Name: Tianxiao Zhao

Email: tzh@kth.se

EL2320 Applied Estimation - Lab 1

Part I – Preparatory Questions

Linear Kalman Filter:

1. What is the difference between a 'control' u_t, a 'measurement' z_t and the state x_t? Give examples of each?

The 'control' $\boldsymbol{u}_t$ helps change the state $\boldsymbol{x}_t$ from previous state $\boldsymbol{x}_{t-1}$through the relation $\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{u}_t) + \boldsymbol{\varepsilon}_t$, where $\boldsymbol{\varepsilon}_t$ denotes the noises in the dynamic system. One example of a 'control' $\boldsymbol{u}_t$ could be the velocity of a robot (set point); The 'measurement' $\boldsymbol{z}_t$ provides information about a momentary state of the environment. For example, the camera images provide us with the relative positions of different objects at time t. $\boldsymbol{z}_t$ could be expressed as the function of the current state $\boldsymbol{x}_t$: $\boldsymbol{z}_t = g(\boldsymbol{x}_t) + \boldsymbol{\delta}_t$, which means that the measurements depend on the current state $\boldsymbol{x}_t$. Usually, $\boldsymbol{u}_t$ is applied in the prediction step and $\boldsymbol{z}_t$ is incorporated in the update step in the linear Kalman filter; The state $\boldsymbol{x}_t$ can be regarded as the collection of all aspects of the object studied, such as the location of people, the location of walls, the velocity of the robot and so on.

2. Can the uncertainty in the belief increase during an update? Why (or not)?

No. According to the update step in the linear Kalman filter, the covariance at time t should satisfy the relation:

$$\Sigma_t = (C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1})^{-1} \Leftrightarrow \Sigma_t^{-1} = C_t^T Q_t^{-1} C_t + \overline{\Sigma}_t^{-1}$$

$\overline{\Sigma}_t$ represents the predicted covariance matrix before update. Because the matrix $C_t^T Q_t^{-1} C_t$ is positive semidefinite, $\Sigma_t^{-1} \geq \overline{\Sigma}_t^{-1}$, which means that $\Sigma_t \leq \overline{\Sigma}_t$ should hold. So, the uncertainty in the belief is expected to decrease or at least remain the same during an update.

3. During update what is it that decides the weighing between measurements and belief?

During the update step, it is the Kalman gain that decides the weighing between measurements and belief. To be specific, the Kalman gain has the equation below:

$$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1}$$

Then we can see that Q_t has an impact on the value of the Kalman gain when C_t is considered to be constant during this process. In other words, the covariance of the measurement noise Q_t decides the weighing between measurements and belief.

4. What would be the result of using a too large a covaraince (Q matrix) for the measurement model?

If the covariance matrix Q_t at time t is too large, the Kalman gain K_t will decrease when other parameters are held constant. As K_t becomes smaller, Σ_t should increase based on the equation $\Sigma_t = (I - K_t C_t)\,\bar{\Sigma}_t$, which means that the estimates behave worse in this case. It will take more time and more measurements for the result to converge.

5. What would give the measurements an increased effect on the updated state estimate?

The Kalman gain K_t specifies the degree to which the measurement is incorporated into the new state estimate. To increase the effect of the measurements, we need to increase $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$, which indicates that the covariance of the measurement noise Q_t should be decreased.

6. What happens to the belief uncertainty during prediction? How can you show that?

The belief uncertainty during prediction could be expressed as below:

$$\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

When $A_t \geq I$, then $A_t \Sigma_{t-1} A_t^T \geq \Sigma_{t-1}$ should hold. $\bar{\Sigma}_t$ with R_t added should be larger than Σ_{t-1}, which means the belief uncertainty increases during prediction; When $A_t < I$, $\bar{\Sigma}_t$ may be smaller than Σ_{t-1} theoretically. But in practice, A_t is usually larger than I. Besides, R_t could be a dominant part of the equation above in real systems. Therefore, the belief uncertainty during prediction tends to increase in a practical sense.

7. How can we say that the Kalman filter is the optimal and minimum least square error estimator in the case of independent Gaussian noise and Gaussian priori distribution? (Just describe the reasoning not a formal proof.)

Suppose we have a linear estimator of the state x_i with three parameters to be determined:

$$\hat{x}_i = a_i \hat{x}_{i-1} + b_i u_i + c_i z_i$$

To get the optimal estimator, we want to minimize the expected value of the squared error e_i :

$$E\{(e_i)^2\} = E\{(x_i - \hat{x}_i)^2\} = E\{(x_i - (a_i \hat{x}_{i-1} + b_i u_i + c_i z_i))^2\}$$

To do the minimization with respect to each parameter, we differentiate the equation above and set the result to zero:

$$\frac{\partial E\{(e_i)^2\}}{\partial a_i} = -2E\{(x_i - (a_i\hat{x}_{i-1} + b_iu_i + c_iz_i))\hat{x}_{i-1}\} = -2E\{e_i\hat{x}_{i-1}\} = 0$$

$$\frac{\partial E\{(e_i)^2\}}{\partial b_i} = -2E\{(x_i - (a_i\hat{x}_{i-1} + b_iu_i + c_iz_i))u_i\} = -2E\{e_iu_i\} = 0$$

$$\frac{\partial E\{(e_i)^2\}}{\partial c_i} = -2E\{(x_i - (a_i\hat{x}_{i-1} + b_iu_i + c_iz_i))z_i\} = -2E\{e_iz_i\} = 0$$

First, we use the first equation to find an expression for a_i. By applying several algebraically equivalent transformations, we get:

$$E\{(x_i - (a_i\hat{x}_{i-1} + b_iu_i + c_iz_i))\hat{x}_{i-1}\} = 0$$

$$\Leftrightarrow E\{(x_i - (a_i\hat{x}_{i-1} - a_ix_{i-1} + a_ix_{i-1} + b_iu_i + c_iz_i))\hat{x}_{i-1}\} = 0$$

$$\Leftrightarrow E\{(x_i - b_iu_i - c_iz_i)\hat{x}_{i-1}\} = a_iE\{(\hat{x}_{i-1} - x_{i-1} + x_{i-1})\hat{x}_{i-1}\}$$

$$\Leftrightarrow E\{(x_i - b_iu_i - c_i(hx_i + Q_i))\hat{x}_{i-1}\} = a_iE\{e_{i-1}\hat{x}_{i-1}\} + a_iE\{x_{i-1}\hat{x}_{i-1}\}$$

Because $E\{e_{i-1}\hat{x}_{i-1}\} = E\{e_{i-1}(a_{i-1}\hat{x}_{i-2} + b_{i-1}u_{i-1} + c_{i-1}z_{i-1})\} = 0$ and the previous estimate is uncorrelated with the measurement noise, which means $E\{Q_i\hat{x}_{i-1}\} = 0$, we get (the linear system has state transition equation $x_i = ax_{i-1} + bu_i + R_i$):

$$E\{(x_i(1 - c_ih) - b_iu_i)\hat{x}_{i-1}\} = a_iE\{x_{i-1}\hat{x}_{i-1}\}$$

$$\Leftrightarrow E\{((ax_{i-1} + bu_{i-1})(1 - c_ih) - b_iu_i)\hat{x}_{i-1}\} = a_iE\{x_{i-1}\hat{x}_{i-1}\}$$

$$\Leftrightarrow (a(1 - c_ih) - a_i)E\{x_{i-1}\hat{x}_{i-1}\} + (b(1 - c_ih) - b_i)E\{u_i\hat{x}_{i-1}\} = 0$$

Similarly, we do the same transformation for the second differentiation above and derive the equation below:

$$(a(1 - c_ih) - a_i)E\{x_{i-1}u_i\} + (b(1 - c_ih) - b_i)E\{u_i^2\} = 0$$

We can write them in matrix form:

$$\begin{bmatrix} E\{x_{i-1}\hat{x}_{i-1}\} & E\{u_i\hat{x}_{i-1}\} \\ E\{x_{i-1}u_i\} & E\{u_i^2\} \end{bmatrix} \begin{bmatrix} a(1 - c_ih) - a_i \\ b(1 - c_ih) - b_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which could be simplified as AX = 0. Then we have X = 0 (only if $\hat{x}_{i-1}$ and x_{i-1} are independent can matrix A = 0). In other words, $a_i = a(1 - c_ih)$, $b_i = b(1 - c_ih)$ need to be satisfied. Then we have our optimal estimator below:

$$\hat{x}_i = a(1 - c_ih)\hat{x}_{i-1} + b(1 - c_ih)u_i + c_iz_i$$

$$\Leftrightarrow \hat{x}_i = a\hat{x}_{i-1} + bu_i + c_i(z_i - h(a\hat{x}_{i-1} + bu_i))$$

This optimal estimator has the same form with the Kalman filter. Therefore, we can say that the Kalman filter is the optimal and minimum least square error estimator in the case of independent Gaussian noise and Gaussian priori distribution.

8. In the case of Gaussian white noise and Gaussian priori distribution, is the Kalman Filter a MLE

and/or MAP estimator?

It depends on whether the Gaussian priori distribution is applied in the estimation process. If we don't know or we don't incorporate the priori distribution of the state into the estimation, then the Kalman filter is a MLE estimator; Otherwise, it is a MAP estimator.

Extended Kalman Filter:

9. How does the extended Kalman filter relate to the Kalman filter?

The extended Kalman filter is the generalized version of the linear Kalman filter. It is usually applied to the nonlinear system which is linearized at the stationary point. For the EKF, the state transition function becomes nonlinear from $x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$ to $x_t = G(x_{t-1}, u_t) + \varepsilon_t$, where $G(\cdot)$ is a nonlinear function; Besides, the measurement model changes from $z_t = C_t x_t + \varepsilon_t$ to $z_t = H(x_t) + \varepsilon_t$, where $H(\cdot)$ is a nonlinear function.

10. Is the EKF guaranteed to converge to a consistent solution?

No. If the initial estimate of the state is wrong, or if the nonlinear state transition model is not precise, then the EKF may diverge; Besides, the underestimated covariance matrix of the noise may cause the result to be inconsistent.

11. If our filter seems to diverge often can we change any parameter to try and reduce this?

Yes. We could change the covariance matrix of the noise Q_t and R_t, so that the convergence may become slower and more stable (e.g. increasing Q_t).

Localization:

12. If a robot is completely unsure of its location and measures the range r to a known landmark with Gaussian noise what does its posterior belief of its location $p(x, y, \theta|r)$ look like? So a formula is not needed but describe it at least.

Based on the condition, the heading θ, position x, y are uncertain. The heading θ will have a uniform distribution between -180° and 180°. The position will be a circle around the known landmark with a uniform distribution.

13. If the above measurement also included a bearing how would the posterior look?

According to the geometric relation between a bearing, a heading and an angle around the circle, one of them could be expressed as a function of the other two. This means that the heading and the angle around the circle are heavily correlated when the bearing is measured. So, the posterior should be a circle around the known landmark. The heading and the angle around the circle should both have Gaussian distribution with a correlated covariance.

14. If the robot moves with relatively good motion estimation (prediction error is small) but a large initial uncertainty in heading θ how will the posterior look after traveling a long distance without seeing any features?

According to the circumstances, the update step in EKF will be ignored because no measurements are received. The posterior will be a C shape trace since the good motion estimation guarantees that the robot could go for a fixed distance, but the uncertainty in heading θ makes the direction of the robot uncertain. So, the heading θ should be correlated with position on the trace.

15. If the above robot then sees a point feature and measures range and bearing to it how might the EKF update go wrong?

If the robot sees a point feature and incorporates the measurements into EKF, the linearization could generate a straight-line uncertainty region. Since the uncertainty area of Gaussian distribution cannot cover the feature, the update will diverge finally.

Part II – Matlab Exercises

Question 1

According to the state-space model, the process noise ε_k should have the same dimension with x_k. So, ε_k should be a $M\times1$ vector, where M denotes the number of the state variables. The measurement noise δ_k shares the same dimension with z_k, which means that δ_k should have a dimension of $N\times1$, where N is the number of measurements. In this case, $M = 2$ and $N = 1$. So, ε_k is a 2×1 vector and δ_k is a scalar.

To uniquely characterize a white Gaussian in a scalar case, we need to know the mean μ and the variance σ^2 of the distribution. Then, this normal distribution can be uniquely represented by $\mathcal{N}(\mu, \sigma^2)$. When the distribution is a vector, the mean μ becomes a single-column matrix and the variance is replaced by the covariance matrix Σ. Because the white noise is independent of other signals, this covariance matrix Σ should be a diagonal matrix. Besides, the white noise usually has a zero mean, which means that $\mu = 0$.

Question 2

Variable	Role / Usage
x	The true state of the car system
xhat	The estimated state of the car system by Kalman filter
P	The covariance matrix of the estimation by Kalman filter
G	The matrix which contains factors of the process noise in the system
D	The factor of the measurement noise in the system
Q	The variance of the measurement noise
R	The covariance matrix of the process noise
wStdP	The true deviation of the noise on simulated position
wStdV	The true deviation of the noise on simulated velocity
vStd	The true deviation of the measurement noise
U	The control applied to the system (acceleration)
PP	The matrix which contains reshaped covariance matrices of the estimation over a time period

Question 3

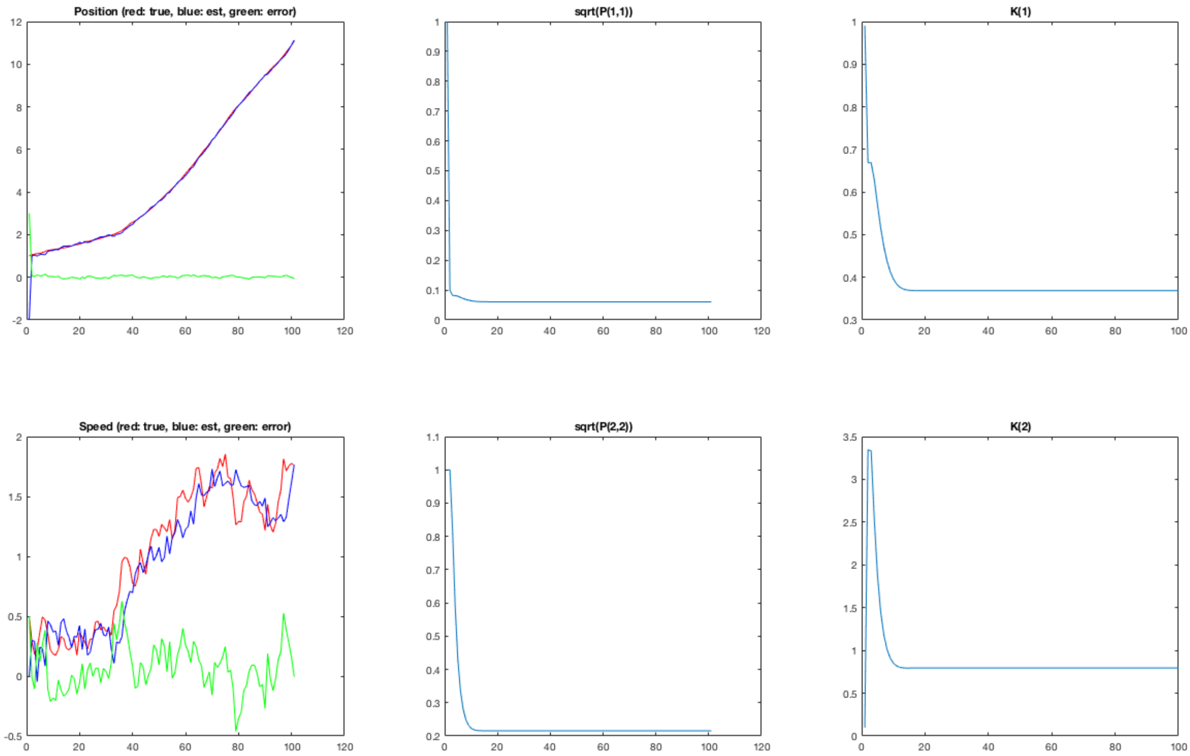

Figure 1: Estimation error, covariance and Kalman filter gain (Q and R default)

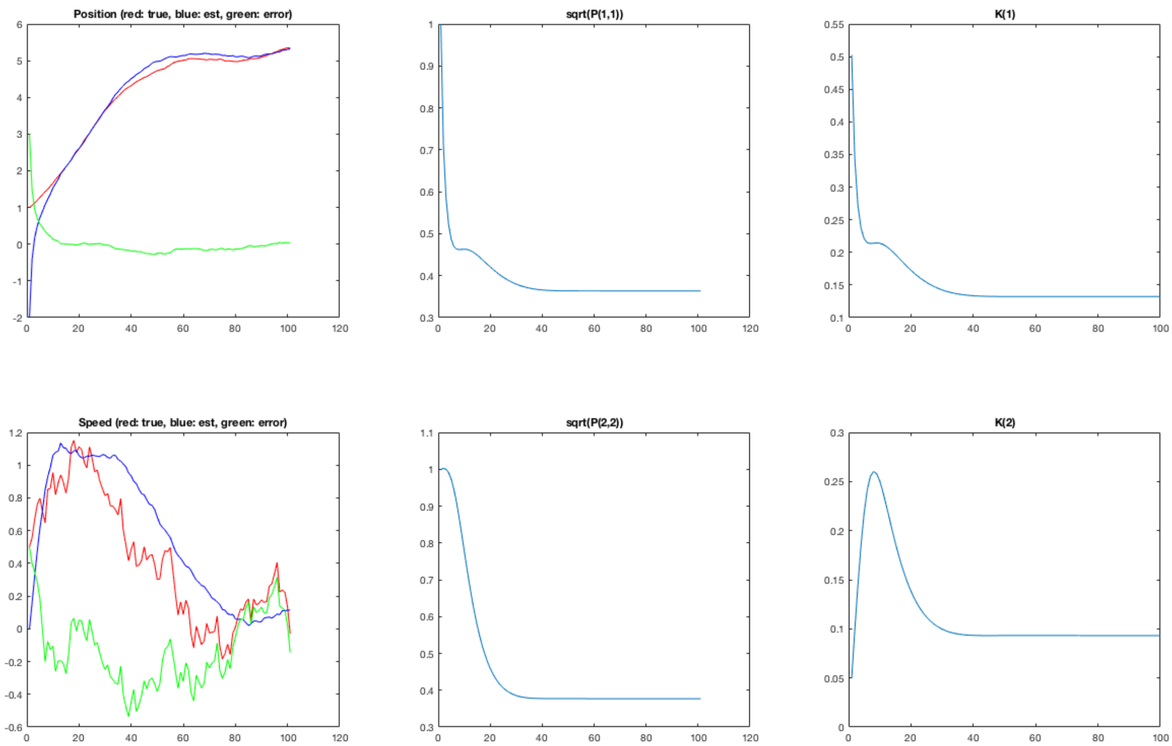

Figure 2: Estimation error, covariance and Kalman filter gain (Q $\times$ 100, R default)

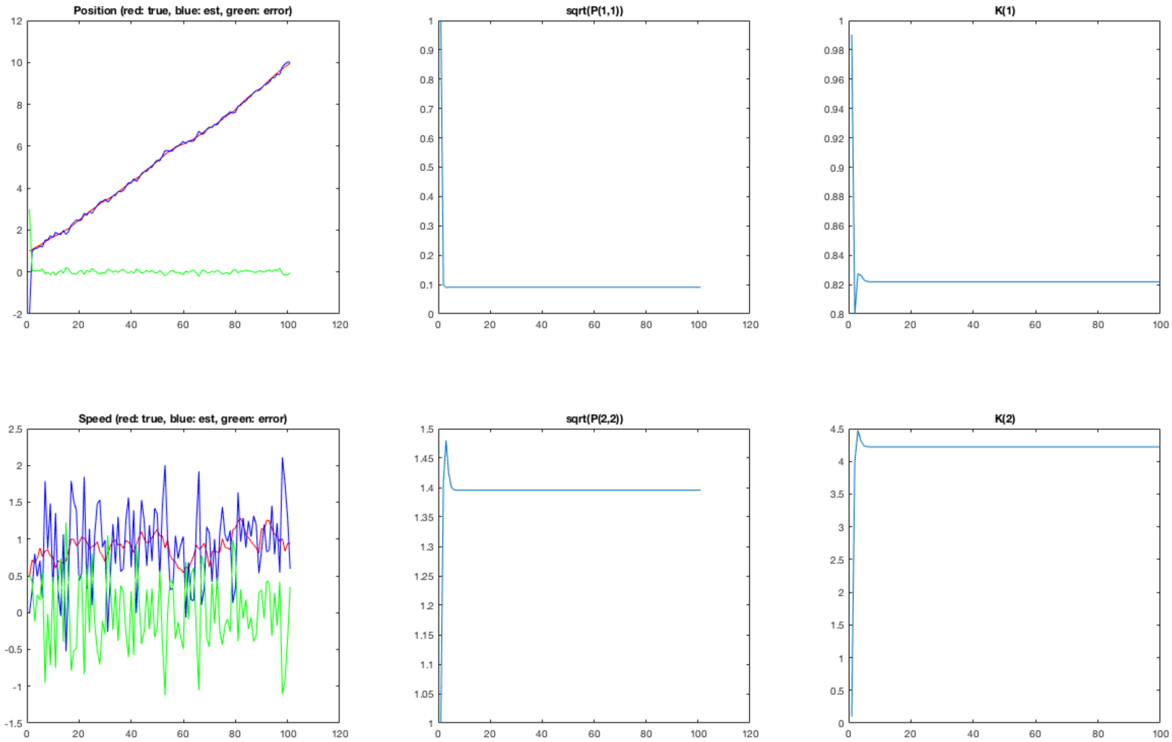

Figure 3: Estimation error, covariance and Kalman filter gain (Q default, R $\times$ 100)

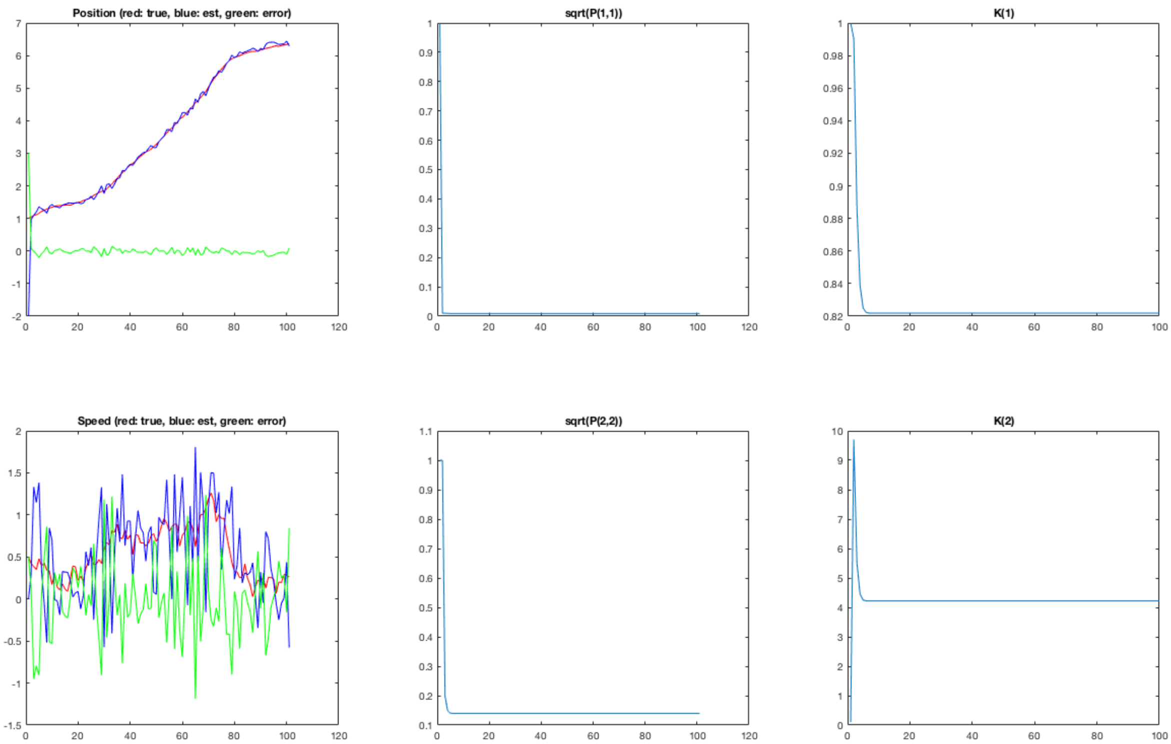

Figure 4: Estimation error, covariance and Kalman filter gain (Q / 100, R default)

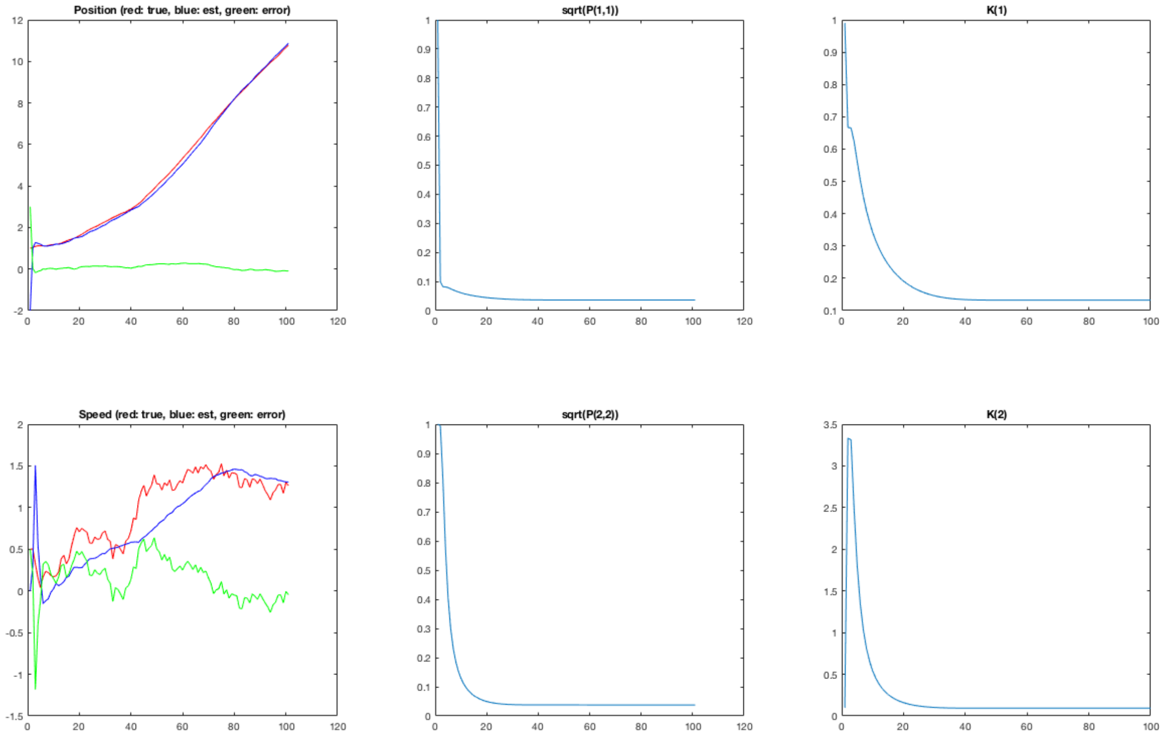

Figure 5: Estimation error, covariance and Kalman filter gain (Q default, R / 100)

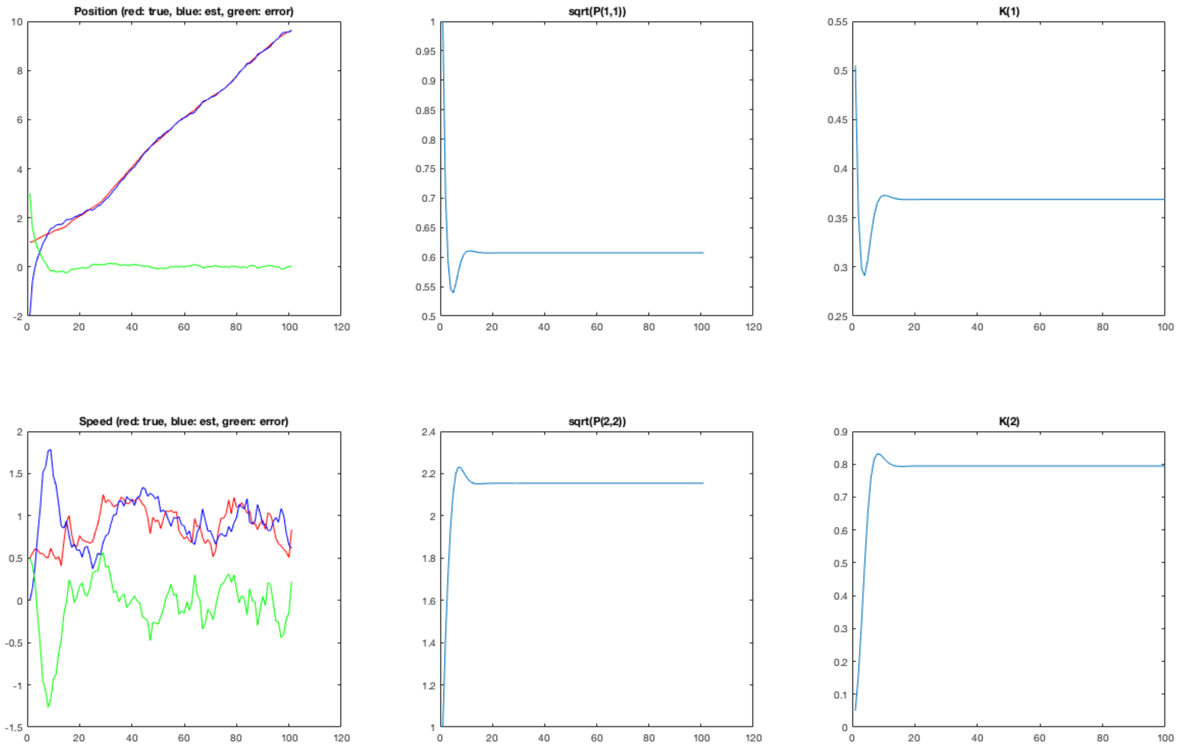

Figure 6: Estimation error, covariance and Kalman filter gain (Q $\times$ 100, R $\times$ 100)

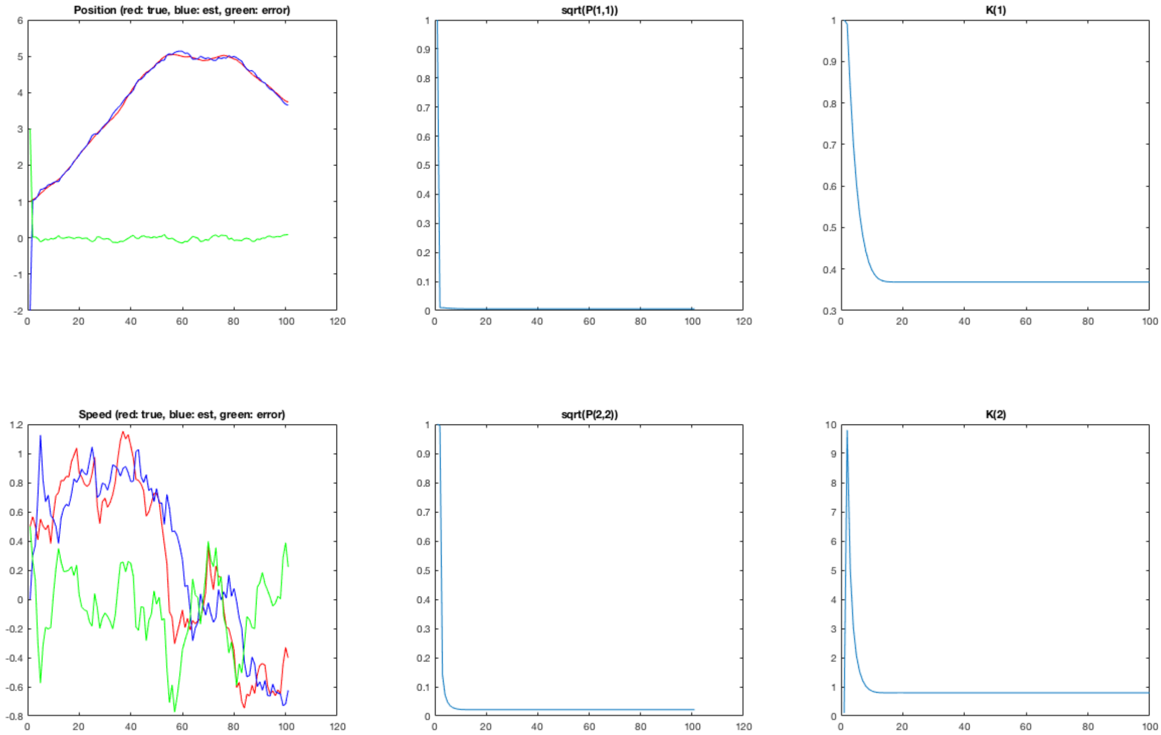

Figure 7: Estimation error, covariance and Kalman filter gain (Q / 100, R / 100)

When we increase the covariance of the measurement noise (Q) and keep other parameters unchanged, the Kalman gain should decrease according to the equation:

$$K(k+1) = P(k+1|k) \cdot C^T (CP(k+1|k)C^T + DQD^T)^{-1} \quad (1)$$

Since the Kalman gain decreases, meaning the measurements are trusted and incorporated less than before, the covariance of the estimate should increase based on the equation below. And it will also take more time to converge for the Kalman filter. These conclusions could be supported by two groups of figures {Figure 1, Figure 2} and {Figure 3, Figure 6}; And when Q decreases, we could expect that the effects mentioned above are totally the opposite. This could also be verified by another two groups of figures {Figure 1, Figure 4} and {Figure 5, Figure 7 }.

$$P(k+1|k+1) = (I - K(k+1) \cdot C) \cdot P(k+1|k) \quad (2)$$

On the other hand, if we increase the covariance of the process noise (R) and keep other parameters unchanged, the predicted error covariance $P(k+1|k)$ will also increase. Then we could derive that the covariance of the estimation $P(k+1|k+1)$ increases based on the following equation. This also means that the convergence will take more time.

$$P(k+1|k+1) = (CQ^{-1}C^T + P(k+1|k)^{-1})^{-1} \quad (3)$$

Furthermore, the Kalman gain would increase with the increase of the covariance of the estimation. This could be explained by the relation below. This is exactly the case as is shown in two different groups of figures {Figure 1, Figure 3} and {Figure 2, Figure 6}. And the opposite conclusion could be expected and backed up with the figures {Figure 1, Figure 5} and {Figure 4, Figure 7}.

$$K(k+1) = P(k+1|k+1) \cdot C^T \cdot Q^{-1} \tag{4}$$

When both Q and R increase together, the Kalman gain will decrease because the denominator increases faster than the numerator as is shown in Equation (1). Since $P(k+1|k)$ increases, according to Equation (2), the covariance of the estimation $P(k+1|k+1)$ should increase. The convergence will be relatively slower. This could be proven by Figure 1 and Figure 6. The combination of Figure 1 and Figure 7 shows that the Kalman filter will perform in an opposite way when both Q and R decrease together.

Question 4

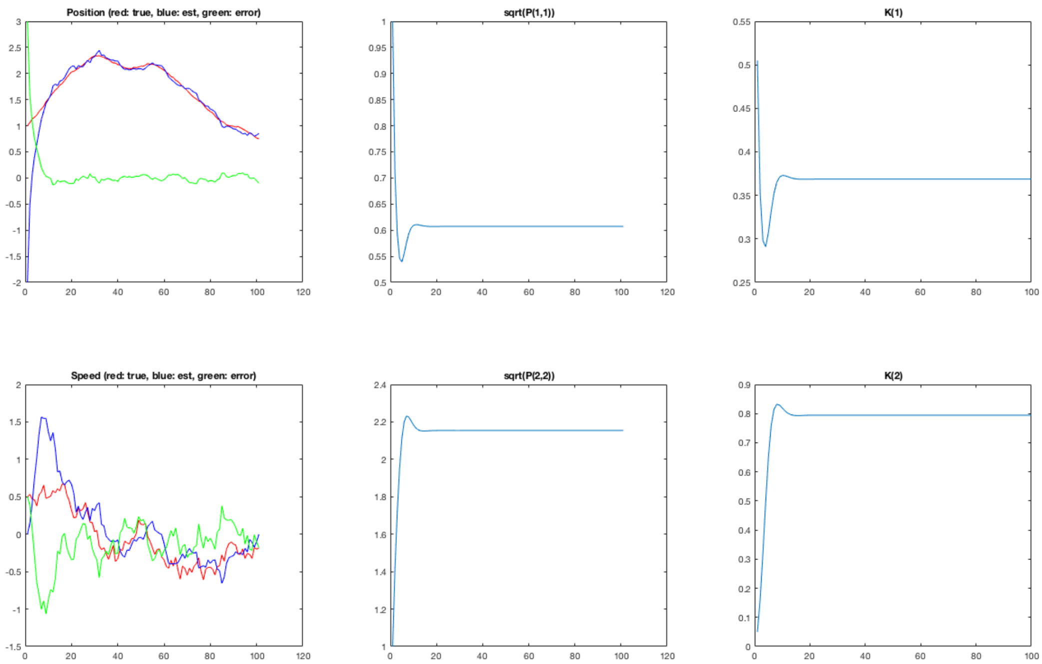

Figure 8: Estimation error, covariance and Kalman filter gain (P default)

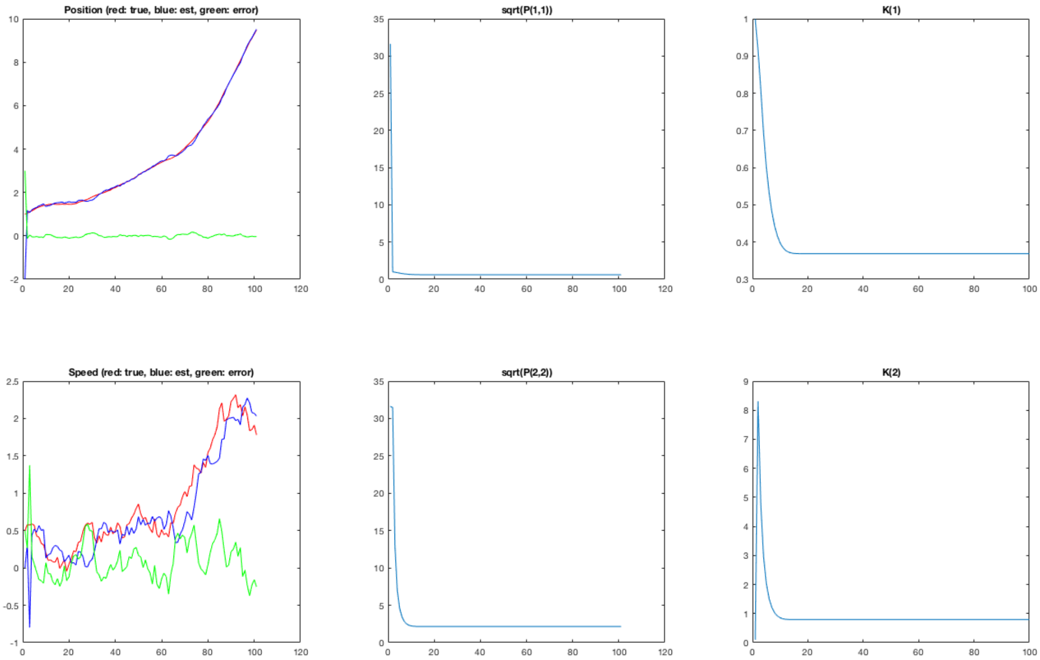

Figure 9: Estimation error, covariance and Kalman filter gain (P $\times$ 1000)

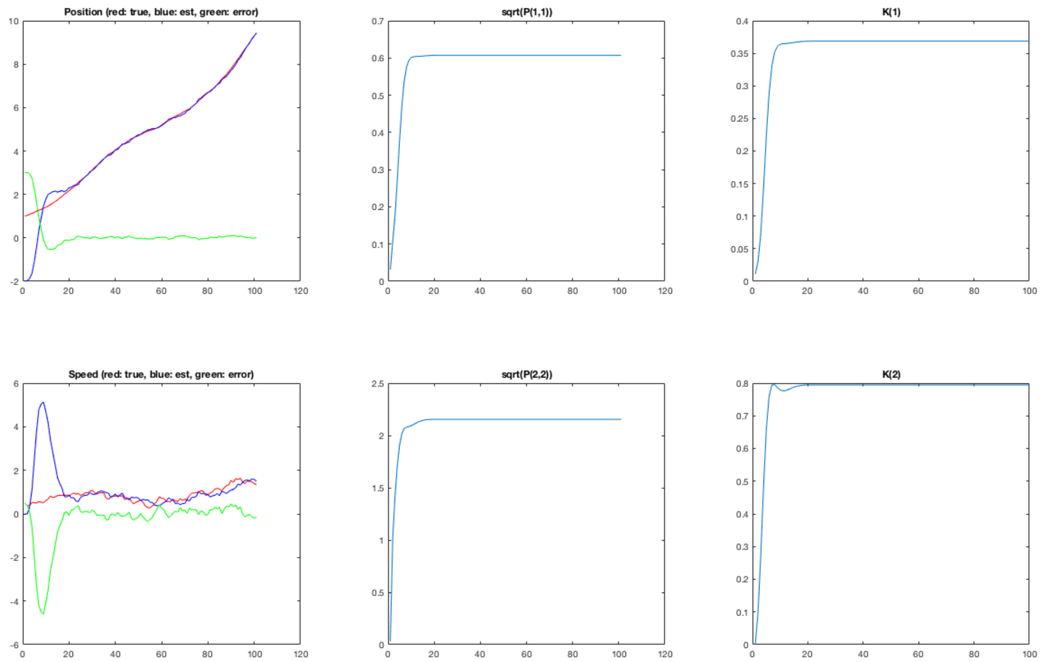

Figure 10: Estimation error, covariance and Kalman filter gain (P / 1000)

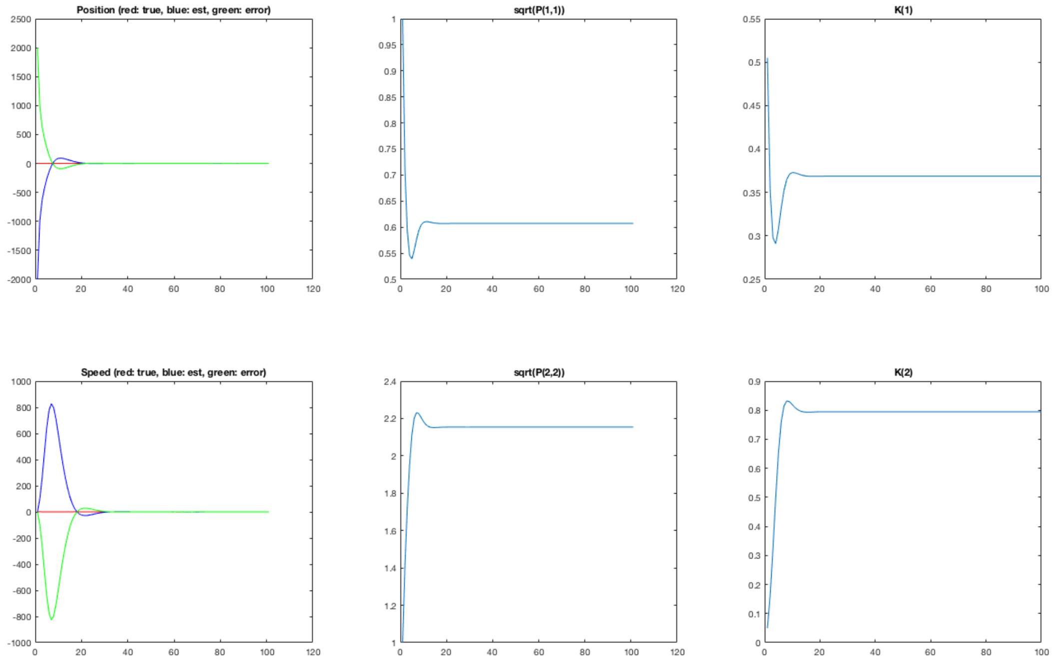

Figure 11: Estimation error, covariance and Kalman filter gain (xhat × 1000)

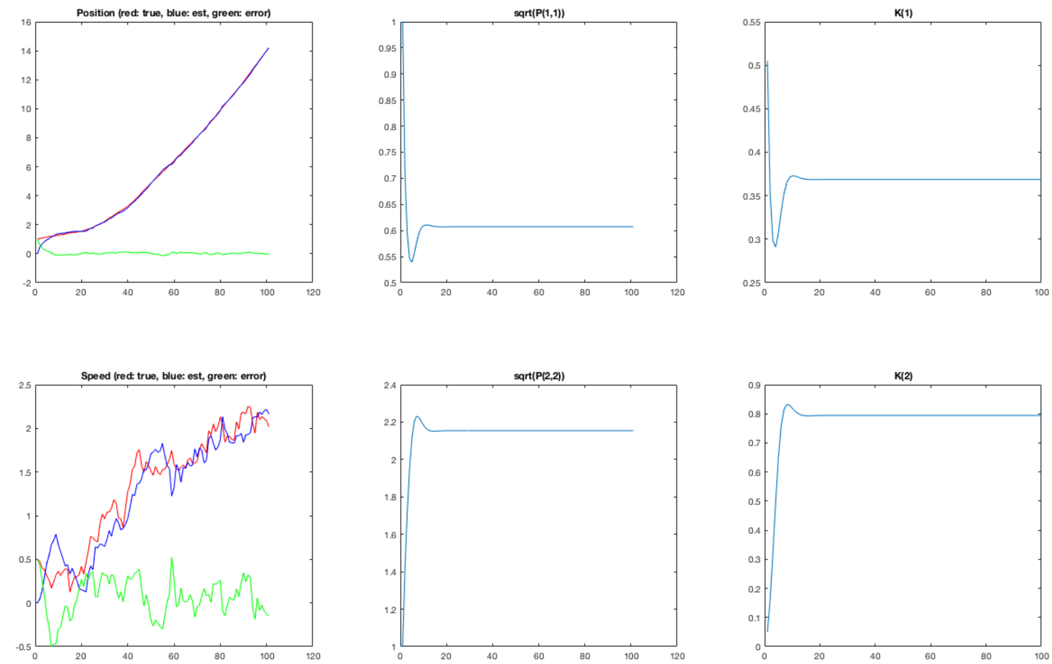

Figure 12: Estimation error, covariance and Kalman filter gain (xhat / 1000)

When P is very large and xhat is kept unchanged, this means that the uncertainty of the true state is still dominant and the prediction part is not reliable enough. Then the Kalman gain should rise rapidly at this moment since the measurements for update are trusted more. From Figure 8 and Figure 9, we could notice that the rates of convergence in both cases are approximately the same, and the covariance converges to nearly the same level at the end; When P is very small, the initial estimate should be trusted more and the measurements will relatively be incorporated less. So, the Kalman gain at the beginning should be very small. During the estimation process, the Kalman gain will rise slowly, which guarantees that the convergence will be slower than before. The estimate error in this case is also regulated to the same level, see Figure 8 and Figure 10.

When xhat is changed to a very large and very small value, it seems that a very minor impact is applied to the system. The changes of the rate of convergence and the estimate error are negligible, see Figure 8, Figure 11 and Figure 12. This makes sense because changing the initial mean value will not affect the Kalman gain and the covariance of the estimate.

Question 5

$$bel(x_t) = p(x_t|u_{1:t}, z_{1:t}, \bar{x}_0, M) = \eta p(z_t|x_t, M)\overline{bel}(x_t) \tag{5}$$

$$\overline{bel}(x_t) = p(x_t|u_{1:t}, z_{1:t-1}, \bar{x}_0, M) = \int p(x_t|u_t, x_{t-1})\ bel(x_{t-1})dx_{t-1} \tag{6}$$

The prediction step uses the belief on the last time step $bel(x_{t-1})$, and the update step incorporates the measurements z_t. Therefore, Equation (5) is responsible for update step and Equation (6) is for prediction step.

$$p(x_t|u_{1:t}, z_{1:t}, \bar{x}_0, M) = \overbrace{\eta p(z_t|x_t, M) \underbrace{\int p(x_t|u_t, x_{t-1})\, p(x_{t-1}|u_{1:t-1}, z_{1:t-1}, \bar{x}_0, M)dx_{t-1}}_{\text{Prediction step}}}^{\text{Update step}}$$

And in the equation above, the integral part is responsible for prediction. After multiplying $\eta p(z_t|x_t, M)$, the whole becomes the update step.

Question 6

Theoretically, it is a valid assumption. This is based on the fact that EKF is first-order Markov model. But in reality, this assumption is not valid because the measurements at certain moment are correlated with the state at that point. In general, this is a trade-off effect since including more state transitions will cause an increase on the complexity of the model and an unreliability of the results of the maximum likelihood method.

Question 7

The bound of δ_M is $[0, 1]$ because δ_M is a probability. Based on the relation $\lambda_M = X_2^{-2}(\delta_M)$ and the ascending monotonicity of the X_2^{-2}, an increase of δ_M will cause a larger λ_M. If the threshold λ_M increases, the previous outliers with large Mahalanobis distance may be classified as inliers. Then, there will be fewer outlier rejections. And a smaller δ_M will cause a higher probability of outlier rejections. If the measurements are reliable, then the covariance matrix Q will be small, and the Mahalanobis distance D_M of these measurements will be large according to the equation below:

$$D_M = \left(\bar{v}_t^i\right)^T (H_{t,j}\bar{\Sigma}_t\left(H_{t,j}\right)^T + Q)^{-1}\bar{v}_t^i$$

To accept more reliable measurements, we need to set a large threshold λ_M, which means δ_M should be given a large value; If the measurements are unreliable, we could set the probability δ_M to a lower value in order to create a smaller threshold λ_M.

Question 8

Suppose the first measurement is noisy in the sequential update algorithm. Then the mean value after update will have a large deviation from the estimate with low-noise measurements, and it will pollute the following prediction and update step. And this problem can be solved by the batch update algorithm because the noisy measurements are in minority of the batch data and they have little influences on the updated mean value. In this way, the effects of the noise in measurements will be largely reduced.

Question 9

In Algorithm 4, the first three lines in the nested iterations do not include variable i, which means the values of $\hat{z}_{t,j}$, $H_{t,j}$ and $S_{t,j}$ don't depend on the value of i for each iteration. These three lines will be computed for $\text{size}(z_t) \times \text{size}(M)$ times, where $\text{size}(z_t)$ means the number of observations and $\text{size}(M)$ means the number of the landmarks. If we store the values of $\hat{z}_{t,j}$, $H_{t,j}$ and $S_{t,j}$ when $j = 1$, and skip computing them as the value of j goes up, we will end up executing the above computation only for $\text{size}(z_t)$ times. In this way, we could avoid redundant re-computations.

Question 10

For each value of i, $\bar{v}_t^i$ has the same dimension with $\hat{z}_{t,j}$, which is a $2{\times}1$ matrix. Since $\bar{v}_t = [(\bar{v}_t^1)^T\ (\bar{v}_t^2)^T\ \ldots\ (\bar{v}_t^n)^T]^T$, $\bar{v}_t$ should be a $2n{\times}1$ matrix. Since the Jacobian matrix is a $2{\times}3$ matrix, $\bar{H}_{t,i}$ should have the same size. Then we could derive that the dimension of $\bar{H}_t$ should be $2n{\times}3$ because of the expression: $\bar{H}_t = [\left(\bar{H}_{t,1}\right)^T\ \left(\bar{H}_{t,2}\right)^T\ \ldots\ \left(\bar{H}_{t,n}\right)^T]^T$. In the sequential update algorithm, $\bar{v}_t$ and $\bar{H}_t$ have a dimension of $2{\times}1$ and $2{\times}3$, respectively. This makes sense because the batch algorithm performs one update after associating observations with landmarks at the same time, and the sequential algorithm treats each single data separately and repeats the update until all the data is processed.

Data sets

1. map_o3.txt + so_o3_ie.txt

Because the odometry information has an un-modeled noise of approximately 1 cm and 1 degree per time step, we choose R and Q in the form shown below. We type **runlocalization_track('so_o3_ie.txt', 'map_o3.txt', 1, 1, 1, 2)** on the command window of Matlab and run the program. The results of the EKF using batch algorithm are also listed, see Figure 13-Figure 15. We could see that the absolute mean errors are all below 0.01 on all dimensions, which means that our noise models can satisfy the requirement.

$$R = \begin{bmatrix} (0.01)^2 & 0 & 0 \\ 0 & (0.01)^2 & 0 \\ 0 & 0 & (\frac{\pi}{180})^2 \end{bmatrix},\ Q = \begin{bmatrix} (0.01)^2 & 0 \\ 0 & (\frac{\pi}{180})^2 \end{bmatrix}$$

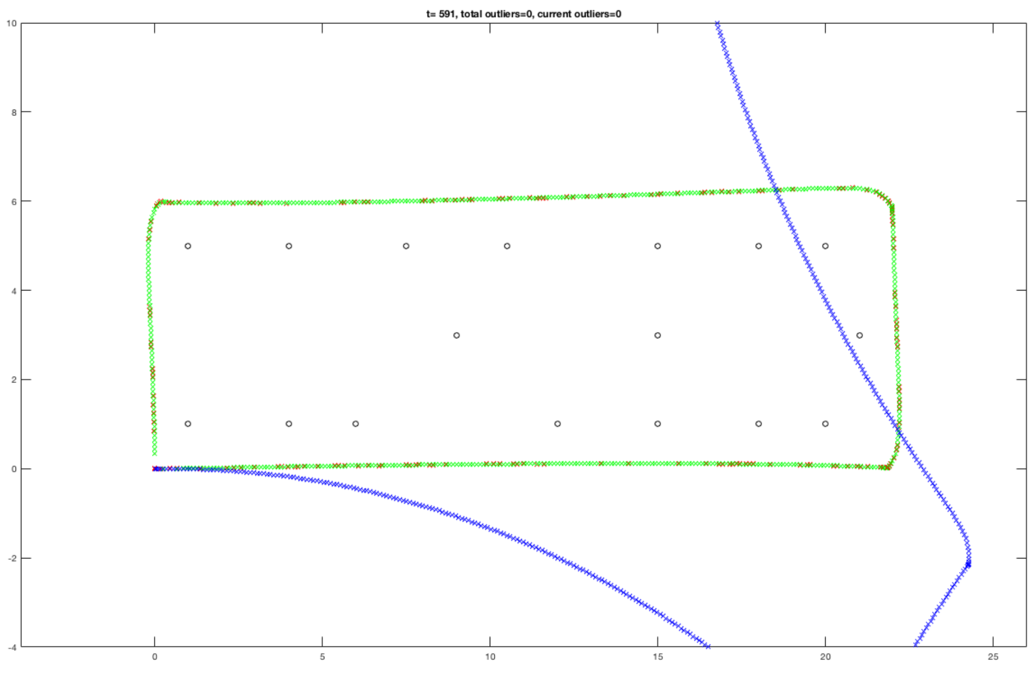

Figure 13: Output of the EKF using batch update algorithm

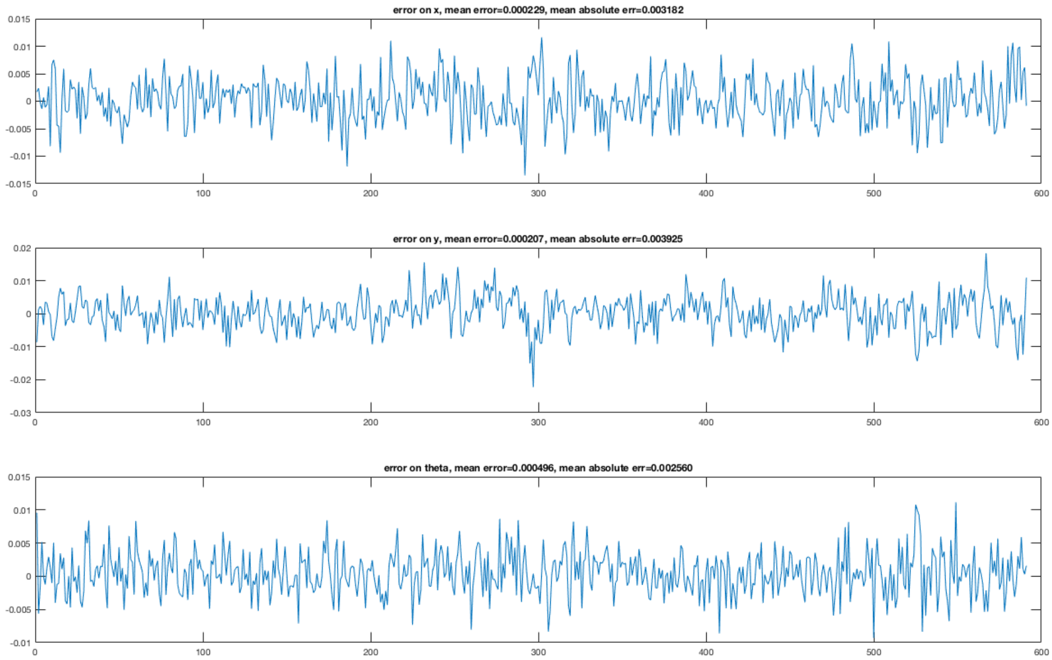

Figure 14: Error information of the estimates

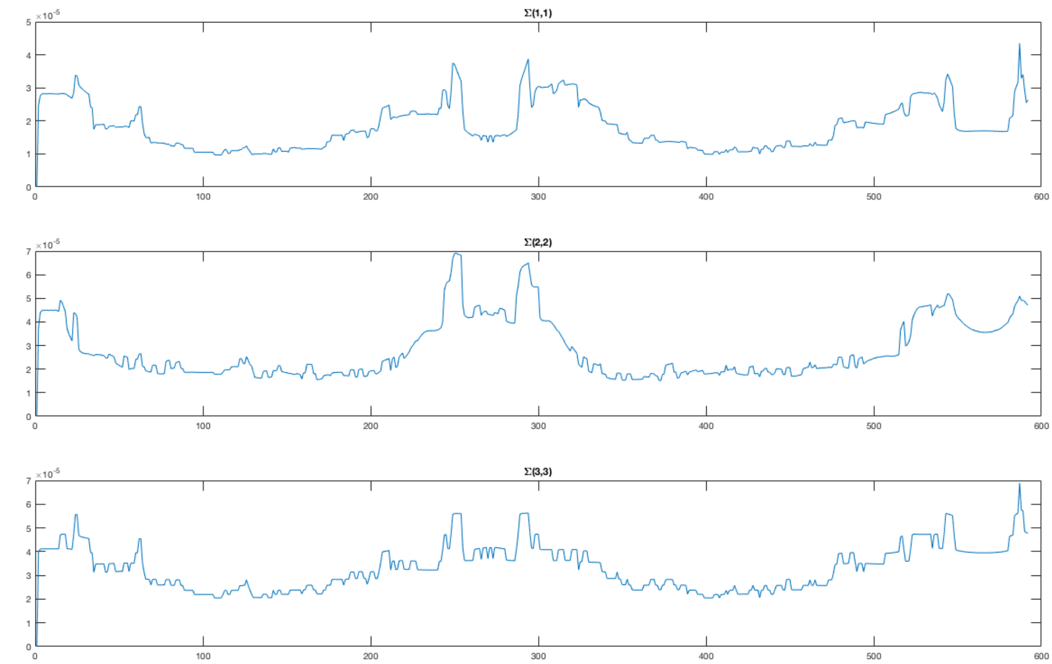

Figure 15: Covariance information of the estimates

The printout information by MATLAB:

mean error(x, y, theta)=(0.000229, 0.000207, 0.000496)

mean absolute error=(0.003182, 0.003925, 0.002560)

total_time =12.936613

2. map_pent_big_10.txt + so_pb_10_outlier.txt

In this case, we keep the R matrix as above and change the Q matrix according to the instruction:

$$R = \begin{bmatrix} (0.01)^2 & 0 & 0 \\ 0 & (0.01)^2 & 0 \\ 0 & 0 & (\frac{\pi}{180})^2 \end{bmatrix}, Q = \begin{bmatrix} 0.2^2 & 0 \\ 0 & 0.2^2 \end{bmatrix}$$

We type **runlocalization_track('so_pb_10_outlier.txt', 'map_pent_big_10.txt', 1, 1, 1, 2)** on the command window of Matlab and run the program. The outputs of the EKF are shown below, see Figure 16-Figure 18. As is required, the mean absolute errors are all smaller than 0.06. The increased covariance of the measurements causes an increased outliers to be detected.

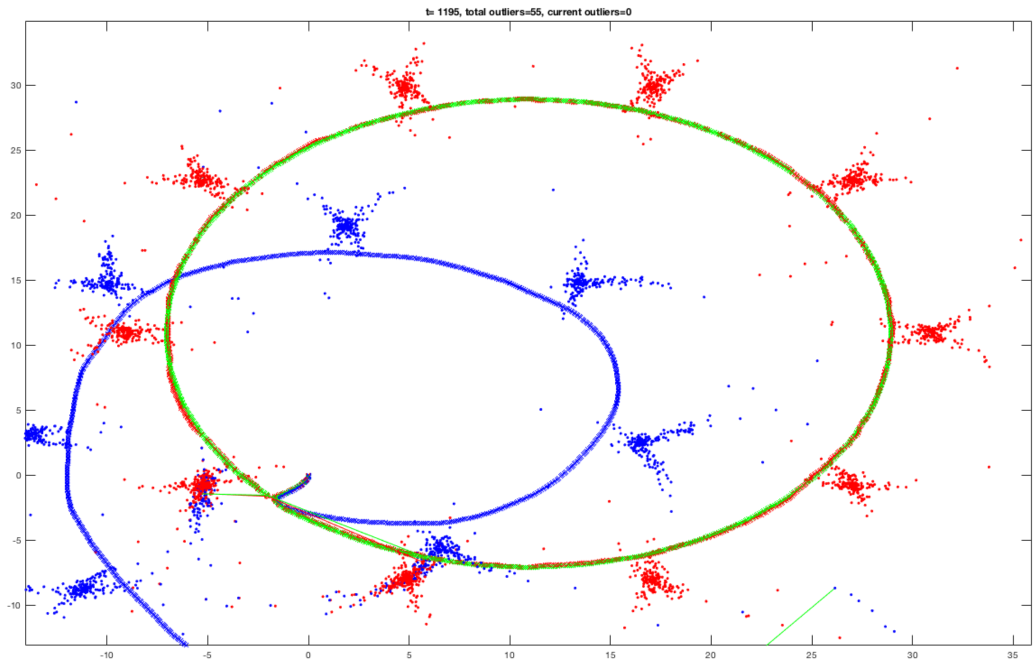

Figure 16: Output of the EKF using batch update algorithm

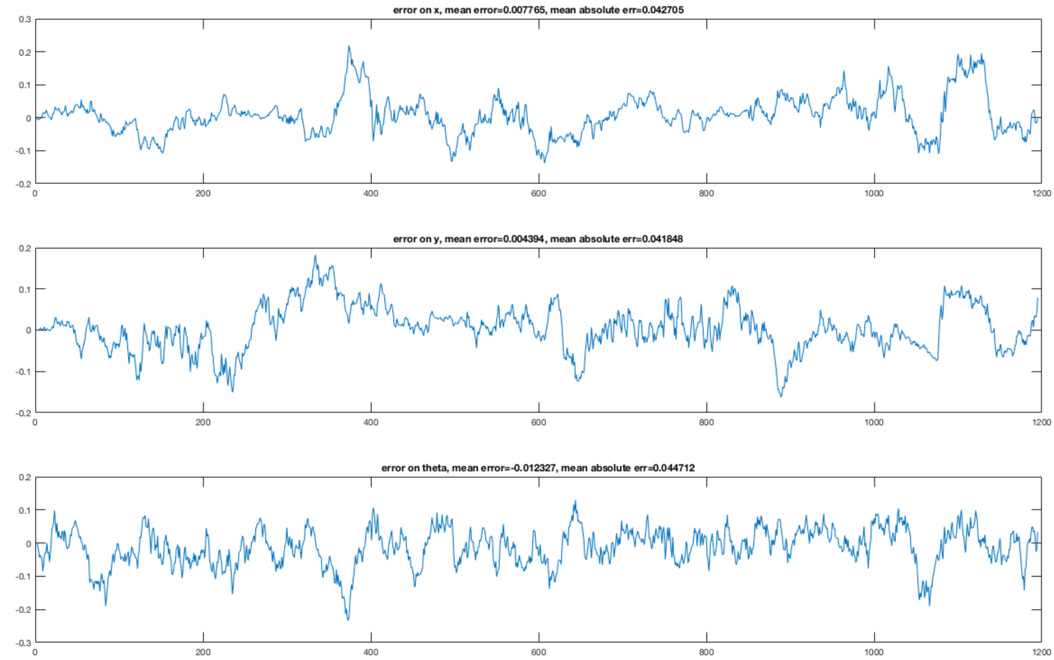

Figure 17: Error information of the estimates

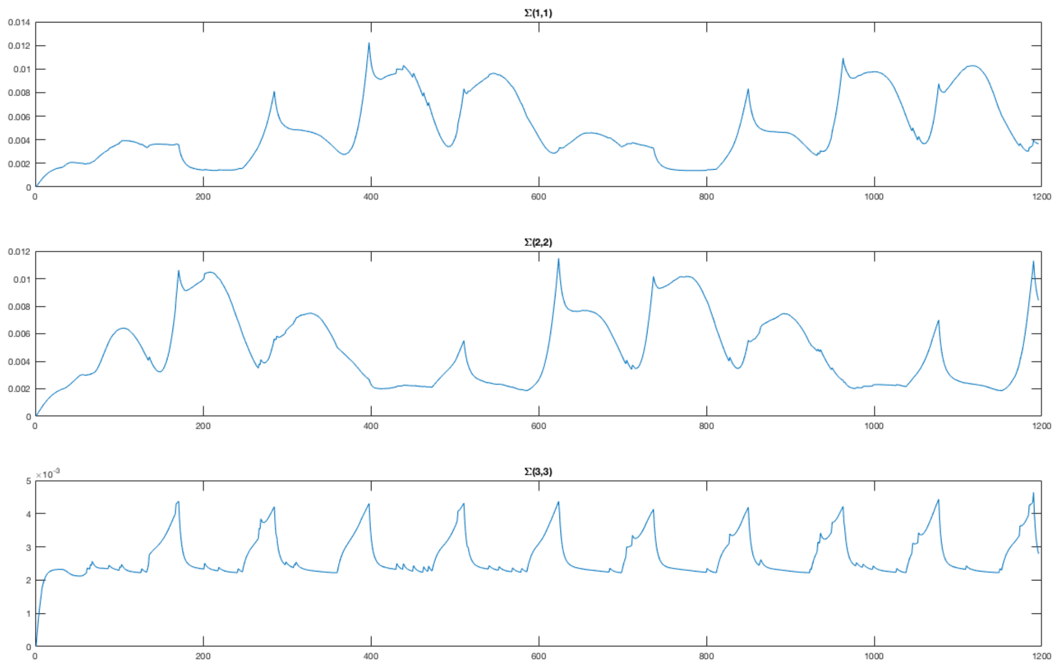

Figure 18: Covariance information of the estimates

The printout information by MATLAB:

mean error(x, y, theta)=(0.007763, 0.004417, -0.012343)

mean absolute error=(0.042680, 0.041876, 0.044715)

total_time =90.634219

3. map_pent_big_40.txt + so_pb_40_no.txt

3.1 Sequential update algorithm

According to the instruction, we set Q and R in the following way:

$$R = \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 1^2 & 0 \\ 0 & 0 & 1^2 \end{bmatrix}, Q = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix}$$

After executing **runlocalization_track('so_pb_10_outlier.txt', 'map_pent_big_10.txt', 1, 1, 1, 2)** on the command window, we obtain the results of the EKF using sequential update algorithm, see Figure 19-Figure 21. Since the odometry information is not included in this case, the mean error is relatively large. From the Figure 19, we could also notice that the red trace falls behind the green trace.

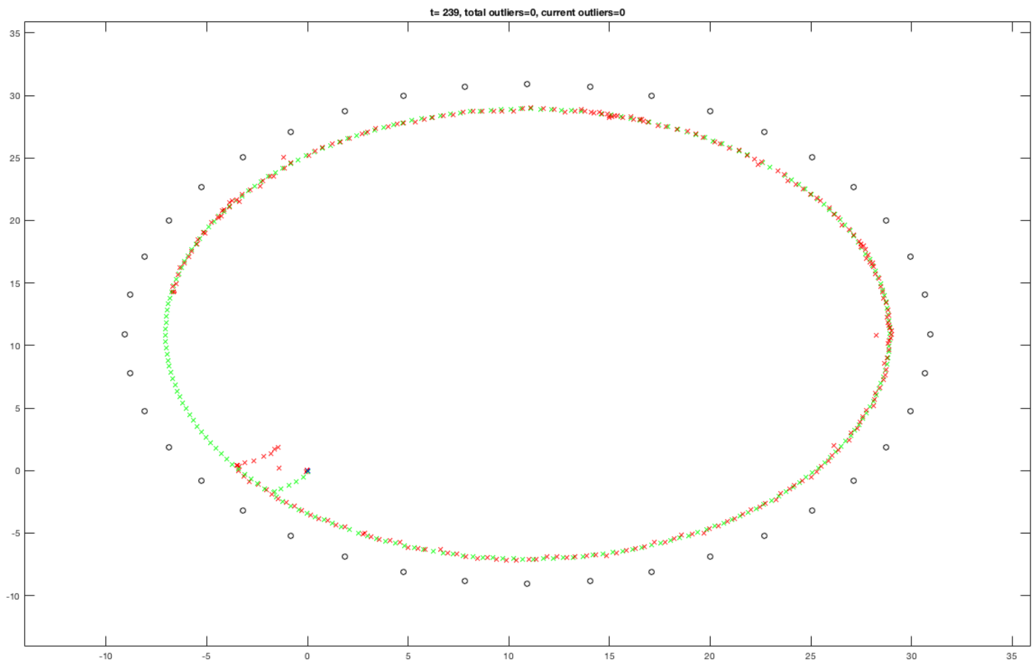

Figure 19: Output of the EKF using sequential update algorithm

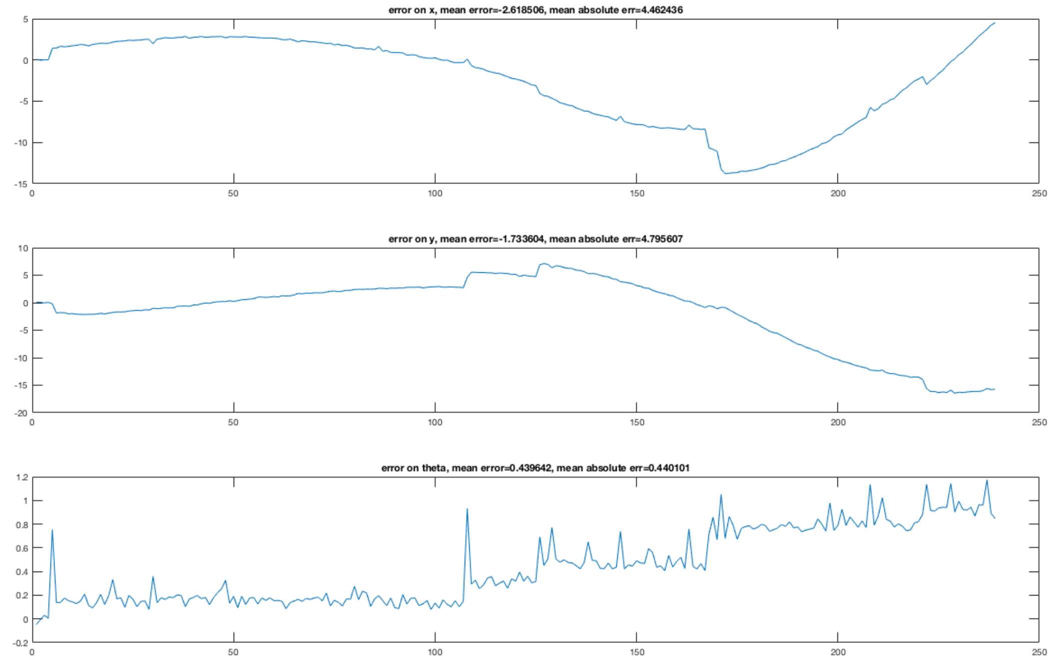

Figure 20: Error information of the estimates

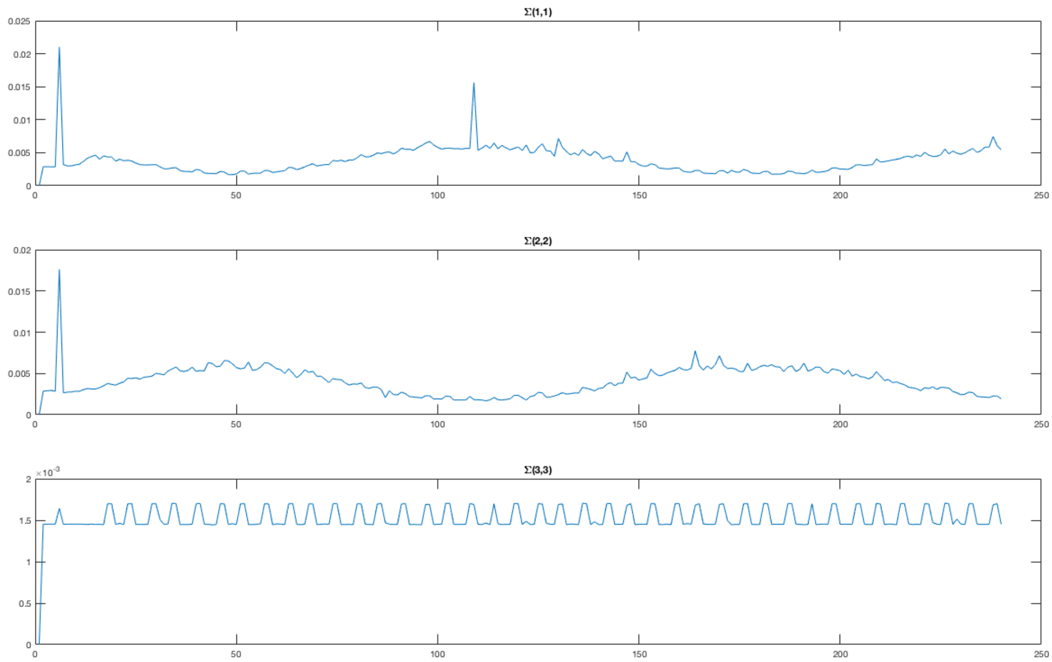

Figure 21: Covariance information of the estimates

The printout information by MATLAB:

mean error(x, y, theta)=(-2.618506, -1.733604, 0.439642)

mean absolute error=(4.462436, 4.795607, 0.440101)

total_time =10.946638

3.2 Batch update algorithm

Different from the sequential update algorithm, the batch update algorithm associates all the measurements with the landmarks at the same time, which proves to be very effective to resist noise. The absolute mean errors are below 0.1, and the program takes less time to obtain the results than that with sequential update algorithm. Figure 22-Figure 24 shows the results for this experiment.

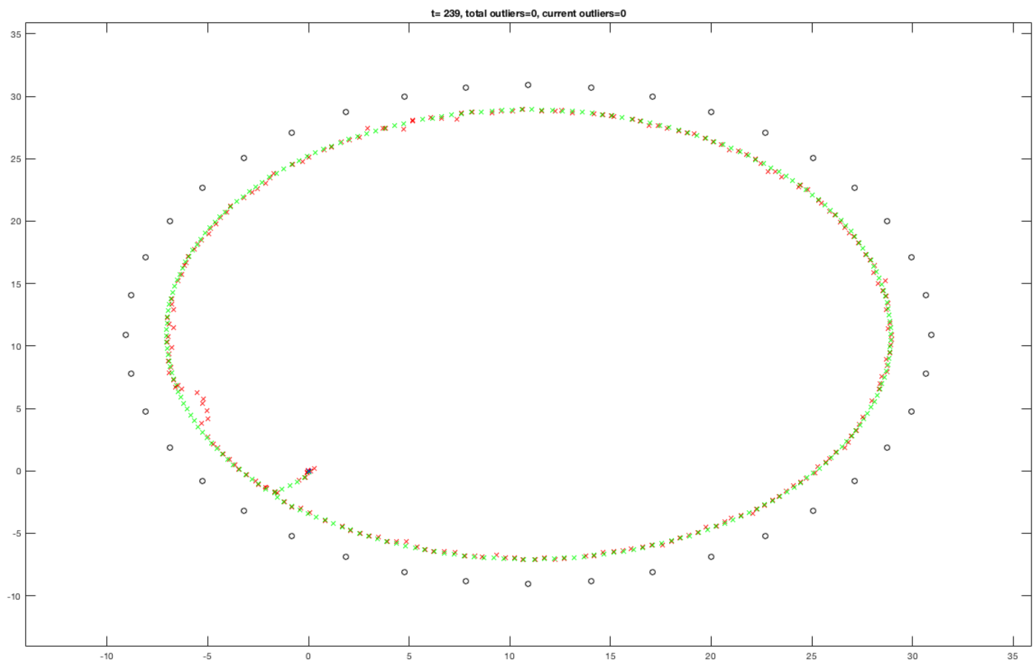

Figure 22: Output of the EKF using batch update algorithm

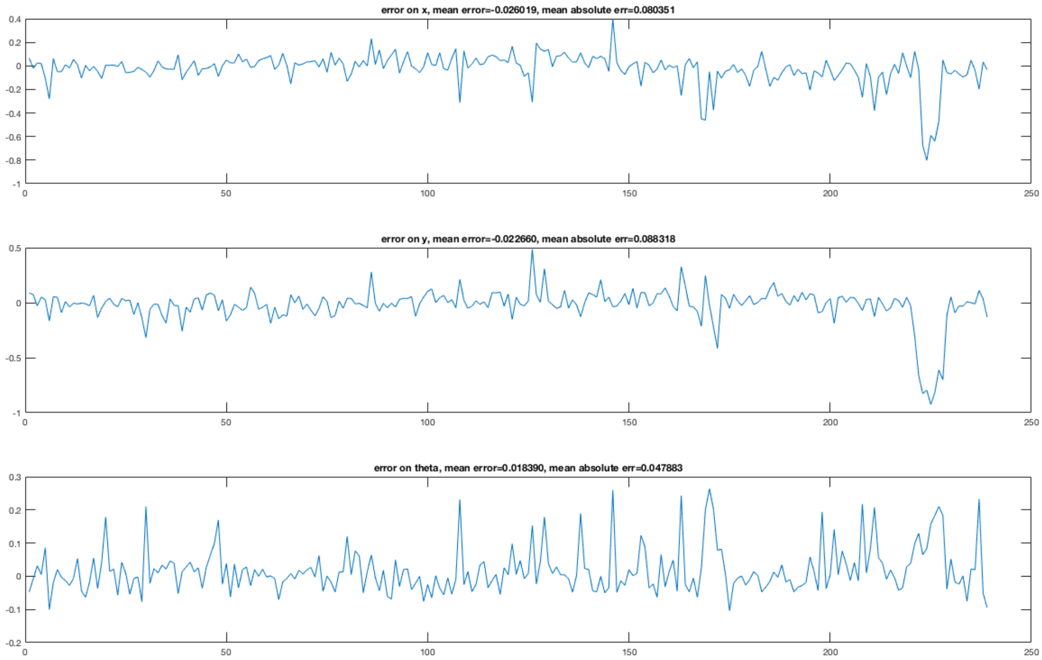

Figure 23: Error information of the estimates

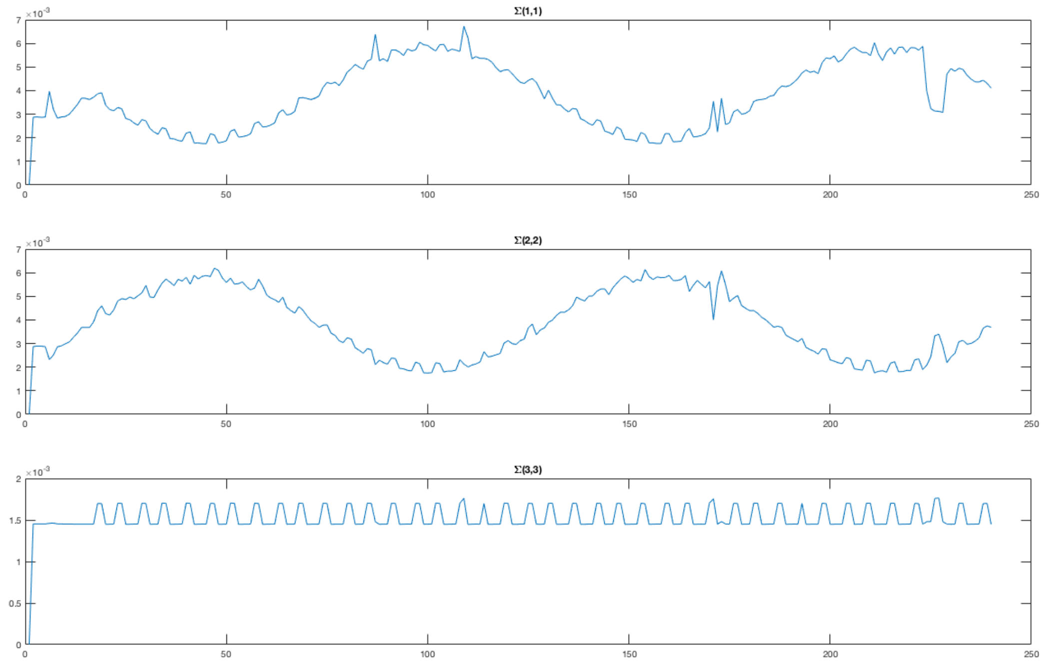

Figure 24: Covariance information of the estimates

The printout information by MATLAB:

mean error(x, y, theta)=(-0.026019, -0.022660, 0.018390)

mean absolute error=(0.080351, 0.088318, 0.047883)

total_time =6.399119