Applied Estimation lab 1

Badi Mirzai

November 2019

1 Part 1 - Preparatory Questions

1. A control u_t is the control input of the systems, for example the heading and acceleration of a car, while the measurement z_t is what we measure of the actual state x_t of the system. For example can the state x_t be the position/state of the robot at the current time t, while the measurement z_t can for example be the distance measured from sensors to detect the surroundings.

2. The uncertainty of the belief can not increase during the update since this is when new measurement z_t are being captured. This means that the previous belief has just been tested and verified by the recent measurement and thus the next belief decreased since the measurement is fresh in its relevance. This is because the variance of the measurement is smaller when this happens.

3. It is the Kalman gain K_t that decides how much the measurements should be weighted relative to the belief during the update step.

4. If the covariance matrix Q_t is too large then the Kalman gain K_t will be small enough to be neglected relative to the belief, since the measurement noise in the data is high. This means that since the uncertainty of the measurement is large it will have almost no impact (relative to the belief) on the prediction, meaning our measured data z_t will not result in our predictions.

5. The Kalman gain K_t would give the measurements an increased effect on the measurements. The measurements are affected by the Kalman gain K_t.

6. The belief uncertainty often increases during the prediction step. This is since the covariance of the belief is increased during line 2 and 3 in the pseudo code of the Kalman algorithm.

7. The Kalman filter is optimal in the sense of having the minimum expected variance from the state estimate. The problem that is being

minimized in the Kalman filter is to minimize a function in the form $J(x) = (y - Hx)^T W(y - Hx)$. (see equation 3.26 in the book). In the case of Kalman optimization, the same thing occurs but with exponent, due to Gaussain distributions. Thus the problem becomes to minimize $e^{J(x)}$, which is essentially the same as mention before becaise it has its minimum at the same x as min $J(x)$.

8. MAP, since we use the prior knowledge of the previous distribution in combination with the measurement to estimate the position.

2 Extended Kalman filter:

9. The extended Kalman filter is an generalization of the linear ("normal") Kalamn filter, where it can handle none linear state transitions. This is done by linearizing the non linear function around the mean of the estimate and treating it as the linear Kalman filter around that linearization point.

10. No, it is not guarenteed to converge to a consistent solution. If the initial guess of the position is too far away from our linear point, then the linearization will not be accurate since it may differ a lot from the its actual value and therefor the error will be large enough to cause it not to converge.

11. The Kalman gain K may be increased in order to take the measurment data more into consideration. We may also need to decrease the covariance so that the model becomes more accurare, since the error will thus be smaller. The initial value of the estimate may be moved more toward the mean so that is does not converge to an inconsistent solution.

3 Localization:

12. It will be a circle with radius R around the landmark and it has an Gaussian distribution around the circle.

13. With the bearing it will be the same as before, only with the additional information of the robots orientation θ adn the heading along with the angle around the ring which is also gaussian.

14. The posterior will have the shape of a circle segment with radius R and the length of the circle segment is dependent on the variance. If it travels an incredibly large distance, then it will get closer to an circle that is thinner at one part and much thicker at the opposite part. But in general it will have a ARC shaped form.

15. The EKF update might go wrong if the noise in the model is not of Gaussian or if the motion model is not linear then there is a chance that the model will converge from its actual value.

4 Part 2 - MATLAB Exercises

4.1 Warm-up problem: Linear Kalman filter

- Question 1: ϵ_k has the same dimension as the state vector x_t, since it is the error term in the stet of the system. In this case ϵ_k is a $2 * 1$ matrix, due to the dimensions of A and B. δ_k has the same dimension as the measurement z_k, which in this case is just a scalar. To define a white Gaussian noise, we only need the mean and the co-variance of the noise.
  - Question 2:

Variables	Meaning
x	The actual state of the system in the simulated world.
xhat	Describes the prediction of the next step of the system.
P	The covariance of the predicted step xhat i.e how much we can trust the prediction.
G	Weight of the process noise
D	Weight of the measurement noise
Q	The covariance matrix of the Gaussian distribution (measurement noise).
R	Covariance matrix of process noise.
WStdP	Weight of the noise on simulated position.
WstdV	Weighted noise of the simulated velocity.
vStd	The simulated measurement noise of the position.
u	Control input is the system. For example steering angle and acceleration.
PP	Matrix storing the covariances for the positions during the Kalman filteer process in the prediction of the position. Estimate error of the covarinace.

- Question 3:

  When simulating the system with the normal values given, the following plots were obtained: (show some plots of normal case).

  When the covariance matrix Q of measurement noise were increased by a factor of 100, it could be seen from the plots that the Kalman gain was decreased in both its position and velocity. This is because the system relies less on the measured data and more on its modeled behaviour. The velocity was also deviating more its true value, with a smooth curve. This is since the system relies much more on the predictions, which resulted for a longer time for the position to converge.

  When the covariance R of the process noise was increased by a factor of $x100$, it could be seen that the system acts more like a controller. With this I mean that the velocity oscillates much more around its

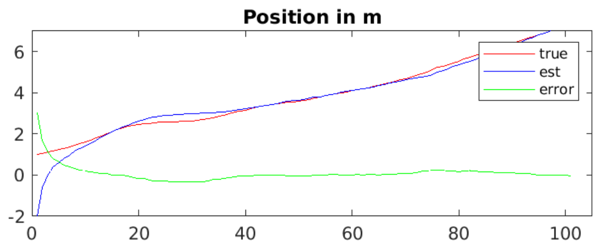


Figure 1: Position with x100 increased covariance of measurement noise.

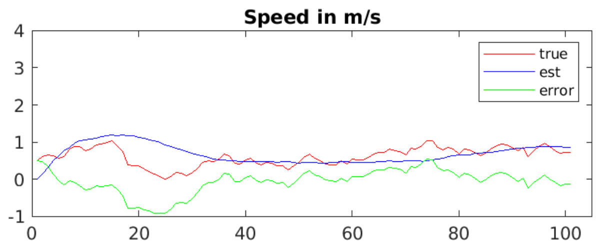


Figure 2: Speed with x100 increased covariance of measurement noise.

true value, since it's trying to compensate from the measured data (high Kalman gain). The speed will therefor have a large variance and a higher error, but the position will therefor be much more stable and the position will also have a smaller variance. The error of the position will therefor be much less, oscillating much less around zero. The Kalman gain was therefor much higher in this case, since the system takes much more into account the measurement, while less into account the modeled in the Kalman gain.

Both of these cases were aligned with my predictions of the systems behaviour, which can also be seen looking at the Kalman filter algorithm.
With the case of when both of the covariances Q and R were increaed the same amount, the variance (standard deviation) of the position didn't change as much from the normal case. But the variance of the velocity increaed and ended up convering to a much higher value than before, since the measurement noise and process noise increased.

- Question 4: If the initial $\hat{x}$ is shifted too far away from its initial actual position, it will take longer time for the Kalman filter to converge. Hence the error is bigger in the beginning but converges eventually to zero.
  If P is decreased, then the rate of convergence will increase, since the probability that the car is withing its mean is less likely and thus it need more time to correct the position. If P is small, and $\hat{x}$ close to

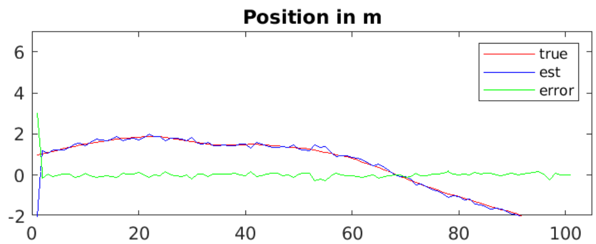


Figure 3: Position with x100 increased covariance of process noise.

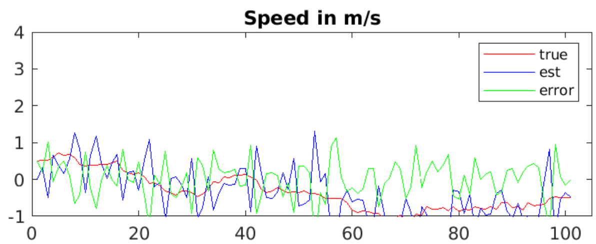


Figure 4: Speed with x100 increased covariance of process noise.

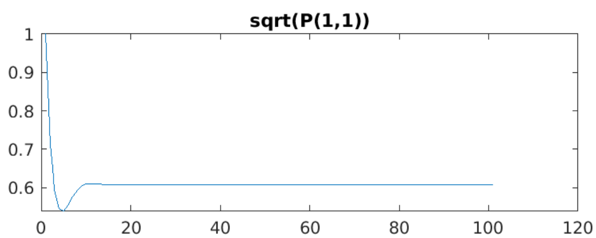


Figure 5: Standard deviation of position with x100 increased covariance of process noise and measurement noise.

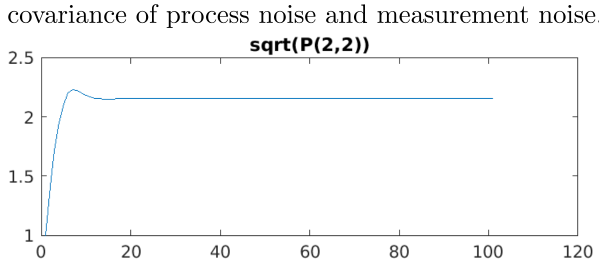


Figure 6: Standard deviation of velocity with x100 increased covariance of process noise and measurement noise..

real value, we the system will converge faster, since we are close to the real value with a small covariaince. The opposite occurs when P and $\hat{x}$ are large.

4.2 Main problem: EKF Localization

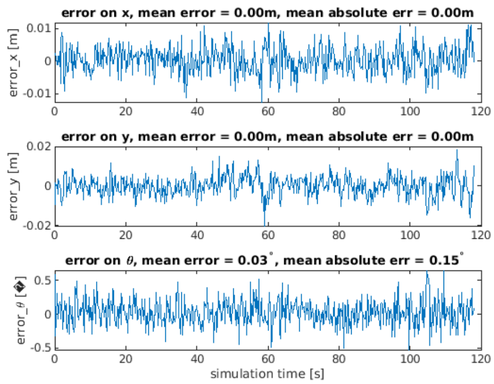
- Question 5: The prediction step is $\bar{bel}(x_t) = P(x_t|u_{1:t}, z_{1:t}, \bar{x_0}, M) = \int p(x_t, u_t, x_{t-1})bel(x_{t-1})dx_{t-1}$, where we predict the nest step with the previous measurement along with the our expected model.
  The update step corresponds to $bel(x_t) = p(x_t|u_{1:t}, z_{1:t}, x_0, M)$, where we update our state based on the new measurement. $bel(x_0)$ is our initial update based on our initial guess of the state x_0 at the beginning of the algorithm.
- Question 6: Yes, this is a valid assumption, since each sampling of the measure in the algorithm are not dependent of each other.
- Question 7: The bounde of δ_M is $[0, 1]$, since it represents a probability. if $_M$ is high, then we will take into account that the landmarks are outliner, where if we choose it low we will discard the possibility of outliners. If we will have reliable measurements, we can assume that λ_M is a high threshold of our distance for detecting outiners.
- Question 8: The first measurement will always be there, since there is nothing to compare it with and it will therefor be added to all the other measurements. This is dangerous, since it might give incorrect measurement for the entire sequence.
- Question 9: One could use the Sparse command in Matlab to reduce the computations for matrix multiplication with matrices that have many zeroes (diagnoal elements). A stupid solution would be to ignore all the ourliners, and assume thet our measurements are reliable. One could also stop going through all the other landmarks when there are 4 that have given an estimated position.
- Question 10: $\bar{\nu}$ has dimensions $2nx1$ and $\bar{H}$ has dimensions $2nx3$ in the batch update algorithm. In the sequential update algorithm, $\bar{H}$ has dimensions $2x3$ and $\bar{\nu}$ has $2x1$. Looking at the algorithms, one could see that batch update uses all the features to decide the outliners, while the sequential only takes them one at a time.

4.3 Simulating data sets

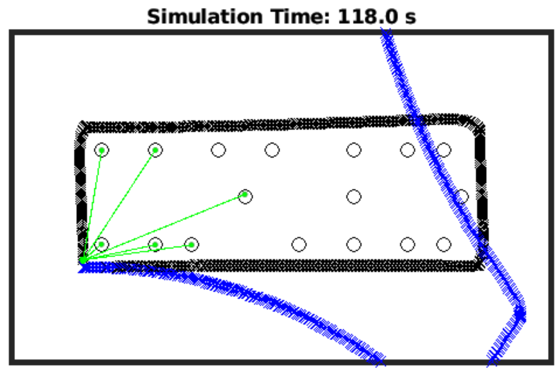
All the functions were written according to the algorithms on the instructions, and tested before simulating the data sets.
The first data set was simulated and the following results were obtained,

In the data set 1 (see figure 10a and 10c) it would be seen that the implemeted algorithms were correct, since the absolute mean error were small enough to be accepted.

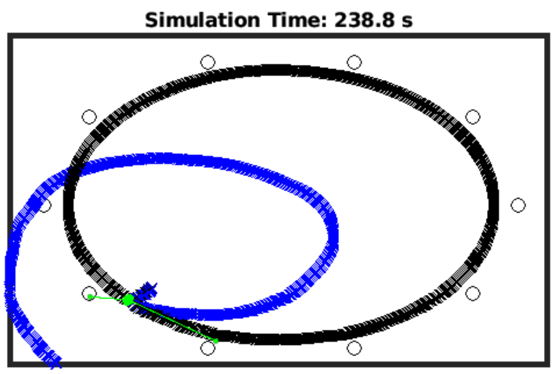
One coule see from the data set 3 that mean absolute errors were muchhigher when using only the sequential update than the batch update algorithm, which was expected and also discussed earlier in the lab.



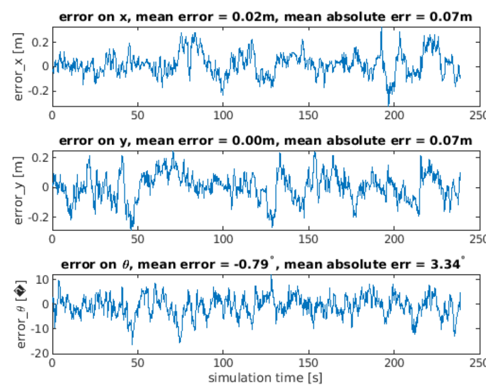
(a) Errors for the data set 1



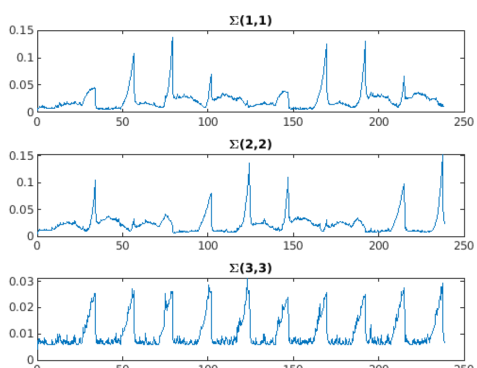
(b) Path for data set 1.



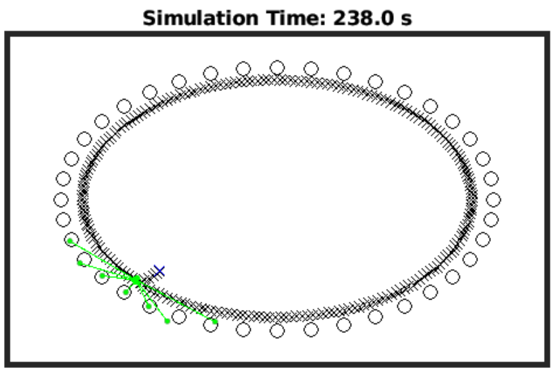
(a) Path for dataset 2



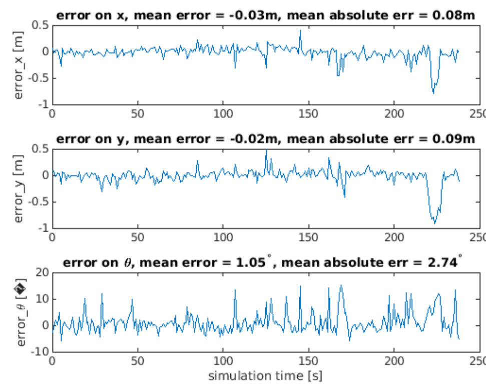
(b) Error for dataset 2.



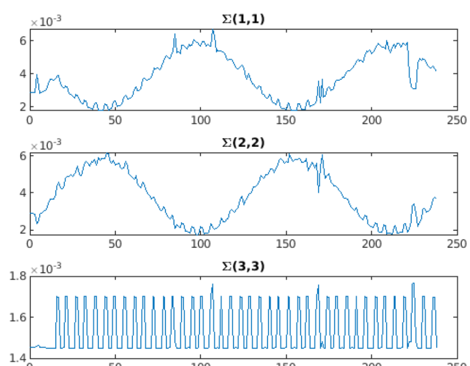
(c) Covariance for data set 2.



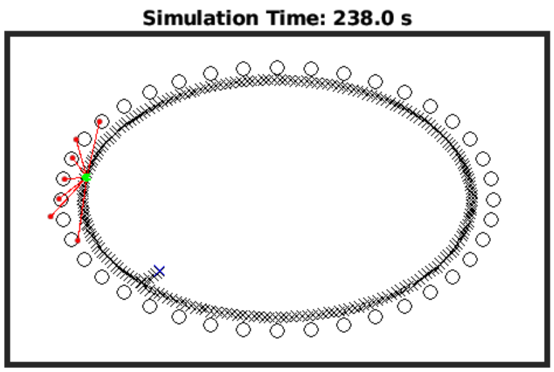
(a) Path for data set 3 with only batch update.



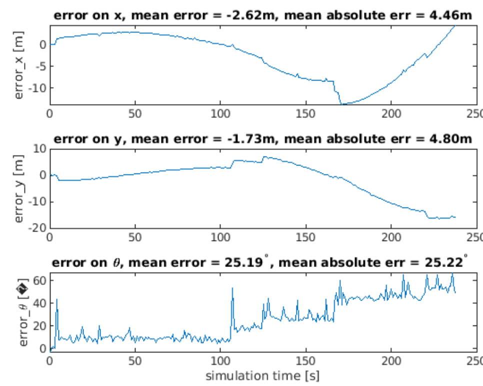
(b) Error for data set 3 with only batch update.



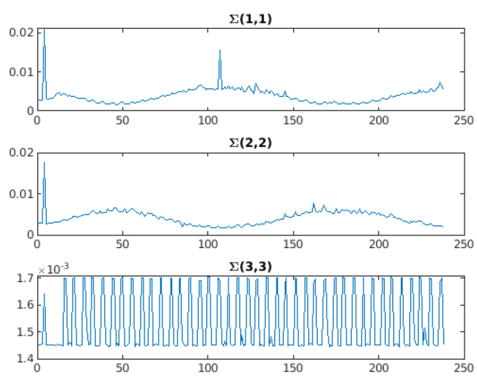
(c) Covariance for data set 3 with only the batch update.



(a) Path for data set 3 with only sequential update.



(b) Error for data set 3 with only sequential update.



(c) Covariance for data set 3 with only the sequential update.