State Estimation for Robotics
Assignment 3

Jiaming Li —— 0501271

December 26, 2023

1 Variances of the Noise

1. Base on the data, is the assumption of zero-mean Gaussian noise reasonable? What values of the variances Q_k and R_k^l, should we use?

Solution:
if we think that the process noise $\mathbf{w}_k$ can be explicitly written as $\mathbf{w}_k = [w_{v_k}, w_{w_k}]^T$, where w_{v_k} and w_{w_k} are independent noises. The same process can be applied to $\mathbf{n}_k^l$, so we have $n_k^l = [n_{r_k^l}, n_{\phi_k^l}]^T$.
Since the means of $w_{v_k}, w_{w_k}, n_{r_k^l}, n_{\phi_k^l}$ are all very close to 0, we can approximately take their distributions as zero-mean Gaussian distributions. According to the variance given in dataset.mat, we have:

$$\mathbf{w}_k \sim \mathcal{N}(0, Q_k) \tag{1}$$

$$\mathbf{n}_k^l \sim \mathcal{N}(0, R_k^l) \tag{2}$$

where

$$\mathbf{Q}_k = \begin{bmatrix} 0.044 & 0 \\ 0 & 0.0082 \end{bmatrix}_{2\times 2}, \tag{3}$$

$$\mathbf{R}_k^l = \begin{bmatrix} 9.3006 \times 10^{-4} & 0 \\ 0 & 6.7143 \times 10^{-4} \end{bmatrix}_{2\times 2} \tag{4}$$

2 Jacobians Calculation

Write out expressions for all the Jacobians(of the motion and observation models) taht are required in an EKF.

Solution:

(a) Expression for $\mathbf{F}_{k-1}$:

$$\mathbf{h} = \begin{cases} x_k = x_{k-1} + Tcos\theta_{k-1}v_k + Tcos\theta_{k-1}n_{v_k} \\ y_k = x_{k-1} + Tsin\theta_{k-1}v_k + Tsin\theta_{k-1}n_{v_k} \\ \theta_k = \theta_{k-1} + Tw_k + Tn_{w_k} \end{cases} \tag{5}$$

Therefore, we have:

$$\mathbf{F}_{k-1} = \begin{bmatrix} \frac{\partial h_1}{\partial x_{k-1}} & \frac{\partial h_1}{\partial y_{k-1}} & \frac{\partial h_1}{\partial \theta_{k-1}} \\ \\ \frac{\partial h_2}{\partial x_{k-1}} & \frac{\partial h_2}{\partial y_{k-1}} & \frac{\partial h_2}{\partial \theta_{k-1}} \\ \\ \frac{\partial h_3}{\partial x_{k-1}} & \frac{\partial h_3}{\partial x_{k-1}} & \frac{\partial h_3}{\partial \theta_{k-1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T(v_k + n_{v_k})sin(\theta_{k-1}) \\ \\ 0 & 1 & T(v_k + n_{v_k})cos(\theta_{k-1}) \\ \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

(b) Expression for $\mathbf{G}_k$:

$$\mathbf{g} = \begin{cases} \sqrt{(x_l - x_k - dcos\theta_k)^2 + (y_l - y_k - dsin\theta_k)^2} \\ atan2(y_l - y_k - dsin\theta_k,\ x_l - x_k - dcos\theta_k) - \theta_k \end{cases} \tag{7}$$

let $X = x_l - x_k - dcos\theta_k$, $Y = y_l - y_k - dsin\theta_k$.
Therefore we have:

$$\mathbf{G}_k = \begin{bmatrix} \frac{\partial g_1}{\partial x_k} & \frac{\partial g_1}{\partial y_k} & \frac{\partial g_1}{\partial \theta_k} \\ \\ \frac{\partial g_2}{\partial x_k} & \frac{\partial g_2}{\partial y_k} & \frac{\partial g_2}{\partial \theta_k} \end{bmatrix} = \begin{bmatrix} -\frac{X}{g_1(x_k)} & -\frac{Y}{g_1(x_k)} & \frac{dXsin\theta_k - dYcos\theta_k}{g_1(x_k)} \\ \\ \frac{Y}{g_1^2(x_k)} & -\frac{X}{g_1^2(x_k)} & -\frac{dXcos\theta_k + dYsin\theta_k}{g_1^2(x_k)} \end{bmatrix} \tag{8}$$

(c) Expression for $\mathbf{Q}'_k$:

$$\mathbf{J}_{Q_k} = \begin{bmatrix} \frac{\partial h_1}{\partial n_{v_k}} & \frac{\partial h_1}{\partial n_{w_k}} \\ \\ \frac{\partial h_2}{\partial n_{v_k}} & \frac{\partial h_2}{\partial n_{w_k}} \\ \\ \frac{\partial h_3}{\partial n_{v_k}} & \frac{\partial h_3}{\partial n_{w_k}} \end{bmatrix} = \begin{bmatrix} Tcos\theta_{k-1} & 0 \\ \\ Tsin\theta_{k-1} & 0 \\ \\ 0 & T \end{bmatrix} \tag{9}$$

Therefore we can write $\mathbf{Q}'_k$ as

$$\mathbf{Q}'_k = \mathbf{J}_{Q_k}\mathbf{Q}_k\mathbf{J}_{Q_k}^T \tag{10}$$

(d) Expression for $\mathbf{R}'_k$:

$$\mathbf{J}_{R_k} = \begin{bmatrix} \frac{\partial g_1}{\partial n_{r_k}} & \frac{\partial g_1}{\partial n_{\phi_k}} \\ \\ \frac{\partial g_2}{\partial n_{r_k}} & \frac{\partial g_2}{\partial n_{\phi_k}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \\ 0 & 1 \end{bmatrix} \tag{11}$$

Therefore we can write $\mathbf{R}'_k$ as

$$\mathbf{R}'_k = \mathbf{J}_{R_k} \mathbf{R}_k \mathbf{J}_{R_k}^T = \mathbf{R}_k \tag{12}$$

3 Modify and Rewrite Equations

Modify and write the equations of the extended kalman filter, modified to handle a variable number of exteroceptive measurements at each timestep.

Solution: Since we have already know $F_{k-1}, G_k, Q_k^{'}, R_k^{'}$, so we put the typical equations of EKF:

$$\textbf{Prediction} = \begin{cases} \check{P}_k = F_{k-1}\hat{P}_{k-1}F_{k-1}^T + Q_k^{'} \\ \check{x}_k = f(\hat{x}_{k-1}, v_k, 0) \end{cases} \tag{13}$$

$$\textbf{Correction} = \begin{cases} K_k = \check{P}_k G_k^T (G_k \check{P}_k G_k^T + R_k^{'})^{-1} \\ \hat{P}_k = (\mathbf{1} - K_k G_k)\check{P}_k \\ \hat{x}_k = \check{x}_k + K_k(y_k - g(\check{x}_k, 0)) \end{cases} \tag{14}$$

As we can see in the assignment, the rangefinder can not always see all landmarks at one time, so we need to skip the invisible landmarks and only consider the landmarks within a manually set range limit. Here we stack all the $\mathbf{G_k^l}$ as $\mathbf{G_k}$, a 34×3 matrix. Correspondingly we also stack all the Kalman gains $\mathbf{K_k^l}$ as $\mathbf{K_k}$, a 3×34 matrix. Lastly, we also stack all the measurements $\mathbf{y_k^l}$ as $\mathbf{y_k}$.

When a landmark is invisible at one step, we should set the corresponding Kalman gain to 0. When we try to estimate the next step using the measurements, the invisible landmarks won't be taken into account.

$$\mathbf{G_k} = \begin{bmatrix} G_k^1 \\ \\ G_k^2 \\ \\ \vdots \\ \\ G_k^{17} \end{bmatrix} \tag{15}$$

$$\mathbf{K_k} = \begin{bmatrix} K_k^1 & K_k^2 & \dots & K_k^{17} \end{bmatrix} \tag{16}$$

4 Results

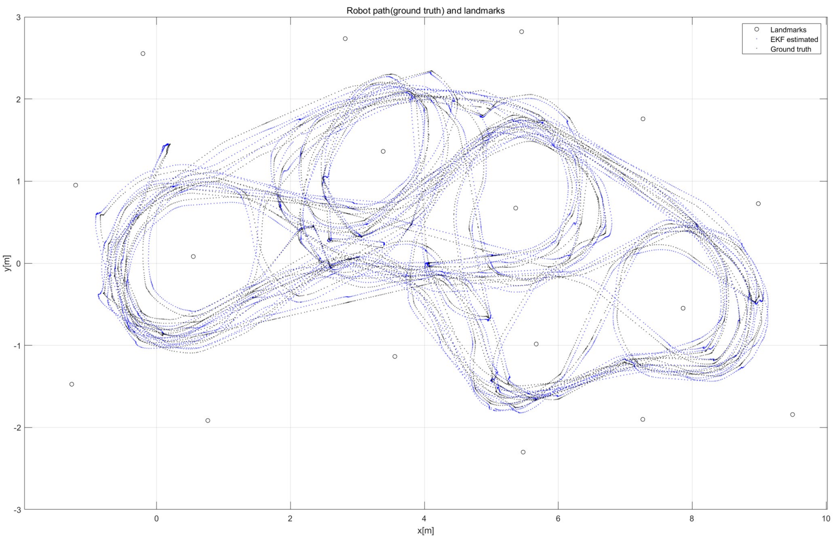
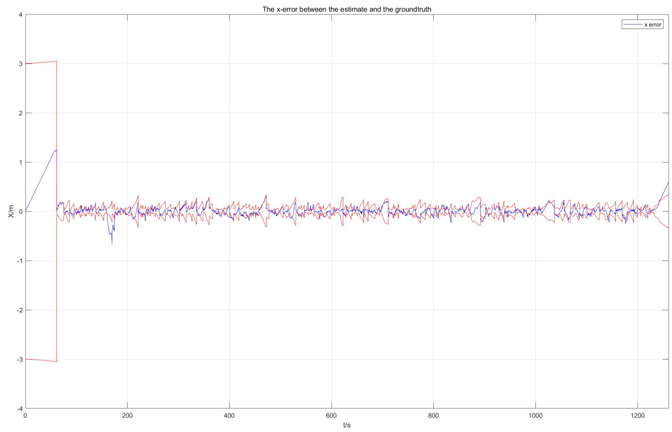
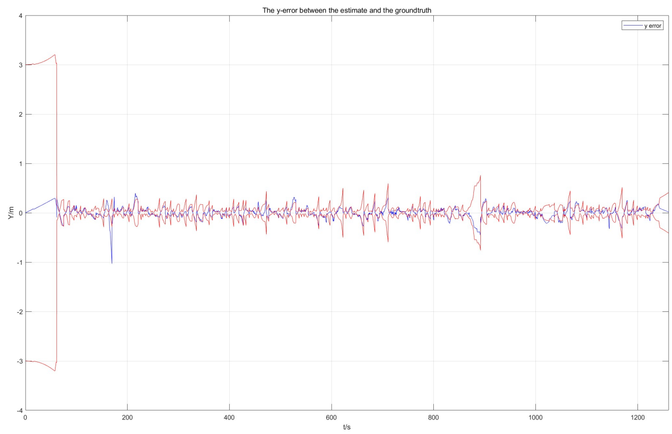


Figure 1: Robot's path: ground truth VS. estimated curve with an Extended Kalman Filter(EKF). The 17 landmarks of the map are also shown using black circles ($r_{max} = 5m$)

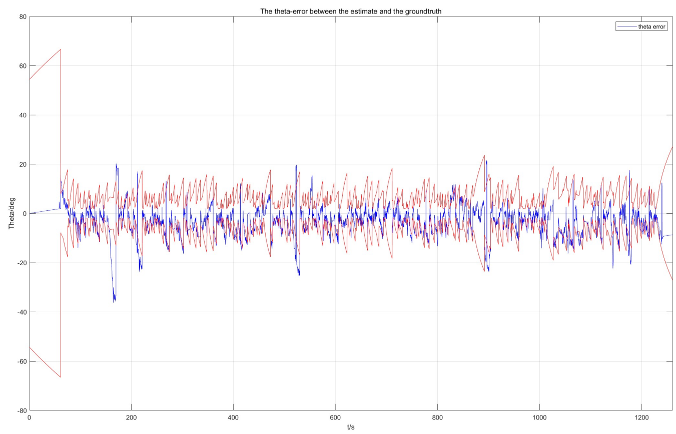
From Fig. 1 we find that the estimated curve still has visible error comparing to the ground truth, which is the explicit disadvantage of Extended Kalman Filter with only one iteration.
Then we use different range limits to run the EKF in order to find the differences between them. The errors are shown in the following figures.



(a) Error in x

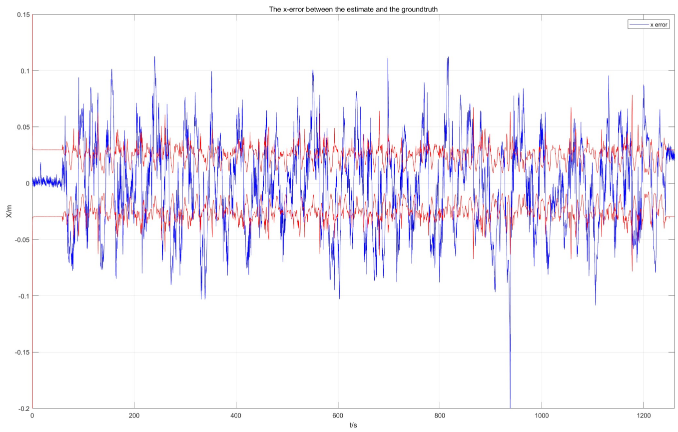


(b) Error in y

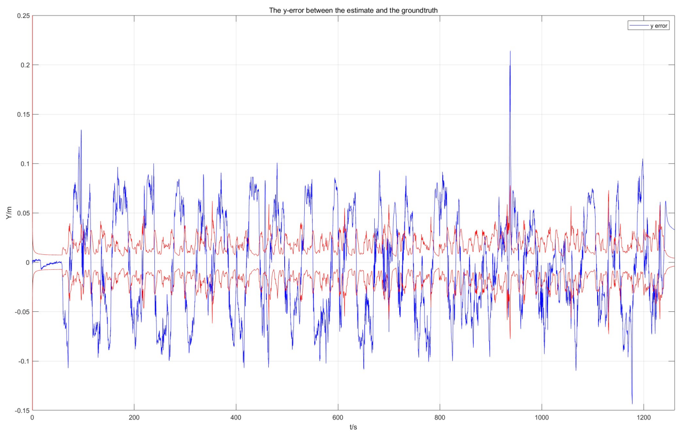


(c) Error in θ

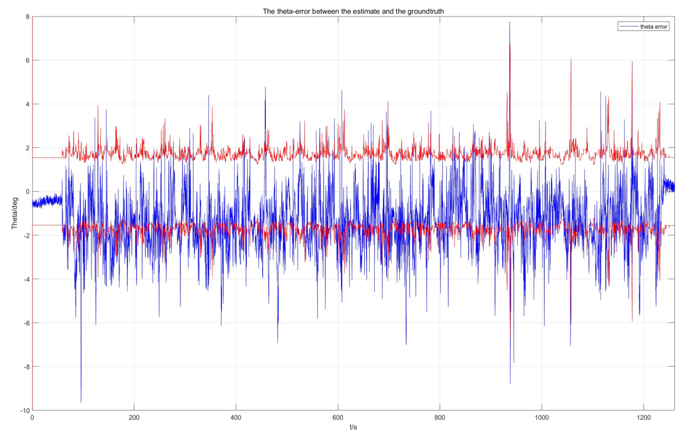
Figure 2: Part 4.a using only landmarks closer than $r_{max} = 1m$



(a) Error in x

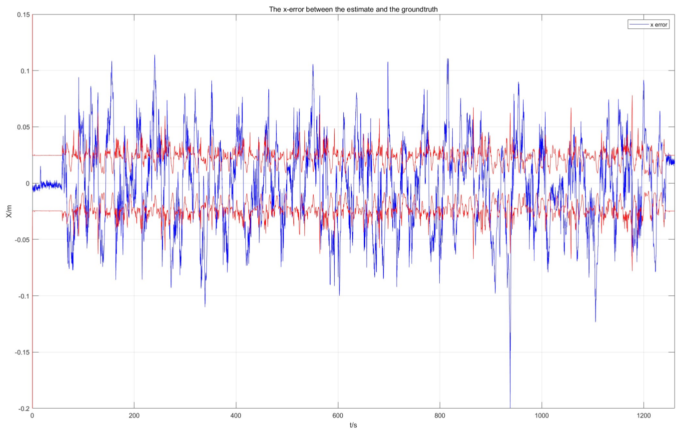


(b) Error in y

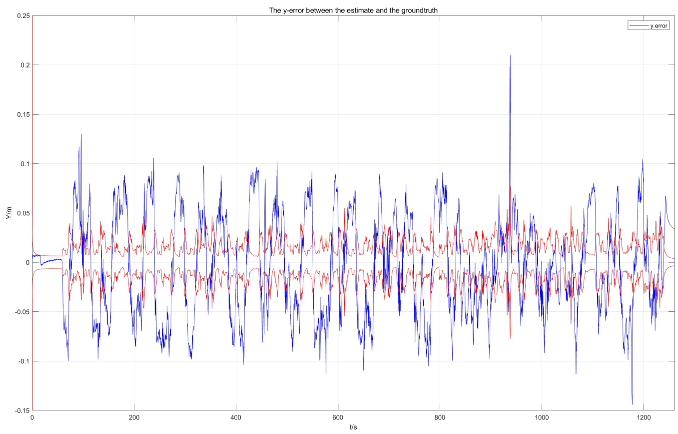


(c) Error in θ

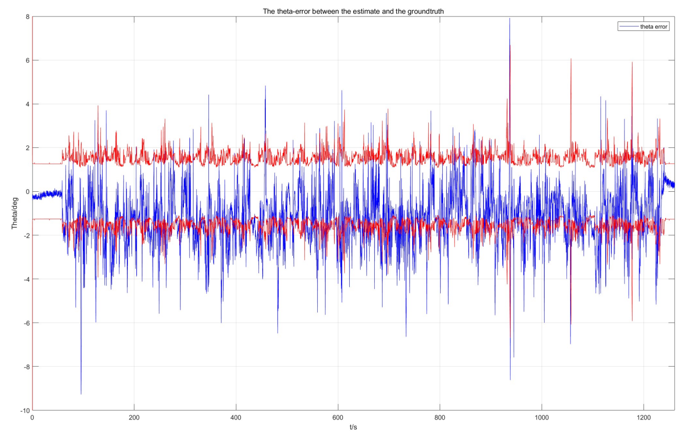
Figure 3: Part 4.a using only landmarks closer than $r_{max} = 3m$



(a) Error in x



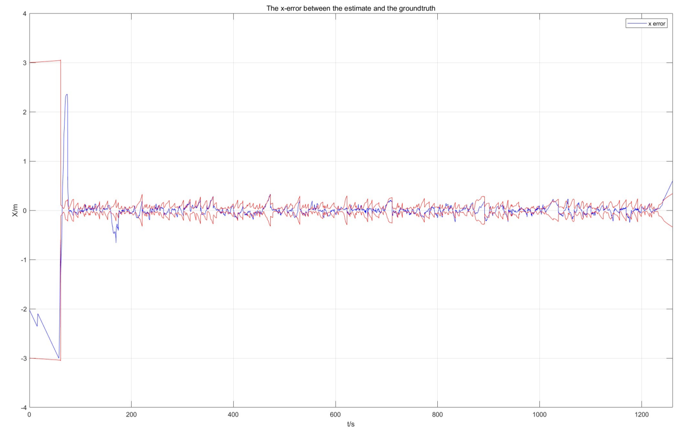
(b) Error in y



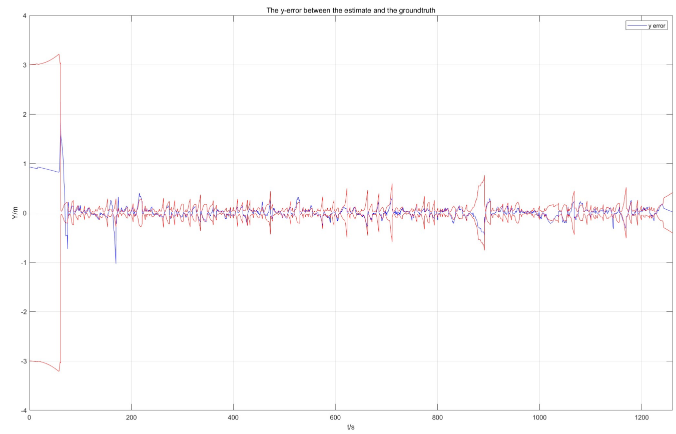
(c) Error in θ

Figure 4: Part 4.a using only landmarks closer than $r_{max} = 5m$

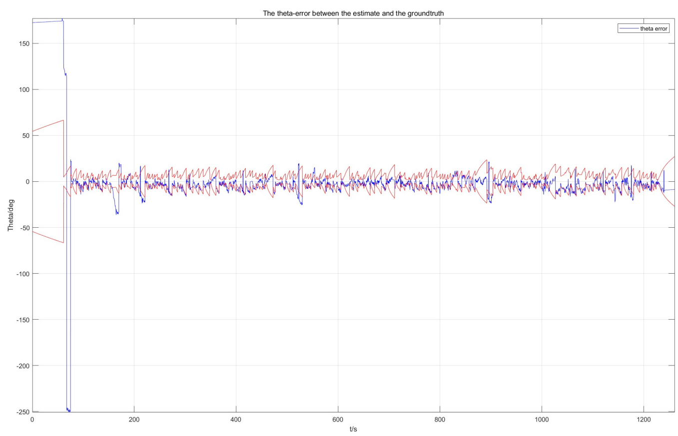
Here we can see that the errors don't stay in the uncertainty envelop, especially when we use bigger user defined range limit. This implies that EKF doesn't converge to the mode of the posterior, and the linearization of the nonlinear model introduces errors.



(a) Error in x

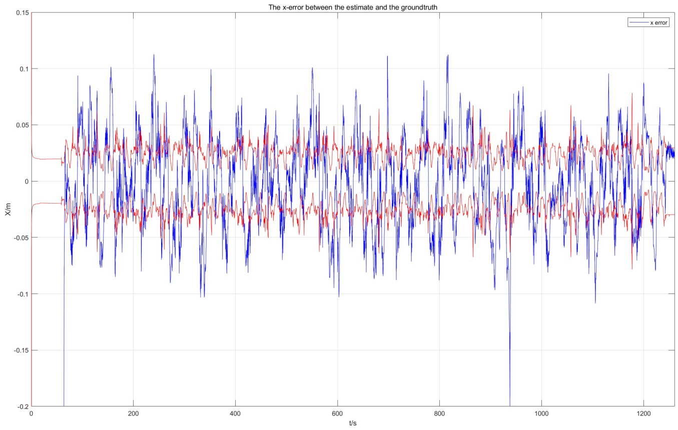


(b) Error in y

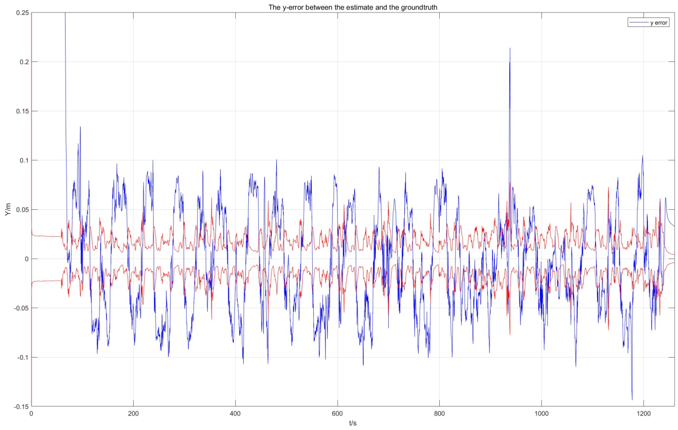


(c) Error in θ

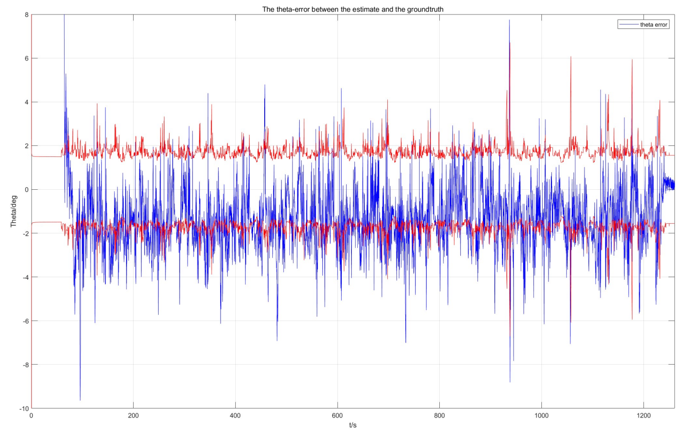
Figure 5: Part 4.b using only landmarks closer than $r_{max} = 1m$ (bad initial state)



(a) Error in x

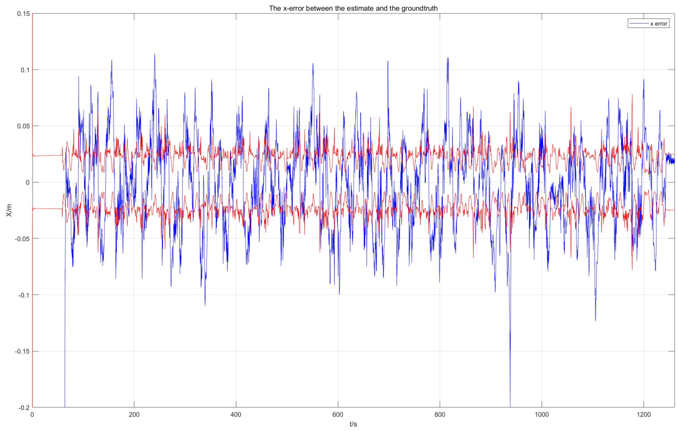


(b) Error in y

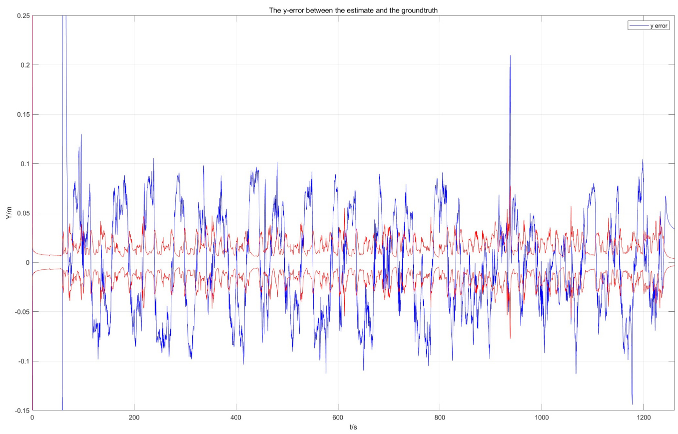


(c) Error in θ

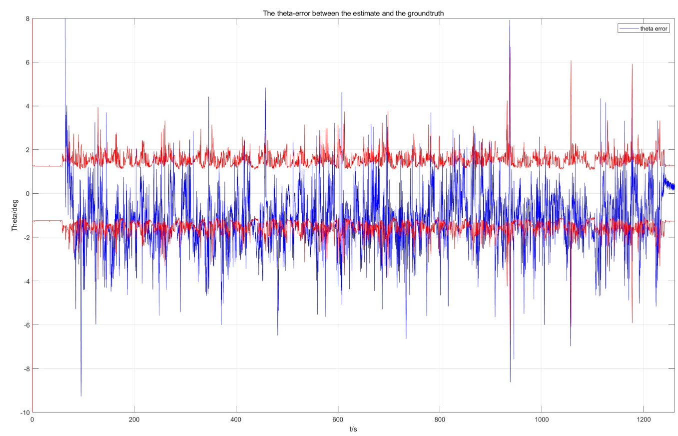
Figure 6: Part 4.b using only landmarks closer than $r_{max} = 3m$ (bad initial state)



(a) Error in x



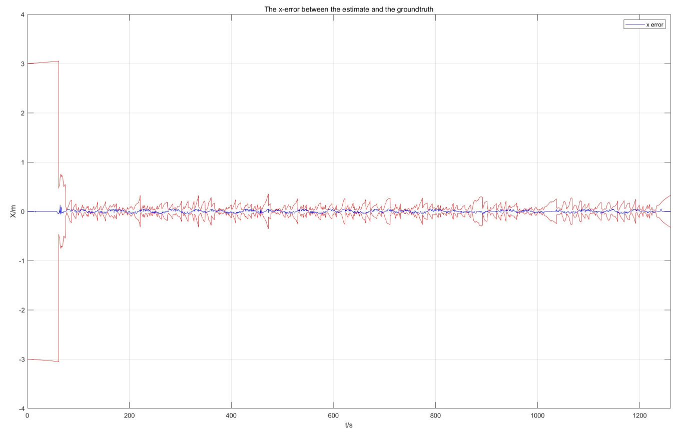
(b) Error in y



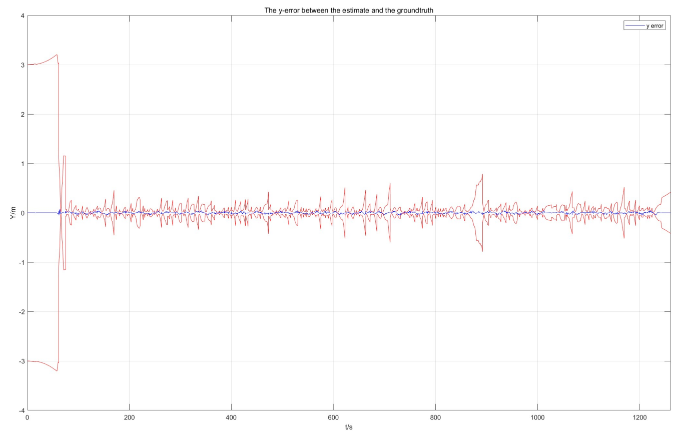
(c) Error in θ

Figure 7: Part 4.b using only landmarks closer than $r_{max} = 5m$(bad initial state)

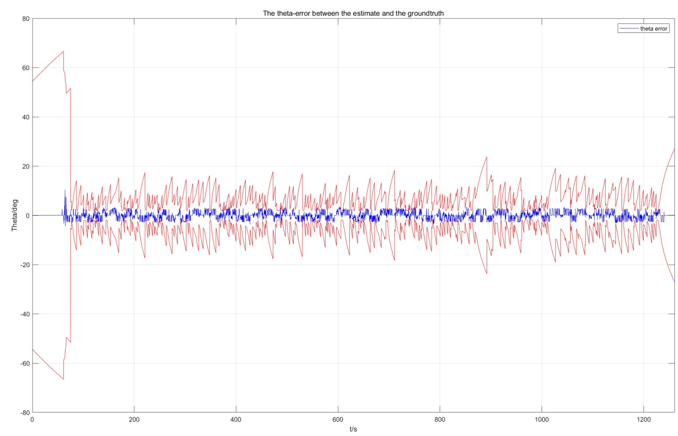
When we use the true robot state to evaluate all the Jacobians, we get the CRLB version of the EKF. The errors are shown as follow:



(a) Error in x

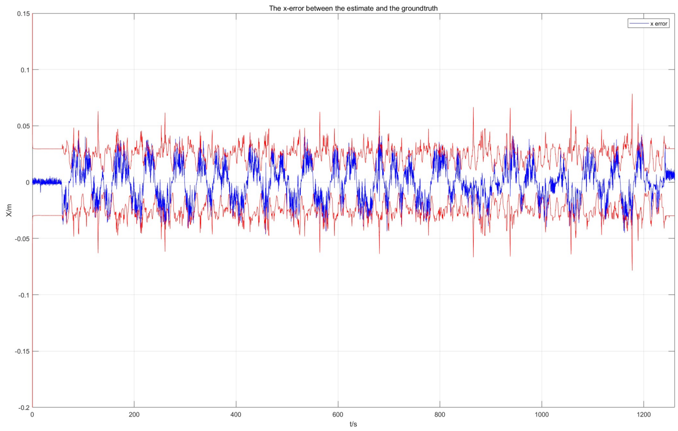


(b) Error in y

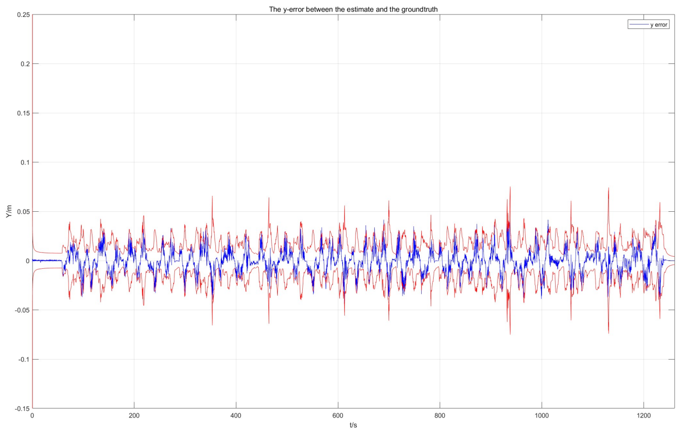


(c) Error in θ

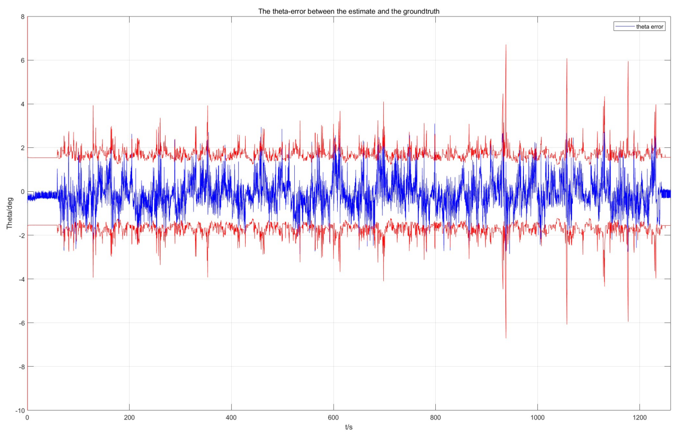
Figure 8: Part 4.c using only landmarks closer than $r_{max} = 1m$ (true robot state)



(a) Error in x

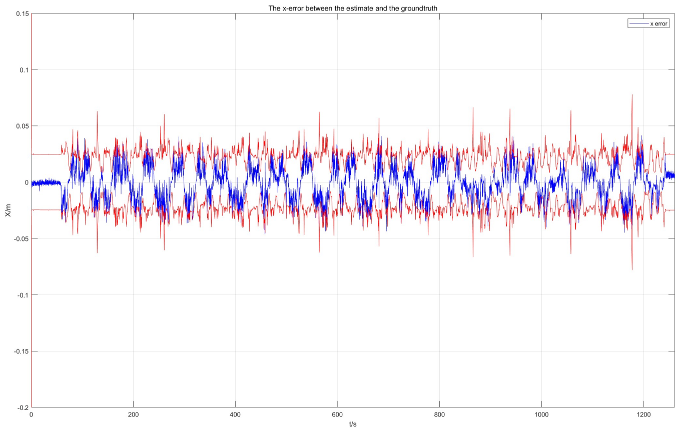


(b) Error in y

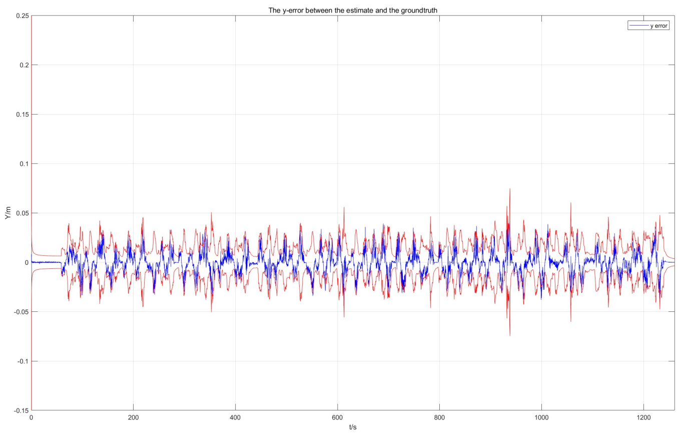


(c) Error in θ

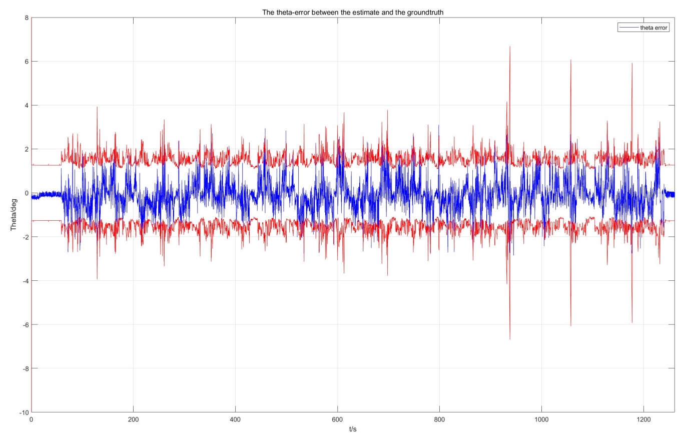
Figure 9: Part 4.c using only landmarks closer than $r_{max} = 3m$ (true robot state)



(a) Error in x



(b) Error in y



(c) Error in θ

Figure 10: Part 4.c using only landmarks closer than $r_{max} = 5m$(true robot state)

All the errors stay within the uncertainty envelop, since we use the true robot state to evaluate the Jacobians, the estimate errors don't accumulate along the timesteps.