Applied Estimation - Laboration 1

Ilian Corenliussen

November 2019

Preparation Task

1: Algorithm Kalman filter $(\mu_{t-1}, \Sigma_{t-1}, u_t, z_t)$
2: $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
3: $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
4: $K_t = \bar{\Sigma}_t C_t^T \left(C_t \bar{\Sigma}_t C_t^T + Q_t \right)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t \left(z_t - C_t \bar{\mu}_t \right)$
6: $\Sigma_t = \left(I - K_t C_t \right) \bar{\Sigma}_t$
7: return μ_t, Σ_t

Linear Kalman Filter

1.

The difference between the $\mathbf{z}_t, \mathbf{u}_t$ and $\mathbf{x}_t$ can be described as the following,

- $\mathbf{u}_t$ can be described as the input signal or the control signal, which can be interpreted as the change from the current state to the next. For example $\mathbf{u}_t$ can be joint angles for a robotic arm.
- $\mathbf{z}_t$ could be the odometry for a moving robot, which is the measured data from motion sensors.
- $\mathbf{x}_t$ could be the predicted state for a moving robot, i.e. where the robot thinks its located and current velocities.

2.

The update step decreases the uncertainty of the robots belief, by taking the measured data into account and corrects some of the error from the prediction step. As can be seen in line 6, if C_t is semidefinite the error can only decrease during the update step.

3.

The weighting between the measurement and the belief is decided by the Kalman gain factor.

4.

The weighting between the measurement and the belief gets smaller for larger covariance Q, which implies that the update step would have a smaller impact on the measurement model.

5.

Smaller covariance Q would increase the effect on the updated state estimation, by the same reason as stated above. For smaller covariance Q the weighting between the measurement and the belief get larger.

6.

During the prediction step the uncertainty increases. Due to A is a positive semidefinite matrix and that R_t is a strict positive matrix, see Kalman Filter line 2 under section Preparation Task.

7.

If we have independent Gaussian noise as error and x is a Gaussian prior distribution, we know that the Kalman filter is optimized and further MLS error estimate. I.e. we know that $\mu = \mu_{better}$.

8.

If we start from a known distribution i.e. Gaussian prior distribution then we know the Kalman filter is a MAP estimator. Otherwise, we know it is an MLE estimator.

Extended Kalman Filter

9.

Extended Kalman filter is used when having non-linear models, by linearize the model and then apply the Kalman filter.

10.

No, it depends on how well the linearized model converges to the real model and is therefore not guaranteed to converge.

11.

This could be caused by a too optimistic filter, which could be tuned by changing the covariance variable. Picking a smaller covariance variable makes the model less optimistic and could fix this issue.

Localization

12.

The initial distribution should then be described as a uniform distribution for all points on a circle with the radius r, around the landmark.

13.

The initial distribution with bearing can be described by the same distribution above except that the orientation would be known due to the bearing.

14.

Because we have a good motion estimate but a large initial uncertainty the posterior would look like an arc or also more intuitively could be described as "banana-shaped" area.

15.

The linearized model cant form the arc shape. Therefore, can the model diverge.

PART II - MATLAB Exercises

Warm-up porblem: Linear Kalman filter

Question 1:

ε_k have dimension 2, due to the dimension of x_{k+1} has dimension 2. δ_k have dimension 1, to match the dimension of z_k which has the dimension 1.

Question 2:

x	State
$xhat$	The estimated state
P	Covariane matrix for the belief
G	Use to make covariance matrix for the model noise.
D	Use to make covariance matrix for the sensor noise.
Q	Noise from the sensors, in the update step.
R	Noise from the model, in the prediction step.
$wStdP$	Noise on simulated position
$wStdV$	Noise on simulated velocity
$vStd$	Simulated measurement noise on position
u	Control signal
PP	It is a matrix collecting all P.

Table 1: Roles/usages of the variables.

Question 3:

When we increase the covariance matrix of the modeled process noise by a factor of a hundred (see Figure 1), one could notice that the estimate fluctuates around the true value rapidly. This could be interpreted as the model is optimistic to the prediction, as could be expected when the covariance is increased the Kalman gain gets larger and the correction from the measurement has a greater effect.

When increasing the covariance matrix of the modeled measurement noise(see Figure 2) the estimates get a lot smother, which could be explained by the Kalman gain decreases when the expected noise from the measurement gets larger, i.e. that the effect of the measurements gets smaller.

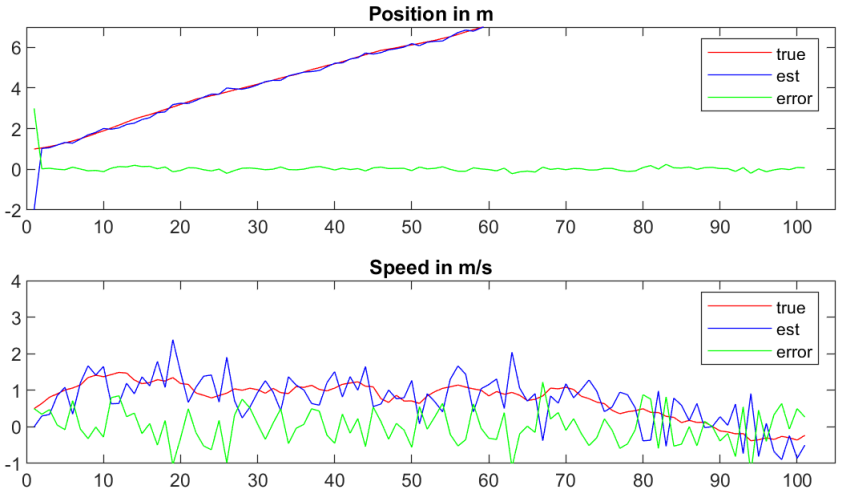

Figure 1: Process noise covariance scaled with 100.

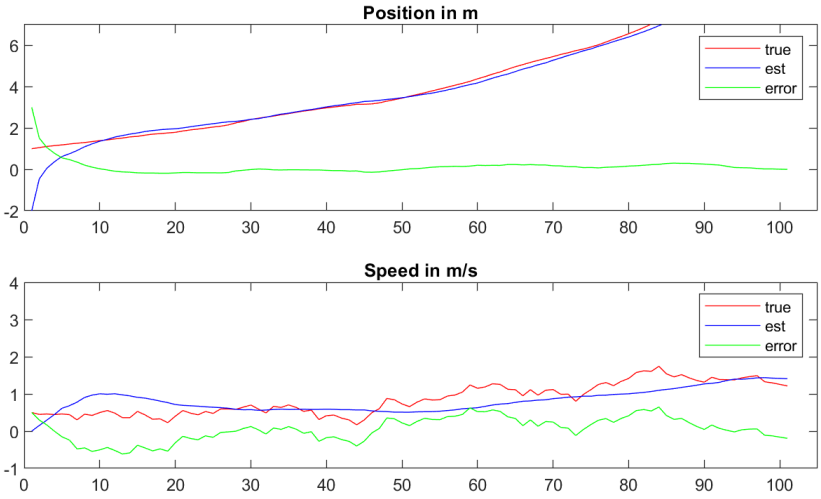

Figure 2: Measurement noise covariance scaled with 100.

Question 4:

For large and small $xhat$ the initial value is far from the true value, but after few iterations, it converges and gets corrected from the measurements update. For large p the system looks normal and for smaller p we get a big overshoot but after around 20 iterations it converges to the real value.

Question 5:

The prediction is the integral of the probability for the earlier state, $t-1$, and the probability for the current, t. Which in (2) and (3) is written as,

$$\text{Prediction} = \overline{\text{bel}}\left(\mathbf{x_t}\right) = \int p\left(\mathbf{x_t}|\mathbf{u_t}, \mathbf{x}_{hbft-1}\right) \text{bel}\left(\mathbf{x_{t-1}}\right) \mathbf{dx}_{t-1}.$$

The update takes the probability for the measurement multiplied with the prediction for the current state at time t and the update step from (2) and (3) is identified as,

$$\text{Update} = p\left(\mathbf{x_t}|\mathbf{u_{1:t}}, \mathbf{z_{1:t}}, \overline{\mathbf{x}}_\mathbf{0}, M\right) = \eta p\left(\mathbf{z_t}|\mathbf{x_t}, M\right) \overline{\text{bel}}\left(\mathbf{x_t}\right).$$

Question 6:

Each measurement is done separately and the noise should be independent between each measurement. Therefore, it should indicate that the measurements are independent and that the assumption is valid.

Question 7:

Since, χ_2^2 is a probability distribution δ_m in $\delta_m = \chi_2^2(\lambda_m)$ can take any value between 0 and 1, e.i. $\delta_m \in [0, 1]$. δ_m will be a threshold for which measurements depending on the probability that will be included. For example, if $\delta_m = 0.1$ we will keep all measurements that have more than 0.1 probability. λ_m for a reliable measurement all arising from features in or map could be selected to $\lambda_m = \infty$ or a very large value. The drawback of choosing $\lambda_m = \infty$ is that the model will be overconfident and none modeled disturbance will make the model to fail. For a scenario with unreliable measurements with many arising from so called clutter or spurious measurements, I would suggest to choose a smaller λ_m.

Question 8:

For Algorithm (3) with a sequential update approach, the error will propagate throughout the estimation. So if the first initial measurement is noisy the following updates and error will be large and not be corrected, due to drawbacks using sequential updates.

Question 9:

One could reduce the re-computation in Algorithm (4) by assembling the $\bar{\nu}_t$ and $\bar{H}_t$ directly in the for loop above instead of doing it after.

Question 10:

$\bar{\nu}_t$ and $\bar{H}_t$ has the dimension $dim(\bar{\nu}_t) = (2n, 1)$ and $dim(\bar{H}_t) = (2n, 3)$ respectively for Algorithm (4). For the sequential update we have the dimensions $dim(\bar{\nu}_t) = (2, 1)$ and $dim(\bar{H}_t) = (2, 3)$. This tells me that the Algorithm (4) is for a batch of states (n).

Dataset 1:

For dataset 1 (see Figure 3) the absolute mean error was below 0.01(m,rad) with the inital given uncertainties, i.e. with motion uncertainties $(x, y) = 0.01m$, $\theta = 1^o$ and sensor uncertainties $r = 0.01m$, $\theta = 1^o$.

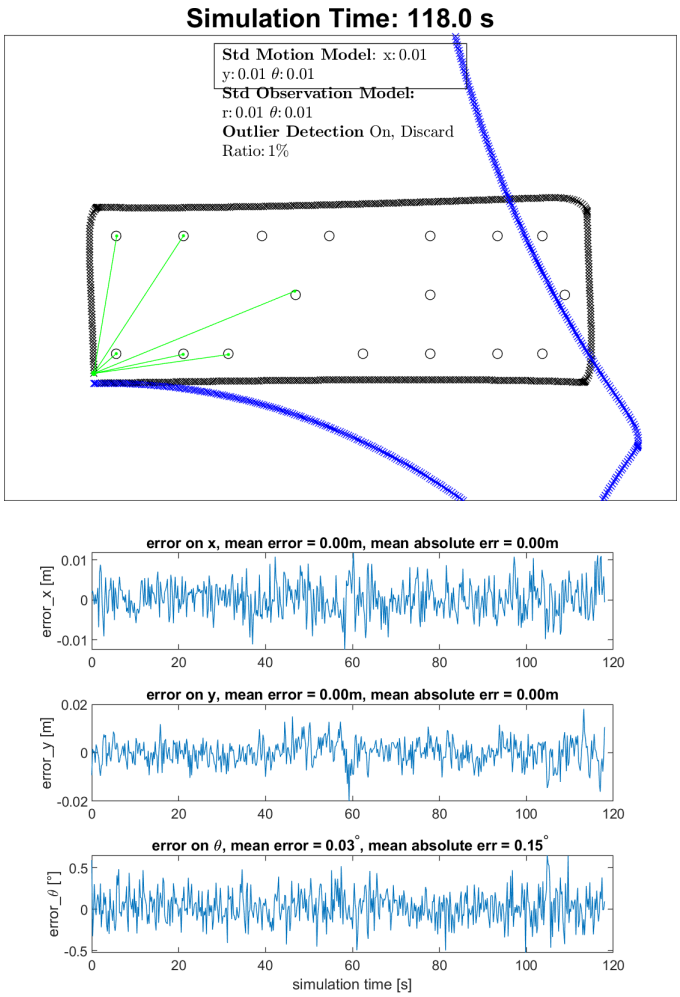

Figure 3: Dataset 1.

Dataset 2:

Due to uncertainties in the measurement, the values in the covariance matrix had to be altered to make the outlier rejection to work properly, i.e. to reject invalid points that shouldn't be used to estimate the position. A mean error below 0.06(m, rad) could be achieved by increasing sensor uncertainties r with 0.04m and θ with 0.03^o, see Figure 4.

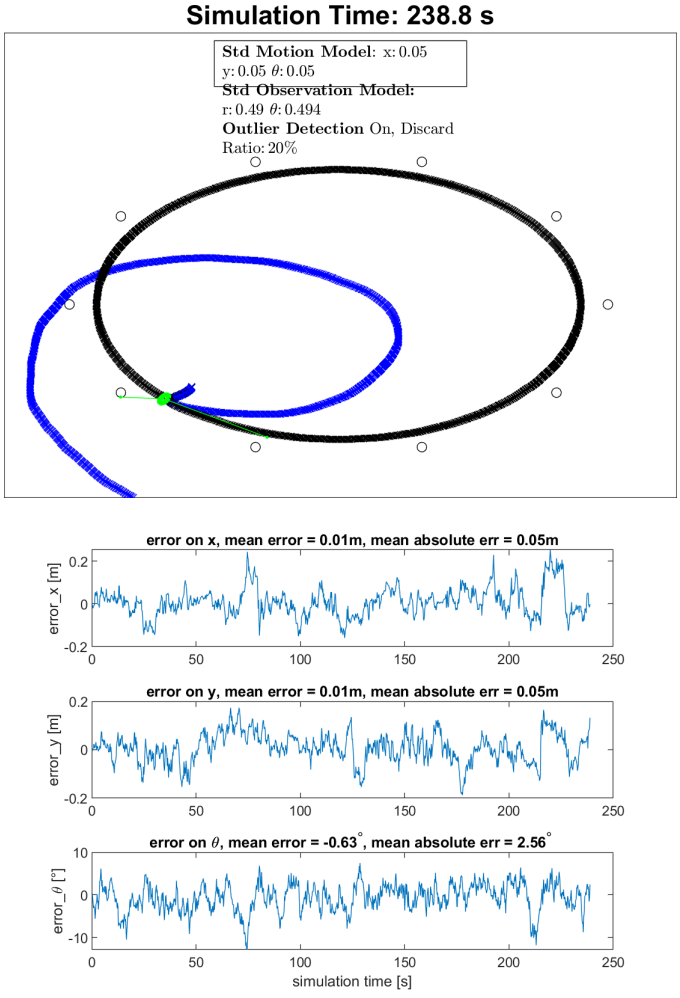

Figure 4: Dataset 2.

Dataset 3:

Between batch update algorithm (see Figure 5) and sequential update algorithm (see Figure 6) one could observe a clear decrease of performance for the sequential update algorithm compared to the batch update algorithm. This is due to the error propagates as the sequential update algorithm runs.

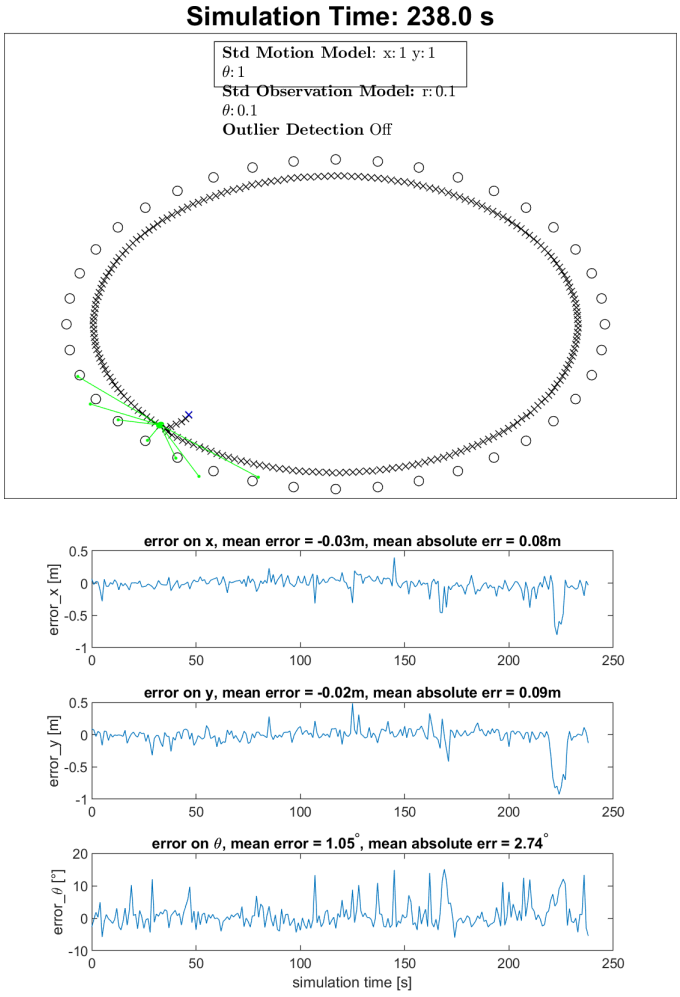

Figure 5: Dataset 3 with batch update algorithm.

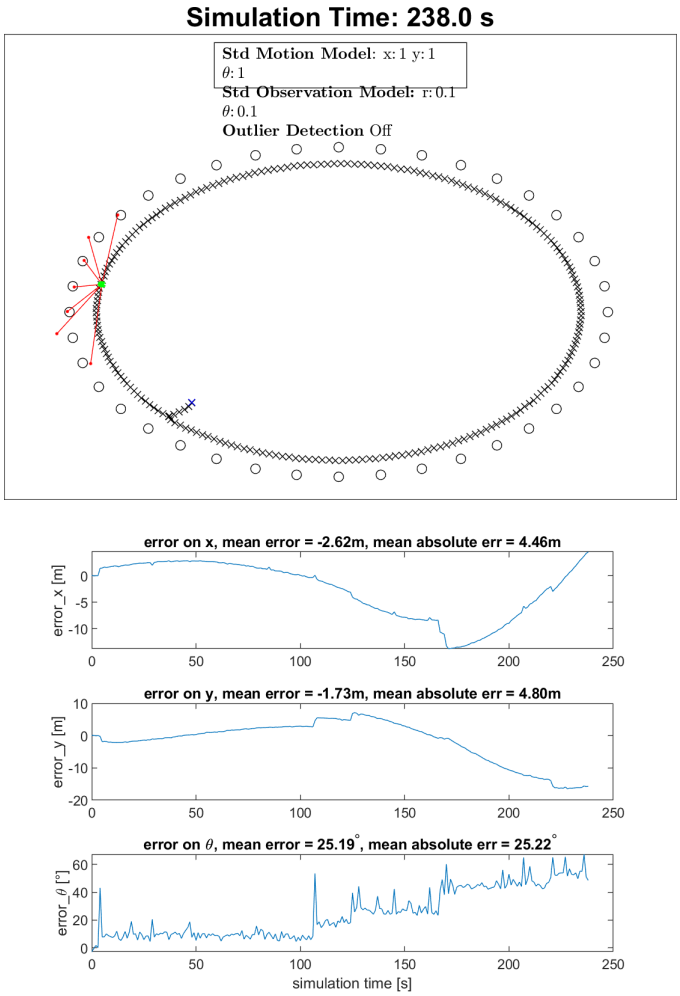

Figure 6: Dataset 3 with sequential update algorithm.