EL2320 Applied Estimation - Lab 1: EKF

Wenjie Yin

yinw@kth.se

1 PART I – Preparatory Questions

1.1 Linear Kalman Filter

1. **What is the difference between a 'control' $\mathbf{u}_t$, a 'measurement' $\mathbf{z}_t$ and the state $\mathbf{x}_t$? Give examples of each?**

   The 'control' $\mathbf{u}_t$: According to the relation: $\mathbf{x}_t = g(\mathbf{u}_t, \mathbf{x}_{t-1}) + \varepsilon_t$, the control is a part of dynamic between $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$, which changes the state from previous state and is used to predict. And ε_t is the noise.

   The 'measurement' $\mathbf{z}_t$: According to the relation: $\mathbf{z}_t = h(\mathbf{x}_t) + \delta_t$, the measurement provides the information about the current state $\mathbf{x}_t$.

   The 'state' $\mathbf{x}_t$: The state is the parameters represent the system, which is depend on both the past 'state' $\mathbf{x}_{t-1}$ and 'control' $\mathbf{u}_t$.

   For example, there is a room with a heater and a thermometer. $\mathbf{u}_t$ can be the heating watts which is produced by a heater to change the temperature of a room. $\mathbf{z}_t$ can be the reading of the thermometer. $\mathbf{x}_t$ can be the real temperature in the room.

   The difference is the 'control' $\mathbf{u}_t$ is used for prediction and the 'measurement' $\mathbf{z}_t$ is used to update.

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

   No!

   According to the update in Kalman filter,

$$K_t = \bar{\Sigma}_t C_t^T \left(C_t \bar{\Sigma}_t C_t^T + Q_t \right)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t \left(z_t - C_t \bar{\mu}_t \right)$$
$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

   Then we can infer

$$\Sigma_t^{-1} = \left(\bar{\Sigma}_t^{-1} + C_t^T Q_t^{-1} C_t \right)$$

As for $C_t^T Q_t^{-1} C_t$ is positive semidefinite, we can infer $\bar{\Sigma}_t^{-1} \geq \bar{\Sigma}_{t-1}^{-1}$, which is equal to $\bar{\Sigma}_t \leq \bar{\Sigma}_{t-1}$. Hence the uncertainty in the belief will not increase during an update.

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

   According to:

$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$$

   We can infer during update is Kalman gain that decides the weighing between measurements and belief. The definition of Kalman gain is:

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

   So the size of Kalman gain is influenced by the predicted covariance $\bar{\Sigma}$ and the covariance of the measurement noise Q_t.

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

   The definition of Kalman gain is:

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

   If the covariance Q is too large, the Kalman gain will be too small. Then it will cause the estimation converges slower. For example, if Q tends to infinity, then Kalman gain tends to zero, which cause the measurement has no influence during update.

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

   Just as what I said in previous question, the covariance Q will influence Kalman gain, and Kalman gain will influence the effect of measurements during the update. So in order to give an increased effect of measurement on the updated state estimate, we need to increase Kalman gain, which means we need to decrease the covariance Q.

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

   According to the equation:

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

   Mostly, $A \geq I$ and $A_t \Sigma_{t-1} A^T$ added with a positive matrix R_t, we can infer $\bar{\Sigma}_t$ is larger than $\bar{\Sigma}_{t-1}$. So the uncertainty during prediction will increase. If $A \leq I$, the uncertainty is not sure, it depends on both A_t and R_t.

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.)**

Because the result of Kalman filter is the minimum of the quadratic function, which means the result is optimal and minimum least square error estimator.

8. **In the case of Gaussian white noise and Gaussian priori distribution, is the Kalman Filter a MLE and/or MAP estimator?**

   The difference between MLE and MAP is MAP has a probability of prior and MLE only has posterior.

$$y_{MAP} = arg\max_{y \in \gamma} P(x|y)P(y)$$
$$y_{MLE} = arg\max_{y \in \gamma} P(x|y)$$

   So if we know a prior distribution and take it into estimation, the Kalman Filter is a MAP estimator, on the contrary, the Kalman Filter is a MLE estimator.

1.2 Extended Kalman Filter

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

   The Kalman filter (KF) is a technique for filtering and prediction in linear Gaussian systems. The extended Kalman filter (EKF) can be applied to non-linear Gaussian systems.

   In EKL:

$$\mathbf{x}_t = \mathbf{g}(\mathbf{u}_t,\ \mathbf{x}_{t-1}) + \varepsilon_t$$
$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \delta_t$$

   Then linearize it:

$$\mathbf{x}_t \approx G_t(x_{t-1} - \mu_{t-1}) + \mathbf{g}(\mathbf{u}_t,\ \mu_{t-1}) + \varepsilon_t$$
$$\mathbf{y}_t = \mathbf{z}_t - \mathbf{h}(\bar{\mu}_t) \approx H_t(\mathbf{x}_t - \bar{\mu}_t) + \delta_t$$

$$G_t \Longleftrightarrow A_t$$
$$\mathbf{g}(\mathbf{u}_t,\ \mu_{t-1}) \Longleftrightarrow A_t\mu_{t-1} + B_t\mathbf{u}_t$$
$$H_t \Longleftrightarrow C_t$$

   Treat this as a linear system then can do Kalman predict and update.

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

    NO!

    Because the distortion of Gaussian by the non-linear transform and the original noise may not be Gaussian to begin with. The EKF may can not converge to the correct valve, diverge or become inconsistent.

    If the a posteriori distribution is incorrectly estimated, the estimator also will be inconsistent.

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

    If the filter often diverge, may caused by too small covariance. we can inject noise by increasing the measurement covariance *Q* to make estimator more stable.

1.3 Localization

12. **If a robot is completely unsure of its location and measures the range r to a know landmark with Gaussain 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.**

    Because the robot completely unsure its location, so can not determine the position (x, y) and the heading θ. The position will be a series of circles whose center's position is *r* to the landmark, and the distribution in the circle is Gaussian distribution. The heading θ is unsure and is a uniform distribution between –180° to 180°.

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

    If the measurement include a bearing, the robot will heading to a specific direction. So the position (x, y) is the same as previous, but the heading θ is also a Gaussian distribution.

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?**

    Because of the good motion estimation, we can determine the distance it moves. But with the uncertainty in heading, so can not determine the orientation. So the result will be a curve.

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 there exists spurious sensor reading, errors in processing the sensor data or poor feature recognition, EKF update will go wrong.

2 PART II – Matlab Exercises

2.1 Warm up problem with Standard Kalman Filter

Question 1:

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ \Delta t \end{bmatrix},\ C = [1 \quad 0]$$

$$x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix},\ u_k = a_0,\ x_{k+1} = Ax_k + Bu_k + \varepsilon_k,\ z_k = Cx_k + \delta_k$$

ε_k is the process noise, δ_k is the measurement noise. According to the above state-space model, we know the dimensions of ε_k is the same as the dimension of x_k. And the dimensions of δ_k is the same as the dimension of z_k. So the dimension of ε_k is 2×1, the dimension of δ_k is 1×1.

In order to uniquely characterize a white noise, we need to define two parameters. When the white noise is scalar, we need to define the mean value μ and the variance σ^2, and the mean value of white zero-mean noise is 0, we can represent it by $\mathcal{N}(\mu, \sigma^2)$.When the noise is a vector, the mean value $\boldsymbol{\mu}$ is a single column matrix, the variance is replaced by a covariance matrix $\boldsymbol{\Sigma}$. As for white is independent of other noise, the covariance matrix is a diagonal matrix, we can represent it by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Question 2:

According to the code and the model, we can make the following table:

Table 1: The Roles and Usages of variables

Variable	Roles/Usages
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
G	An identity matrix used for keeping the matrix dimensionality of process
D	An identity matrix used for keeping the matrix dimensionality of measurement
Q	The variance of the measurement noise
R	The covariance matrix of the process noise
wStdP	The true standard derivation of the position noise
wStdQ	The true standard derivation of the velocity noise
vStd	The true standard derivation of the measurement noise
U	The control signal (acceleration) of the system
PP	A matrix stores the updated covariance matrix over a period

Question 3:

According to the code and the model, we can make the following table:

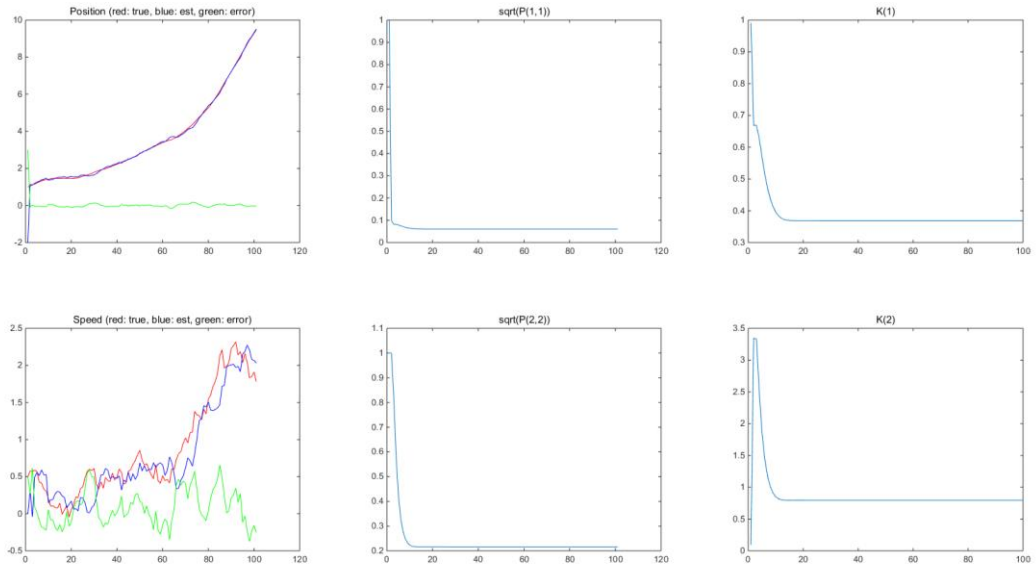

Figure 1: Position, Speed, Estimated error covariance and Kalman gain (Q_default, R_default)

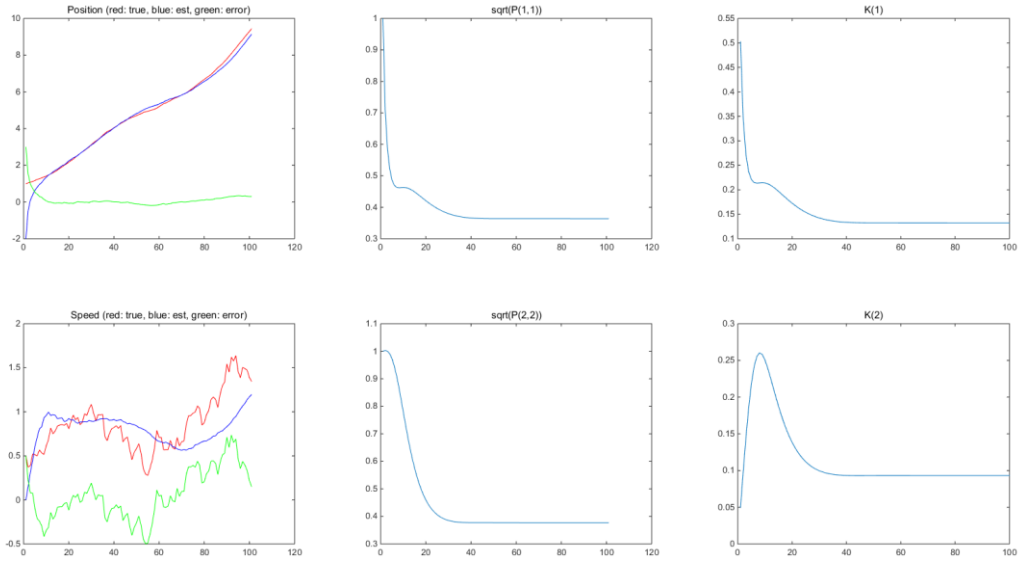

Figure 2: Position, Speed, Estimated error covariance and Kalman gain (Q_×100, R_default)

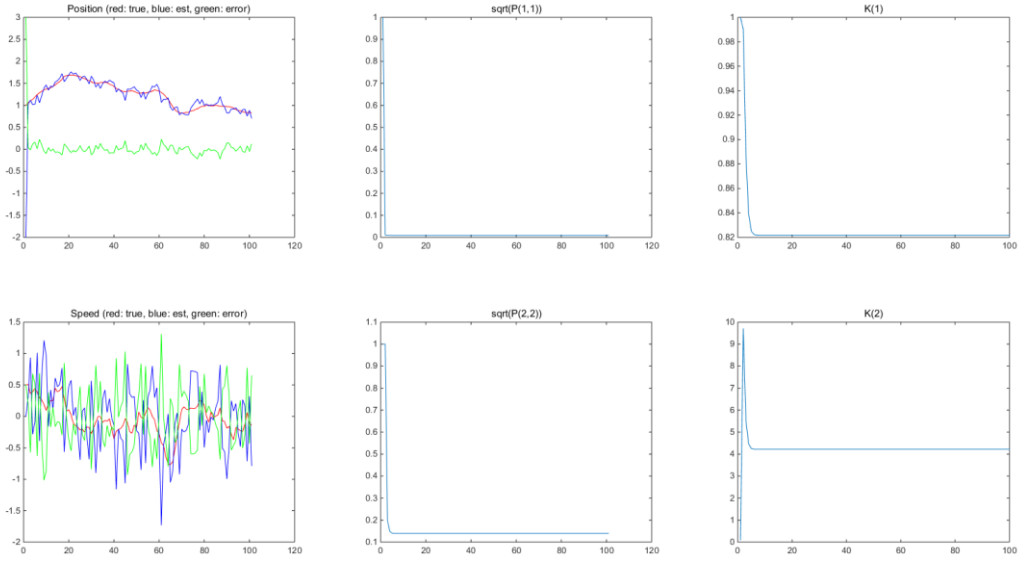

Figure 3: Position, Speed, Estimated error covariance and Kalman gain (Q_/100, R_default)

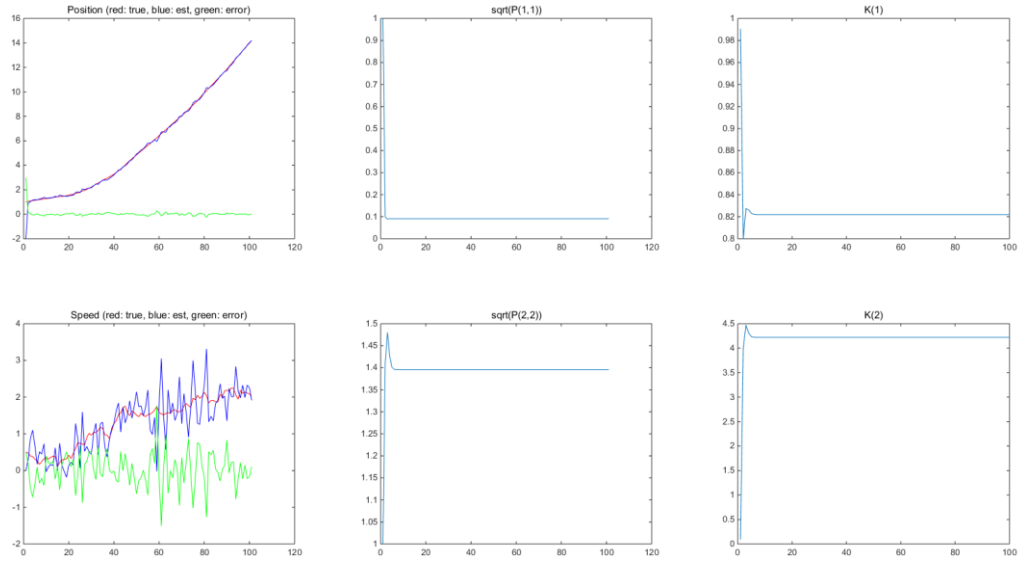

Figure 4: Position, Speed, Estimated error covariance and Kalman gain (Q_default, R_×100)

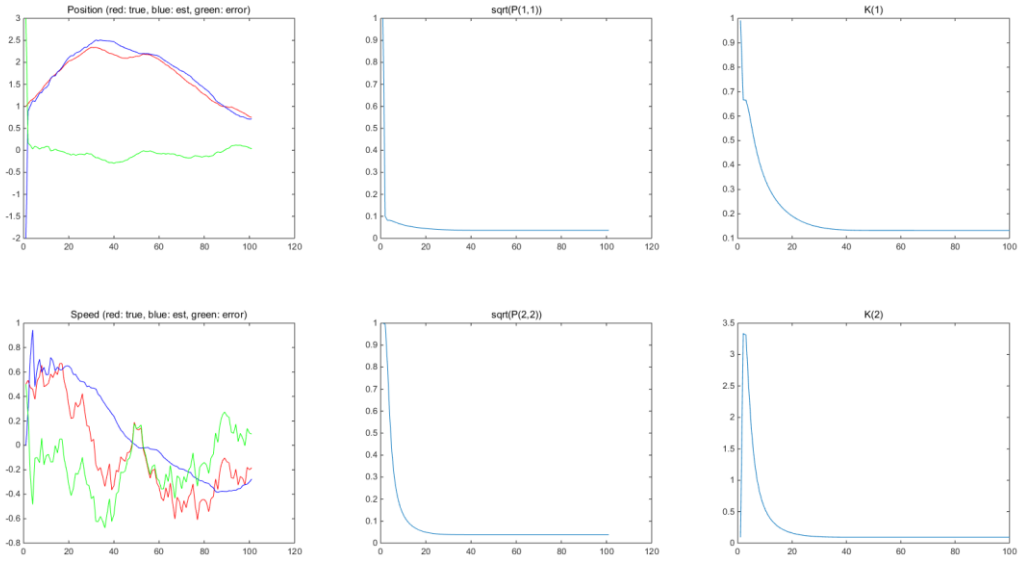

Figure 5: Position, Speed, Estimated error covariance and Kalman gain (Q_default, R_/100)

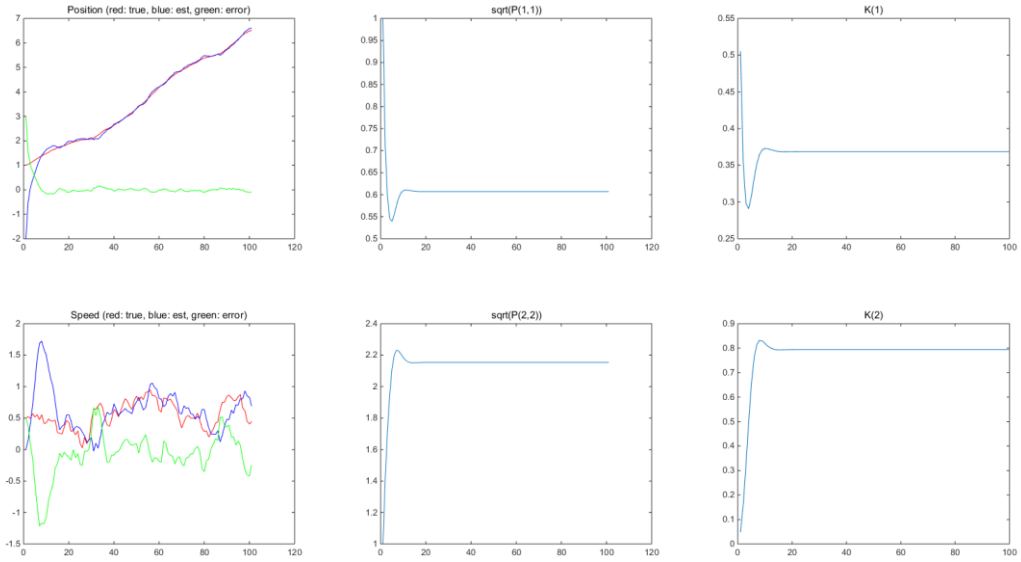

Figure 6: Position, Speed, Estimated error covariance and Kalman gain (Q_×100, R_×100)

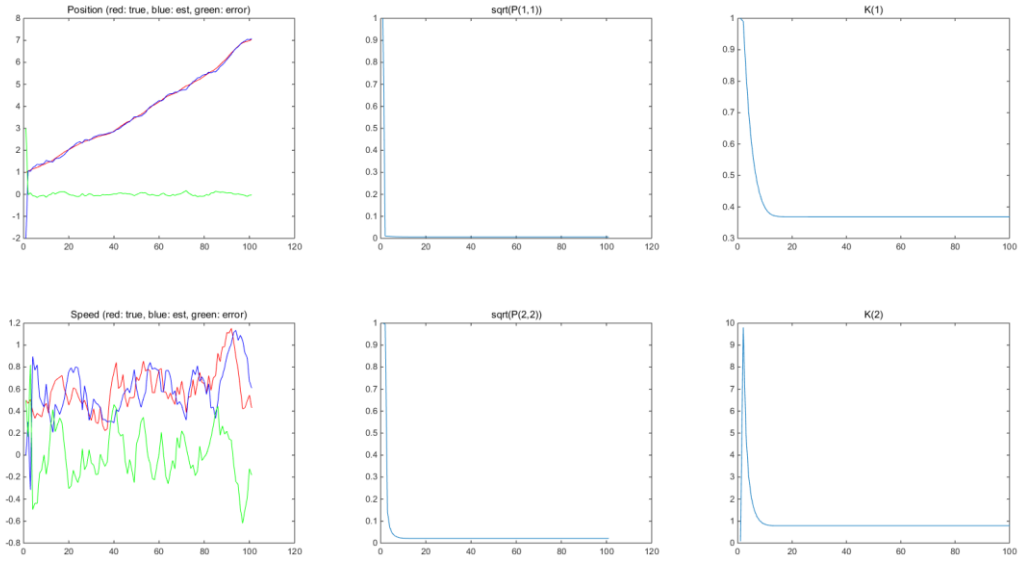

Figure 7: Position, Speed, Estimated error covariance and Kalman gain (Q_/100, R_/100)

We change the covariance matrix of the modeled process noise and measurement noise 100 times. When we increase the covariance matrix of the measurement noise Q, according to the definition:

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

When Q increase, the Kalman filter gain will decrease, which means the belief of measurement will decrease, and it will cause the estimation converges slower. And because of:

$$P(k+1|k+1) = P(k+1|k) - K(k+1)CP(k+1|k)$$

The estimated error covariance will increase for the Kalman gain decrease. We also can conclude these result by comparing Figure 1 and Figure 2 or Figure 4 and Figure 6. When the Q decrease, the results are totally opposite, the Kalman gain will increase, the estimation converges faster and the estimated error covariance will decrease. We can see the results by comparing Figure 1 and Figure 3 or Figure 5 and Figure 7.

When we increase the covariance of the process noise R, according to the definition:

$$P(k+1|k) = APA^T + GRG^T$$

We can deduce the predicted error covariance will increase, and in this way, the covariance of the estimation will increase. And because the belief of estimating true state decrease, the Kalman gain will increase, for the belief of measurement increase. We also can conclude these result by comparing Figure 1 and Figure 4 or Figure 2 and Figure 6. Oppositely, when R decrease, the Kalman gain will decrease, the covariance of predicted error will decrease, and the covariance of estimation will decrease. We can see these results by comparing Figure 1 and Figure 5 or Figure 3 and Figure 7.

When the covariance of measurement and process both increase, according to the definition above, the Kalman gain will decrease the covariance of predicted error will increase, so the covariance of estimation will increase. Oppositely, when the covariance of measurement and process both decrease, the Kalman gain will increase the covariance of predicted error will decrease, so the covariance of estimation will decrease. We can conclude these result by comparing Figure 1, Figure 6 and Figure 7.

Question 4:

According to the code and the model, we can make the following table:

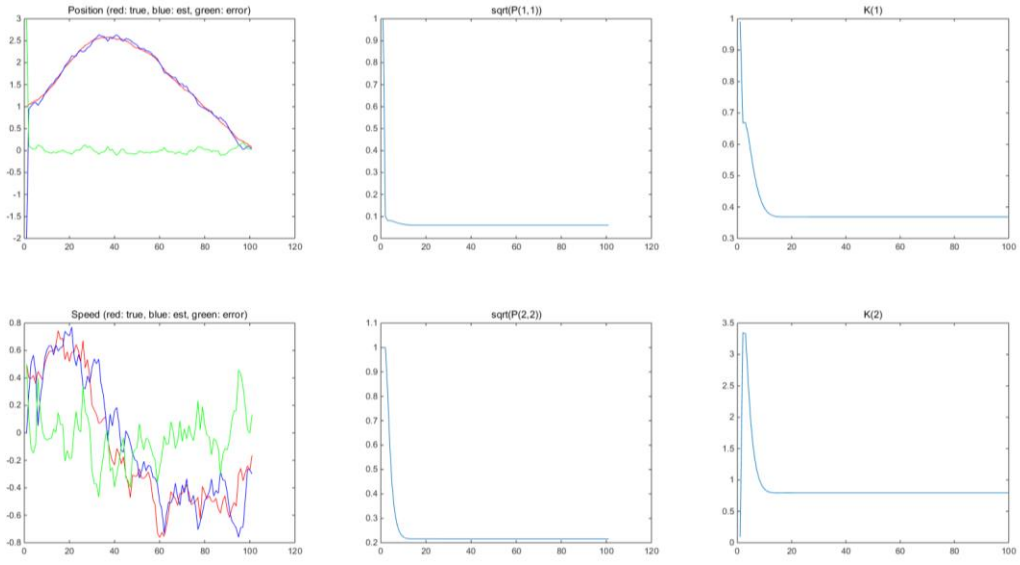

Figure 8: Position, Speed, Estimated error covariance and Kalman gain (P_default, xhat_default)

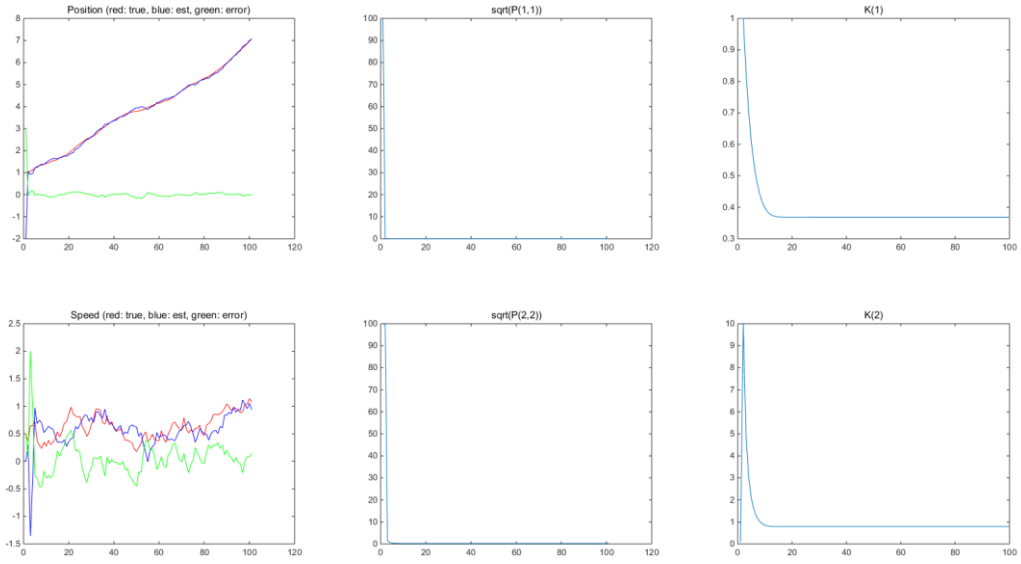

Figure 9: Position, Speed, Estimated error covariance and Kalman gain (P_×10000, xhat_default)

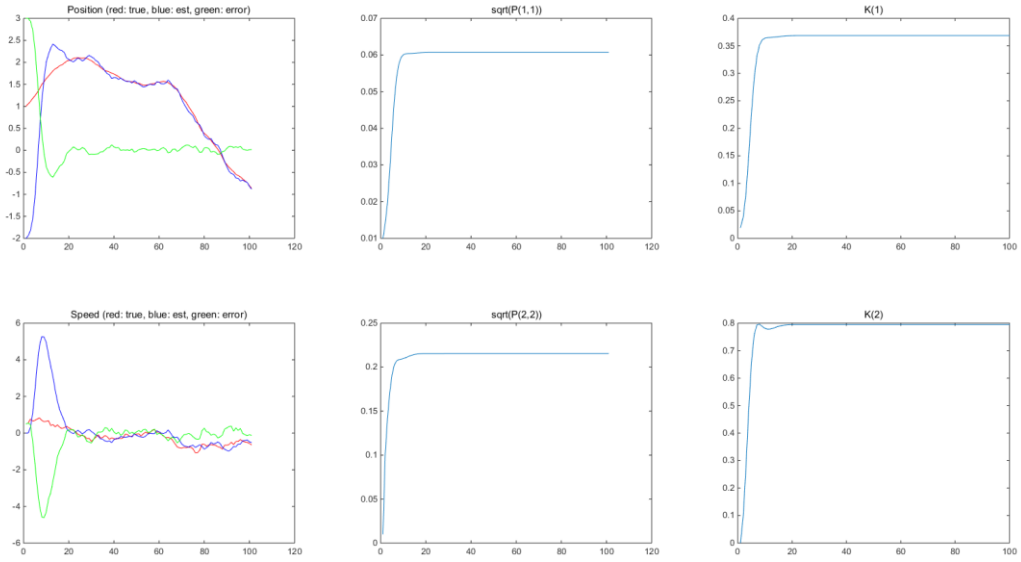

Figure 10: Position, Speed, Estimated error covariance and Kalman gain (P_/10000, xhat_default)

In Figure 8 to Figure 10, P is changed and xhat is the default value. When P is 10000 times, P is very large, which means the true state is uncertainty, and under this condition the Kalman gain will be large because the measurement is reliable relatively. So the rate of convergence is faster

On the contrary, when P is very small, the estimation is more reliable than measurement, so at first, the Kalman gain will very small, and it will rise gradually. So the rate of convergence is slower.

As for the estimate error, according to Figure 8 to 10, the error is approximately in the same level.

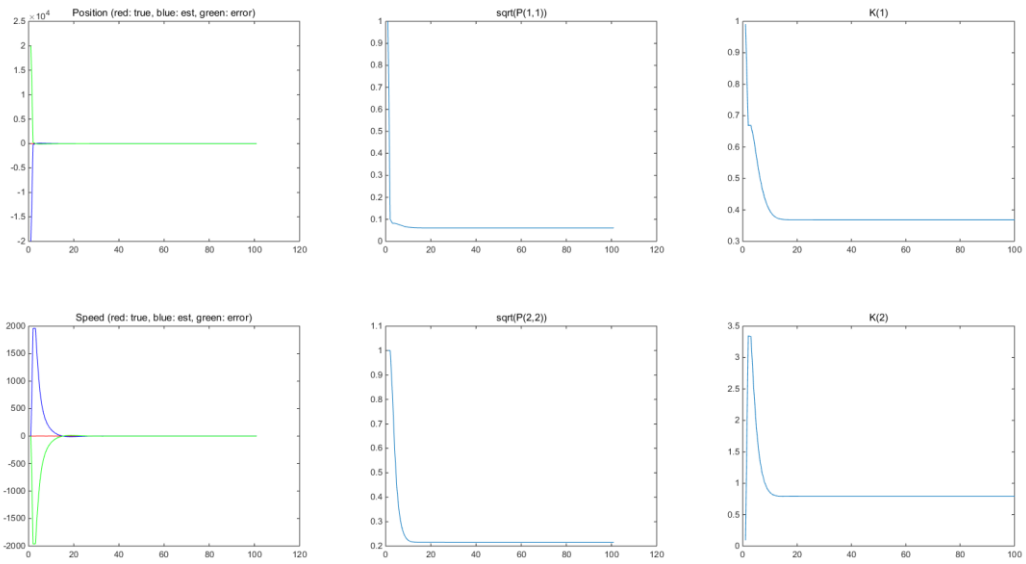

Figure 11: Position, Speed, Estimated error covariance and Kalman gain (P_default, xhat_×10000)

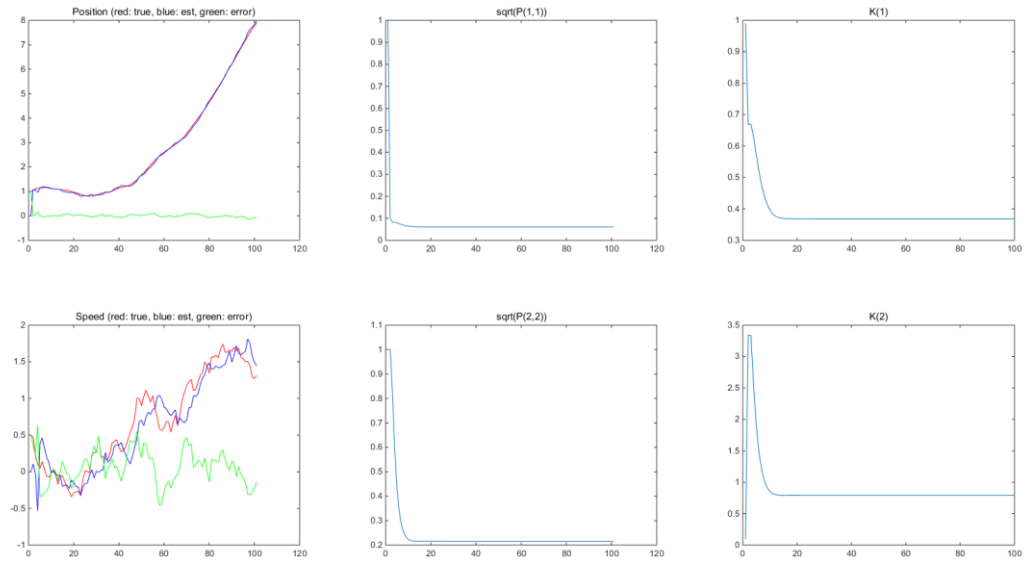

Figure 12: Position, Speed, Estimated error covariance and Kalman gain (P_default, xhat_/10000)

In Figure 11 to Figure 12, P is default and xhat is the changed. When we change the initial estimated value, the influence is very small and can be neglected. Because the change of initial does not influence the Kalman gain or the covariance, so it has no influence to the Kalman filter.

2.2 Main problem: EKF Localization

Question 5:

In the following equation:

$$\begin{cases} bel(\mathrm{x_t}) = p(\mathrm{x_t}|\mathrm{u_{1:t}}, \mathrm{z_{1:t}}, \overline{\mathrm{x}}_0, M) = \eta p(\mathrm{z_t}|\mathrm{x_t}, M)\, \overline{bel}(\mathrm{x_t}) \\ \overline{bel}(\mathrm{x_t}) = p(\mathrm{x_t}|\mathrm{u_{1:t}}, \mathrm{z_{1:t-1}}, \overline{\mathrm{x}}_0, M) = \int p(\mathrm{x_t}|\mathrm{u_t}, \mathrm{x_{t-1}})\, bel(\mathrm{x_{t-1}})\, \mathrm{dx_{t-1}} \\ bel(\mathrm{x_0}) = p(\mathrm{x_0}|\overline{\mathrm{x}}_0) = \delta(\mathrm{x_0} - \overline{\mathrm{x}}_0) \end{cases}$$

The part responsible for prediction is $\overline{bel}(x_t)$, use the belief on previous step. And the part responsible for update is $bel(x_t)$, which also use the measurements z_t.

In the following equation:

$$\begin{cases} p(\mathrm{x_t}|\mathrm{u_{1:t}}, \mathrm{z_{1:t}}, \overline{\mathrm{x}}_0, M) = \eta p(\mathrm{z_t}|\mathrm{x_t}, M) \int p(\mathrm{x_t}|\mathrm{u_t}, \mathrm{x_{t-1}})\, p(\mathrm{x_{t-1}}|\mathrm{u_{1:t-1}}, \mathrm{z_{1:t-1}}, \overline{\mathrm{x}}_0, M)\, \mathrm{dx_{t-1}} \\ p(\mathrm{x_0}|\overline{\mathrm{x}}_0) = \delta(\mathrm{x_0} - \overline{\mathrm{x}}_0) \end{cases}$$

The part responsible for prediction is the integral part:

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

And the part responsible for update is combined with the measurement z_t. So the update part is:

$$\eta p(\mathrm{z_t}|\mathrm{x_t}, M) \int p(\mathrm{x_t}|\mathrm{u_t}, \mathrm{x_{t-1}})\, p(\mathrm{x_{t-1}}|\mathrm{u_{1:t-1}}, \mathrm{z_{1:t-1}}, \overline{\mathrm{x}}_0, M)\, \mathrm{dx_{t-1}}$$

Question 6:

Yes. The measurements are independent of each other is a valid assumption. For the process is a first order Markov process, so the past has no influence on the future. And the measurement noise is white noise, which means these noise is uncorrelated. To be honest, one measurement is correlated with the previous in some degree, but these correlations can be neglected if there are many measurements at time t. If measurements are independent we can simplify the calculation.

Question 7:

The threshold on the Mahalanobis distance follows the inverse chi square cumulative distribution with 2 degrees of freedom. So, the threshold λ_M is given by:

$$\lambda_M = X_2^{-2}(\delta_M)$$

And because δ_M is a probability, so the bounds for δ_M is [0, 1].

Because the inverse chi square cumulative distribution with 2 degrees of freedom χ_2^{-2} is monotonically increasing, so larger δ_M will cause larger λ_M, so the threshold is larger. So, a

larger δ_M will cause less outlier rejections. On the contrary, a smaller δ_M will cause smaller λ_M, then will cause more outlier rejections.

If the measurements of all features in our map are reliable, then the uncertainty is low, which means the probability value δ_M is high. Thus, we can use a high threshold to use more measurements, only throw a few unreasonable data.

On the contrary, if the measurements of scenario is unreliable, arising from clutter or spurious measurements, the uncertainty is high, and the probability will be low. Thus, we should use a threshold with small value to make sure the measurements we use are reliable.

But according to the extra material about outlier, in some real applications the measurements are few and far between, so it really hurts to throw away any data. Thus, we need to consider the cost of both false positives and false negatives and then trade off. We can adjusts the threshold to get an estimator can work on test data.

Question 8:

The sequential update approach has some down-sides when the first noisy measurements, it will cause some problems if get outliers into the system and perform data association for every new measurement processed. Because the noisy measurement will influence the mean value and covariance, the mean value is shifted wrongly, and the covariance decreased incorrectly, which is called inconsistency. As the covariance decreased, then the threshold will increase incorrectly, which will cause some measurements may be incorrectly discarded as outliers. Then all following measurements will be affected. This disadvantage of sequential update can avoid by using batch update for batch update is less sensitive to outliers.

Question 9:

According to the Algorithm 4, we can observe in each observation, we compute $\hat{c}_{t,j}$, $H_{t,j}$ and $S_{t,j}$ for one times. So there exists redundant re-computations. For every landmarks j, we compute z_t times. Totally, we compute these value for $z_t \times M$ times. In fact, we only need to compute once for each landmark.

We can modify this algorithm by compute $\hat{c}_{t,j}$, $H_{t,j}$ and $S_{t,j}$ before the loop. In this way we only compute these value for M times.

Question 10:

In algorithm 4, because z_t is a 2×1 matrix, so each $\bar{v}_t^i$ also is a 2×1 matrix, because $\bar{v}_t = [(\bar{v}_t^1)^T\ (\bar{v}_t^2)^T\ \ldots\ (\bar{v}_t^n)^T\ ]^T$, so the dimensions of $\bar{v}_t$ is $2\mathrm{n} \times 1$.

As for $\bar{H}_{t,i}$ is a 2×3 matrix, then according to $\bar{H}_t = [(\bar{H}_{t,1})^T\ (\bar{H}_{t,2})^T\ \ldots\ (\bar{H}_{t,n})^T\ ]^T$, we know the dimensions of $\bar{H}_t$ is $2\mathrm{n} \times 3$.

In the sequential update algorithm, the dimensions of $\bar{v}_t$ is 2×1, and the dimensions of $\bar{H}_t$ is 2×3.

This is because in batch algorithm we associate all observations with landmarks simultaneously

and then performs one update for all the observations in each time step. However, in the sequential algorithm we update each observation separately. According to the dimensions we the complexity of batch algorithms is higher than sequential algorithms.

2.3 Data sets

1. map_o3.txt + so_o3_ie.txt

Because the laser scanner has an accuracy of (1 cm, 1 degree), and the odometry information has an un-modeled noise of approximately 1cm and 1 degree per time step, so we set R and Q like this:

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

Then the results are shown in Figure 13 to Figure 15.

The printout is:

```
10 th measurement was labeled as outlier, t=89
mean error(x, y, theta)=(0.000237, 0.000209, 0.000498)
mean absolute error=(0.003173, 0.003925, 0.002562)
total_time =21.392657
```

The mean absolute error of the estimation is less than 0.01 on all dimensions, so the results are good.

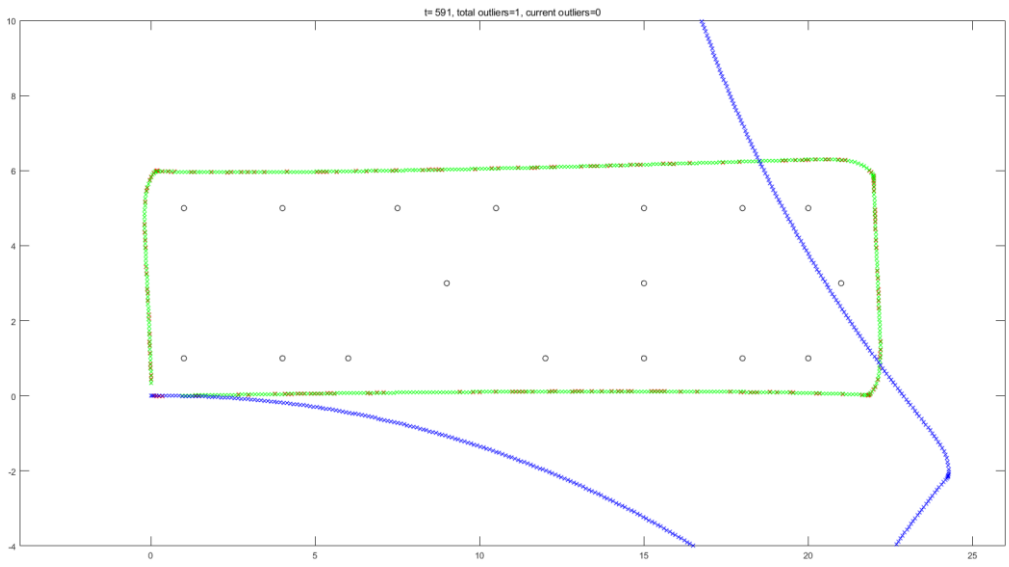

Figure 13: output map of the first dataset

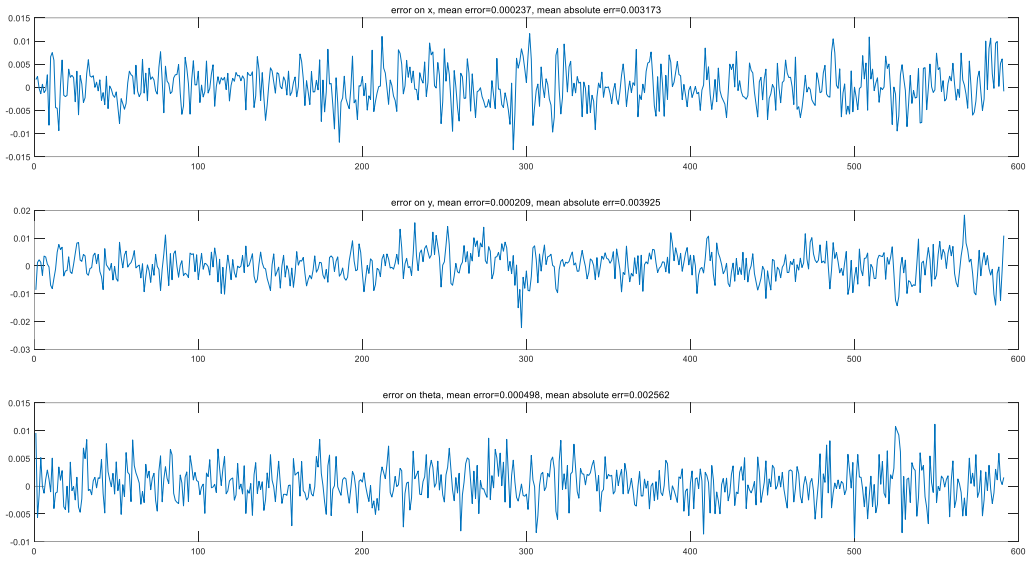

Figure 14: mean absolute error of the first dataset

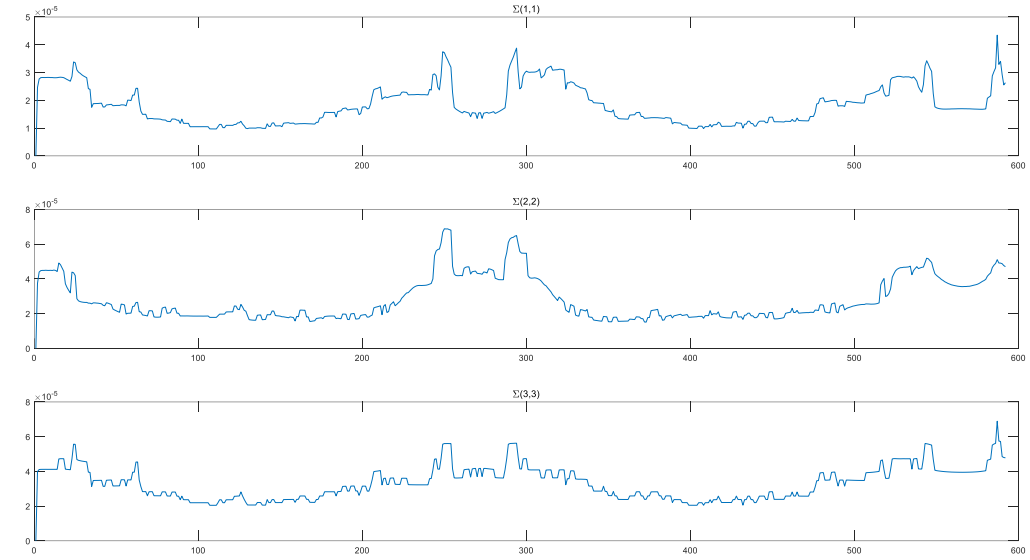

Figure 15: covariance of the first dataset

2. map_pent_big_10.txt + so_pb_10_outlier.txt

In this dataset, the measurements have undergone a white Gaussian with the standard deviation of 0.1(m, rad), so there are certain outliers need to be detected. Here we set R and Q as:

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

Then the results are shown in Figure 16 to Figure 18. In Figure 16 we can see how noise affects the measurements. In this dataset, the measurements have white noise with large error, so we need to increase the uncertainty of the covariance of measurements noise, or the results will diverge.

The printout is:

```
mean error(x, y, theta)=(0.007164, 0.004776, -0.012892)
mean absolute error=(0.041994, 0.042097, 0.044507)
total_time =168.418984
```

So the mean absolute error of the estimation is less than 0.06 on all dimensions.

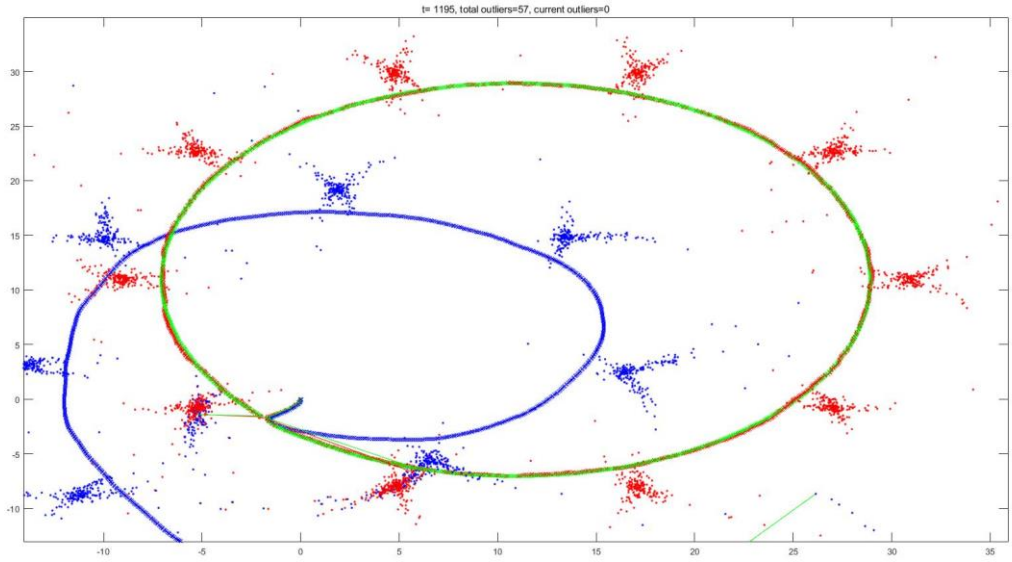

Figure 16: output map of the second dataset

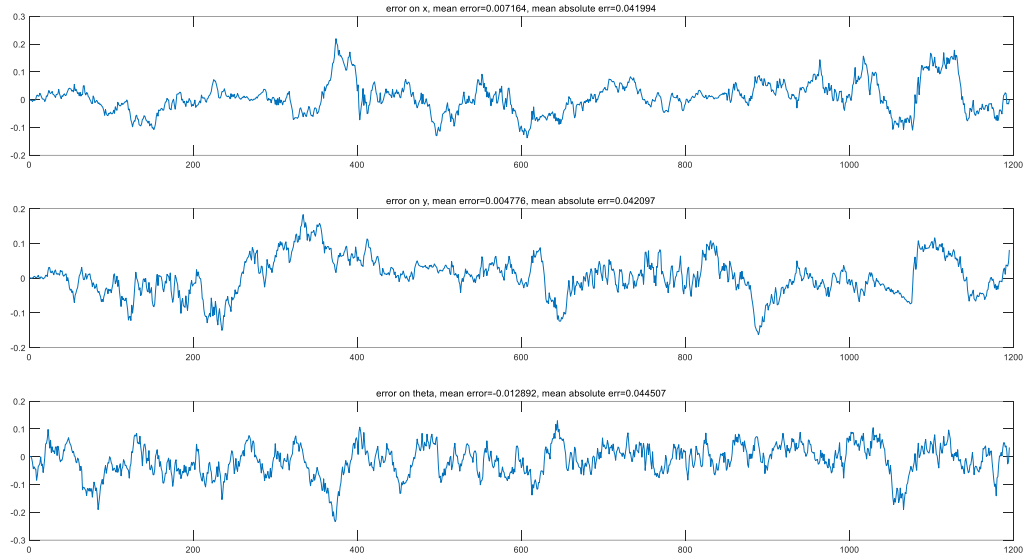

Figure 17: mean absolute error of the second dataset

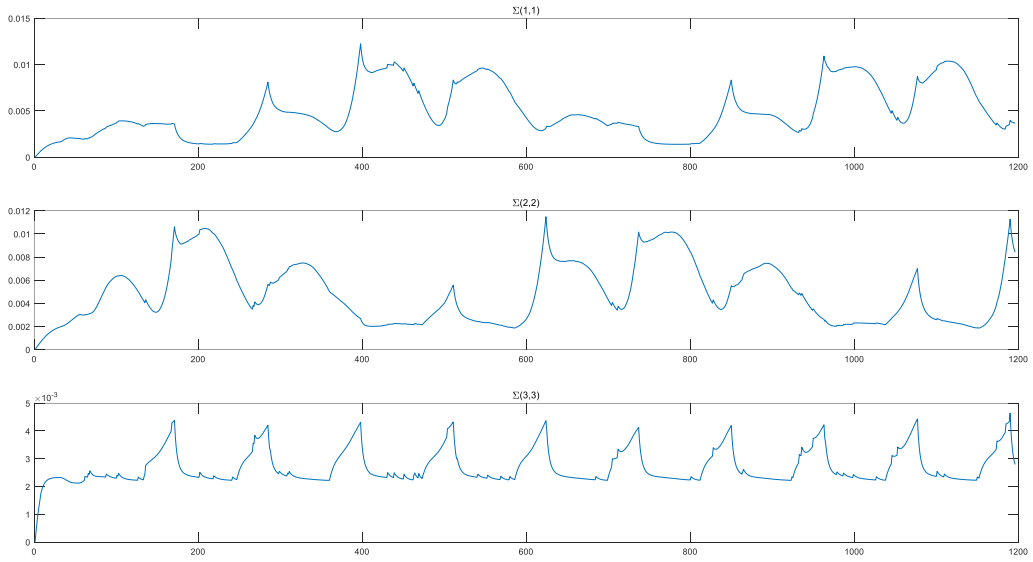

Figure 18: covariance of the second dataset

3. map_pent_big_40.txt + so_pb_40_no.txt

In this dataset, we don't have any odometry information, thus we compensate it by considering a big standard deviation standard deviation for the process noise of 1(m, m, rad) and model the measurement noise with a standard deviation of 0.1(m, rad). So the Q and R are shown below:

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

After running the sequential update algorithm on the dataset, the results are shown in Figure 19 to Figure 21.

The printout is:

```
mean error(x, y, theta)=(-2.618506, -1.733604, 0.439642)
mean absolute error=(4.462436, 4.795607, 0.440101)
total_time =8.396633
```

Because this dataset does not include any odometry information and the outliers detection is disabled, the mean error is large, and cannot estimate the ground truth well enough.

After running the batch update algorithm on the dataset, the results are shown in Figure 22 to Figure 24.

The printout is:

```
mean error(x, y, theta)=(-0.026019, -0.022660, 0.018390)
mean absolute error=(0.080351, 0.088318, 0.047883)
total_time =7.443440
```

Because the batch update algorithm associates all observations with landmarks simultaneously and performs one update for all the observations in each time step, so the result is better. From the results we can see the mean absolute error of the batch update algorithm being less than 0.1 on all dimensions.

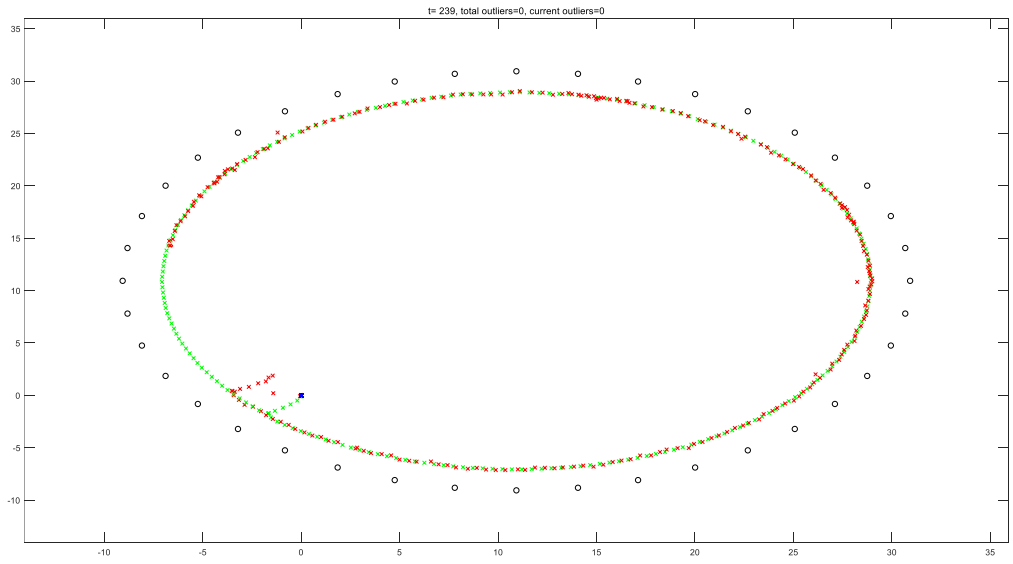

Figure 19: output map of the third dataset using sequential update.

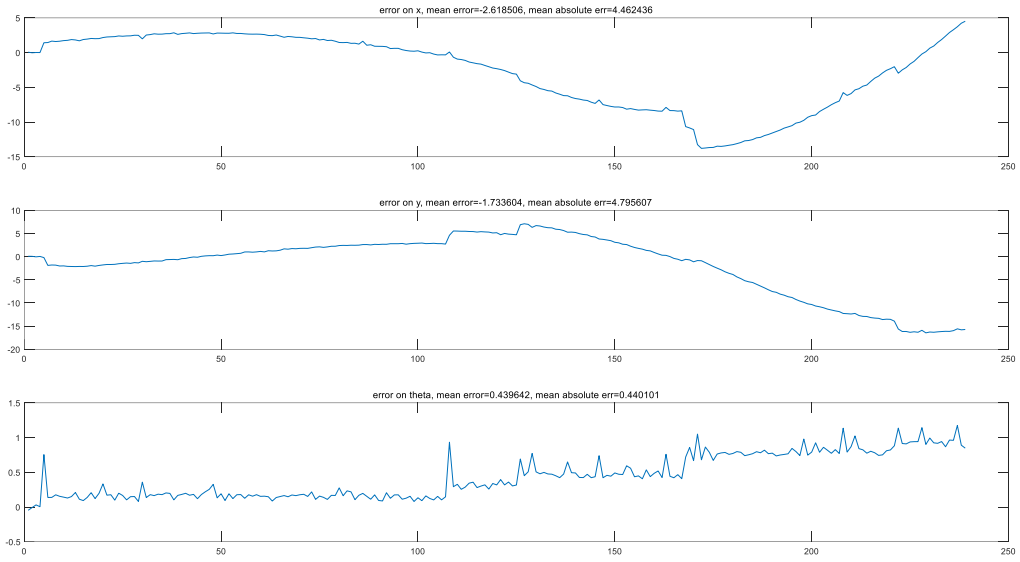

Figure 20: mean absolute error of the third dataset using sequential update.

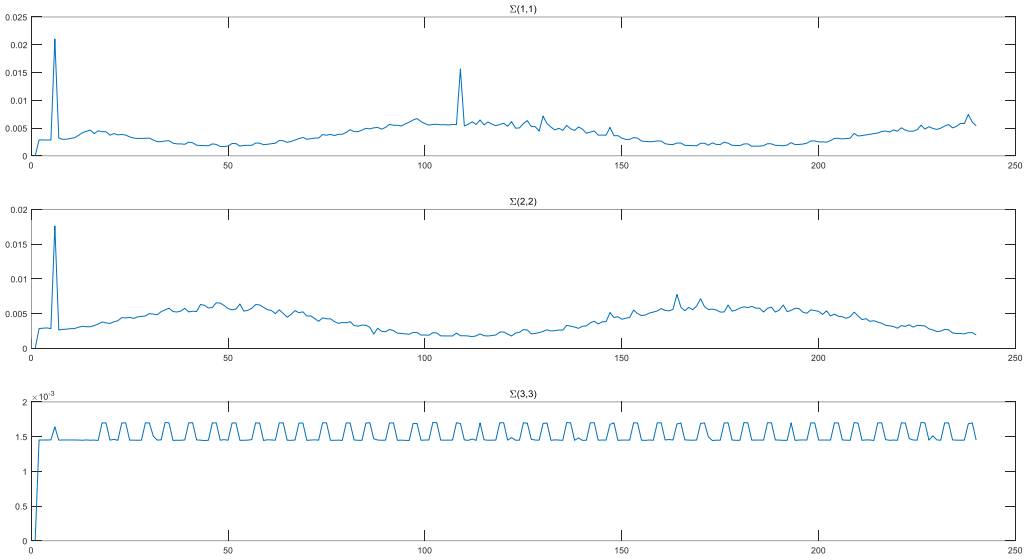

Figure 21: covariance of the third dataset using sequential update.

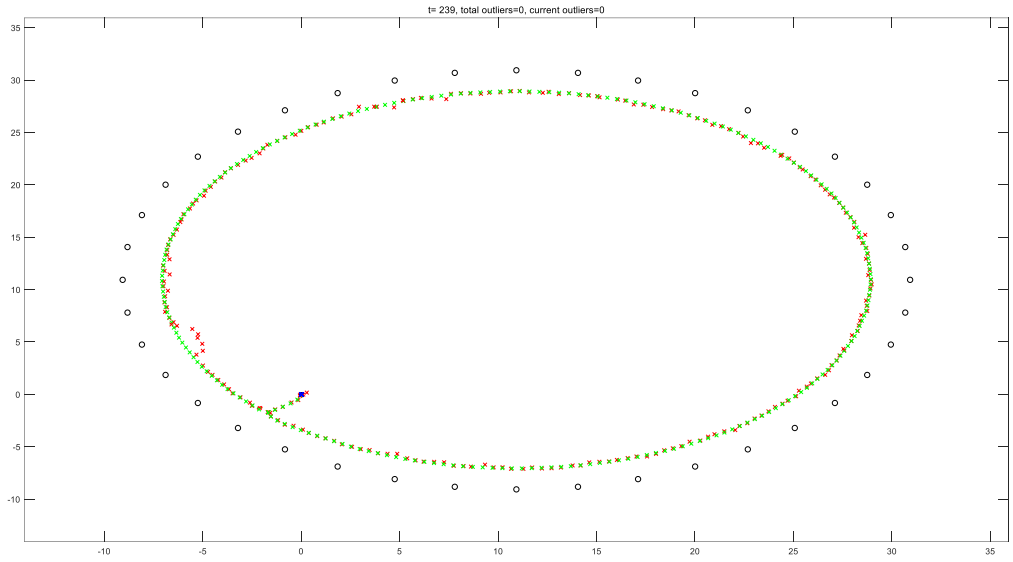

Figure 22: output map of the third dataset using batch update.

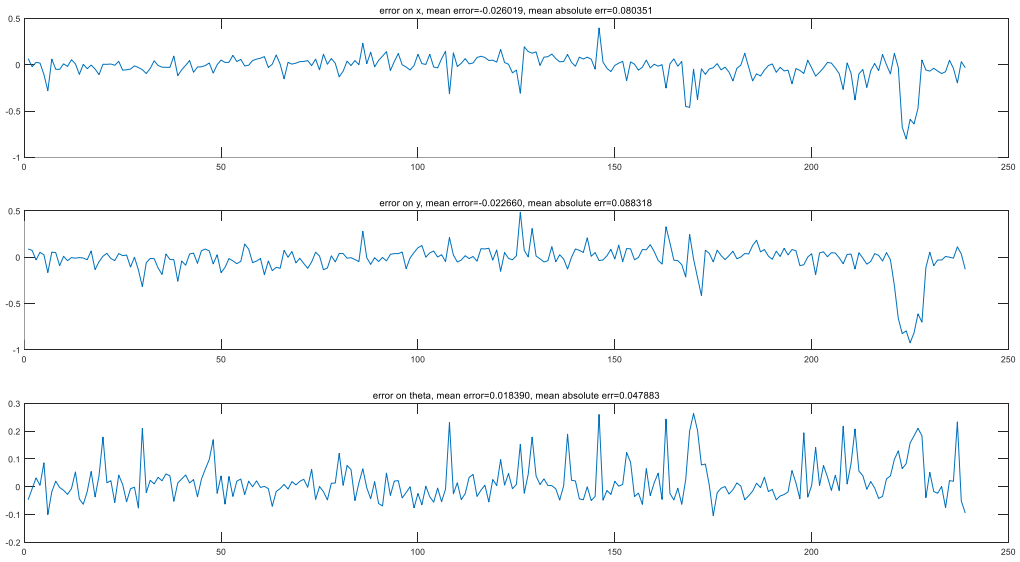

Figure 23: mean absolute error of the third dataset using batch update.

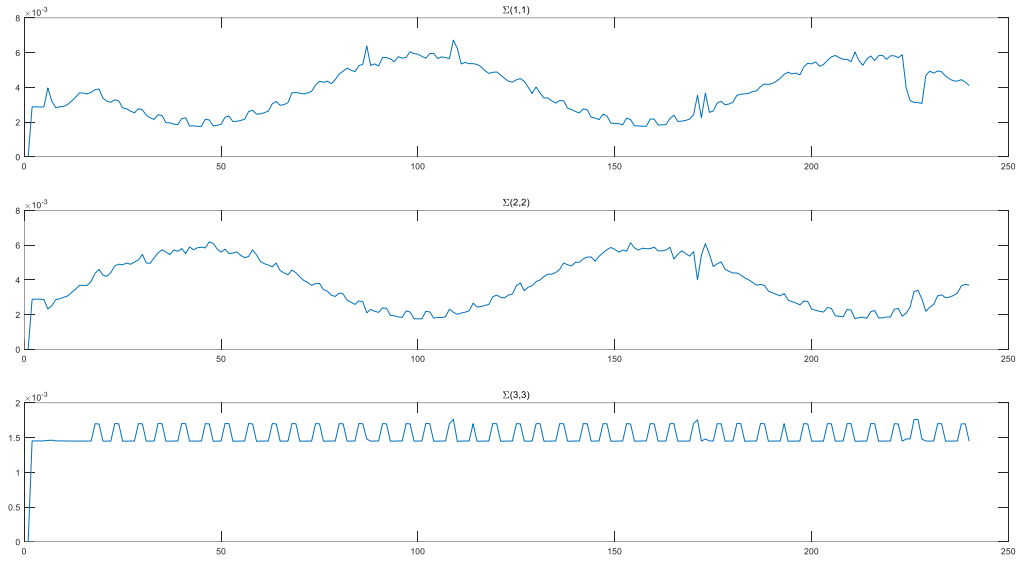

Figure 24: covariance of the third dataset using batch update.