EL2320 Applied Estimation Lab 1

Huijie Wang, huijiew@kth.se

November 2016

2 Preparitory 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?*

   A state x represents the variables of robot (e.g. pose, velocity, etc.) as well as its environment (e.g. location and features of surrounding objects). In probabilistic model x_t is modeled depending on u_t and x_{t-1}, namely $x_t = g(u_t, x_{t-1}) + \epsilon_t$.
   A measurement is the result of sensors obtaining information about the state of its environment. For example, camera images and a range scan. z_t is modeled depending on x_t.
   A control is a setting that change the state. A typical example is velocity of a robot. u_t is one of the dependent factors to x_t.

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

   No. According to the correction step in KF:

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

   As $(Q_t)^{-1}$ is inverse of the covariance matrix of measurement, $C_t{}^T Q_t{}^{-1} C_t$ is postive semidefinite, which means that it is always larger or equal to zero. As a result, the inverse of $\Sigma_t{}^{-1}$ can never decrease, so that uncertainty of belief denoted by Σ_t can never increase.

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

   Kalman gain $K_t = \Sigma_t C_t{}^T Q_t^{-1}$ is calculated to define the decides the weighing between measurements and belief. In which Q_t gives large influence on this term.

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

   In the term of **Kalman gain**,

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

   , a too large covariance leads to a very small K_t. Meanwhile, K_t has a large influence in the following correction step in Kalman filter,

   $$\mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t)$$

   $$\Sigma = (I - K_t C_t)\overline{\Sigma}_t$$

   , as a result, a very small K_t will lead to a very small update of μ and Σ.

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

   Updated state estimate is effected by uncertainty of a measurement. A lower uncertainty will lead to a faster update of states. The uncertainty of a measurement is encoded by covariance Q_t. In the steps of updates, as stated in Question 4, a smaller Q_t can decrease the uncertainty of measurement, which leads to a larger **Kalman Gain**.

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

   Blief is represented by μ_t and Σ_t, in which uncertainty is encoded in Σ_t. During perdiction, $\overline{\Sigma}_t = A_t \Sigma_{t-1} {A_t}^T + R_t$. If covariance of noise, i.e. R_t is large enough to cancel out the scale effect of A, the uncertainty will increase.

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.)*

   The state is represented by $x_t = A_t x_{t-1} + B_t u_t + \epsilon_t$, which is a Gaussian distribution with $\mu_t = A_t x_{t-1} + B_t u_t$ and covariance $\Sigma_t = R_t$. As covariance is depend on noise, the estimation is only depend on μ_t.

   Assume that we can find a better estimation μ', thus, applying least square error estimator method:

   $$\int (x-\mu')^2 G(\mu_t, \Sigma_t)dx = \int x^2 G(\mu_t, \Sigma_t)dx - 2\mu' \int xG(\mu_t, \Sigma_t)dx + \mu'^2 \int G(\mu_t, \Sigma_t)dx \tag{1}$$

   In order to find μ' with minimum error, we need to calculate the derivative of equation (1) with respect to μ'. It is obvious that the first term is a constant to μ', and in the second term, $\int xG(\mu_t, \Sigma_t)dx$ is the expectation of x, which equals μ_t, and in the third term, $\int G(\mu_t, \Sigma_t)dx = 1$. So the equation can now be represented by $-2\mu'\mu_t + \mu'^2$. Taking the first derivative and assigned 0: $-2\mu_t + 2\mu' = 0$. So $\mu' = \mu_t$, we can say it is optimal.

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

   The Kalman Filter is a MAP estimator, since during the iteration, we always gives a prior belief $bel(x_{t-1})$. However, if we give a non-informative prior initially, it is almost the same as MLE (only in the initial step).

Extended Kalman Filter

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

   In Kalman filter we assume that state probability and measurement probability must be linear functions in its arguments with added Gaussian noise. Extended Kalman filter extends the application to non-linear problems:

   $$x_t = g(u_t, x_{t-1}) + \epsilon_t$$
   $$z_t = h(x_t) + \delta_t$$

   As a result, A_t is replaced by G_t-Jacovian of g with respect to the x_{t-1} at μ_{t-1} and C_t is replaced by H_t-Jacovian of h evaluated at $\overline{\mu}_t$.

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

    No. The underlying Riccati equation are not guaranteed to be positive definite [1]. It's convergence is mainly depend on its significance of nonlinearity.

[1] Wikipidea, Extended Kalman filter, https://en.wikipedia.org/wiki/Extended_Kalman_filter

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

    The model is risked given a wrong initial estimate of the state or a process model. This is controlled by noise terms, with covariance Q_t and R_t. So a more uncertain estimation can take less risk at the cost of lower convergence. For example, a larger Q_t leads to a smaller K_t so that the update can be slower but less risk.

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.*

    Since the only known information is the range r to a know landmark with Gaussian noise, and the direction with respect to the landmark is unknown, thus, the location of the robot is a circle with radius r and center at the landmark. And after adding Gaussian noise it becomes ring. And θ is unknown with uniform distribution.

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

    It would look the same. The only difference is that if a bearing is included, the heading and the angle of the circle would be correlated.

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?*

    As it starts with an uncertain initial heading, with a good motion, it would be certain about the distance with respect to the initial point. However, the direction is uncertain because of the uncertainty of heading. As a result, the posterior will be a 'C' shape, with central angle heading θ.

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

    It can go wrong due to its non-linearity. In the process of Jacoboian linearization, the update direction can become straight line instead of curve. As a result, Gaussian is not appropriate to describe.

3 PART II - Matlab Excercises

3.1 Warm up problem with Standard Kalman Filter

1. *What are the dimensions of ϵ_k and δ_k? What parameters do you need to define in order to uniquely characterize a white Gaussian?*

    According to the function that $x_{k+1} = Ax_k + B\mu_k + \epsilon_k$, the dimension of ϵ_k is the same as x_{k+1}, that is (2×1). Similarly, the dimension of δ_k is same as $z_k = Cx_k + \delta_k$, where dimension of C is (1×2) and dimension of x_k is (2×1). So the dimension of δ_k is (1×1).

    A white Gaussian is defined as $N(0, C)$, which is a Gaussian model with zero-mean and Covariance C. So in order to uniquely characterize a white Gaussian, parameter C is needed.

2. *Make a table showing the roles/usages of the variables(x, xhat, P, G, D, Q, R, wStdP, wStdV, vStd, u, PP). To do this one must go beyond simply reading the comments in the code to seeing how the variable is used. (hint some of these are our estimation model and some are for simulating the car motion).*

Variables	Roles/Usage
x	A two dimensional vector recording the true state of position and velocity.
xhat	Indicates the estimated state before and after incorporating measurement y, namely $\overline{\mu}_t$ and μ_t respectively.
P	Covariance matrix describing uncertainty of estimated state before and after incorporating measurement y, namely $\overline{\Sigma}_t$ and Σ_t respectively.
G	Scalar matrix
D	Scalar factor
Q	The covariance of the measurement noise. In this case is also the standard deviation as Q is one dimensional.
R	Covariance of process noise. Describe the randomness of estimated state before incorporating measurement.
wStdP	The standard deviation of noise in the first dimension of state—simulated position.
wStdV	The standard deviation of noise in the first dimension of state—simulated velocity.
vStd	The standard derivation of noise of simulated measurement.
u	Control signal. Here is always 0.
PP	Estimated covariance after incorporating measurement y_k over time.

3. *Please answer this question with one paragraph of text that summarizes broadly what you learn/deduce from changing the parameters in the code as described below. Chose two illustrative sets of plots to include as demonstration. What do you expect if you increase/decrease the covariance matrix of the modeled (not the actual simulated) process noise/measurement noise 100 times(one change in the default parameters each time) for the same underlying system? Characterize your expectations. Confirm your expectations using the code (save the corresponding figures so you can analyze them in your report). Do the same analysis for the case of increasing/decreasing both parameters by the same factor at the same time. (Hint: It is the mean and covariance behavior over time that we are asking about.)*

   Figure 1, 2 and 3 shows the result when we increase/decrease process noise and measurement noise.

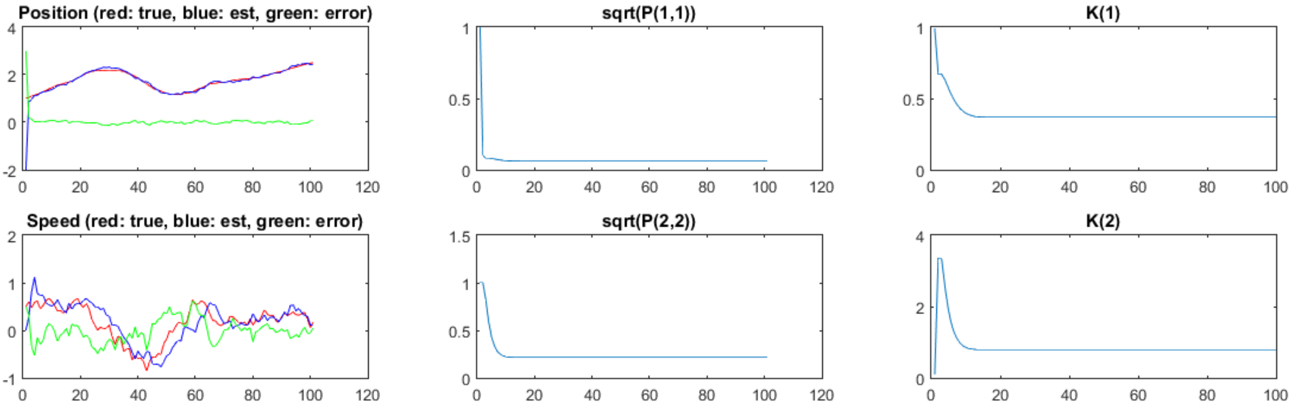
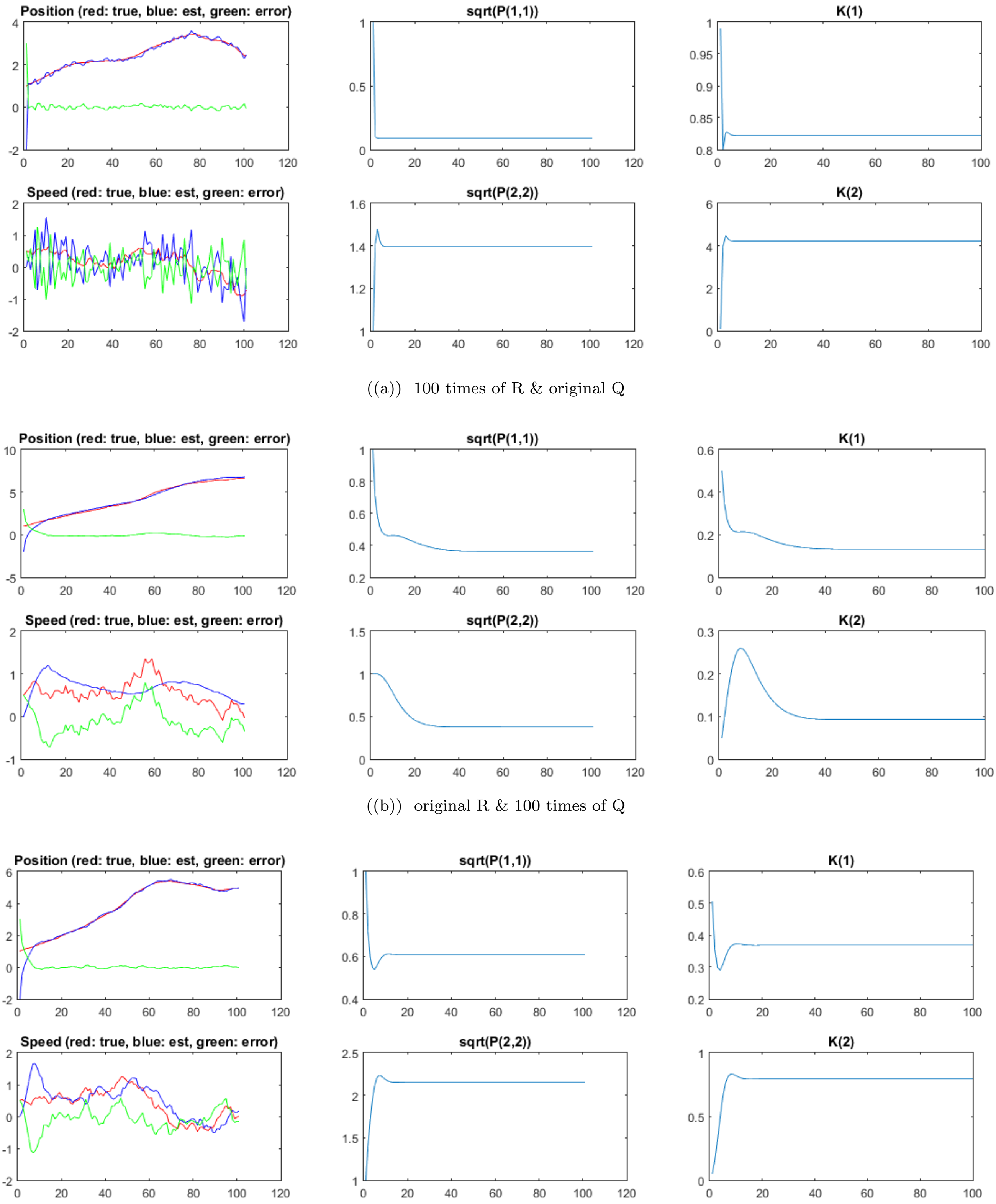


Figure 1: Simulation with riginal process noise R & measurement noise Q

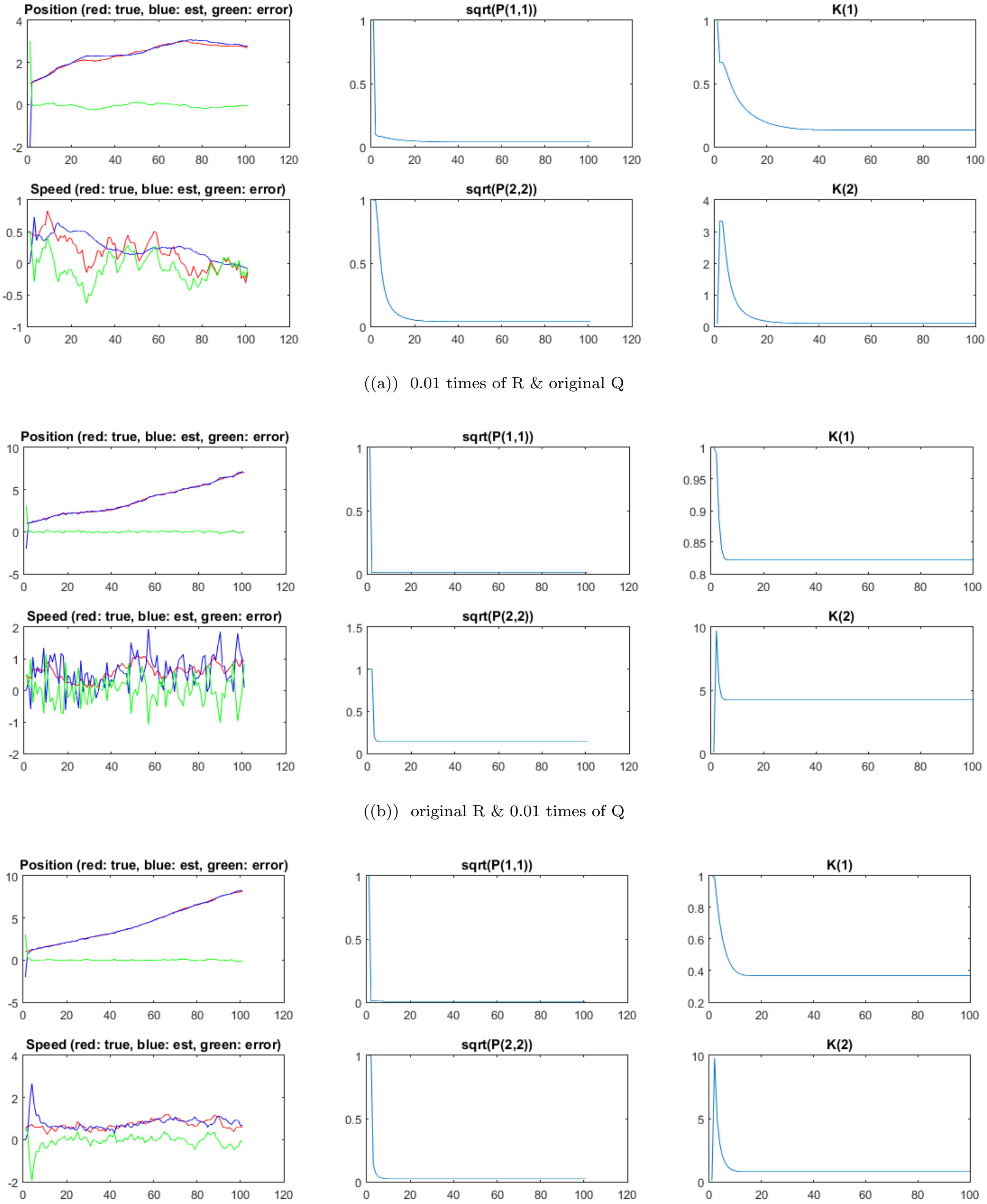


((a)) 100 times of R & original Q

((b)) original R & 100 times of Q

((c)) 100 times of R & 100 times of Q

Figure 2: Increasing of noises



((a)) 0.01 times of R & original Q

((b)) original R & 0.01 times of Q

((c)) 0.01 times of R & 0.01 times of Q

Figure 3: Decreasing of noises

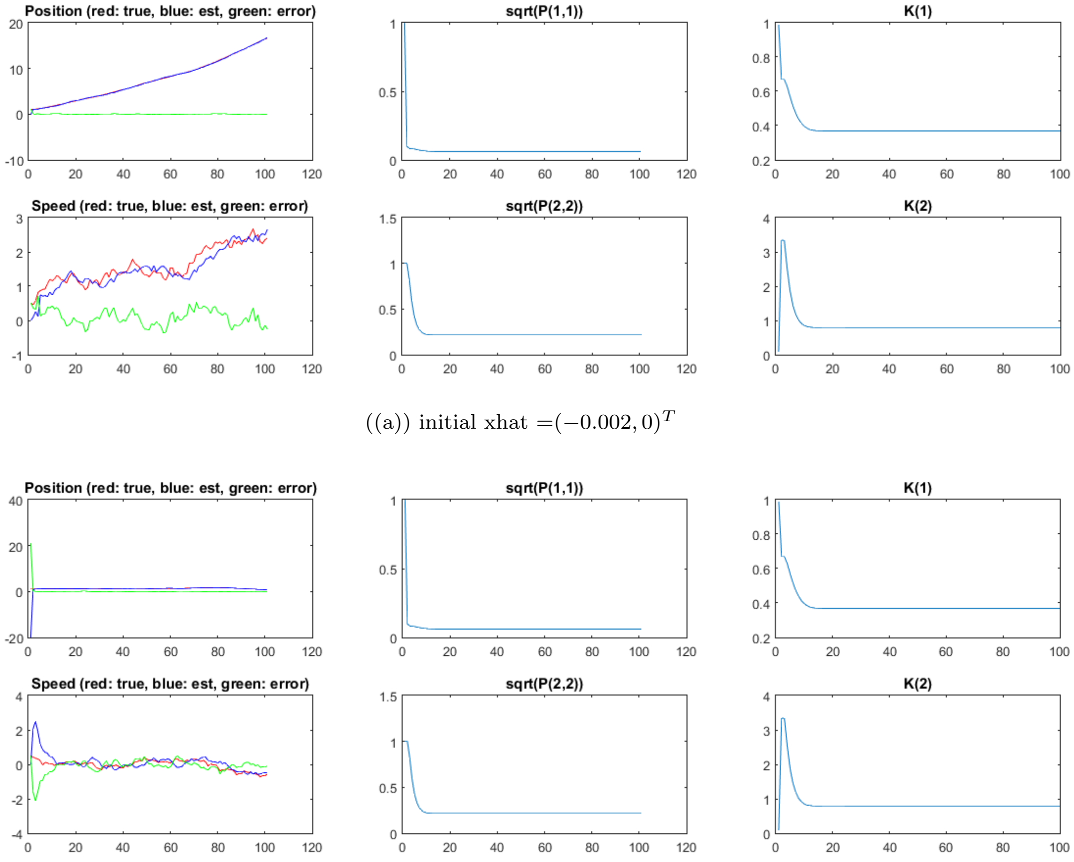
As can be seen in Figure 2(b), when we increase the measurement noise Q, the estimation of position and state become smoother compared to the pattern shown in 1. That is because Q has a direct influence on Kalman Gain K. Increasement of Q compresses K so that it leads to a slower update of μ_t and Σ. It can also been seen from the second column that the convergence is slower, but the variance of P is also smaller. On the other hand, a decreasement of Q has an inverse effect, as shown in Figure. 3(b).

Meanwhile, as can be seen in Figure 2(a), when we increase the process noise R, the estimation of position and state rapidly oscillates around true state compared to the pattern shown in 1. It also leads to a faster convergence but a higher variance of P. That is because when R_t increases, the $\overline{\Sigma}_t$ also increases. According to the fact that $\Sigma = ({C_t}^T {Q_t}^{-1} C_t + \overline{\Sigma}_t)^{-1}$, and $K = \Sigma_t {C_t}^T {Q_t}^{-1}$, an increasement of R_t indirectly leads to growth of K. So that the update is severe and more likely to deviate from the truth. So that the error rate is also relatively higher. Also similarly, a decrease of R_t also lead to an inverse effect on update, shown as Figure. 3(a).

Finally, it is interesting to notice that the estimation shows a similar pattern of the update of μ_t and Σ_t with the original one as R_t and Q_t increase or decrease with a same rate. Their effects to some extend cancel out each other. However, changing them at the same time still has a significant influence on converged variance. A large value seems to cause unstability of system, as shown in Figure. 2(c), where the variance of $sqrt(P)$ and Kalman Gain K are both higher. In contrast, as shown in Figure. 3(c), the variance is much more smaller and update is somehow slower.

4. *How do the initial values for P and xhat affect the rate of convergence and the error of the estimates (try both much bigger and much smaller)?*

   Initial values of xhat and P represent for estimated state $\overline{\mu}_0$ and uncertainty $\overline{\Sigma}_0$ respectively. The influence of $\overline{\mu}_0$ can be seen as Figure. 4, where in the upper figure the initial value is originally $(-2, 0)^T$ and changed to $(-0.002, 0)^T$, and in the lower figure it is changed to $(-200, 0)^T$. It is shown that changing initial xhat has not much effect on EKF system.

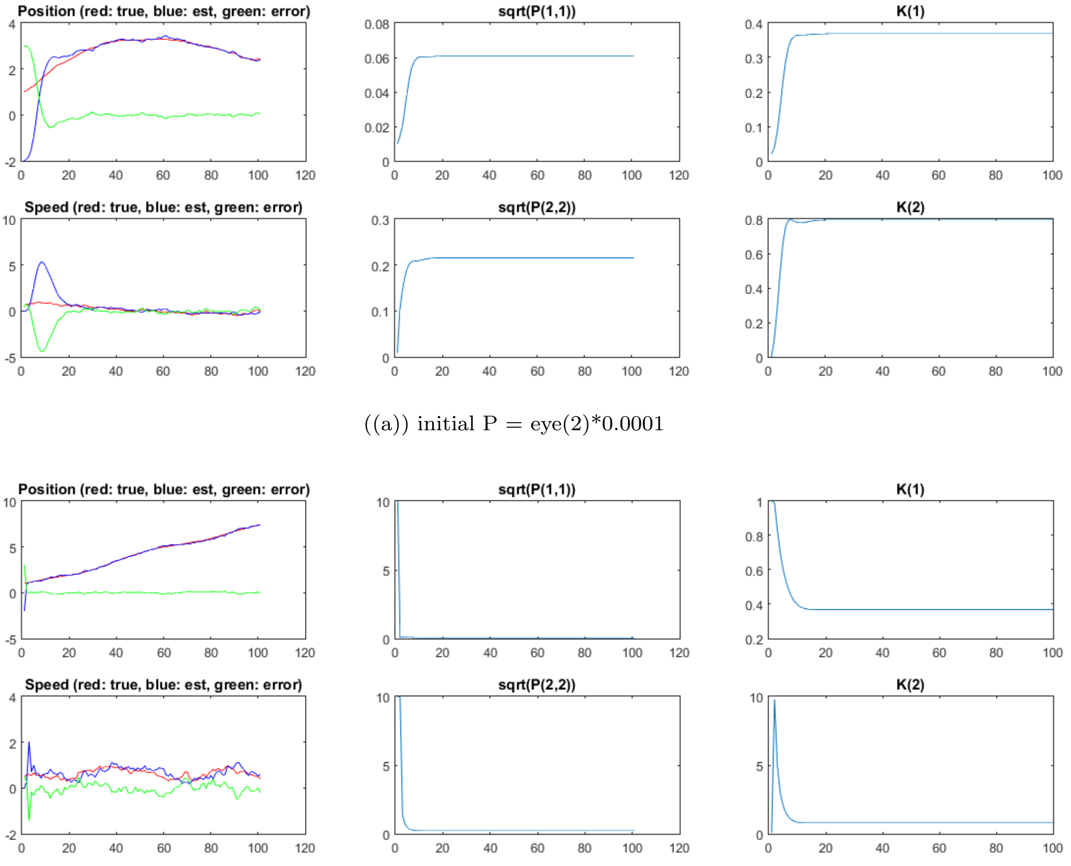


((a)) initial xhat $=(-0.002, 0)^T$

((b)) initial xhat $=(-200, 0)^T$

Figure 4: Influence of initial xhat

On the other hand, the initial value of P has significant influence on the system. Figure. 5(a) shows the pattern when $\overline{\Sigma}_0$ is changed to 0.0001 of original value and Figure. 5(b) changed to 100 times of the original value. As can be seen, a large initial P leads to a rapid increasement of Kalman Gain in the first few steps. That is because a large P increases the uncertainty of estimation of $\hat{x}$. In contrast, with a smaller P, Kalman Gain starts with a very low value and increases gradually to a converged value. That is because, with a small P, the uncertainty of estimation is small so that a smaller Kalman Gain is calculated which makes a slower update. It also causes slower convergence.



((a)) initial P = eye(2)*0.0001

((b)) initial P = eye(2)*100

Figure 5: Influence of initial P

3.2 Main problem: EKF Locolization

5. *Which parts of (2) and (3) are responsible for prediction and update steps?*

   In equation(2),

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

   , in which $\int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z_{1:t-1}, u_{1:t-1}, \overline{x}_0, M) dx_{t-1}$ is prediction, and incorporating it with $\eta p(z_t|x_t, M)$ is update.

   In equation(3),
   $\overline{bel}(x_t) = \int p(x_t|u_t, x_{t-1}, z_{1:t} bel(x_{t-1}) dx_{t-1}$ is prediction, and $bel(x_t) = \eta p(z_t|x_t, M)\overline{b}el(x_t)$ is update step.

6. *In the maximum likelihood data association, we assumed that the measurements are independent of each other. Is this a valid assumption? Explain why.*

   In this system, we assume that the model is first order so that a measurement is only depend on its state x_t, i.e. given state x_t, it is independent from previous states and measurements, $p(z_t|z_{1:t-1}, x_{1:t}, u_{1:t}) = p(z_t|x_t)$. In reality, usually the assumption is not valid. However, it is a trade-off that considering the dependency of measurements, it can largely increase the complexity of the model by modeling dependency as $p(z_t|x_t, z_{1:t-1})$. As a result, in the maximum likelihood method, $ML = argmax_{z_t} p(x_t, z_{1:t-1}|z_t)$ could be unreliable. Mathematically, it is a good assumption especially when the environment is static. It also enable us to easily add more measurement feature information into the model.

7. *What are the bounds for δ_M in (8)? How does the choice of δ_M affect the outlier rejection process? What value do you suggest for λ_M when we have reliable measurements all arising from features in our map, that is all our measurements come from features on our map? What about a scenario with unreliable measurements with many arising from so called clutter or spurious measurements?*

   As λ_M built is based on probability, it would be efficient to define δ_M within $[0, 1]$. As it is defined with a inverse-chi-squared distribution, λ_M monotonously increase as δ_M increase. On the other hand, value of measurement covariance Q effects D_M since D_M is calculated from $(\overline{S}_{t,i})^{-1}$. In directly, a larger Q decrease the value of D_M. That means, a larger uncertainty makes a larger D_M. Since λ_M acts as a threshold, we want a larger threshold when uncertainty is small. As a result, we should set a larger δ_M when Q is large and vice versa.

8. *Can you think of some down-sides of the sequential update approach(Alg 3)? Hint: How does the first [noisy] measurements affect the intermediate results?*

   The sequential update is sensitive to noises. As μ_t and Σ_t update with every measurement, a noisy data will significantly update its value as Kalman Gain will increase. So that we are more likely to get an more biased belief in next time step. If data is very noise, it may cause a slow convergence.

9. *How can you modify Alg 4 to avoid redundant re-computations?*

   In the function, the calculation of $\hat{z}_{t,j}$, $H_{t,j}$, and $S_{t,j}$ only rely on parameter j. However, according to the algorithm, because the loop over parameter i, each of them were calculated redundantly. Therefore, if we apply a if on the them, redundant calculation can be avoided. It can ba shown as follow:

```
for all Observations $i$ in $z_t$ do
  for all Landmarks $j$ in $M$ do
    if $i == 1$
      $\hat{z}_{t,j} = h(\overline{\mu}_t, M, j)$
      $H_{t,j} = H(\overline{\mu}_t, M, j, \hat{z}_{t,j})$
      $S_{t,j} = H_{t,j}\overline{\Sigma}_t(H_{t,j})^T + Q$
    endif
    ..............
  endfor
endfor
```

10. *What are the dimensions of $\overline{\nu}_t$ and $\overline{H}_t$ in Alg 4? What were the corresponding dimensions in the sequential update algorithm? What does this tell you?*

    As can be seen in the algorithm, in sequential update, dimension of ν_t is $2 \times N$, where N denotes the number of Landmark, and 2 is corresponding to the dimension of measurement z. In batch update, ν is saved by $\overline{\nu}_t = ((\overline{\nu}_t{}^1)^T, (\overline{\nu}_t{}^2)^T, ...(\overline{\nu}_t{}^n)^T)^T$. In which, the dimension of $\overline{\nu}_t{}^n$ is 2×1 so that the dimension of $\overline{\nu}_t$ is $2n \times 1$, where n is the number of inliners.

    As for $\overline{H}_t$, in sequential update it's dimension is $2 \times 3 \times N$. In batch update, it is saved as $\overline{H}_t = (\overline{H}_t{}^1)^T, (\overline{H}_t{}^2)^T, ..., (\overline{H}_t{}^n)^T)^T$. So that the dimension is $2n \times 3$.

    The differences of dimensions shows the different way of update in the sequential and batch method. In the former case, each data is treated separately, while in the latter method, matrix manipulation is used to calculate data in batch.

3.3 Datasets

3.3.1 map_o3.txt + so_o3 ie.txt

In this dataset, $R = diag(0.1^2, 0.1^2, 0.1^2)$ and $Q = diag(0.1^2, 0.1^2)$ are applied. Using **runlocalization_track('so_o3_ie.txt', 'map_o3.txt', 1, 1, 1, 2);** to simulate, the result is shown as Figure 6:

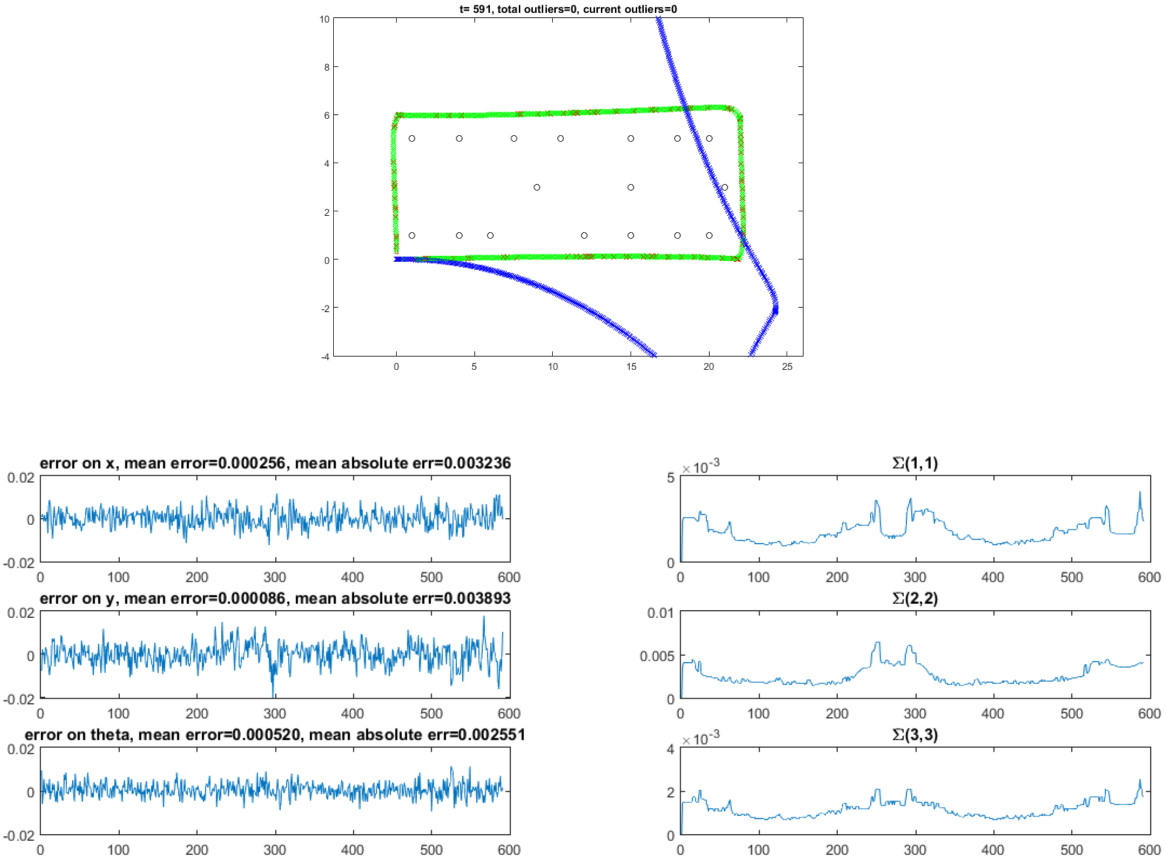


Figure 6: Result of dataset 1

, and the output is shown as following:
mean error(x, y, theta)=(0.000256, 0.000086, 0.000520)
mean absolute error=(0.003236, 0.003893, 0.002551)
total_time =31.436604

3.3.2 map_pent_big_10.txt + so_pb_10_outlier.txt

In this dataset, $R = diag([0.01^2, 0.01^2, (2 * pi/360)^2])$ and $Q = diag([0.2^2, 0.2^2])$ are applied. Using **runlocalization_track('so_pb_10_outlier.txt', 'map_pent_big_10.txt', 1, 1, 1, 3);** to simulate, the result is shown as Figure 7:

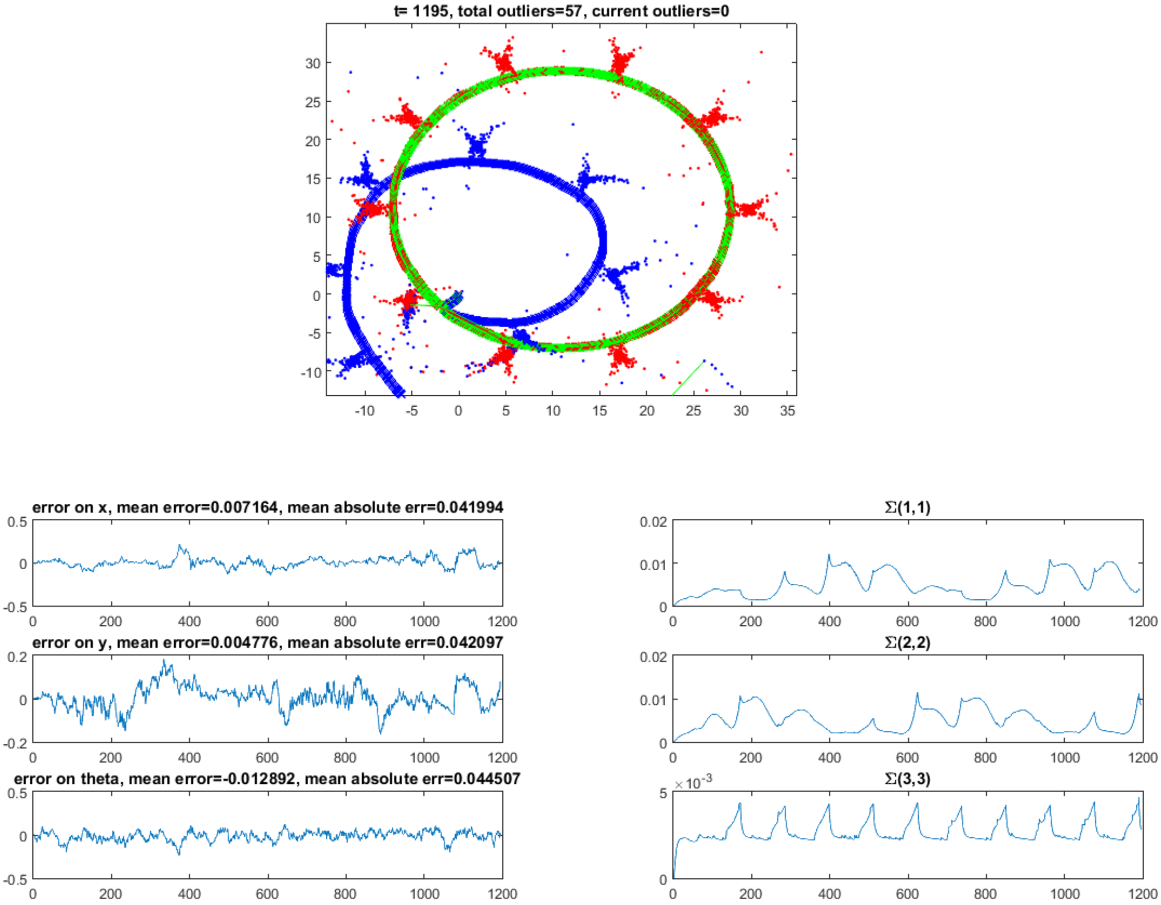
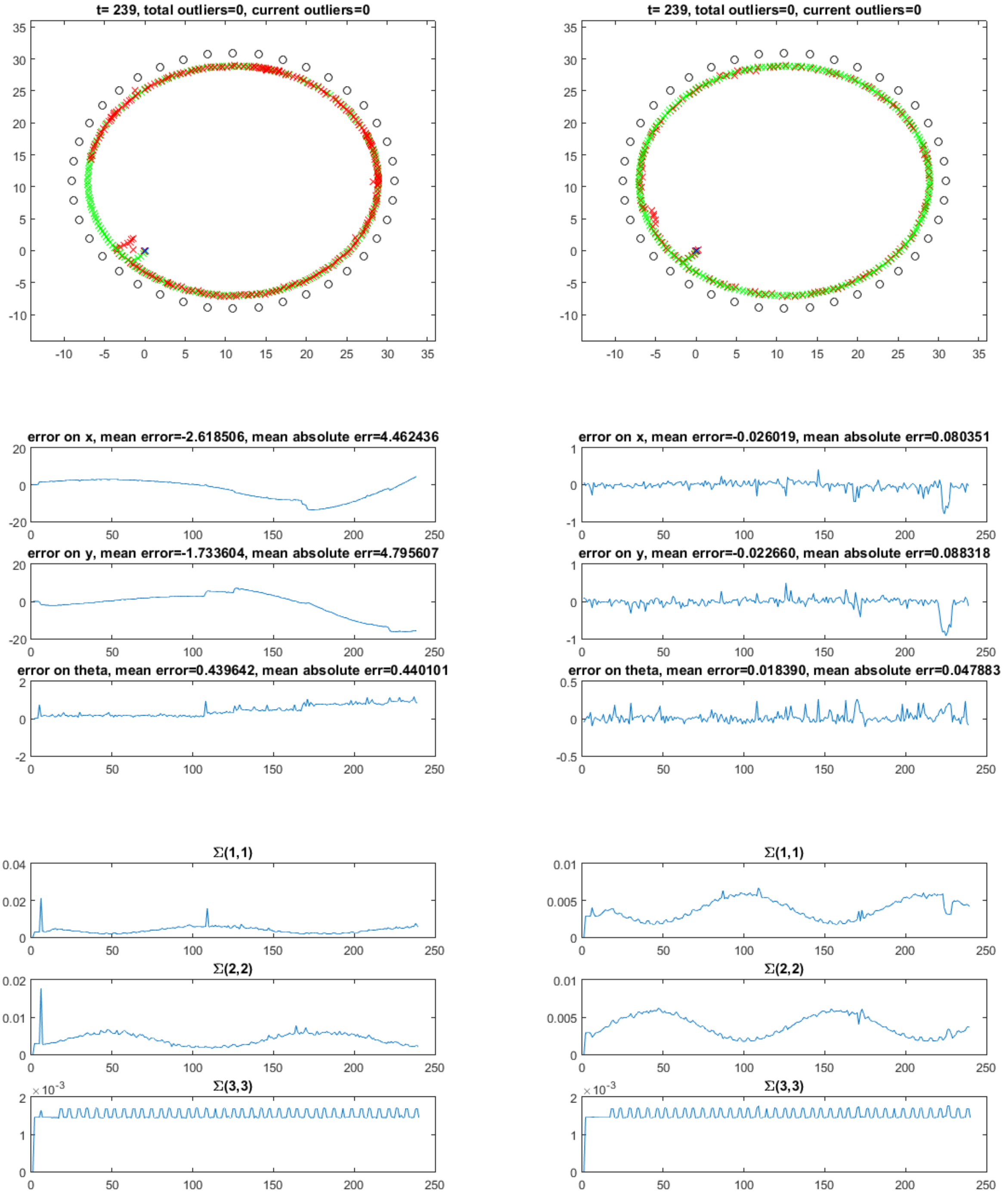


Figure 7: Result of dataset 2

, and the output is shown as following:
mean error(x, y, theta)=(0.007164, 0.004776, -0.012892)
mean absolute error=(0.041994, 0.042097, 0.044507)
total_time =161.844885

3.3.3 map_pent_big_40.txt + so_pb_40_no.txt:

In this dataset, $R = diag([1^2, 1^2, 1^2])$ and $Q = diag(0.1^2, 0.1^2)$ are applied. Furthermore, **delta_m** is set to 1 to avoid detecting outlier and using batch update or not is compared. Using **runlocalization_track('so_pb_40_no.txt', 'map_pent_big_40.txt', 1, 1, 1, 2);** to simulate, the result is shown as Figure 8:



((e)) Without Batch Update

((f)) With Batch Update

Figure 8: Result of dataset 3

, and the output without using batch update is shown as following:

```
..........
warning, 6 th measurement(of landmark 28) was incorrectly associated to landmark 22,
t=239
warning, 7 th measurement(of landmark 29) was incorrectly associated to landmark 23,
t=239
mean error(x, y, theta)=(-2.618506, -1.733604, 0.439642)
mean absolute error=(4.462436, 4.795607, 0.440101)
total_time =9.644177
```

And the output using batch update is shown as following:

```
..........
warning, 1 th measurement(of landmark 23) was incorrectly associated to landmark 29,
t=237
warning, 6 th measurement(of landmark 28) was incorrectly associated to landmark 22,
t=237
mean error(x, y, theta)=(-0.026019, -0.022660, 0.018390)
mean absolute error=(0.080351, 0.088318, 0.047883)
total_time =8.103424
```

In this case, it is obvious that the update without batch is too slow so that it would cause a severe problem of misassociate of landmark in the measurement.