 \\ 0 & (0.4\delta_t)^2 & 0 \\ 0 & 0 & (0.05\delta_t)^2 \end{bmatrix}, \quad Q_k = \begin{bmatrix} Q_{pv} & 0 & \cdots & 0 \\ 0 & 0_{2\times 2} & & \vdots \\ 0 & & \ddots & 0 \\ 0 & \cdots & & 0_{2\times 2} \end{bmatrix}$$

$$R_i = \begin{bmatrix} (0.1)^2 & 0 \\ 0 & (\frac{3\pi}{180})^2 \end{bmatrix}, \quad i \in \{1, \cdots, 35\}$$

As above, the estimated trajectories for the different "measurement zone" radii are shown in Figure 4. Like for beacon-based localization, the estimated robot trajectory gets better with $r > 5$ m.

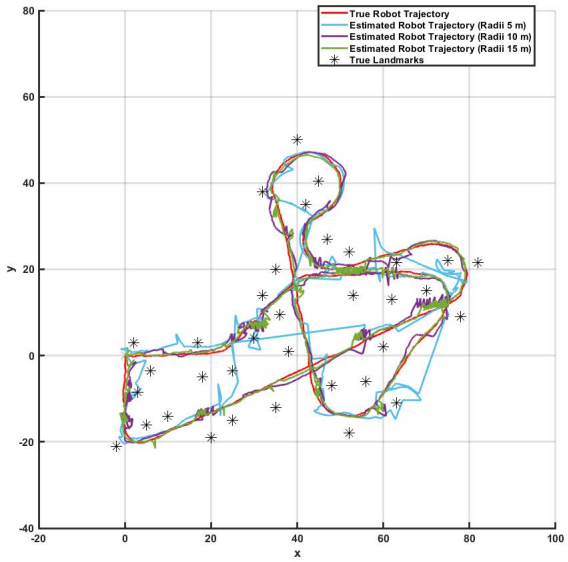

Figure 4: SLAM localization estimated robot trajectories for $r \in \{5, 10, 15\}$ m.

The time at which each uniquely identified landmark is detected for each "measurement

zone" radius is shown in Figure 5. These landmark detection plots should be the same as in the beacon-based localization case since the true robot trajectory and true landmark positions were used for detection in both cases. Notice again, there are several instances where measurements are not transmitted to the robot (e.g., $\sim 90 - 105$ seconds).

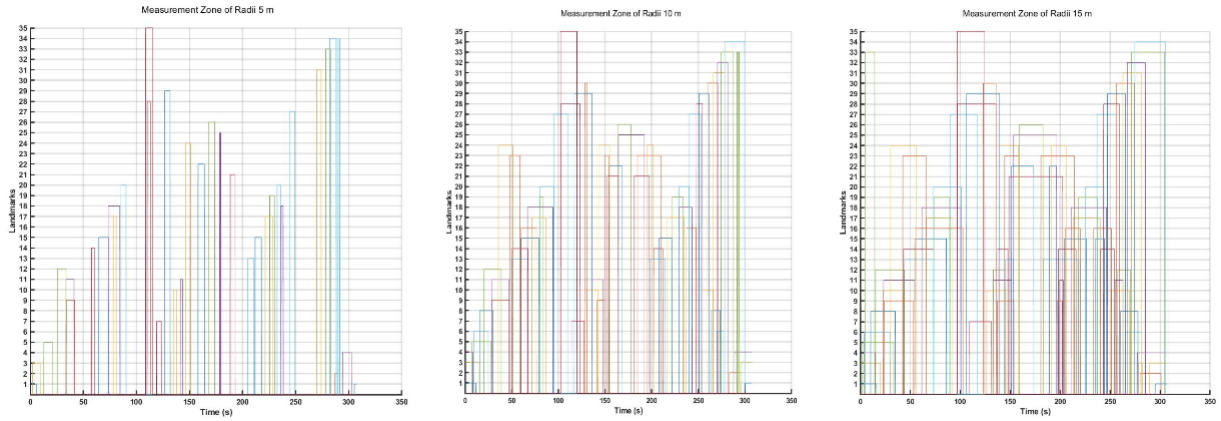

Figure 5: SLAM localization landmark detection times.

The loop-closure error for each case study is summarized below in table 2. We see that the loop-closure gets significantly better when expanding the "measurement zone" from r = 5 m to r = 10 m. Although, notice the loop-closure for r = 10 m has a slightly better performance than for r 15 m. This could arise due to more landmarks needing to be estimated with a greater "measurement zone" which degrades our filter's performance. Also it is worth noting, the loop closure performance in SLAM is quite worse compared to our loop-closure performance in beacon-based localization. This performance degradation is expected as the worse performance is correlated with a higher number of estimated states.

Table 2: SLAM localization loop-closure.

Zone Radii (m)	**Loop-Closure [x, y] Error (m)**	**Loop-Closure Error (m)**
5	[-0.2373, -0.7577]	0.7940
10	[-0.3948, -0.0937]	0.4058
15	[-0.3886, -0.0843]	0.3976

4 Performance Analysis Discussion

This system was a simulation of a unicycle robot in a known (beacon-based localization) and unknown (SLAM localization) environment. There are multiple ways to compare the estimation performance of an EKF in both a single-run and Monte Carlo (MC) setting.

For single-run EKF, it is nice to compare "apples to apples" rather than "apples to oranges." Thus, a system with a fixed random seed for both the measurement and process noise allows us to better compare filter performance. This seed setting could be defined as "controlled randomness" which allows for a better comparison of our filter's performance. Several good metrics for our single-run EKF case are the estimation error trajectories with $\pm 3\sigma$ bounds, normalized estimation error squared (NEES), and normalized innovation squared (NIS). The estimation error plots provide a good metric for determining whether our estimates are biased and how confident we are in the estimates. The NEES and NIS metrics determine whether the estimates and measurements yield a consistent filter. The NEES or NIS will use a chi-squared test to determine if the filter is ill-matched in the dynamics/measurements or in the process/measurement noises.

It would also be good to perform a Monte Carlo (MC) analysis to determine the average performance of the filter over many trials. The root-mean square error (RMSE) metric is a classic example of a test for evaluating the performance of a filter with M independent error MC trials. A point cloud with the absolute error superimposed on the RMSE for the filter will provide good insight into estimation performance.

Note: See provided MATLAB codes for the simulated performance results.

5 Conclusion

This project enabled students to understand the performance of a probabilistic robot from a beacon-based and SLAM-based localization point-of-view. The ability to estimate the internal states (e.g., position and velocity) of a vehicle, or robot, is due to the EKF. If a linearized system is approximately equal to the actual system, then an EKF will provide a sufficient approximation of the true system's states. The EKF is an extremely useful and powerful tool, but there are alternative nonlinear estimators which perform better at the expense of computation effort (e.g., Particle filter, unscented Kalman filter, Gaussian sum filter, ect.). The project could be repeated using other nonlinear estimation filters to compare their performance to the EKF in the future.

References

[1] J. Guivant and E. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[3] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): part ii," *IEEE Robotics Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.