EL2820 Applied Estimation : Lab 1

Chia-Hsuan Chou

Chchou@kth.se

Part 1: Preparitory Questions:

§§Linear Kalman Filter:

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

---

u_r is control data which carry information about the change of state in the environment. For a robot, one example of control data is velocity.

Z_t is measurement data which provides information about a momentary state of the environment. Examples of the measurement data are camera image and range scan.

Xt is the states, which is the collection of all aspects of the robot and its environment that can impact the future. For example, the robot's pose, the location and features of surrounding objects in the environments.

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

No, the uncertainty in the belief decrease during an update while the uncertainty in the belief increase during a prediction step. The prediction step use sensor data (measurement data) to be integrated into the present belief, and decrease the uncertainty. To be more concisely, the update step use measurement data to adjust the prediction to more accurate states.

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

Kalman Gain decides the weighing between measurements and belief. To find Kalman Gain, it needs the two covariance from two Gaussians, which one is from the prediction step while the other one is from the measurement error.

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

Estimates would be pessimistic and convergence would be slower. The updated states would be closer to the prediction states which means u_t dominates the estimation of the states. The uncertainty would be really big, and questions if the belief is trustable.

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

As the answer in question 4, the larger the covariance for measurement model, the less effect on the updated state estimate. Therefore, if we want an increased effect on the updated state estimate, we can choose smaller covariance for the measurement.

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

The belief uncertainty would increase during the prediction because the action is assumed to be stochastic, therefore the uncertainty in prediction step will usually increase based on the action.

Question 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 if represented by $x_t = A_t x_{t-1} + B_t u_r + \varepsilon_t$, which is a Gaussian distribution with $\mu_t = A_t x_{t-1} + B_t u_r$ and the covariance $\Sigma_t = R_t$.If we have correct prediction, which means no error ε_t on it. the mean would be converged exact to where it locates and the covariance would tend to a constant.

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

In the case of Gaussian white noise and Gaussian priori distribution, the Kalman Filter is a MAP estimator since when iterating, we always give it a $bel(x_{t-1})$. If we start with an update from no prior then the Kalman Filter is a MLE.

Extended Kalman Filter:

Question 9: How does the extended Kalman filter relate to the Kalman filter?

Extended Kalman filter deal with the estimation with nonlinear states. EKF assumes the transition probability and measurement probability are governed by nonlinear functions g and h where:

$$x_t = g(x_{t-1}, u_t) + \varepsilon_t$$

$$z_t = h(x_t) + \delta_t$$

Question 10: Is the EKF guaranteed to converge to a consistent solution?

EKF does not guaranteed to converge to a consistent solution.

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

If our filter seems to diverge often we might change our modeled uncertities Q and R. Typically divergence occurs on update and increasing the relative size of the measurement covariance Q will help.

If the divergence was due to a poor data association then we can try to change our matching threshold.

Localization:

Question 12: If a robot is completely unsure of its location and measures the range r to a know 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.

It will have a uniform distribution over heading between $-\pi$ and π. The distribution will be a donut.

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

The posterior belief of its location will look like a big circle with landmark as the center of that circle. The radius of the circle would be r.

Question 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?

It will look the same as a donut but the heading and angle around the donut would be Gaussian with a completely correlated covariance.

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

The jacobian will produce an update on a straight line which can not move the estimate along the curved crescent. The Gaussian will not be able to represent the crescent shape and thus update might diverge.

Part 2: Matlab Exercise:

§§Warm up problem with Standard Kalman Filter :

Question 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?**

ε_k will be 2x1 matrix while δ_k will be 1x1 matrix. We need to define the covariance of ε_k and the covariance of δ_k since they are Gaussian distribution with zero mean and we need to define the covariance.

Question 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/Usages***
x	*The actual states.*
xhat	*The estimated states.*
P	*Covariance of estimated states.*
G	*2 dimensional identity matrix. Used to scale the uncertainty of states.*
D	*1 dimensional identity matrix. Used to scale the uncertainty of states.*
Q	*measurement uncertainty, covariance of measurement noise*

R	*state transition uncertainty, covariance of state transition noise*
wStdP	*Standard deviation of noise on simulated position.*
wStdV	*Standard deviation of noise on simulated velocity.*
vStd	*Standard deviation of Simulated measurement noise.*
u	*Acceleration of the car.*
PP	*The set of covariance of all estimated states.*

Question 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.)**

In my expectation, if we increase the covariance of state transition noise, which is R here, the estimated state will rely on measurement more, therefore K will increase. On the other hand, if we increase the covariance of measurement noise, which is Q here, the estimated state will rely on the measurement less, therefore K will decrease and converge slowly. If we increase both Q and R, then the convergence happens latter and the covariance of estimated states will be large. In contrast, if we decrease both Q and R in the same time, then the covariance of estimated states will be smaller.

Figure 1 shows the Kalman Gain(K) and covariance of estimated states (P) with the Q and R given by the source code, while figure 2 shows K and P with R is 100 times larger fig 3 with Q is 100 times larger.

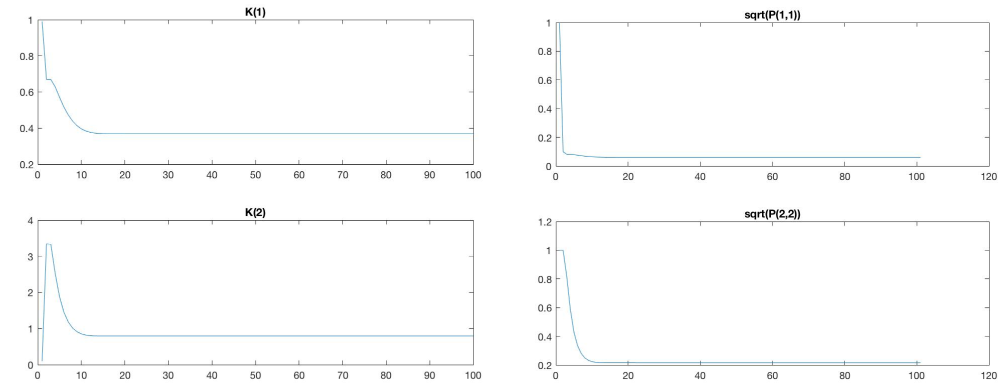


fig (1). K and P with original Q and R

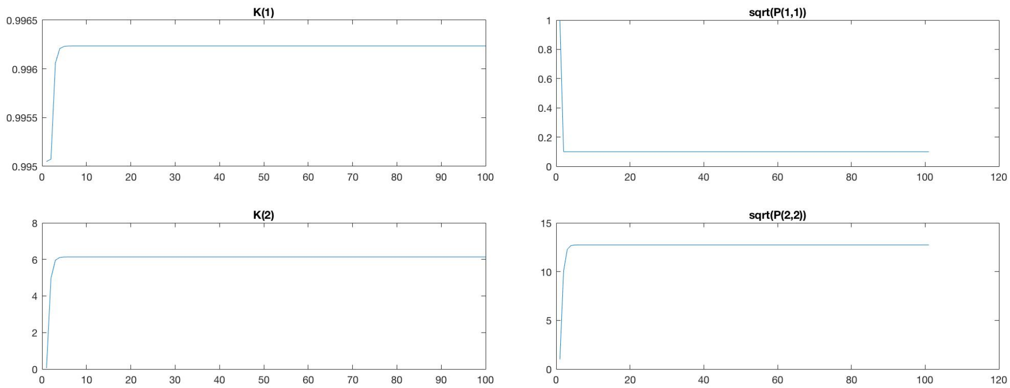
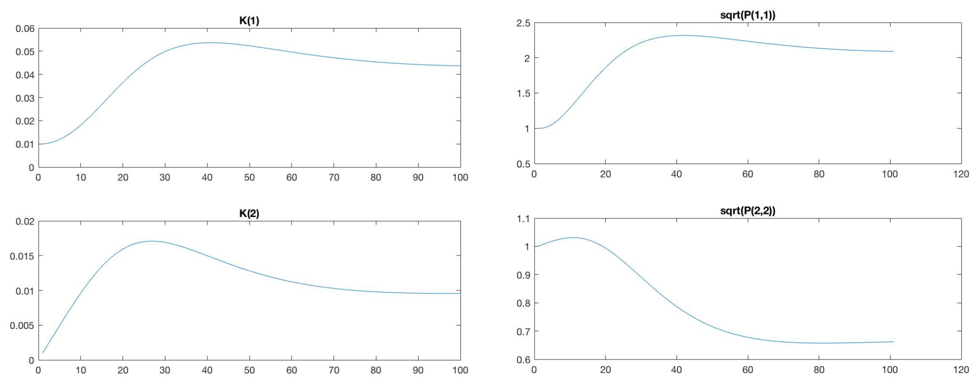


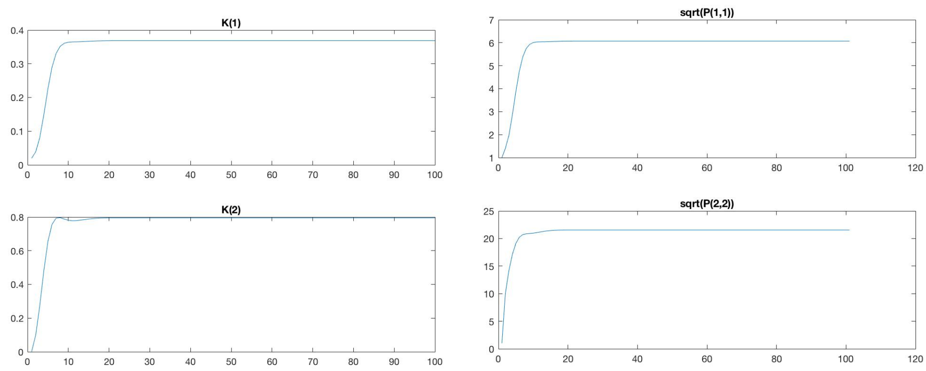
fig (2). K and P with R is 100 times larger



fig(3). K and P with Q is 100 times larger

Compare to figure 1 and 2, we can find that the Kalman Gain become bigger (from around (0,4 ,1) to (0.9 , 6)) when R is 100 times larger. As for figure 3, we know that Kalman Gain become smaller (from around (0,4 , 1) to (0.05, 0.01)) and the time to converge is apparently later.

Figure 4 shows the K and P when we increase 100 times both Q and R:

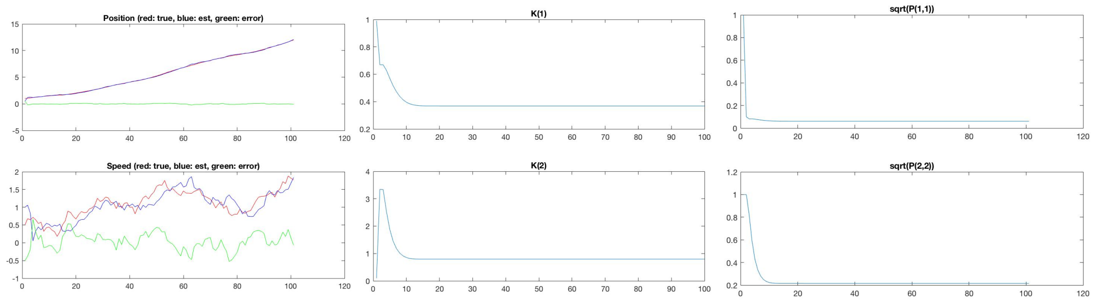


Fig(4). K and P with both Q and R increase 100 times

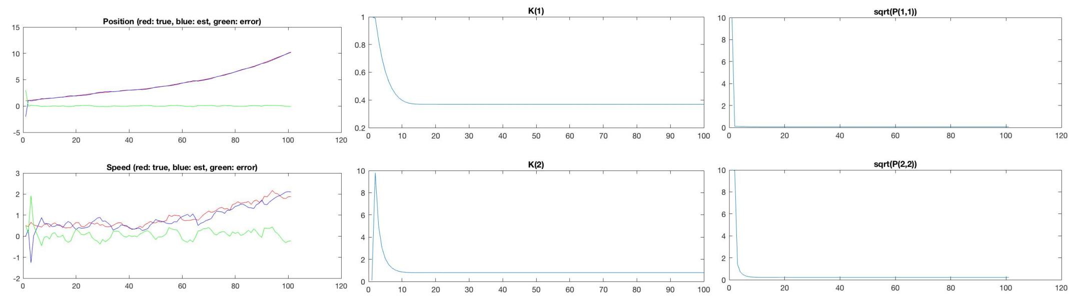
From figure 4 we found that if both Q and R increase, the covariance of estimated states will increase a lot as well, which means that the car is not quite sure exactly where it is.

Question 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)?**

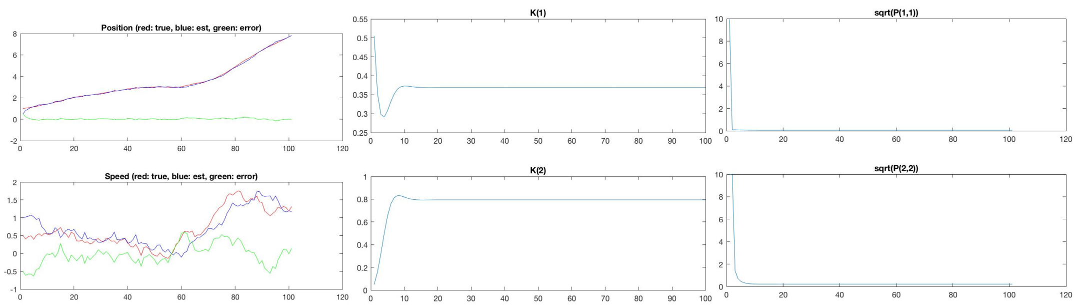
When initial values for P is 100 times larger, the rate of convergence remains proximately the same and the error is approximately the same. Then I set initial values for P is 100 times smaller, the rate of convergence is slower and the error of estimates become smaller as well. Therefore, I change the initial values for P again in order to testify my assumption, I set initial values of P for 10000 times smaller, and the rate of convergence is much slower. Therefore we can say that the smaller the initial value of P is, the smaller the error is and the slower the convergence is. The following sigures show the result of setting different P:



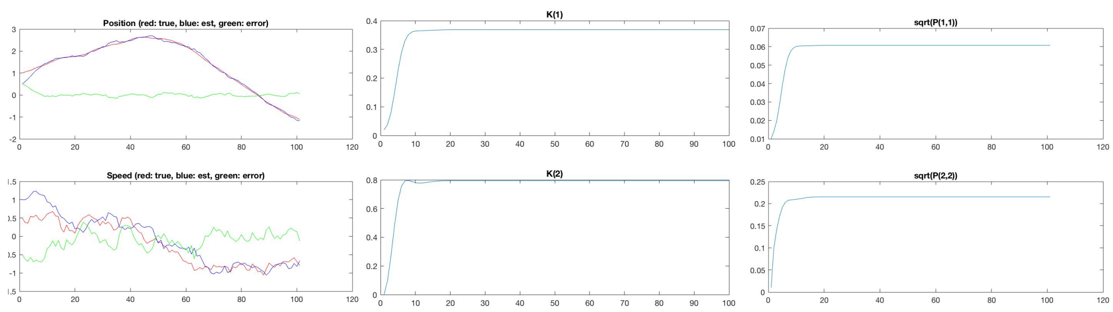
Fig(5) Path, K and P with P = 1



Fig(6) Path, K and P with P = 100



Fig(7) Path, K and P with P = 0.01



Fig(8). Path, K and P with P = 0.0001

As for the estimated initial states (xhat), no matter how big or small, the rate of convergence and the error of estimate has no huge difference.

§§ Main Problem: EKF Localization :

Question 5: Which parts of (2) and (3) are responsible for prediction and update step?

$\int P(X_t|u_t, X_{t-1})P(X_{t-1}|Z_{1:t-1}, u_{i:t-1}, \overline{X_0}, M)dX_{t-1}$ is responsible to the prediction step, and $\eta P(Z_t|X_t, M)$ is responsible to the update step.

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

It is a valid assumption only if we have actual white noise. If we measure two different stable landmarks, when we measure the distance between first landmark and the robot's position, there is no effect on the distance between the second landmark and the robot's position. Therefore, the measurements are independent of each other and this assumption is valid.

Question 7: What are the bounds for δM in (8)? How does the choice of M affect the outlier rejection process? What value do you suggest for 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?

In the lab note we know that δM is probability, therefore it is bounded to [0,1]. If we choose larger δ_M, the threshold will be larger, which means we are more confident on the measurement, and less measurement will be rejected. In the same way, if we choose smaller δ_M, then more measurement will be rejected as we are less confident on our measurement. Therefore, if we have reliable measurements, we can choose larger value of $\delta_{M.}$

Question 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**?

Since we update on every measurement each time, if there is one very noisy measurement, then we will estimate the uncertainty might decrease wrongly, which means the uncertainty does not capture the true situation. To be more concise, the robot misunderstands the current situation and think other correct or acceptable measurements as outliers. On the rest of the update approach, it is hard to update correctly and get a really poor estimation.

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

After the first for-loop in the first measurement, we can narrow down the search on the next measurement and compute for those who has higher possibility landmarks only, because when the robot is on a state, there is a limited on the measurement.

Question 10: What are the dimensions of $\bar{\nu}_t$ and $\overline{H_t}$ in Alg 4? What were the corresponding dimensions in the sequential update algorithm? What does this tell you?

Assuming a range-bearing model from equation (4), we know that in Batch update the dimension of $\bar{\nu}_t$ is 2n x 2 and the dimension of $\overline{H_t}$ is 2n x 3. In the sequential update algorithm, the dimension of $\bar{\nu}_t$ is 2x1 and the dimension of $\overline{H_t}$ is 2 x 3. Comparing to the dimensions of $\overline{H_t}$ and $\bar{\nu}_t$ in two algorithm, we found that sequential update algorithm has less computation however, it cannot consider all the measurement in a time, which means a noisy measurement might lead the uncertainty decrease wrongly. In Batch update, all the measurement in i-th step is put in a matrix and then compute the estimated states. Therefore a noisy measurement data might have less influence on the whole estimation.

§Datasets:

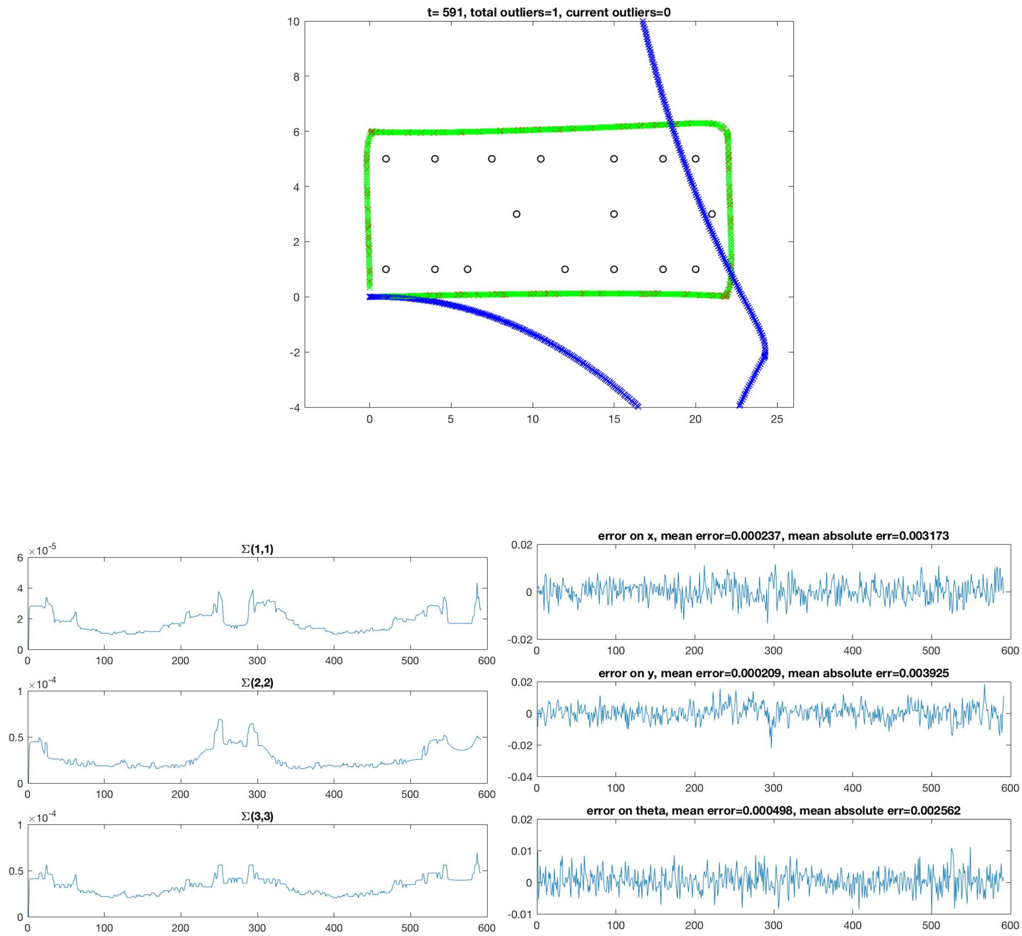
1. map o3.txt + so o3 ie.txt:

In this dataset, Q and R are applied as :

```
R = diag([0.01^2 0.01^2 (2*pi/360)^2]);

Q = diag([0.01^2 (2*pi/360)^2]);
```

Where the laser scanner has an accuracy of (1 cm,1 degree), the odometry information has an un-modeled noise of approximately 1 cm and 1 degree per time step as described in lab note.

Using *runlocalization_track('so_o3_ie.txt', 'map_o3.txt', 1, 1, 1, 2)* ,the result is shown on figure 9:



Fig(9). Results of Dataset 1

and the output:

```
  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 =8.693104
```

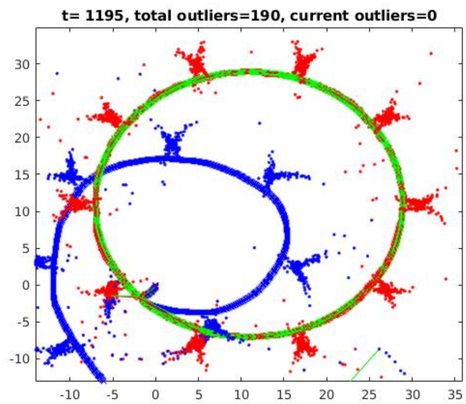
2. **map pent big 10.txt + so pb 10 outlier.txt:**

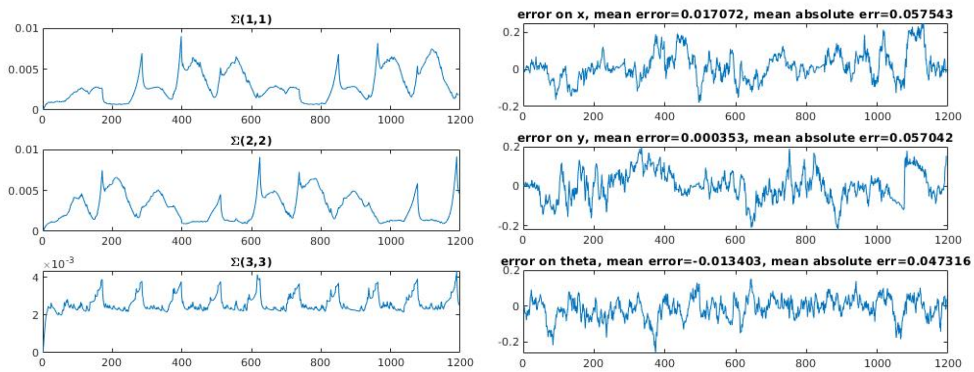
   In this dataset, Q and R are applied as :

```
R = diag([0.01^2 0.01^2 (2*pi/360)^2]);

Q = diag([0.1^2 0.2^2]);
```

   Where the measurement has a white Gaussian with the standard deviation of 0.2(m,rad)

   Using *runlocalization_track('so_pb_10_outlier.txt', 'map_pent_big_10.txt', 1, 1, 1, 3)* , the result is shown on figure 10:





Fig(10). The result of second Dataset

And the output:

```
mean error(x, y, theta)=(0.017072, 0.000353, -0.013403)

mean absolute error=(0.057543, 0.057042, 0.047316)

total_time =79.245822
```

3. map pent big 40.txt + so pb 40 no.txt:

In this dataset, Q and R are applied as:

```
R = diag([1^2 1^2 1^2]);

Q = diag([0.1^2 0.1^2])
```

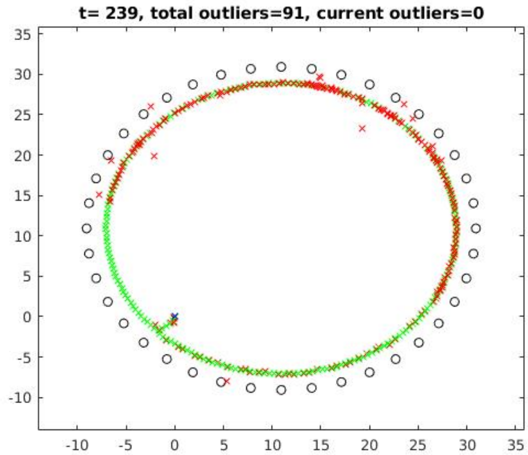
Where measurement noise with a standard deviation of 0.1(m, rad), the process noise with a standard deviation of 1(m, m, rad).

In this section, we need to compare the sequential update algorithm and the batch update algorithm by changing δ_M.The result is shown in following section:

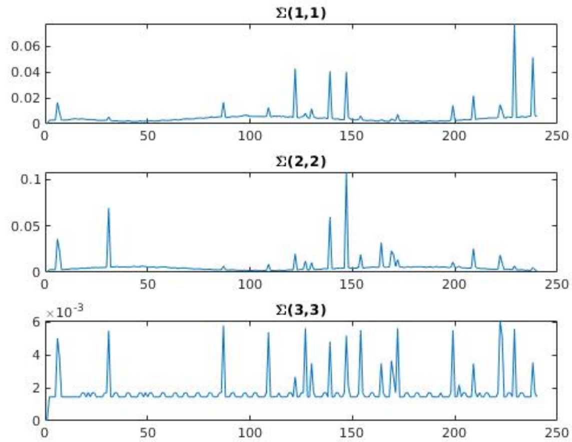
3.1 The batch Update algorithm:

Remain the delta_m as 0.999, we can get the result with Batch Update algorithm:

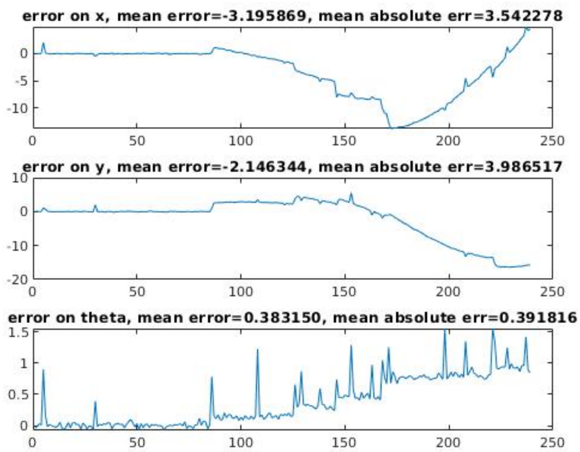
Following figures shows the result of this algorithm:



Fig(11). The path with batch update algorithm



Fig(12) The covariance with batch update algorithm



Fig(13) The error with batch update algorithm

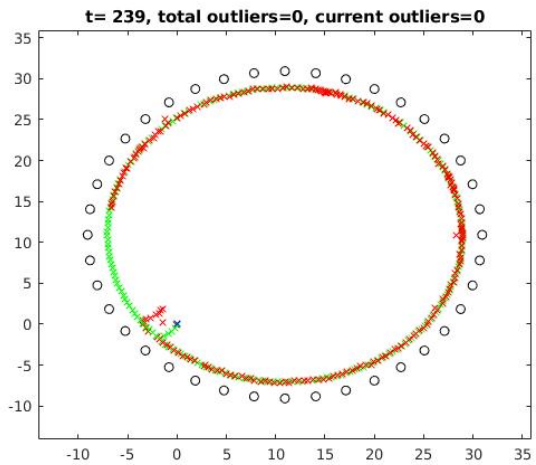
And the output:

```
mean error(x, y, theta)=(-3.195869, -2.146344, 0.383150)

mean absolute error=(3.542278, 3.986517, 0.391816)

total_time =3.056224
```

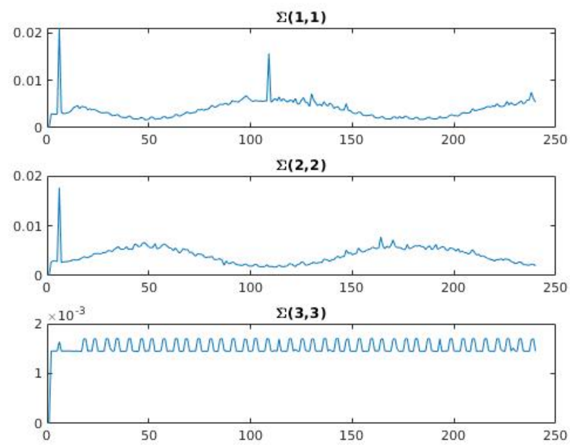
3.2 The sequential update algorithm

Setting delta_m into 1, we can get the result with sequential update algorithm:

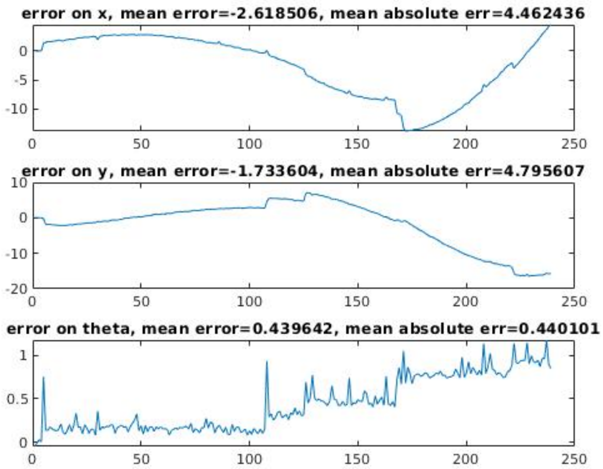
Following figures show the result of this algorithm:



Fig(14). The path with sequential update algorithm



Fig(15). Covariance with sequential update algorithm



Fig(16). Error with sequential update algorithm

And the output:

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

mean absolute error=(4.462436, 4.795607, 0.440101)

total_time =3.370468
```

In this case, it is obvious that the sequential update algorithm is slower than update with batch and has larger value of error.