Applied Estimation lab1

Zhengming Zhu

November 13, 2023

1 Part1

1.1 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?
u_t:change the state of the system, like motors.
x_t: describe the statement of the system.
z_t: the measurement of the system form sensor like camera, range scan.

$$u_t = [dx, dy] \quad x_t = [x, y, \theta] \quad z_t = [x, y]$$

2. Can the uncertainty in the belief increase during an update? Why (or not)?
No, during the update process:

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

because:$C_t^T Q_t^{-1} C_t$ is simi-define, so the covariance updated is smaller the before.

3. During update what is it that decides the weighing between measurements and belief?
Q matrix, R matrix and $\sum$

4. What would be the result of using a too large a covariance (Q matrix) for the measurement model.
The Q matrix represents the measurement noise covariance. If Q matrix is too big, which means a lager uncertainty. The Kalman Gain will decrease, so the updated belief will depend on the measurement model in less degree. And the filter will have a slower convergence.

5. What would give the measurements an increased effect on the updated state estimate?
The Q matrix will decrease, which leads to smaller uncertainty and an increased Kalman Gain. So, during the update process, the updated state will assign more weights on the measurement model.

6. What happens to the belief uncertainty during prediction? How can you show that?
During prediction, the belief uncertainty will increase, because it considers the process noise and the uncertainty of previous state with the transformation.

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

We can monitor the covariance, if the covariance grows, the uncertainty increases.

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.)
First, The Kalman filter is a Bayesian estimator, which uses the prior distribution and the new observational data to produce a posterior distribution. For this linear, independent Gaussian system, the posterior distribution is also a Gaussian distribution. For a Gaussian distribution, the mean and the

covariance can be computed precisely.

8. In the case of Gaussian white noise and Gaussian priori distribution, is the Kalman Filter a MLE and/or MAP estimator?
MLE: Under Gaussian noise, the Kalman filter maximize the likelihood function of the probability of observing the measurements given the state estimates. In this aspect, Kalman filter is a MLE.
MAP: The Kalman filter is a kind of Bayesian estimator, and it not only consider the likelihood of the measurement data, but also the prior probability distribution of the parameters. So, it also a MAP to maximize the posterior probability.

1.2 Extended Kalman Filter

9. How does the extended Kalman filter relate to the Kalman filter?
The extended Kalman filter is suitable for nonlinear system based on standard Kalman filter. EKF approximates the system as linear around the current estimation by a first-order Taylor series expansion. Both of them have the similar update process.

10. Is the EKF guaranteed to converge to a consistent solution?
No, here are some factors regarding the convergence of the EKF like the degree of uncertainty and the degree of local nonlinearity of the function that are being approximated. If the degree of uncertainty and local nonlinearity is too big, EKF can't guarantees to converge to a consistent solution.

11. If our filter seems to diverge often can we change any parameter to try and reduce this?
yes:
- we can replace μ with a state $\tilde{\mu}$ that minimizes when linearize around the point:

$$minimize : |h(\tilde{\mu} - z_t)|^2$$

- adjust the Q and R.
- change the matching threshold.

1.3 Localization

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 be a circle around the landmark with distance r, Also, the circle has width regarding to the standard deviation of the measurement noise. And, the θ will remain uniform because of lacking of information of direction.

13. If the above measurement also included a bearing how would the posterior look?
It will be also a circle around the landmarks. Since the system can get the bearing measurement, we can know the robot facing a specific direction.The posterior will be a ring, but the covariance will change.

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?
The posterior will have an increasing covariance because of increasing and accumulating uncertainty.

15. If the above robot then sees a point feature and measures range and bearing to it how might the EKF update go wrong?
This performance of EKF depends on the linearization of the measurement model, which already has a significant error and uncertainty. Also, because of the big covariance at this time, the EFK may be divergent.

2 Part2

2.1 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?
The dimensions of ϵ_k is same as the dimensions of x_k, and the dimensions of δ_k is same as the dimensions of p_k. So, the dimensions of ϵ_k is 2, the dimensions of δ_k is 1. Furthermore, we can define the mean μ and covariance Σ for a unique white Gaussian.

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).

x	state of system
xhat	estimated state of system
P	estimated error covariance
G	process noise matrix for dimensionality consistency
D	measurement noise matrix for dimensionality consistency
Q	measurement covariance matrix
R	process covariance matrix
wStdP	Noise on simulated position
vStd	Simulated measurement noise on position
u	control input vector
PP	Containers for storing P matrix

Table 1: table1

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.)
Default:
Standard deviation of error in position (second half): 0.067595m
Standard deviation of error in velocity (second half): 0.234694m/s

Increase Q 100 times:
The Kalman gain will decrease, the system will depend less on the measurement model and take more time to converge.
Standard deviation of error in position (second half): 0.072739m
Standard deviation of error in velocity (second half): 0.161340m/s

Increase R 100 times:
The Kalman gain will increase, the system will depend more on the measurement model. And the estimated speed and error increase float drastically
Standard deviation of error in position (second half): 0.083544m
Standard deviation of error in velocity (second half): 0.504529m/s

Increase both R and Q 100 times:
The Kalman gain will decrease, and this system will spend more time to converge.
Standard deviation of error in position (second half): 0.069710m
Standard deviation of error in velocity (second half): 0.246589m/s

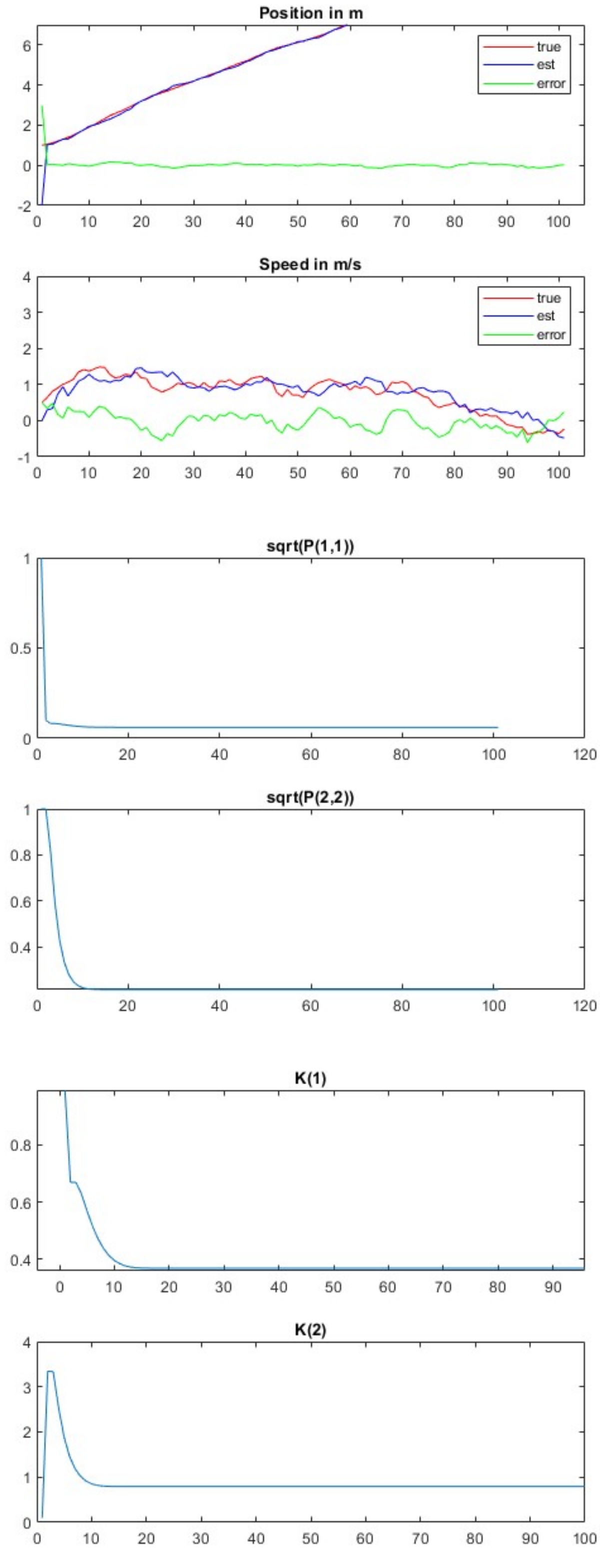

Figure 1: default Q and R

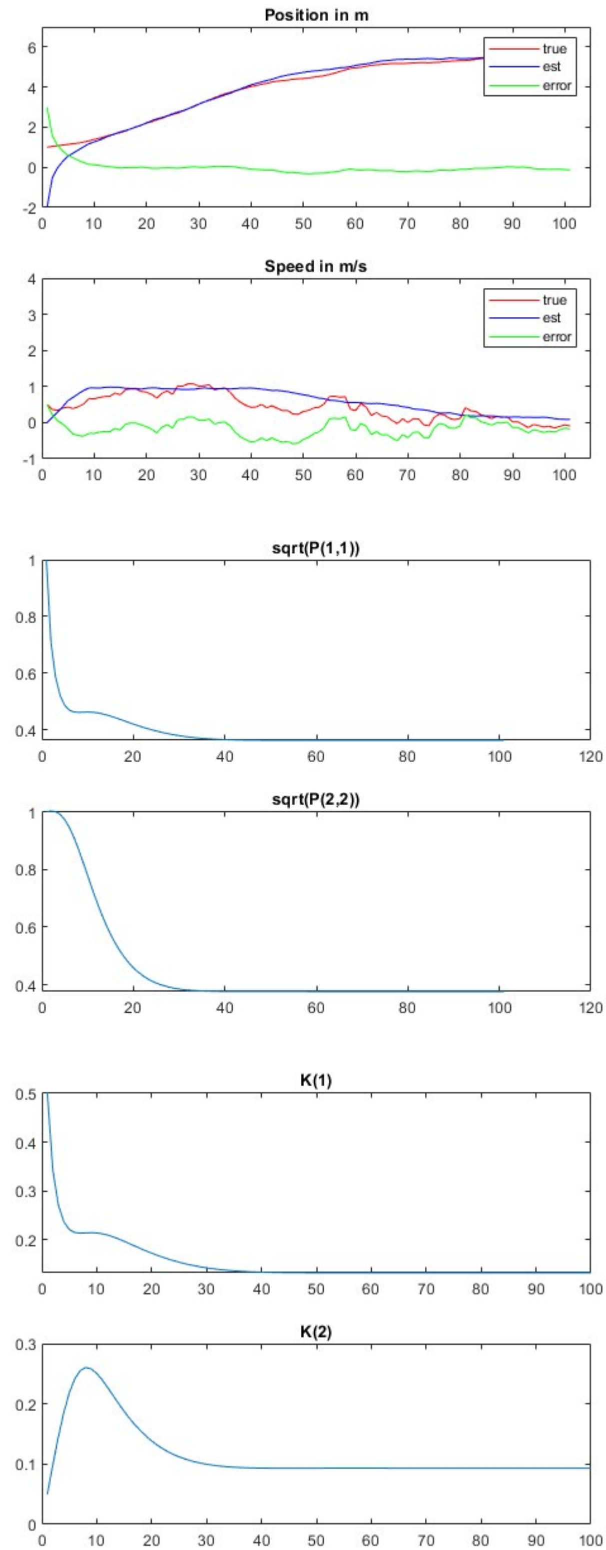

Figure 2: Increase Q 100 times

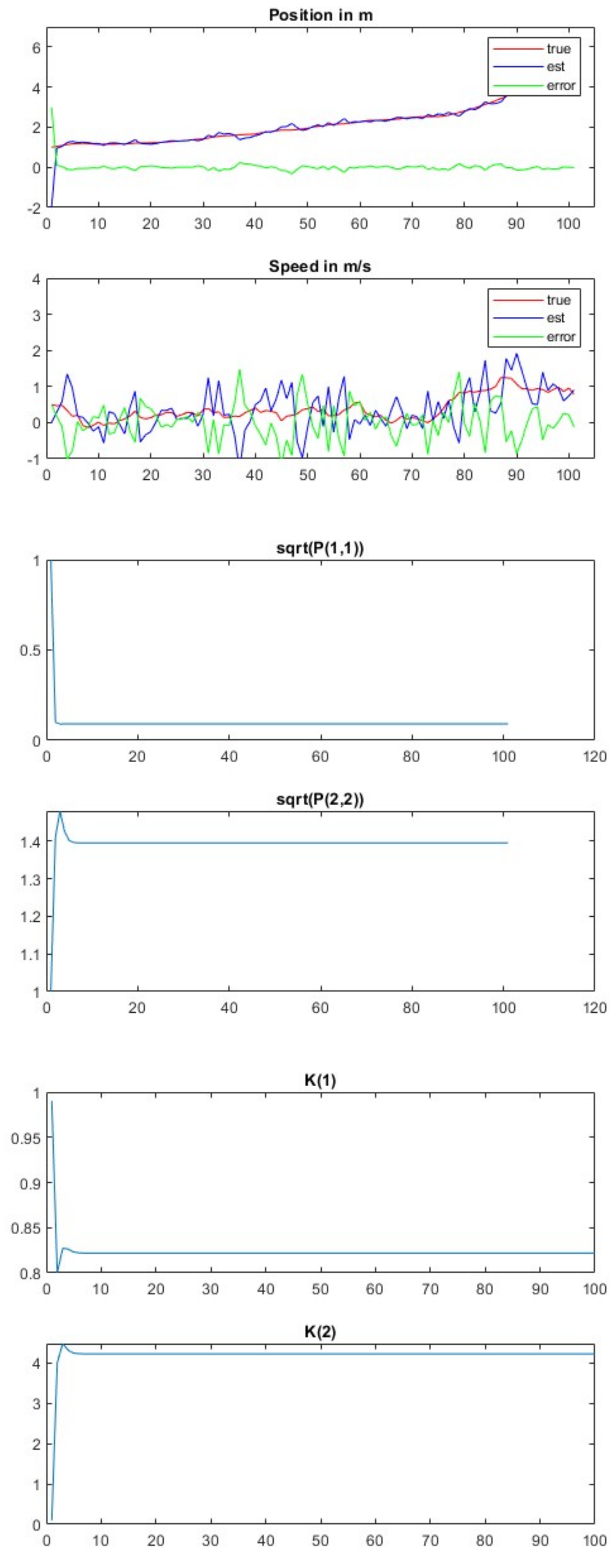

Figure 3: Increase R 100 times

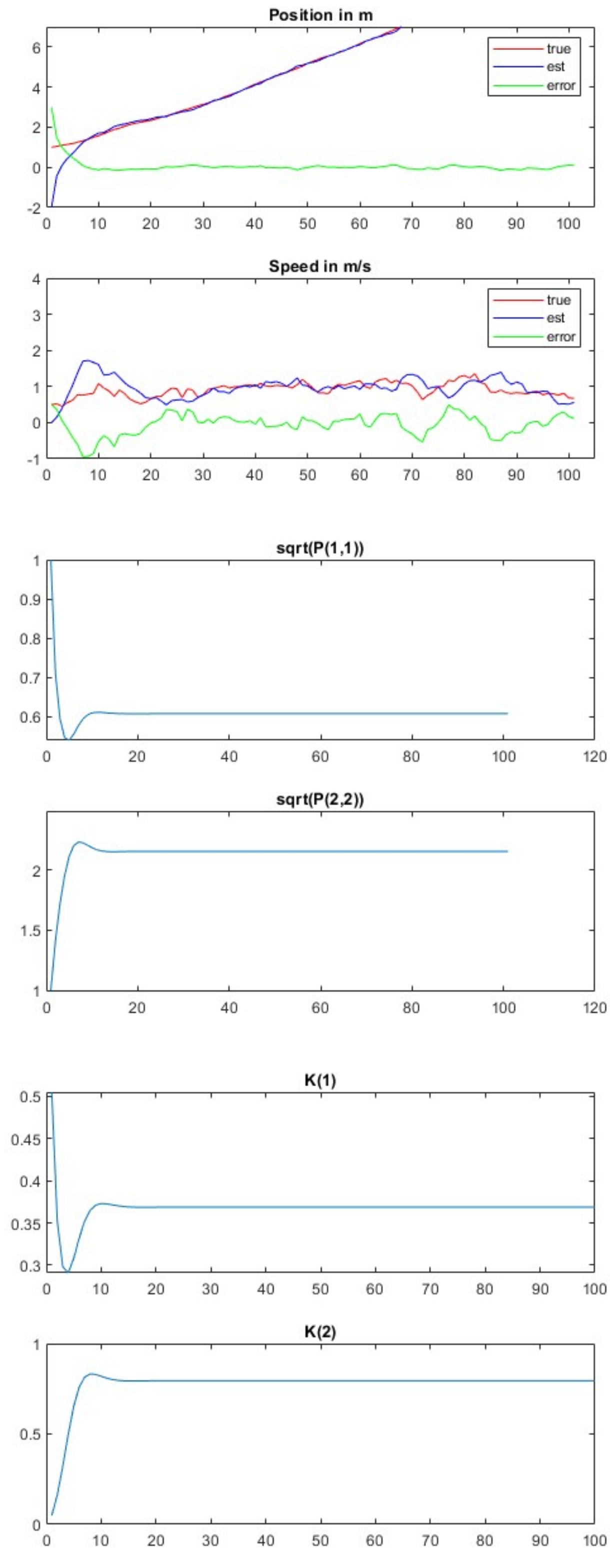

Figure 4: Increase Q and R 100 times

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)?
The too big P means large uncertainty of the true state. The kalman gain will be large, and weight more on the measurement model. The converge time does not change.And the error of the estimates would not be changed much.
The too small P means small uncertainty of the true state. The Kalman gain will be smaller at the beginning, and weight more on the prediction model. The converge time will be longer. And, the error of the estimates would not be changed much.

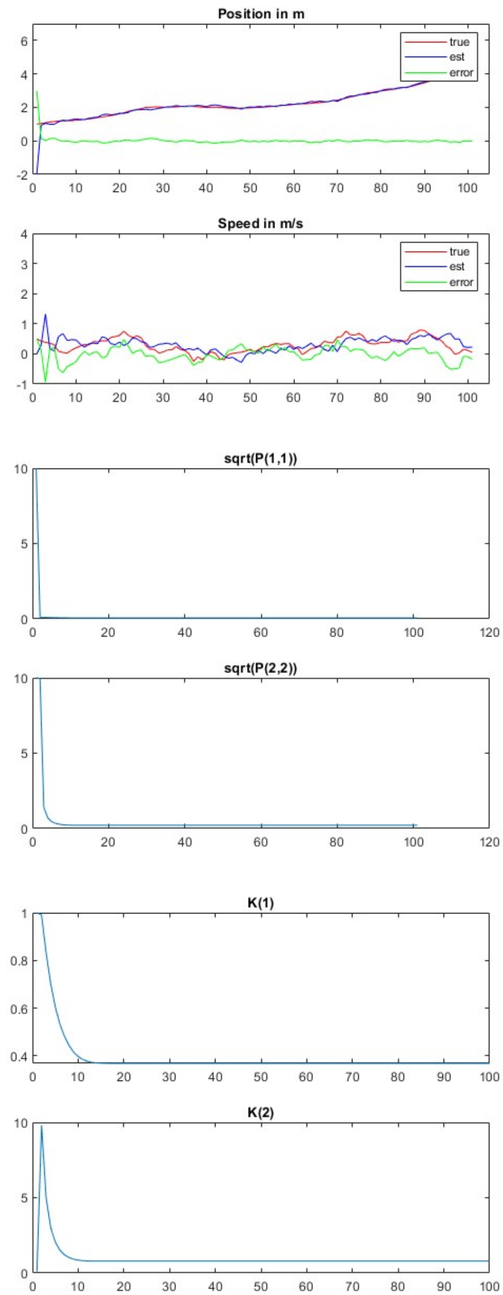

Figure 5: Increase P 100 times

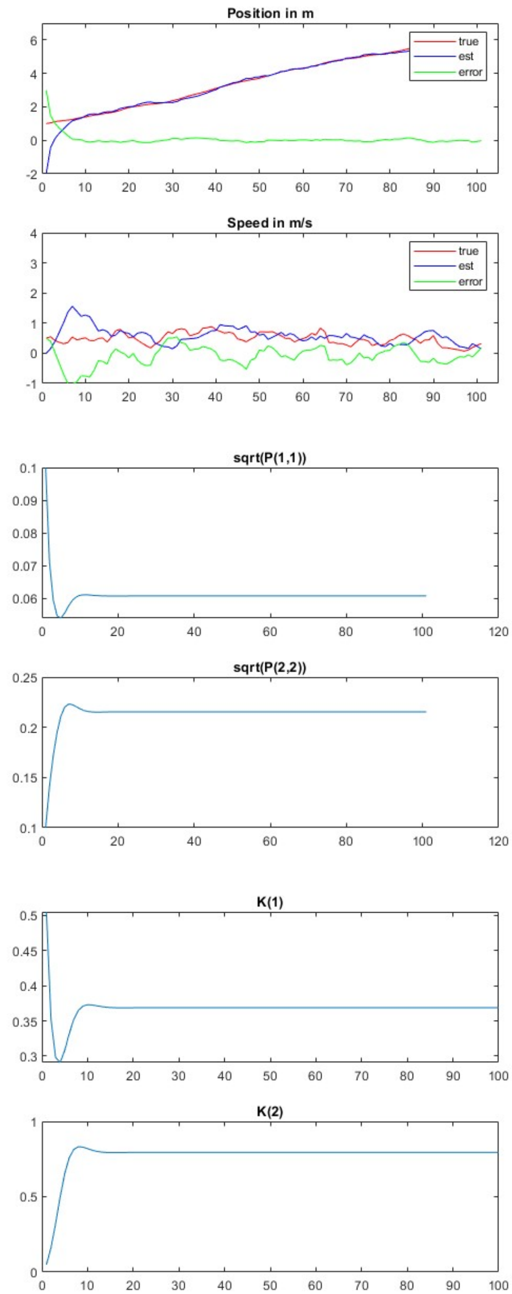

Figure 6: Decrease P 100 times

The too big initial value of xhat will extend the time of convergence and does not change the error much.
The too small initial value of xhat will decrease the time of convergence and does not change the error much.

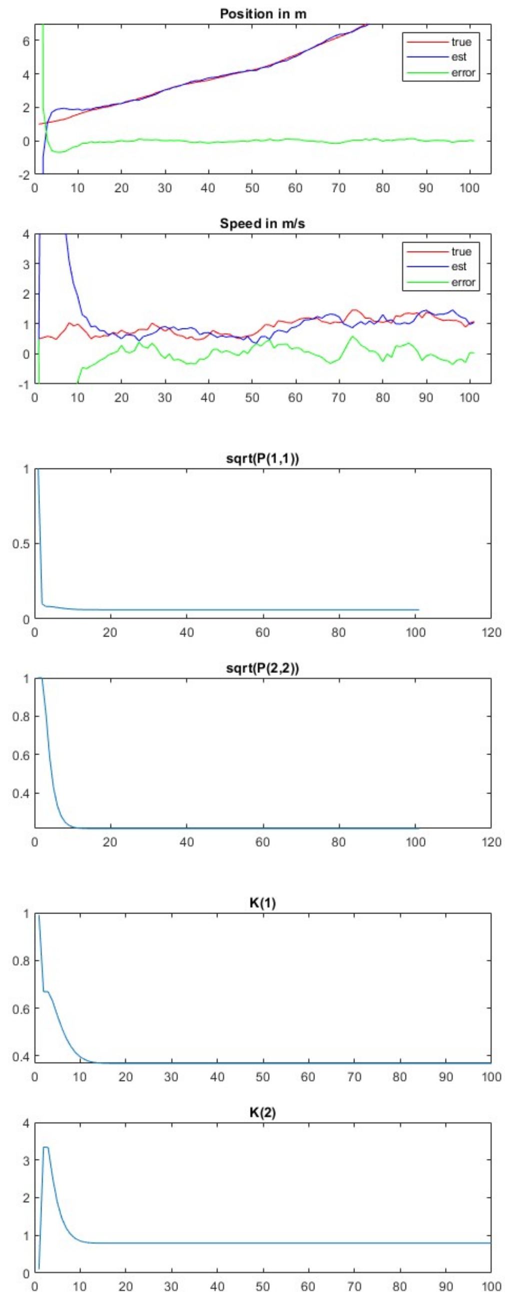

Figure 7: Increase xhat 100 times

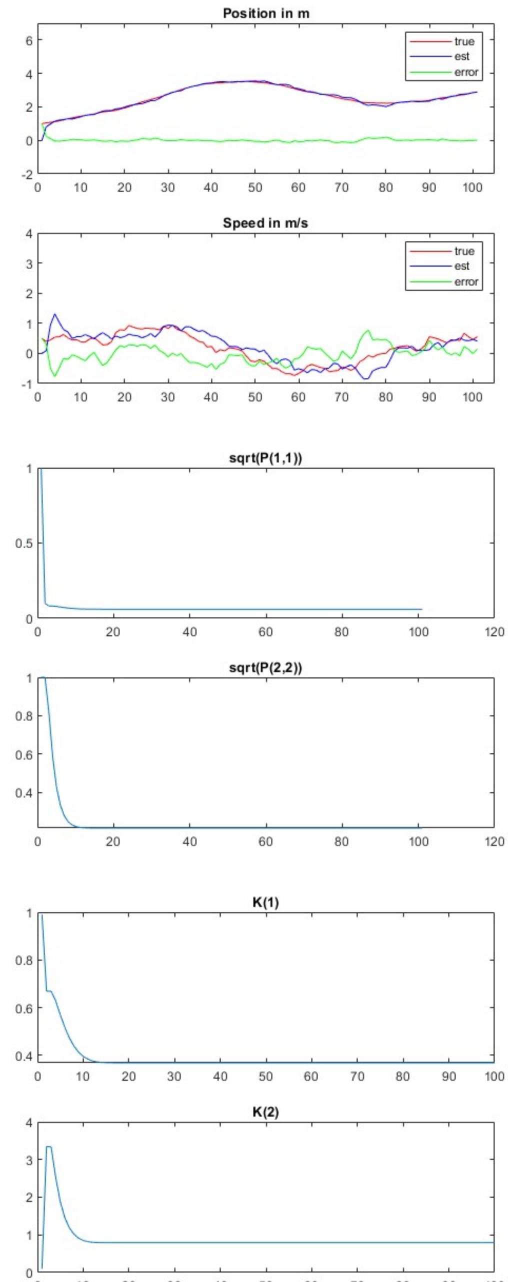

Figure 8: Decrease xhat 100 times

2.2 Main problem: EKF Localization

5. Which parts of (2) and (3) are responsible for prediction and update steps?
For (2):
the first line is responsible for both prediction and update, and this part is responsible for prediction: $\int p(x_t|u_t, x_{t-1})p(x_{t-1}|z_{1:t-1}, u_{1:t-1}, \overline{x}_0, M)dx_{t-1}$
For (3):
prediction: $\overline{bel}(x_t) = p(x_t|u_{1:t}, z_{1:t-1}, \overline{x}_0, M)$
update:$bel_(x_t) = \eta p(z_t|x_t, M)\overline{bel}(x_t)$

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 valid, because the measurement of every landmark is only depends on the position of landmarks in the map. Landmarks have no connection with each other, and each of them is a white Gaussion distribution.

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?
The bounds of δ_M are $\delta_M \in [0, 1]$, and it decides the value of chi square χ_n^{-2} cumulative probability, which is correspond to the value of λ_M. The δ_M increases, the λ_M will also increase. When we have reliable measurements all arising form features in the map, we can set a large threshold λ_M because few measurement should be viewed as outliers. On the other hand, we should set a small λ_M if we have unreliable measurements with many arising from so called clutter or spurious measurements.

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?
In the data association process, the calculated innovation $\overline{v}_t^i$ will be non-zero, and will cause the accumulated error in the $\overline{\mu}_t = \overline{\mu}_t + K_{t,i}\overline{v}_t^i$. Furthermore, it will also cause larger Mahalanobis distance, which leads to more measurements to be throw away as outliers.

9.How can you modify Alg 4 to avoid redundant re-computations?
The $\hat{z}_{t,j}, H_{t,j}, S_{t,j}$ are be calculated in every loop of i in z_t, however, they only depends on the landmarks j in M at time t. So, we can extract $\hat{z}_{t,j}, H_{t,j}, S_{t,j}$ only to the loop Landmarks j in M.

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?
Because the dimensions of $\overline{\nu}_t$ is same as $z_{t,i}$, which is 2×1. And $H_{t,i}$ is 2×3 So, the dimension of $\overline{\nu}_t$ is $2n \times 1$, and the dimension of $\overline{H}_t$ is $2n \times 3$. n is the number of inlier.

2.3 GUI Simulation

2.3.1 Dataset 1

In this situation, we can get accurate information from the laser scanner and the odometry, so we can model it with small uncertainties both process and measurement. As a result, we can get very small mean absolute error in all dimension.
Uncertainties:
Process:$[xy = 0.01m, \theta : 1°]$
Measurement:$[xy = 0.05m, \theta : 5°]$
The mean absolute err = $[0.00m, 0.00m, 0.39°]$ (The precision is 0.01)

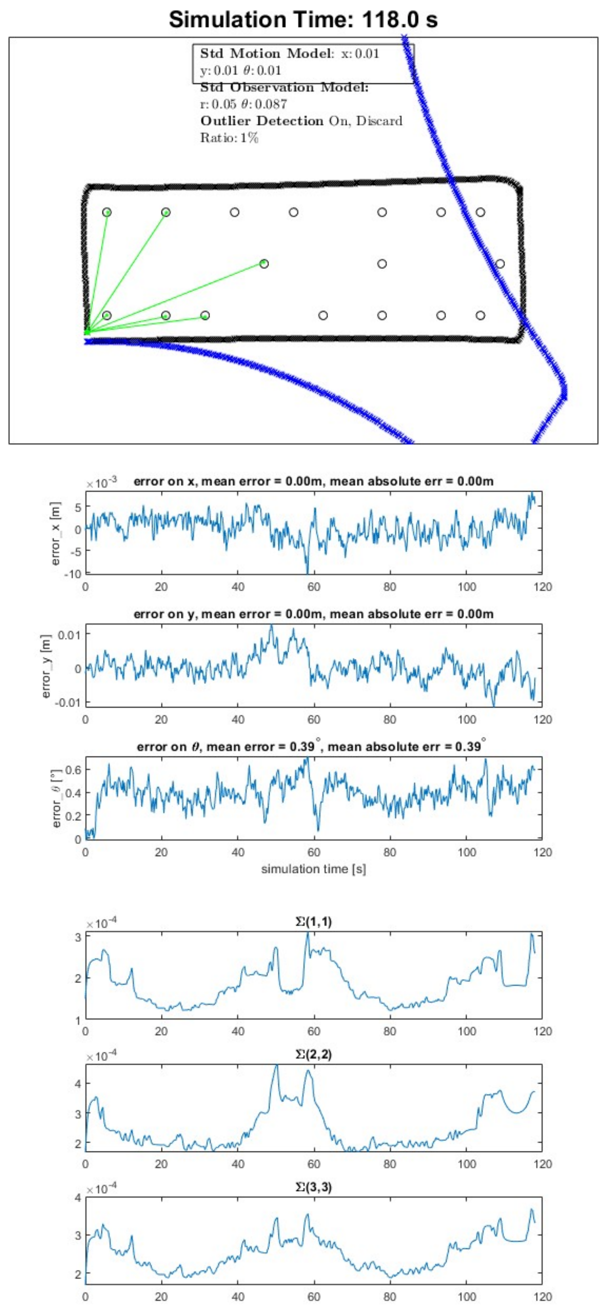

Figure 9: Dataset1: error and covariance

2.3.2 Dataset 2

In this situation, we need to detect and discard many outliers, we need modify the threshold value, because the outlier will effect the performance of EKF dramatically. Uncertainties:
Process:$[xy = 0.02m, \theta : 1°]$
Measurement:$[xy = 0.2m, \theta : 11°]$
Threshold of outlier detection: 5%
The mean absolute err $= [0.05m, 0.05m, 2.68°]$

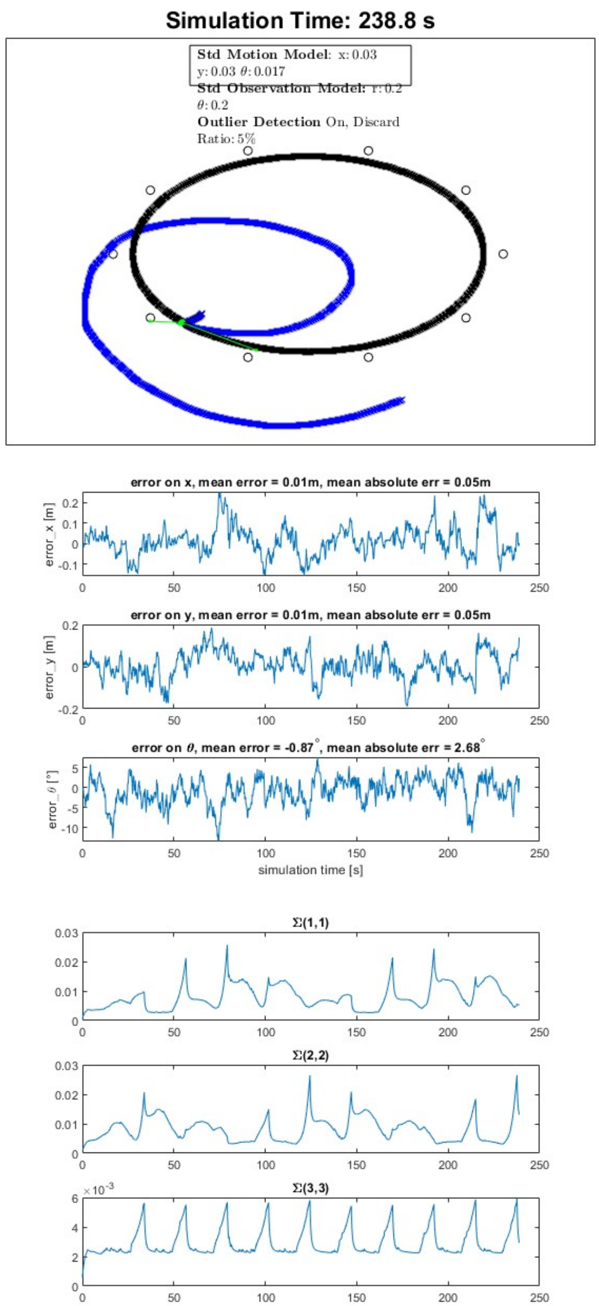

Figure 10: Dataset2: error and covariance

2.3.3 Dataset 3

Uncertainties:
Process:$[xy = 1m, \theta : 57°]$
Measurement:$[xy = 0.1m, \theta : 6°]$
1.Sequential update algorithm:
In this situation, the system does not have any odometry information, which causes large uncertainties. And the sequential update will cause large error because some bad information at the beginning affect the latter prediction and update process cumulatively.
The mean absolute err $= [4.46m, 4.80m, 25.22°]$

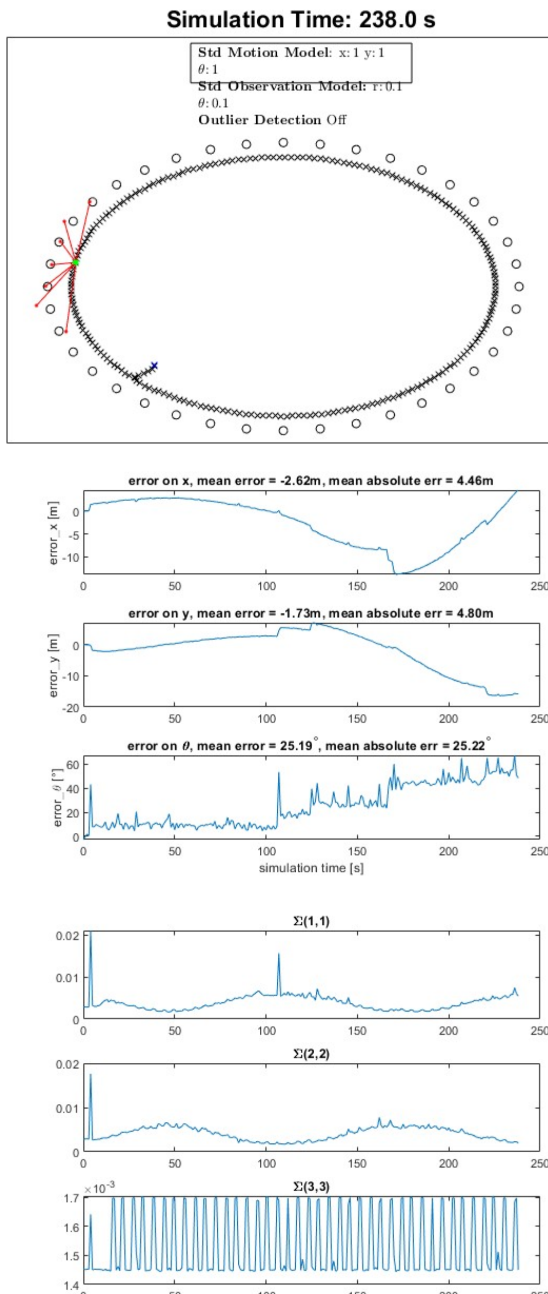

Figure 11: Dataset3 sequential update: error and covariance

2.batch update algorithm:
The mean absolute err $= [0.08m, 0.09m, 2.74°]$

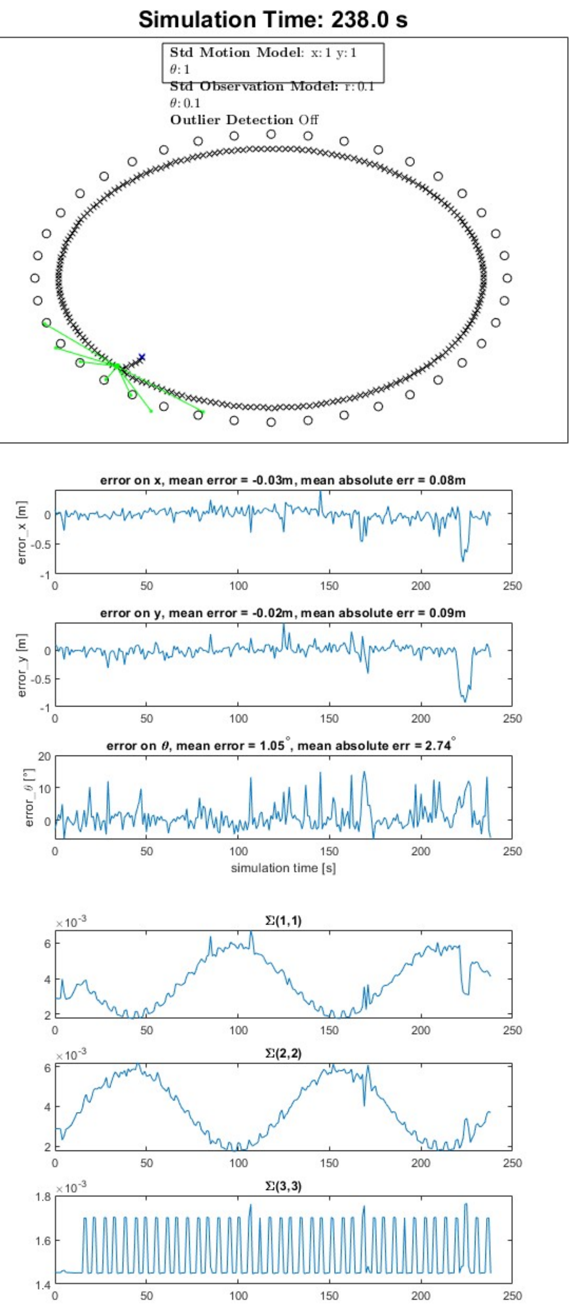

Figure 12: Dataset3 batch update: error and covariance