Applied Estimation Lab 1 - EKF

Johan Lewenhaupt, 930328-3478, johanlew@kth.se

November 19, 2017

1 Preparatory Questions

Answers to questions 2, 7, 11 and 15 were modified, after the session where the correct results were given (lecture 6).

1.1 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?

   Answer: A control u_t gives information about the change of states. It is called control due to the fact that it is an input that we decide (though with uncertainty). An example of a control u_t could be the input current to a dc motor. This motor will give output in rotations of a rod, which will spin faster the more input current we give.

   A state x_t is a number of variables that may impact a robot's future. For example, let's look at an industry robot arm with six degrees of freedom. Each degree of freedom translates to a possible direction of movement, and is in a standard robot arm characterized by a one dimensional change in a degree between to arms, that together sum up to a kinetic system. The states of this robot arm could then be these different angles in the arm, that determines an end-manipulator's position.

   A measurement z_t provides information about a state/states. For example, a robot with position and velocity as its states, a sonar mounted on it, could scan the surroundings and perhaps a wall. This enables a distance calculation between the robot and the wall, which would give positional information. Moreover, by repeating the sonar scan a time later it could find a new distance and thus get information about its velocity.

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

   Answer: No it can't. If the new measurement is bad, the update will not increase.

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

   Answer: The Kalman Gain ($K_t = \vec{\Sigma}_t C_t^T (C_t \vec{\Sigma}_t C_t^T + Q_t)^{-1}$).

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

   Answer: We see in the Kalman equations that we add Q to the predicted future covariance $C_t \vec{\Sigma}_t C_t^T$, and know that Q is simply the uncertainty in our predicted future state. Thus, a large Q will cause a very uncertain solution.

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

   Answer: A large Kalman Gain would mean that the measurement that is incorporated in each iteration will have a larger significance.

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

   Answer: During the prediction, the belief uncertainty increases. If we imagine a robot moving, it can be understood by the fact that we are incorporating a new movement during the prediction, and this new movement brings a new uncertainty that has to be accounted for. The update step however will reduce the belief uncertainty.

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.)

   Answer: As the Kalman Filter will find the correct solution, we know that it is also the optimal one, as no solution can be more correct. This can be showed by trying to find a better μ, which will not be possible.

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

   Answer: The Kalman Filter calculates the posterior probability, using Bayes theorem: $p(X|Y) = \frac{p(Y|X) \cdot p(X)}{p(Y)}$. This theorem can be seen as the following: $posterior = \frac{likehood \cdot prior}{evidence}$. MLE (Maximum Likelyhood Estimation) is maximizing the likelyhood term, and MAP (Maximum a Priori) is maximizing the prior term. The kalman filter calculates the maximum posterior using this formula, which can be done by both MLE and MAP, so a Kalman Filter can be either MLE and MAP.

1.2 Extended Kalman Filter

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

   Answer: The extended Kalman filter is as the name suggests, an extension of the classical Kalman filter. The problem with the Kalman filter is the fact that it incorporates a linear model. In reality, most situations are non-linear (for example a robot that wishes to turn will) which renders the Kalman filter deficient. By instead using a non-linear model that is then linearized (usually via Taylor expansion), we can overcome this issue. This new Kalman filter with a non-linear model is the EKF.

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

    Answer: No, the EKF will not always converge. And when it converges, it might converge to the wrong solution.

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

    Answer: We can try make sure that the initial estimate is close to the real solution, so that we easier find the solution. Moreover, we can increase the measurement noise covariance Q to so certain, which will however reduce convergence speed.

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 $p(x, y, \delta|r)$ look like? So a formula is not needed but describe it at least.

    Answer: Let's take a flying drone for example. The only thing we know is the distance r to the already known landmark. Moreover, since we do not know anything of the map or in which direction we are measuring, we can imagine the measurement as we are somewhere on the edge of this sphere. The belief is then distributed on the edge of this sphere with a gaussian noise, meaning we also uncertain of the radius r.

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

    Answer: We now also know the direction of the measurement (bearing). This means that the sphere belief would be reduced to a belief around a point of distance r to the landmark.

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

    Answer: If the robot has a very large uncertainty in its heading, but little uncertainty in its path, the uncertainty will look like a shell (edge of a sphere in the 3D drone case) as it moves. The shell will be thicker for larger uncertainty in movement estimation.

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

    Answer: The update will linearize the model, which will not match the uncertainty that is of a shell (in the 3D case), which can cause divergence.

2 Matlab Exercises

2.1 Warm Up Problems

1. **Question 1 Answer:**

   The white noise Gaussians are covariance matrices, and must thus be square matrices with the same size as the number of states for the equations to hold up. Here, ϵ_k is the noise of the states, and will thus be the covariance matrix of the noise of the position and the velocity. Secondly, δ_k is the measurement noise and will be the the covariance matrix of the noise measurement of the two states.

   Moreover, the definition of white noise is that it is noise with zero mean. Hence, in order to characterize a white Gaussian we only need the covariance matrix of the states.

2. **Question 2 Answer:**

x	Vector containing the states
xhat	Kalman fiter estimation of the states, position and velocity
P	This is the matrix determining the uncertainty of the states
G	Identity matrix, used for dimensions
D	Scalar value for the measurement noise, giving right amplitude
Q	Noise in the measurement that is used in the Kalman Gain
R	Process noise of the model, grows as we move
wStdP	A white noise that is applied to the position, with variance $(wStdP)^2$
wStdV	A white noise that is applied to the velocity, with variance $(wStdV)^2$
vStd	A white noise that is applied to the measurement of the position, with variance $(wStdp)^2$
u	A discrete time control signal that is equal to zero in our case
PP	A 4 times n+1 matrix that saves the uncertainty matrix P in each row (P is reshaped to a vector)

3. **Question 3 Answer:**

   The original system is illustrated in the images 1-3 below. Now,if the matrices of the modeled process/measurement noise (R and Q respectively) is increased a 100 times, we expect to have a very uncertain solution, which will after iterations get smaller and smaller. This can be seen in picture 4-6, especially at picture 6 we see that the uncertainty of the speed is very large. By decreasing the matrices a 100 times, we are very optimistic about our measurement and model, and expect to have a low uncertainty, which can be seen at picture 9. However, this creates a very fast system that could diverge at nonlinearities. Next, to get a feeling for the system we now change R and Q separately. In figures 10-12, where we have increased Q 100 times, we see that the update is not being very powerful (easily seen on speed at figure 10), which is due to the little effect of the measurement due to high noise. In figures 13-15 we decreased the measurement noise Q, and now see that the system has a hard time converging to a final value at the time limit given. This is due to the measurement being of too large significance. Figures 16-18 show the results of an increased R, which tells us that we have little faith in our model, and thus trust our motion very little, which in turn means that measurements will be more important. This can be seen in the high kalman gain. Lastly, in pictures 19-21, we have the opposite (now a low R), and see instead that the kalman gain is very small.

4. **Question 4 Answer:**

   Theory tells us that a good initial guess means faster convergence rate, as we need less information to find minimum of error. With a bad initial guess we risk falling in the trap of a local minimums here and there, and need more iterations to end up at a real convergence. Thus, a bad xhat should mean slower convergence rate. Moreover, a large P (large uncertainty in the states) will further increase the convergence rate. In pictures 22-23, we see the results, and that the results align with theory.

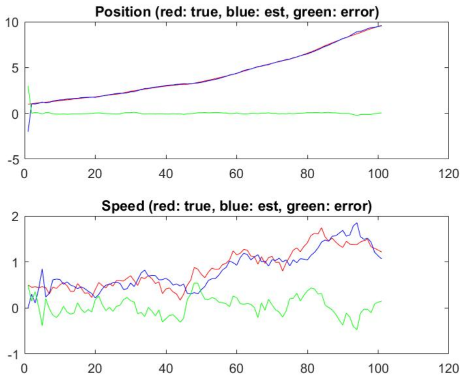

Figure 1: Position and speed of original system.

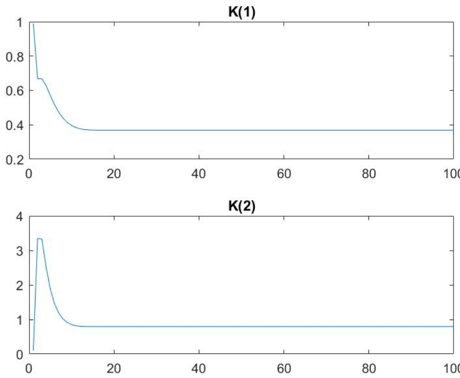

Figure 2: The Kalman Gain at original system.

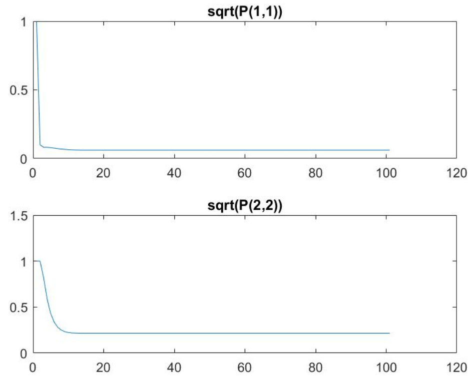

Figure 3: Uncertainty of states at original system.

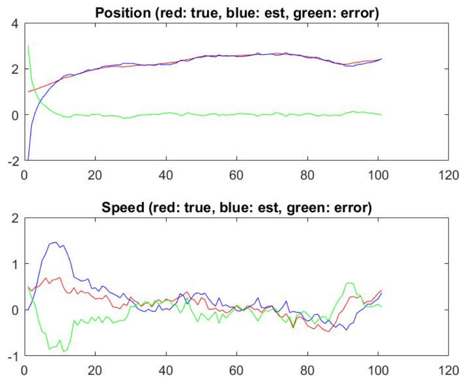

Figure 4: Position and speed of 100 times increased Q and R.

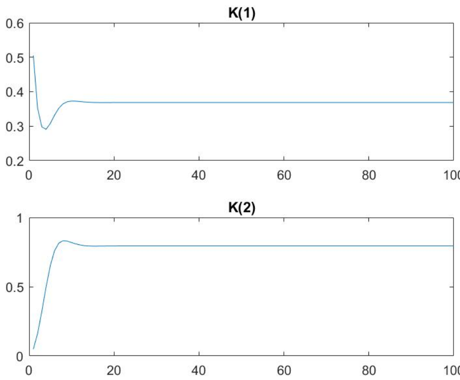

Figure 5: Kalman Gain with 100 times increased Q and R.

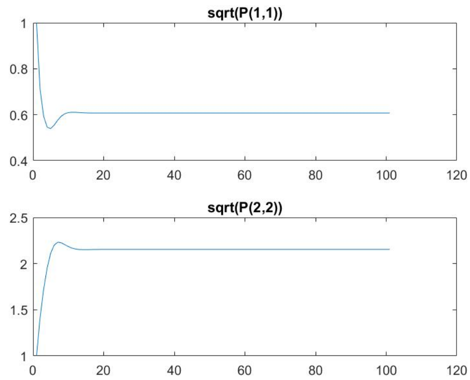

Figure 6: Uncertainty of states with 100 times increased Q and R.

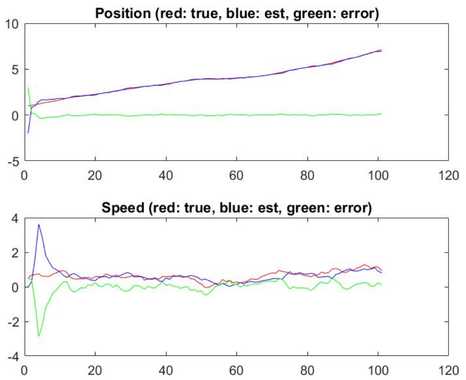

Figure 7: Position and speed of 100 times decreased Q and R.

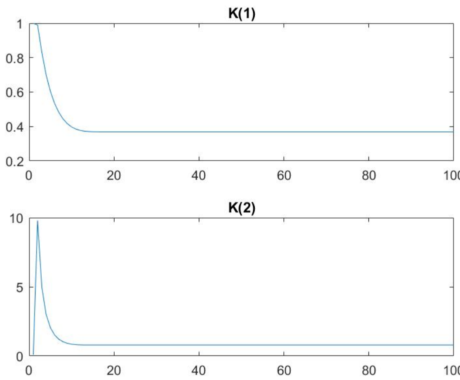

Figure 8: Kalman Gain with 100 times decreased Q and R.

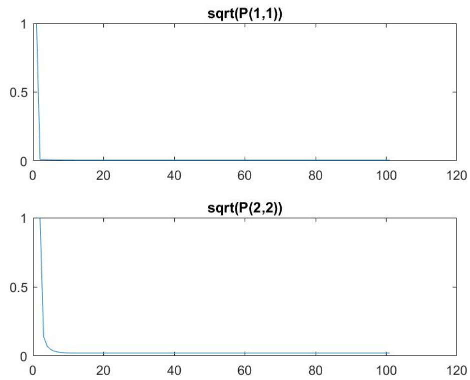

Figure 9: Uncertainty of states with 100 times decreased Q and R.

3 Main Problem: EKF Localization

5. **Question 5 Answer:**

   Looking at the equations below (equation 2 in the lab sheet), we see that the first line is the prediction step, as it contains states from previous steps. This is not the case in the update

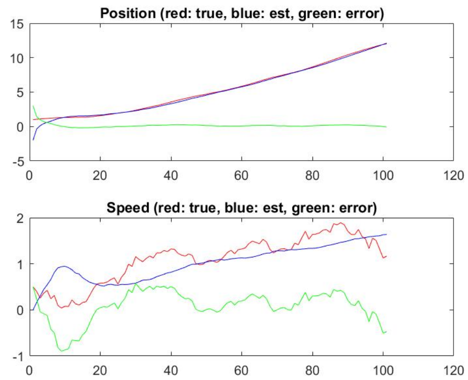

Figure 10: Position and speed of 100 times increased Q.

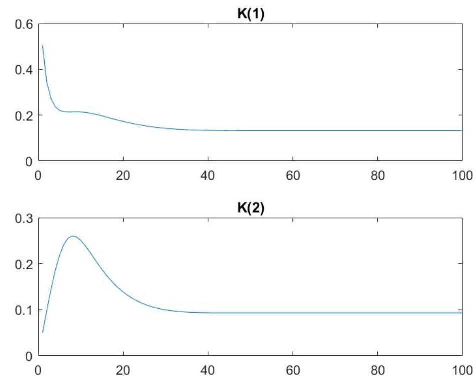

Figure 11: The Kalman Gain with 100 times increased Q.

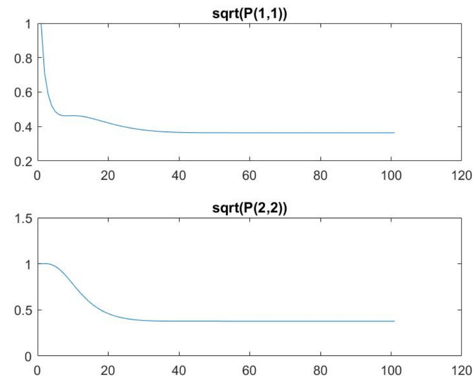

Figure 12: Uncertainty of states with 100 times increased Q.

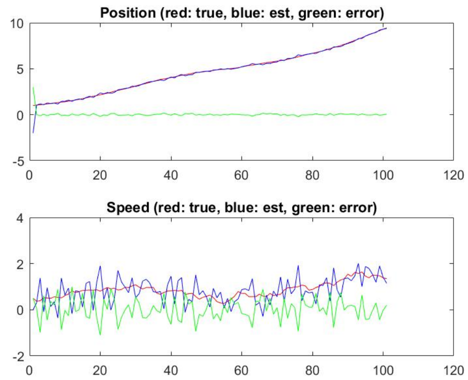

Figure 13: Position and speed of 100 times decreased Q.

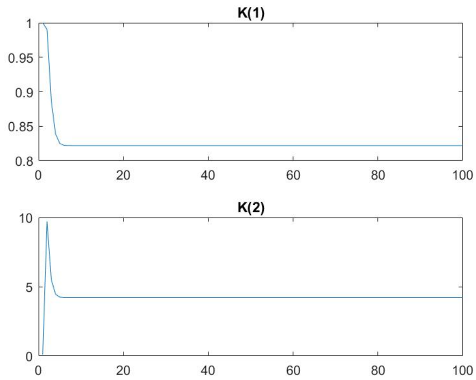

Figure 14: The Kalman Gain with 100 times decreased Q.

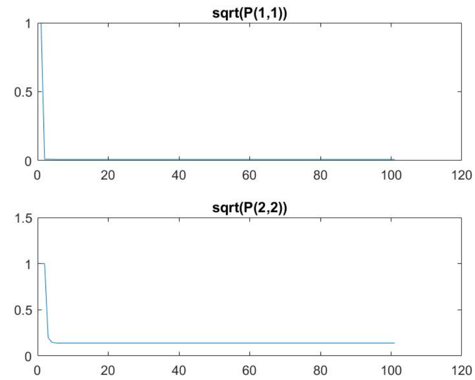

Figure 15: Uncertainty of states with 100 times decreased Q.

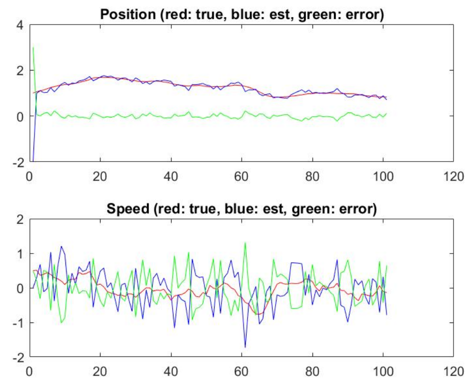

Figure 16: Position and speed of 100 times increased R.

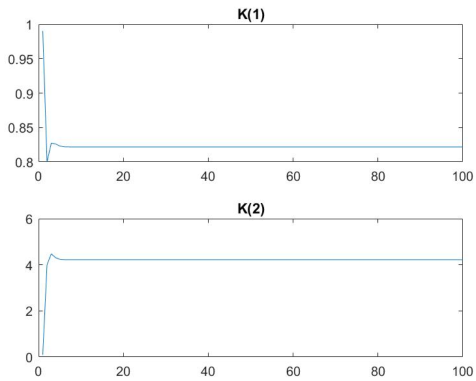

Figure 17: The Kalman Gain with 100 times increased R.

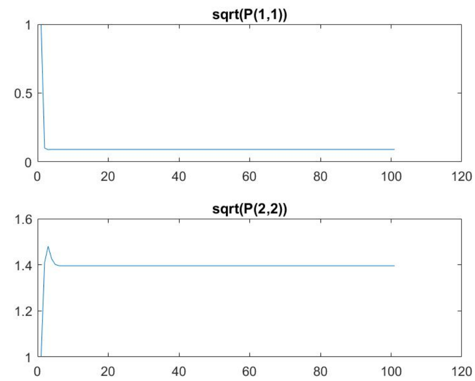

Figure 18: Uncertainty of states with 100 times increased R.

step. We also note that the second line here is not an update.

$$p(x_t|u_{1_t}, z_{1:t}, \vec{x}_0, M) = \eta p(z_t|x_t, M) \int p(x_t|u_t, x_{t-1}) p(x_{t-1}|z_{1:t_1}, u_{1:t_1}, \vec{x}_0, M) dx_{t-1}$$
$$p(x_0|\vec{x}_0) = \delta(x_0 - \vec{x}_0)$$

Again, in the equations below, we see that the second line requires states from previous time

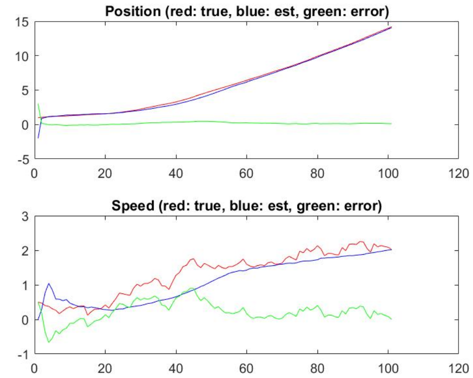

Figure 19: Position and speed of 100 times decreased R.

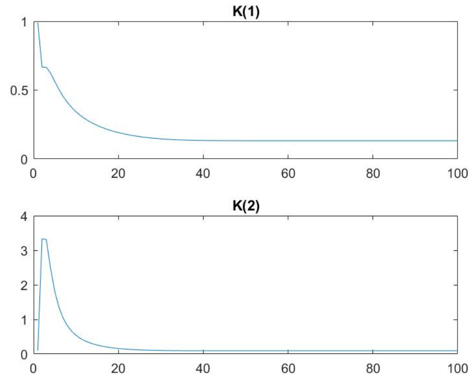

Figure 20: The Kalman Gain with 100 times decreased R.

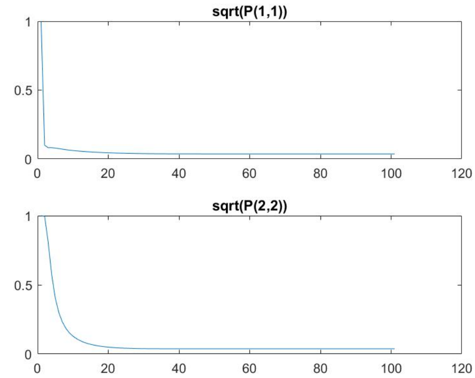

Figure 21: Uncertainty of states with 100 times decreased R.

steps, and thus it is the update step. Moreover, the first line is here an update step.

$$bel(x_t) = p(x_{t-1}|z_{1:t}, u_{1:t}, \vec{x}_0, M) = \eta p(z_t|x_t, M)\vec{bel}(x_t)$$
$$\vec{bel}(x_t) = p(x_{t-1}|z_{1:t-1}, u_{1:t}, \vec{x}_0, M) = \int p(x_t|u_t, x_{t-1})bel(x_{t_1})dx_{t-1}$$
$$bel(x_0) = p(x_0|\vec{x}_0) = \delta(x_0 - \vec{x}_0)$$

6. **Question 6 Answer:**

   Although it is not completely realistic to assume measurement Independence, it is a valid assumption when modeling, as it reduces the computational complexity. It has also been proven to work very well to assume independence between given observations. In fact, this assumption is called "Naive Bayes Assumption".

7. **Question 7 Answer:**

   The bounds for δ_M is found by looking at the inverse of the chi-squared cumulative distribution. This distribution is spread between 0 and 1, which are the boundaries for δ_M. Moreover, we notice from the chi-squared distribution that the choice of δ_M will affect the outlier-determination process. A larger δ_M will mean a larger λ_M, and thus cause acceptance of more values that could otherwise be noise, essentially over-fitting the system. Now, if all the measurements are from actual features, thus no outliers, we could use a larger λ_M to make sure we capture all of the points and not regard any of them as outliers. With unreliable measurements however, high probability of having outliers, we want a smaller λ_M to ensure filter out noise.

8. **Question 8 Answer:**

   When using the first measurements the update will not know what is noise and what is not. Thus, it might conclude that some features that should be outliers are instead landmarks. Since the mean is calculated before the uncertainty, the new mean that now could include outliers might now think of the real landmarks as outliers, which can eventually cause divergence.

9. **Question 9 Answer:**

   The first thing we observe in the batch-update algorithm is the double for-loop, which is very computationally heavy. By computing for all the landmarks at once, for example in a vector, instead of having the inner for-loop compute for each landmark at a time, we could reduce the double loop to a single loop. Moreover, when doing all the values as vectors, we do not have to later compute the vectors as in done in later stages of the update, and thus reduce re-computations.

10. **Question 10 Answer:**

The dimensions of $\vec{v}_t$ is n*I*J, where n is the total inlier number, I is the total number of observations and J the total number of landmarks. In the same sense, the dimension of $\vec{H}_t$ is n*I. In the sequential update algorithm, the number are different. We find that it is 1*J for $\vec{H}_t$, and I*J, where again I is the total number of observations and J the total number of landmarks. We see that the batch update is so computationally heavy as it uses vectors inside matrices.

4 Matlab Results

Firstly, I want to note that I did not get the *batch_associate.m* script to work, which was a problem on the last map. Anyhow,I just could not find any faults, so I proceeded forward. Instead, I gave a theoretic explanation as to what it should be where the script was used (sub-section 4.3).

4.1 map_o3.txt + so_o3_ie.txt

The following pictures (22-24) show the the results of this map. We conclude that the results are very good, our estimated position is pretty much spot on the true. This is done with $R = [0.04^2 \quad 0 \quad 0; 0 \quad 0.05^2 \quad 0; 0 \quad 0 \quad 0.2^2]$ and $Q = [0.25^2 \quad 0; 0 \quad 0.25^2]$. In picture 23 we also the see that the mean absolute error is smaller than 0.01 in all dimensions, which was the criteria.

4.2 map_pent_big_10.txt + so_pb_10_outlier.txt

In this map we now set the verbose flag equal to 3, to illustrate the scanner findings, which makes us see how the noise affects measurements. I ran it again with the same R and Q as previously. The results of this exercise are the pictures 25-27.

4.3 map_pent_big_40.txt + so_pb_40_no.txt

Now using $R = [1^2 \quad 0 \quad 0; 0 \quad 1^2 \quad 0; 0 \quad 0 \quad 1^2]$ and $Q = [0.1^2 \quad 0; 0 \quad 0.1^2]$, with $\delta_M = 1$, we get the results seen in pictures 28-30 with the sequential update algorithm, and the results seen in pictures 31 with the batch update algorithm (I set USE_BATCH_UPDATE equal to 1 to use it). Here, the batch update algorithm should work a lot better, but the reason that was not the case here is that I did not get my script to work fully. However, we know the batch update should be better here, as it processes all the observations at once, and is thus less sensitive to outliers, which caused the problems in the sequential update.

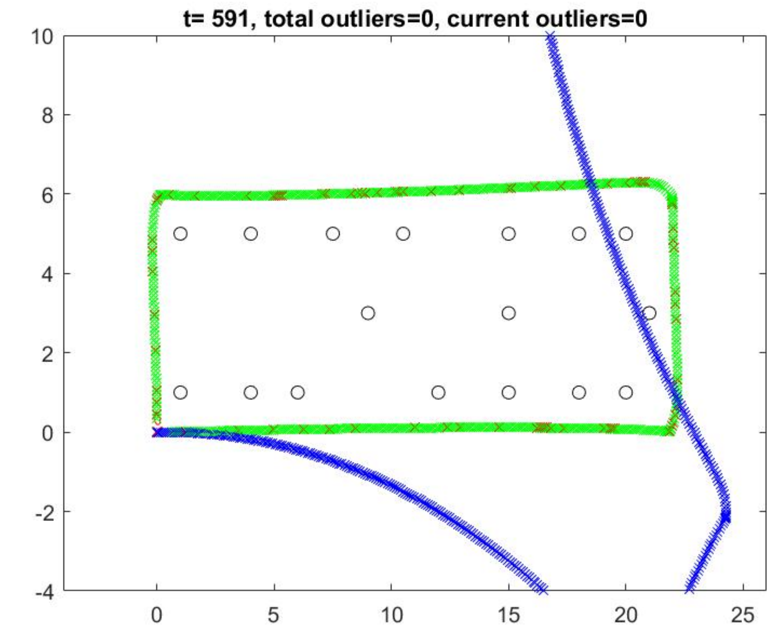

Figure 22: Green: true position, red: estimation, blue: odometry

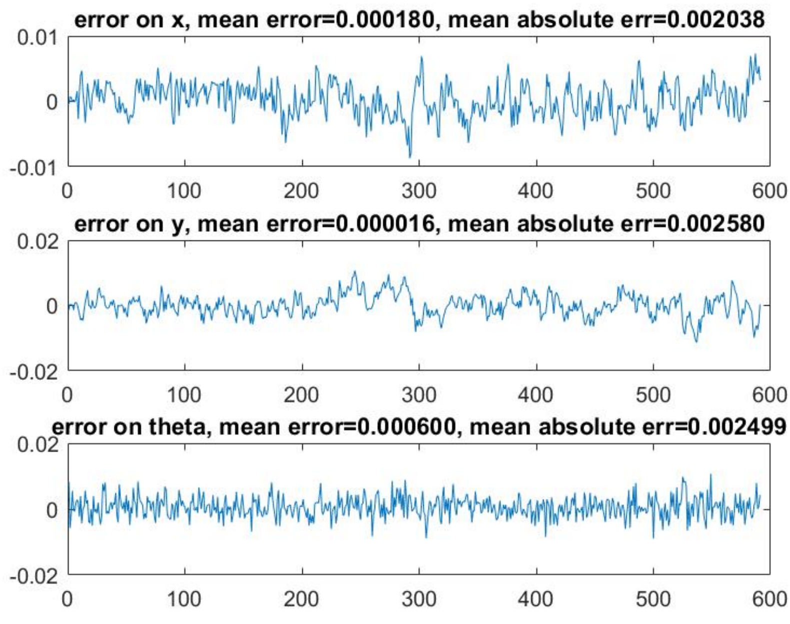

Figure 23: Different errors

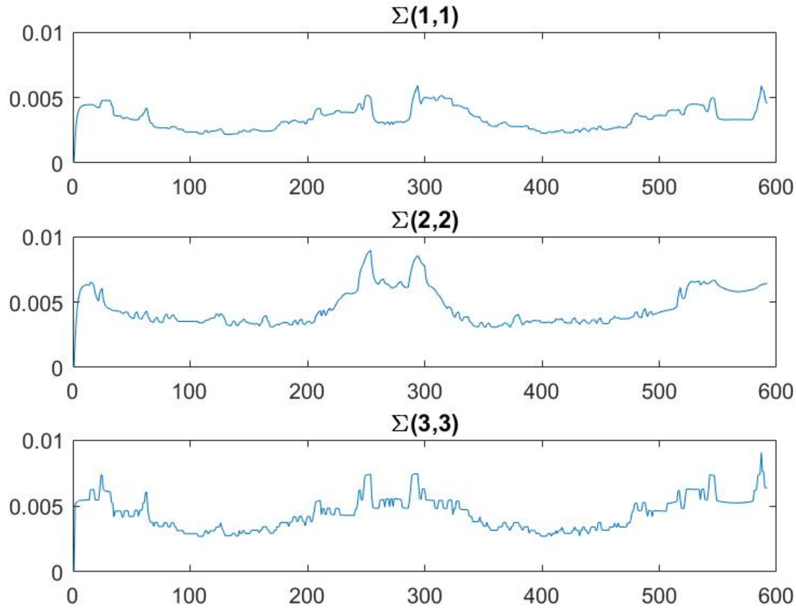

Figure 24: Uncertainties in states

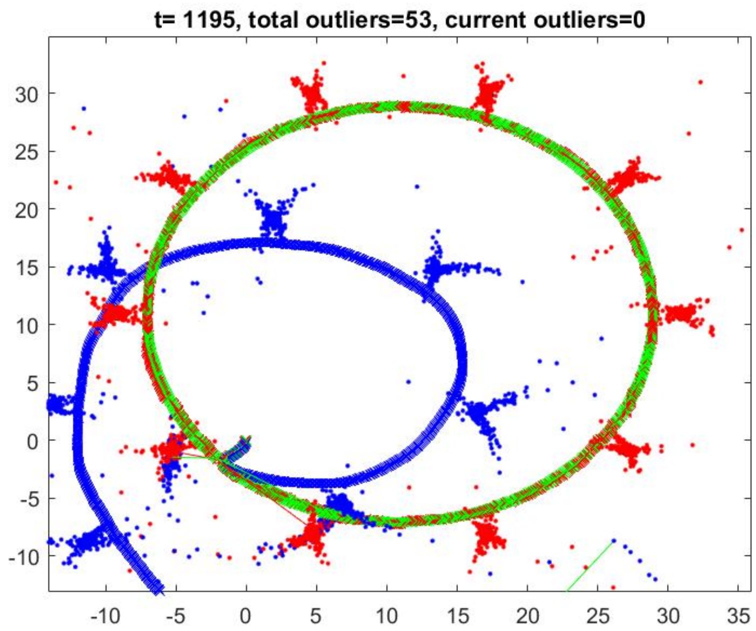

Figure 25: Green: true position, red: estimation, blue: odometry

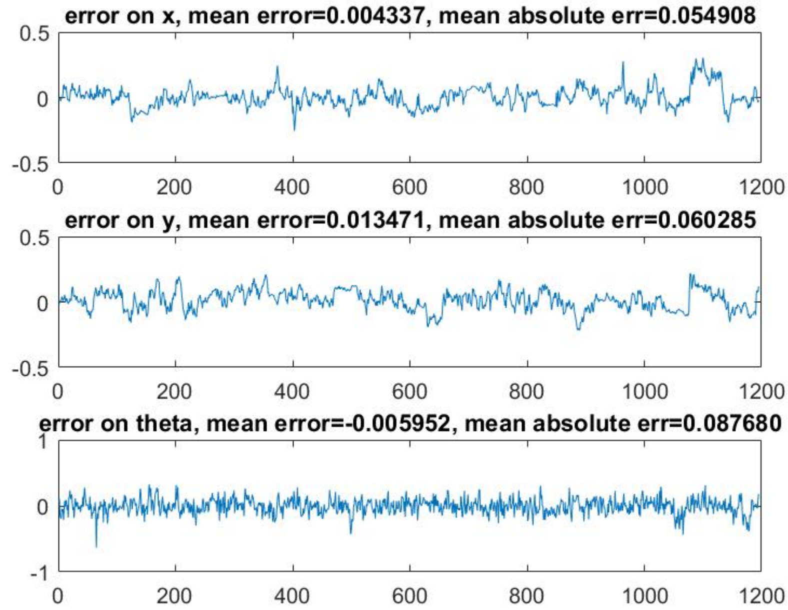

Figure 26: Different errors

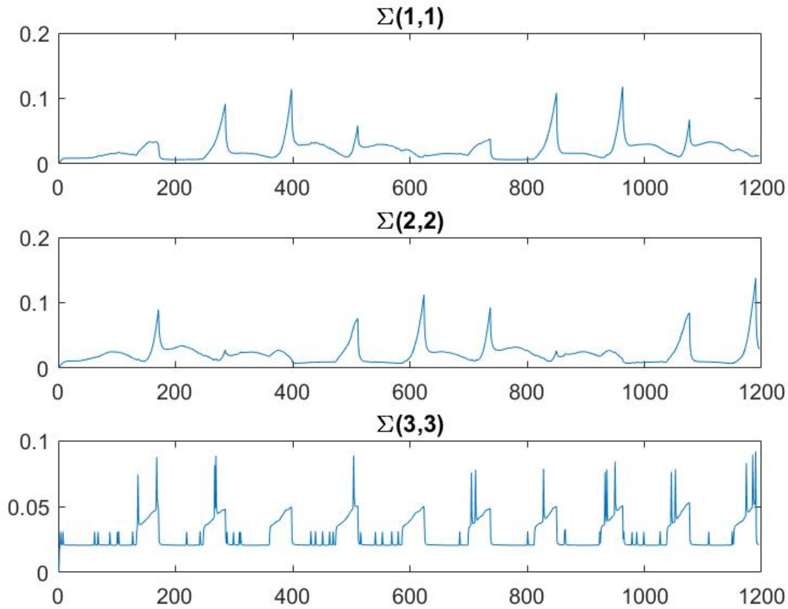

Figure 27: Uncertainties in states

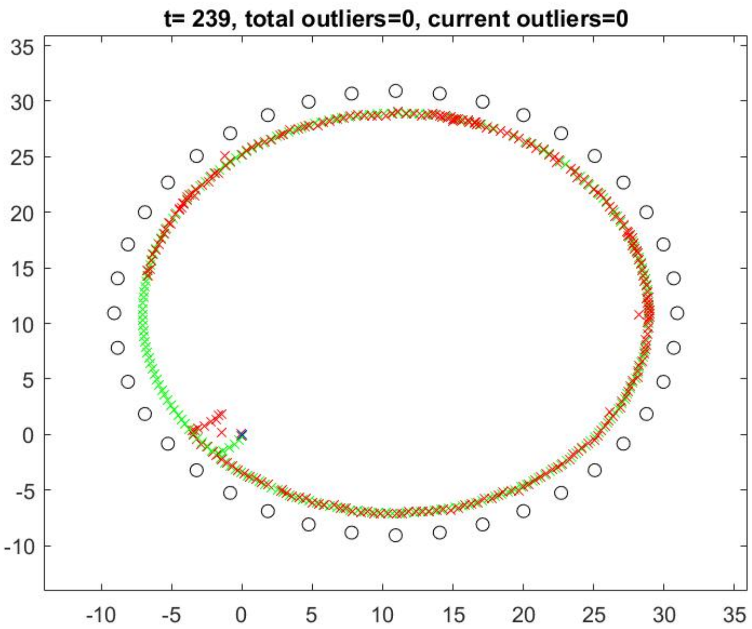

Figure 28: Green: true position, red: estimation

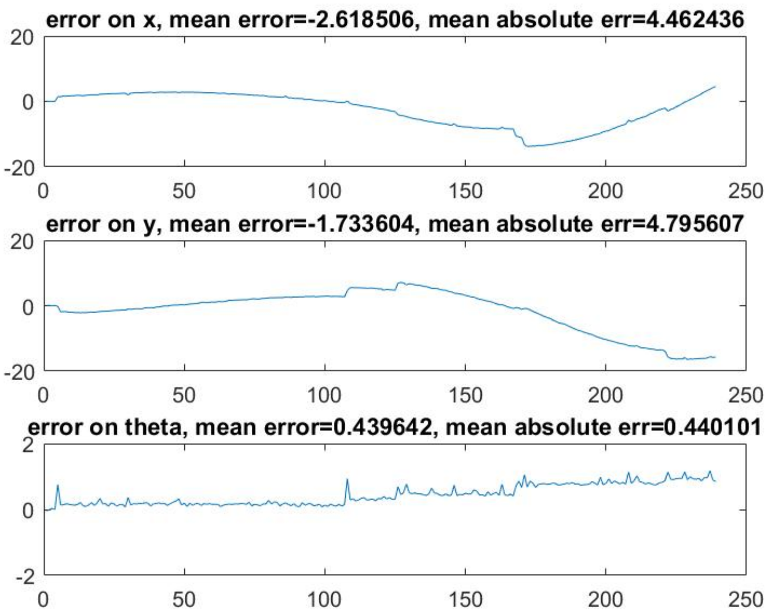

Figure 29: Different errors

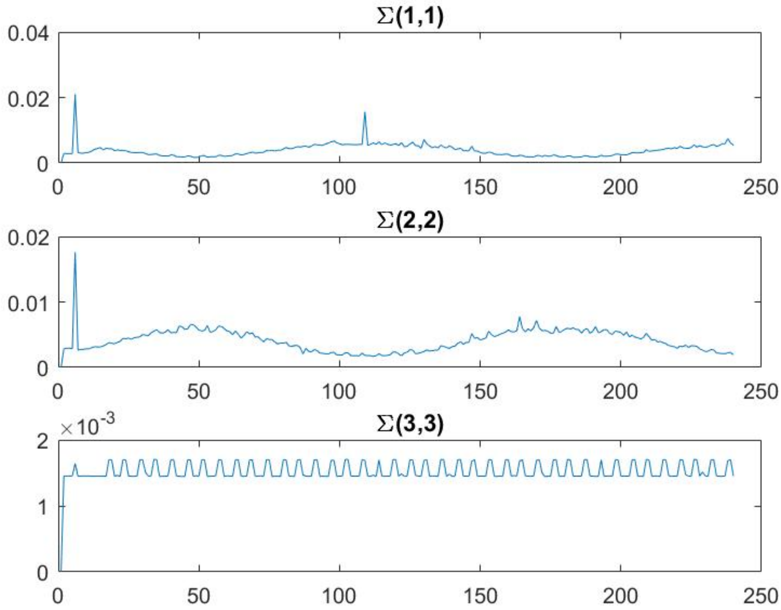

Figure 30: Uncertainties in states

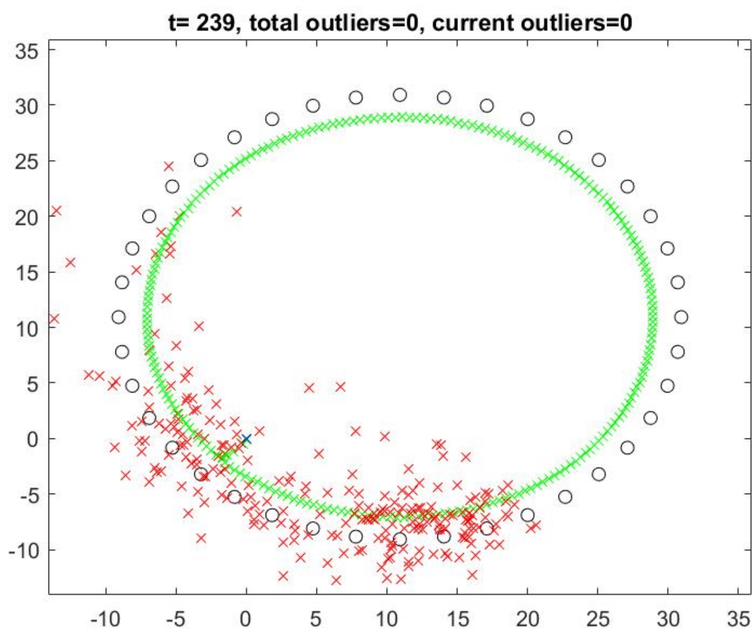

Figure 31: Uncertainties in states