EL2320 – Applied Estimation
SCR15
Martin Isaksson, misakss@kth.se

Laboration 1

2015-11-19

Laboration 1

Part 1

1. Control **u** is the input to the system, e.g. **u** can be the inflow into a water tank. What we measure is **z**, which can be, e.g. the outflow of a water tank. The state is **x**, which we are trying to estimate, i.e. the true value, e.g. the outflow of the water tank.
2. $\sum_t = \overline{\sum_t} - K_t C_t \overline{\sum_t} \leq \overline{\sum_t}$ no, because $K_t C_t \overline{\sum_t} \geq 0$
3. R_t from Kalman gain $K_t = \sum_t C_t^T R_t^{-1}$ decides the weighting between the measurements and the belief.
4. It will be pessimistic, which makes the data association ambiguous.
5. To decrease R_t in the Kalman gain.
6. $\overline{\sum_t} = Q_t + A_t \sum_{t-1} A_t^T$, it increases often/most of the time. Q_t is chosen large in general, and the other term is often positive.
7. Because it's Gaussian. The Kalman filter gives the true posteriori distribution for a linear Gaussian system, i.e. the Kalman filter is the Gaussian we are trying to estimate.
8. If we start with no prior knowledge of the distribution (Gaussian white noise), we have MLE. If we start with prior knowledge of the distribution (Gaussian priori distribution), we have MAP.
9. $G_t \leftrightarrow A_t$, $g(u_t, \mu_{t-1}) \leftrightarrow A_t \mu_{t-1} - B_t u_t$, $H_t \leftrightarrow C_t$, y_t is defined non-linear wrt. $\bar{\mu}_t$
10. No, the distortion of Gaussians by the non-linear transform can be an issue and make it not converge.
11. We can increase the modeled uncertainty, i.e. we can inject noise. Increase R_t, Q_t but not both at the same time.
12. It will look like a donut around the landmark. Because we don't know anything about ϑ, it will be modelled as a uniform distribution over $-\pi$ to π.
13. Now it looks more like a "C".
14. With linearization (EKF) the posteriori will be a straight line. If not EKF, it will be in shape of a half circle.
15. EKF uses linearization (with help of Jacobian), which makes the posteriori a straight line instead of a curved line. Because of this the robot doesn't really know where it is.

Part 2

1. $\dim(\varepsilon_k) = 2$, $\dim(\delta_k) = 1$, μ & σ^2 are the parameter that is needed.
2.

x	position & speed, state-vector
xhat	estimate of x (usually as μ)
P	Σ in Gaussian, the covariance
G	Filter model noise (coefficient)
D	Filter measurement noise (coefficient)
Q	GRG', covariance for v
R	DQD', covariance for w
wStdP	noise coefficient
wStdV	noise coefficient
vStd	noise coefficient
u	control signal
PP	Saves how P changes

Table 1 Parameters

3. I expect that is we increase/decrease the covariance matrix. Increase R but remain Q makes it faster. Increase Q but remain R makes it slower. If we increase both at the same time, the filter looks different, but it's not faster or slower. If the Kalman gain is faster/smaller, we get the estimate faster (& vice versa). This is from the observations below.

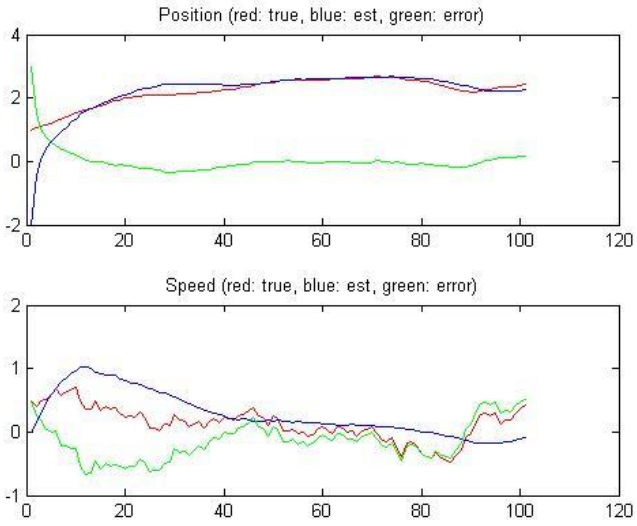


Figure 1 Large Q, the estimate against the true value

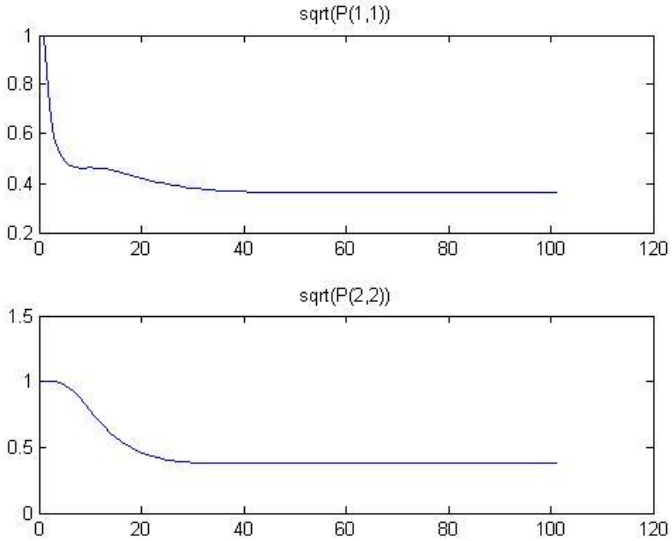


Figure 2 Large Q, the standard deviation

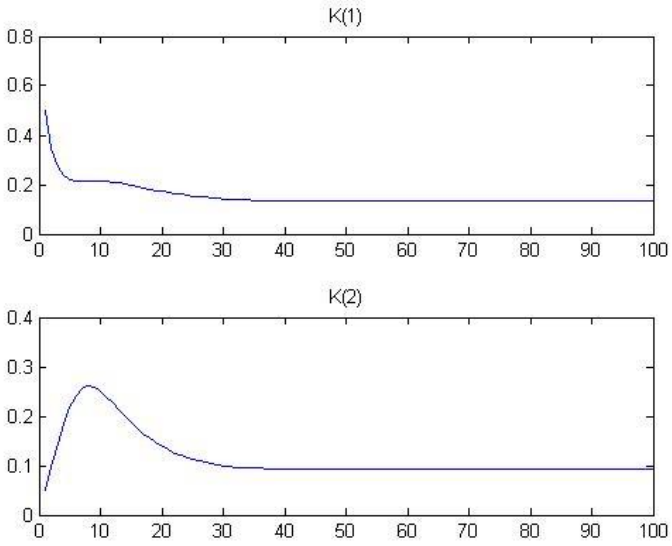


Figure 3 Large Q, the Kalman gain

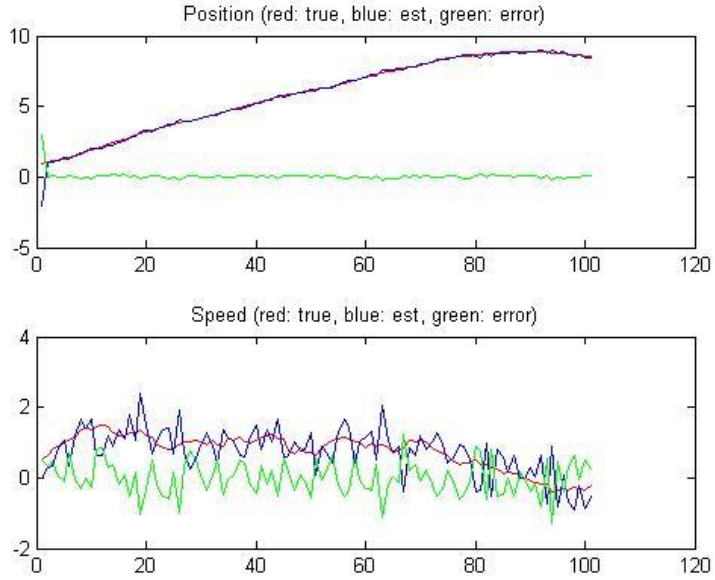


Figure 4 Large R, the estimate against the true value

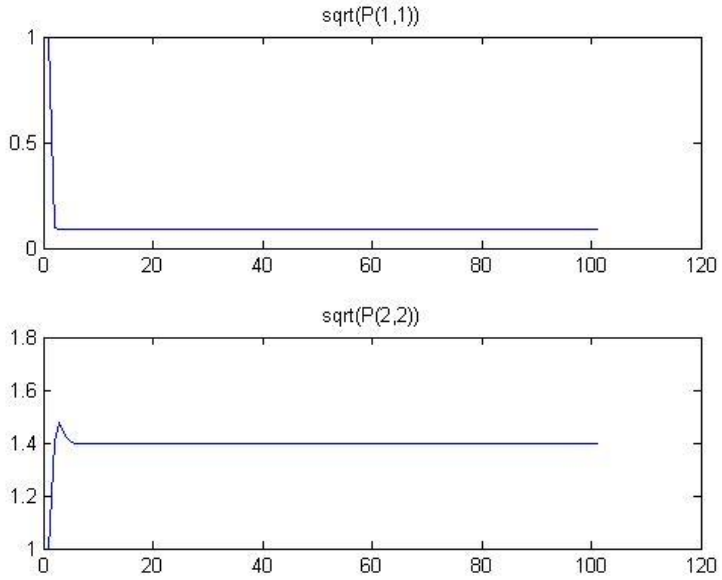


Figure 5 Large R, the standard deviation

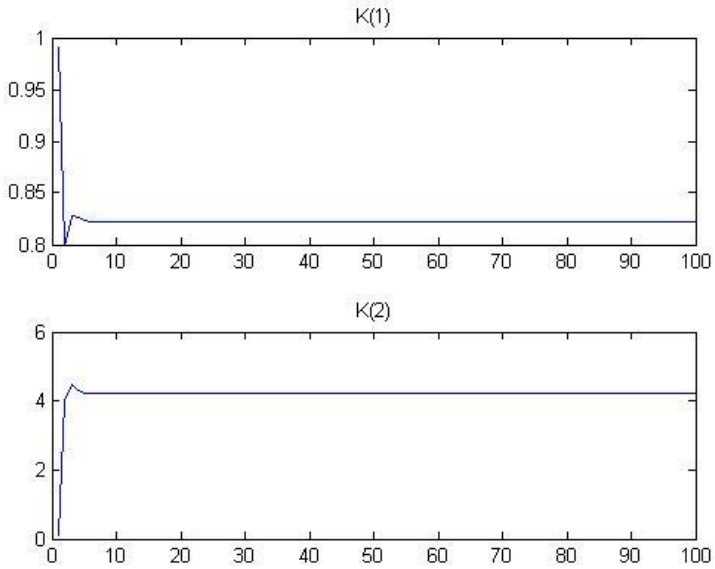


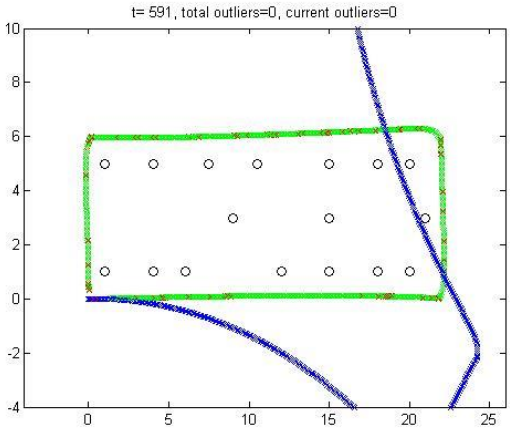
Figure 6 Large R, the Kalman gain

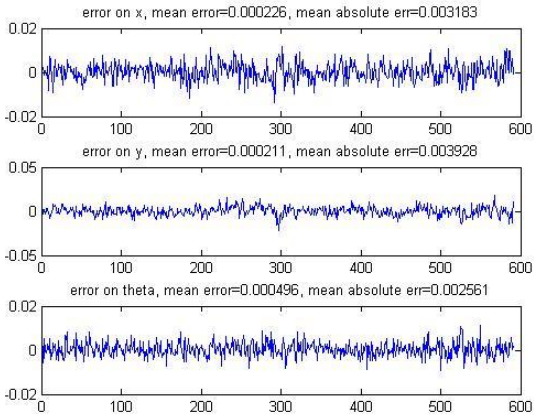
4. To change xhat doesn't change anything noticeable, but to increase P makes it faster and to decrease P makes it slower (i.e. converge faster/slower). The variation in the error in the estimate isn't noticeable for either increasing or decreasing xhat or P.
5. (2): The factor before the integral is the update and the expression within the integral is the prediction.
   (3): The first row/expression is the update and the second row/expression is the prediction.
6. Yes, the measurements can be considered as independent to each other because they only depends on the corresponding current state.
7. $0 \leq \delta_M \leq 1$, $\delta_M = 1$ means that we keep all the measurements. If we use a smaller δ_M we reject more data, e.g. $\delta_M = 0$ means that we reject all data. When we have reliable measurements all arising from features in our map we want all data and we set $\lambda_M = infinity$. With unreliable measurement we want a small λ_M.
8. It updates every loop, which means that the noise also does. It will get more uncertain when getting an outlier before the data association. The uncertainty is getting bigger after each update and this will lead to a bad estimate after all the updating is done.
9. Everything inside the for loop (running for j) that only depends on j can be calculated outside the i-loop.
10. The dimensions are $\dim(\bar{v}) = 2$, $\dim(\overline{H}) = 2$ and in the sequential case $\dim(\bar{v}) = 2$, $\dim(\overline{H}) = 3$.
    H is dependent on number of landmarks in the sequential case and H depends on number of

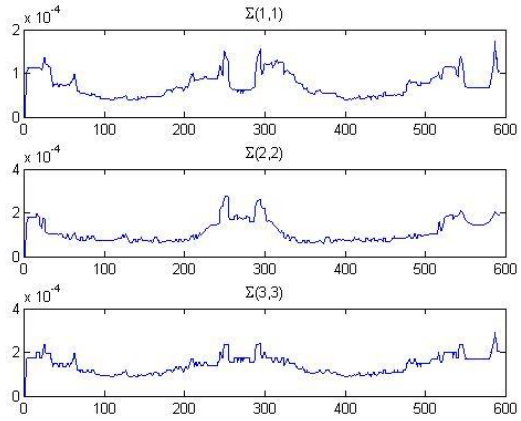
observations in the case with algorithm 4. The sequential case takes one landmark at a time and then update and then it will affect the rest (problem with outliers). Algorithm 4 checks all landmarks and then update all at the same time.

MATLAB

Case 1

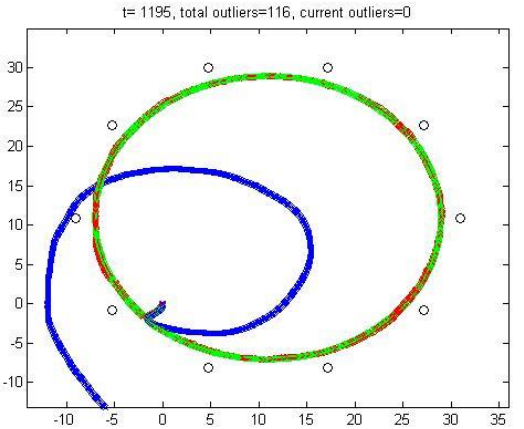


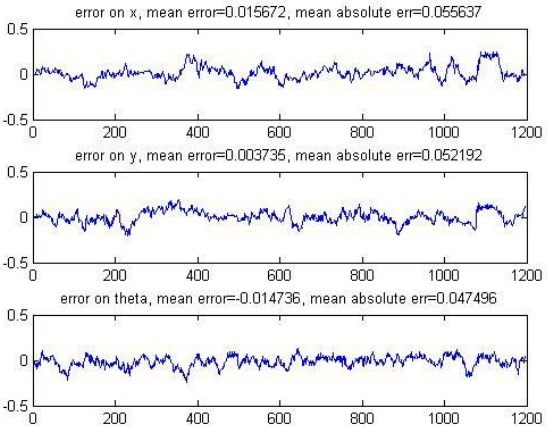


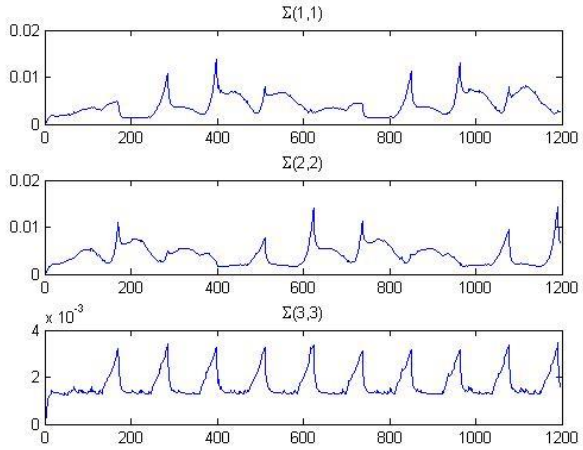



```
mean error(x, y, theta)=(0.000226, 0.000211, 0.000496)
mean absolute error=(0.003183, 0.003928, 0.002561)
total_time =25.648962
```

Case 2



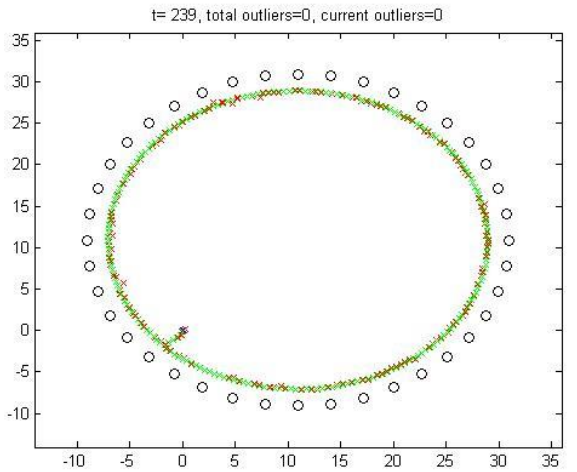


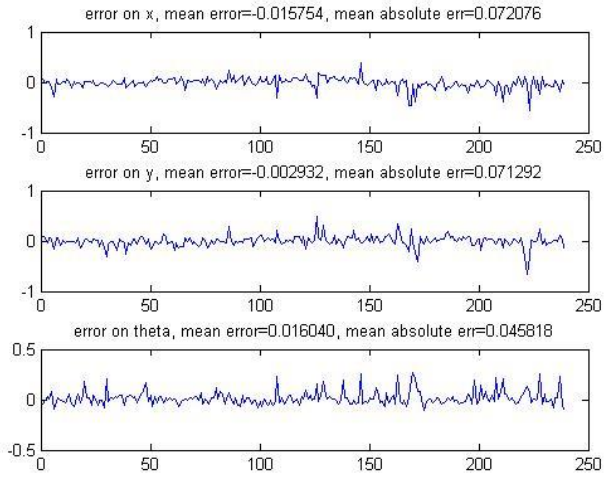


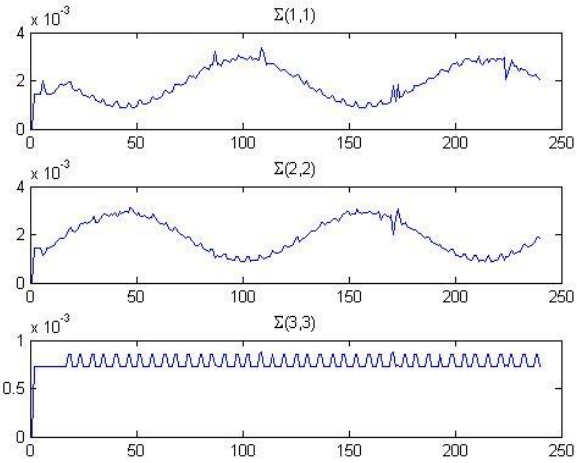

```
mean error(x, y, theta)=(0.015672, 0.003735, -0.014736)
mean absolute error=(0.055637, 0.052192, 0.047496)
total_time =56.560438
```

Case 3

Batch run

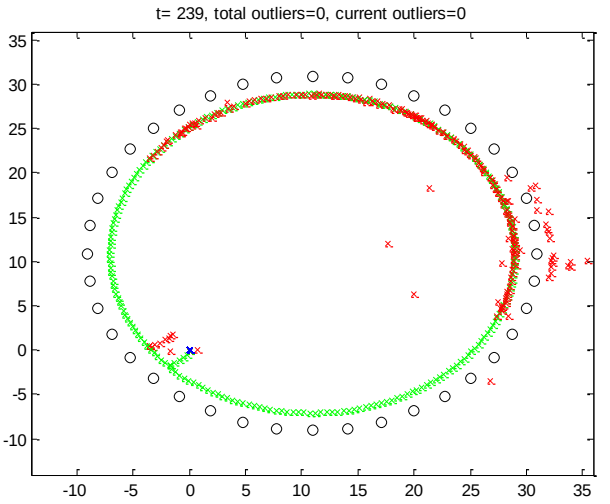


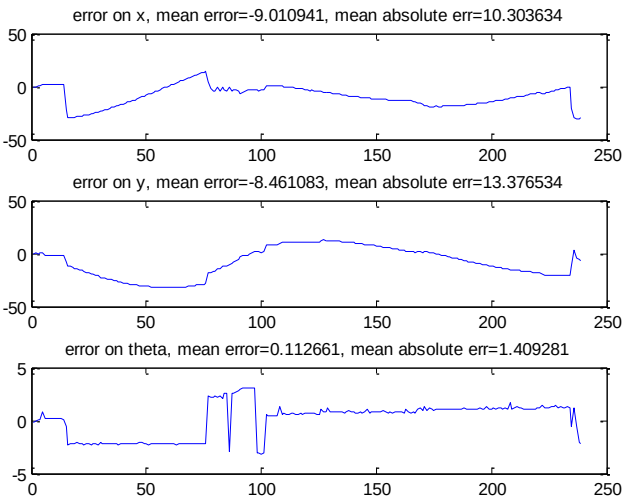


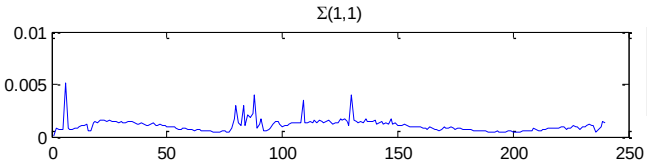



```
mean error(x, y, theta)=(-0.015754, -0.002932, 0.016040)
mean absolute error=(0.072076, 0.071292, 0.045818)
total_time =6.769645
```

Sequence run








```
mean error(x, y, theta)=(-9.010941, -8.461083, 0.112661)
mean absolute error=(10.303634, 13.376534, 1.409281)
total_time =20.844930
```