Control Theory

Mecotron Assignment: Swivel Assignment 3

Team 6:
Mathias Schietecat
Karel Smets

December 2021

Departement Werktuigkunde
Professor J. Swevers
Professor G. Pipeleers

Contents

1 State feedback controller design

1.a state equation

The general continuous-time state equation is shown in Equation 1, with x the states of the system, and u the inputs.

$$\dot{x} = Ax + Bu \tag{1}$$

In our case of the cart, this converts itself into Equation 2, where u is now substituted by velocity v of the cart.

$$\dot{x} = v \tag{2}$$

So $A = 0$ and $B = 1$. However, to make the system implementable, we convert the velocity v into the product of $R\omega$, with R the radius of the wheels and ω the rotational speed of the wheels. The result is then Equation 3.

$$\dot{x} = R\omega \tag{3}$$

So $A = 0$ and $B = R\omega$

This state equation can then be discretized using a forward Euler scheme shown in Equation 4

$$\dot{x} = \frac{x[k+1] - x[k]}{T_s} \tag{4}$$

Filling in the forward Euler scheme results in Equation 5, our final discrete state equation.

$$x[k+1] = x[k] + RT_s\omega[k] \tag{5}$$

So $A = 1$ and $B = RT_s$.

1.b State feedback gain K

The closed loop system is schematically represented in Figure 1. The corresponding discrete pole is given by Equation 6. By transforming to the continuous domain, the pole is represented as in Equation 7.

$$p_d = 1 - RT_sK \tag{6}$$

$$p_c = \frac{ln(1 - RT_sK)}{T_s} \tag{7}$$

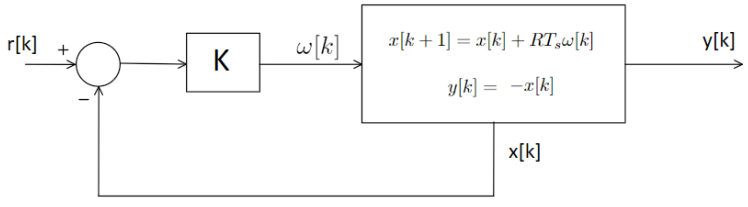


Figure 1: Closed-loop system of the state feedback

Applying the stability criterion to the located discrete pole, $|p_d| < 1$, and to the continuous pole, $Re(p_c) < 0$, gives us the same boundaries for the stability region of the system. K should satisfy $0 < K < \frac{1}{RT_s}$ to obtain a stable system. How the pole moves on a pole-zero map in function of K is illustrated in Figure 2. When increasing K between its allowed boundaries, it moves from the

unit circle to the inside on the real axis. In the continuous Laplace-plane, this corresponds to the pole moving from the origin to the left on the negative Real axis. This will make the response of the system faster, but also more nervous. Choosing $K > \frac{1}{RT_s}$ will result in an unstable system. However, there is also a practical limit to K. Increasing K too much will result in large control actions, eventually saturating the DC-motors even before K hits the stability limit.

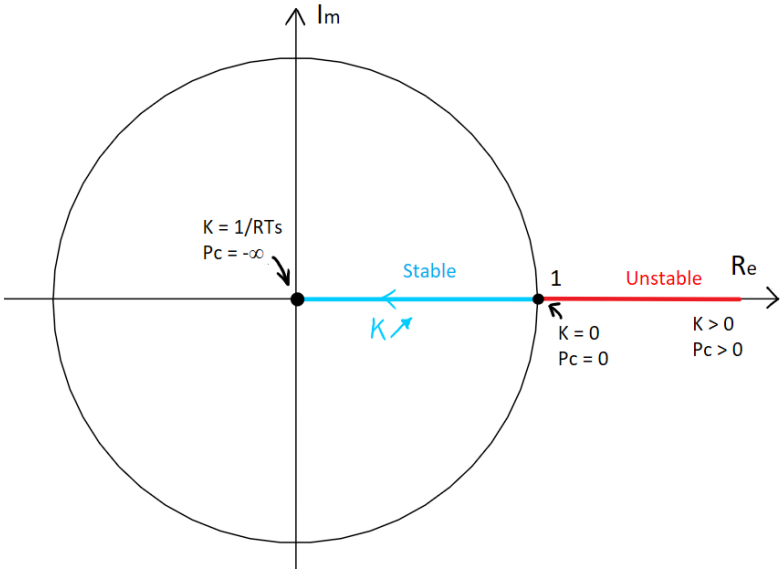


Figure 2: PZ-map of the closed-loop system of the state feedback

2 Validation of Kalman filter principles

2.a Measurement equation

The discrete measurement equation is shown in Equation 8:

$$y[k] = Cx[k] + D = -x[k] \tag{8}$$

So C = -1 and D = 0.

2.b Expression of L_{k+1} as function of $\hat{P}_{k|k}$, Q and R

Since A, B, C and D are matrices of dimension 1x1, they are considered scalars. Using this and the general expression for L_{k+1} results in:

$$L_{k+1} = \frac{C\hat{P}_{k+1|k}}{S_{k+1}} \tag{9}$$

Filling in expressions for $\hat{P}_{k+1|k}$ and S_{k+1} and using the numerical values of A and C results in:

$$L_{k+1} = -\frac{\hat{P}_{k|k} + Q_k}{\hat{P}_{k|k} + Q_k + R_{k+1}} \tag{10}$$

Letting Q and R consecutively go to ∞ results in the following:

$$\lim_{Q\to\infty} L_{k+1} = -1 \tag{11}$$

$$\lim_{R\to\infty} L_{k+1} = 0 \tag{12}$$

These results are as expected. If Q becomes large, the trust in the model decreases relative to the trust in the measurement. This leads the estimator to only use measurement data to estimate the position (= state). In other words, the estimation becomes $x_{k+1|k+1} = Ly_{k+1}$.

If R becomes large, the trust in the measurement decreases relative to the trust in the model. This leads the estimator to only use the model to estimate the position. In other words, the estimation becomes $x_{k+1|k+1} = x_{k|k} + RT_s u_k$.

2.c Expression of $\hat{P}_{k+1|k+1}$ as function of $\hat{P}_{k|k}$, Q and R

Since A, B, C and D are matrices of dimension 1x1, they are considered scalars. Using this and the general expression for $\hat{P}_{k+1|k+1}$ results in:

$$\hat{P}_{k+1|k+1} = \hat{P}_{k+1|k} + \frac{C^2 \hat{P}_{k+1|k}}{S_{k+1}} \tag{13}$$

Filling in expressions for $\hat{P}_{k+1|k}$ and S_{k+1} and using the numerical values of A and C results in:

$$\hat{P}_{k+1|k+1} = \frac{R_{k+1}\hat{P}_{k|k} + R_{k+1}Q_k}{\hat{P}_{k|k} + Q_k + R_{k+1}} \tag{14}$$

Letting Q and R consecutively go to ∞ results in the following:

$$\lim_{Q\to\infty} \hat{P}_{k+1|k+1} = R_{k+1} \tag{15}$$

$$\lim_{R\to\infty} \hat{P}_{k+1|k+1} = Q_k + \hat{P}_{k|k} \tag{16}$$

These results are as expected. As said before, when Q becomes large, the estimator only uses measurement data to estimate the position. Therefore the uncertainty of the estimate will equal that of the measurement.

Similarly, when R becomes large, the estimator only uses the model to estimate the position. Therefore the uncertainty will be determined by the model uncertainty.

2.d Expressions for the steady-state covariance $\hat{P}_\infty$ and related Kalman gain L_∞ as function of Q and R

If $k \to \infty$ then $\hat{P}_{k+1|k+1} = \hat{P}_{k|k} = \hat{P}_\infty$. This expression only makes sense if also $Q_k = Q_{k+1} = Q$ and $R_{k+1} = R_{k+2} = R$ when $k \to \infty$ or if Q and R are time independent. Substituting $\hat{P}_{k+1|k+1}$ and $\hat{P}_{k|k}$ by $\hat{P}_\infty$ in Equation 14 and rearranging terms results in a second order polynomial:

$$\hat{P}_\infty^2 + Q\hat{P}_\infty - QR = 0 \tag{17}$$

This leads to:

$$\hat{P}_\infty = \frac{1}{2}Q\left(\sqrt{1 + \frac{4R}{Q}} - 1\right) \tag{18}$$

where the negative root was discarded due to the fact that a covariance is always positive.

The Kalman gain is found by substituting $\hat{P}_{k|k} = \hat{P}_\infty$ into Equation 10. This results in:

$$L_\infty = -\frac{1 + \sqrt{1 + \frac{4R}{Q}}}{1 + \sqrt{1 + \frac{4R}{Q}} + \frac{2R}{Q}} \tag{19}$$

This steady state Kalman gain should be the same as the fixed LQE gain. The comparison is made in Table 1.

Q[m^2]	R[m^2]	L_∞	L_{LQE}
1e-1	1e-6	-1.0000	-1.0000
1e-6	1e-6	-0.6180	-0.6180
1e-11	1e-6	-0.0032	-0.0032

Table 1: Comparison of the steady-state Kalman gain L_∞ and LQE gain L_{LQE}.

For each proposed value of Q and R, the estimator gains indeed turn out the same. The behaviour predicted in Section 2.b is visible in the table. For relatively larger Q, L becomes -1 and for relative larger R, L becomes small and eventually goes to zero.

2.e LQE closed loop pole and its behaviour

The characteristic equation of the LQE (in continuous time) is given by:

$$det\left(s\mathbf{I} - \mathbf{A} + \mathbf{LC}\right) = 0 \tag{20}$$

Because A, L and C are scalar, the only eigenvalue is $s = A - LC$. This is the closed loop pole of the LQE. The estimator is stable if $A - LC < 0$. Filling in the numerical values for A and C results in:

$$L < 0 \tag{21}$$

Filling in Equation 19, which was proven to be an exact representation of the LQE gain, results in:

$$-\frac{1+\sqrt{1+\frac{4R}{Q}}}{1+\sqrt{1+\frac{4R}{Q}}+\frac{2R}{Q}} < 0 \tag{22}$$

Because Q and R are variances, and thus always positive, the condition of Equation 23 will always be true. It is thus impossible to make the estimator unstable.

The location of the pole in terms of Q and R is:

$$p_e = -\frac{1+\sqrt{1+\frac{4R}{Q}}}{1+\sqrt{1+\frac{4R}{Q}}+\frac{2R}{Q}} = L_{LQE} \tag{23}$$

$$\lim_{\frac{Q}{R}\to\infty} p_e = -1 \tag{24}$$

$$\lim_{\frac{Q}{R}\to 0} p_e = 0 \tag{25}$$

For large values of Q/R the pole moves to $s = -1$. For small values of Q/R the pole moves to $s = 0$. The estimator thus becomes faster for larger Q/R ratios. This is to be expected since a large Q/R means that the measurement is being trusted more than the model. Therefore the estimated state will react faster on a changing physical state. This effect can clearly be seen in Figure 10 in Section 3.e

3 Implementation of state estimator and state feedback controller

3.a Choosing values for K and Q

There are different sources of noise in the system which should be taken into account. Q gives an indication of the process noise, which originates from our identified model of the cart. This model is accurate (enough), but not perfect. Varying resistances or inductances are for example not taken into account in the model. Another source of process noise is the assumption that a perfect velocity controller is used. This is in fact not the case. Also the sensor to measure the front distance is not perfect. There will be an error in the measurement with respect to the real distance. The sensor is imperfect, it has a certain resolution and high frequency noise will always be present. This is taken into account by the value R.

Choosing an appropriate value of R is relativity straight forward. The sensor is placed at a certain distance from a wall, and the sensor output is recorded. This data is then used to calculate the measurement covariance R.

Tuning Q is done by looking at the behaviour of the estimated state, using the previously calculated value of R. The estimated state is plotted against the measurement like in Figure 9. Q should then be chosen such that there is no large lag between the estimate and measurement, and such that the measurement noise is rejected as much as possible out of the state. Making Q larger reduces the lag between the measurement and the state. To reject the measurement noise out of the state estimate Q should be made smaller.

Choosing a value for K again comes down to experimental tuning. When K is too high, overshoot will occur. the cart will pass the reference aimed for, and will have to reverse. Choosing K too low will result in a very slow controller, making the cart approach the reference needlessly slow. It comes down to the designer and the criteria that are subjected on the controller. How much overshoot is allowed? How long do we want the rise time or settling time to be? Again, a balance should be found. More on this in question 3.b.

$P_{0|0}$ is a measure for the uncertainty in the initial placement of the cart. E.g. a covariance of $0.01m^2$ comined with an initial estimate of 15cm means that the car will most probably be placed initially anywhere between 5cm and 25cm from the wall.

The values we chose for the discussed parameters are shown in subsection 3.a

K	55 s^{-1}
Q	1e-8 m^2
R	9.5199e-7 m^2
$P_{0\|0}$	0.01 m^2

Table 2: Numeric values for design parameters of the Estimator and feedback controller

3.b Influence of feedback constant K on the step response

In Figure 3 the step response of the system is plotted for different values of K. The values of R, Q and $P_{k|k}$ are chosen as listed in subsection 3.a. The step input moves the cart from a distance of 15cm to 20cm. As already mentioned in the previous question, the pole in the feedback system becomes faster when increasing K. This can be recognized in the rise time becoming shorter. The cost of this decreased risetime is an increased overshoot, resulting in longer settling time. A value of K=55 is chosen to optimize the controller. For the system at hand, it provides the smoothest response, minimizing settling time and limiting overshoot.

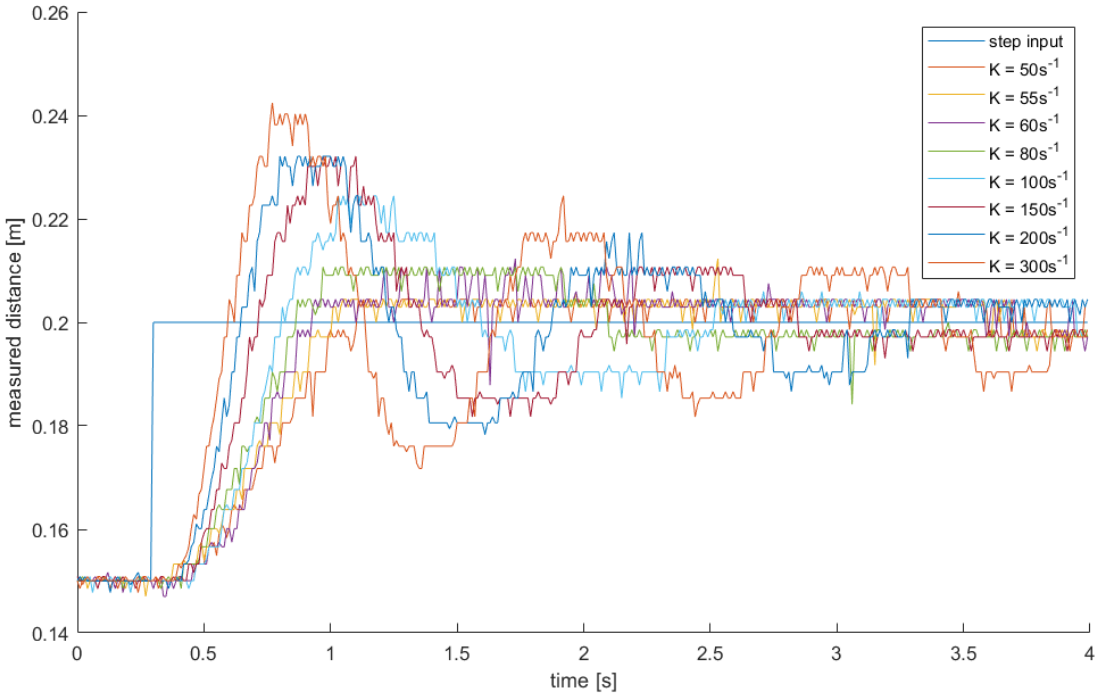


Figure 3: Step response of the system with varying feedback constant K

The control signal of motor A during these experiments are shown in Figure 4. The same remarks made above can be repeated here. Rise time becomes smaller, because the control signal is higher, thus allowing for a faster acceleration of the cart. But again, this comes at the price of increased overshoot and settling time. It is also important to notice that if K is chosen too high, the motor voltage will saturate. Another limit for K is given by the stability criterion as mentioned in Section 1.b

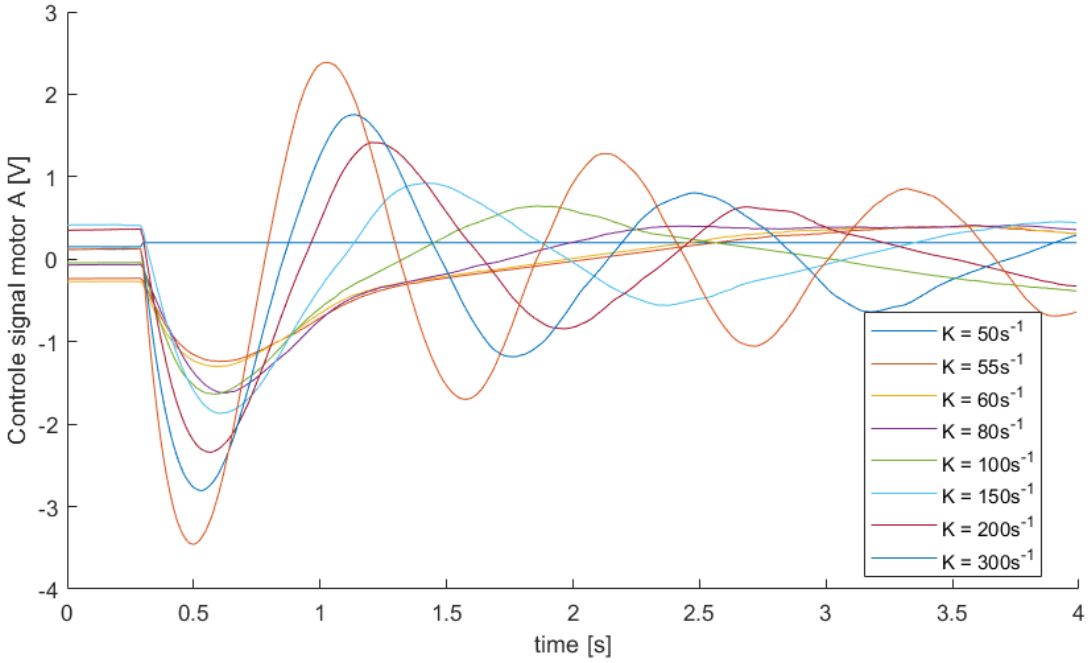


Figure 4: Control signal (motor A) of the system during step response with varying feedback constant K

3.c Influence of Q and R on the evolution of $\hat{P}_{k|k}$ and L_k

To analyse the influence of Q and R on $\hat{P}_{k|k}$ and L_k, the cart is positioned at a distance of 30cm. When the controller and estimator are enabled, the reference distance is set to 15 cm. Two sets of measurements were made. The first one keeps R at a constant value of $R = 9.5199e-7m^2$, varying Q between 1e-1m^2 and 1e-11m^2. The results are visualised in the upper half of Figure 5. The second set shows data for Q constant and equal to 1e-8m^2, with R varying from 1 to 1e-6m^2. Results are plotted in the lower half of Figure 5.

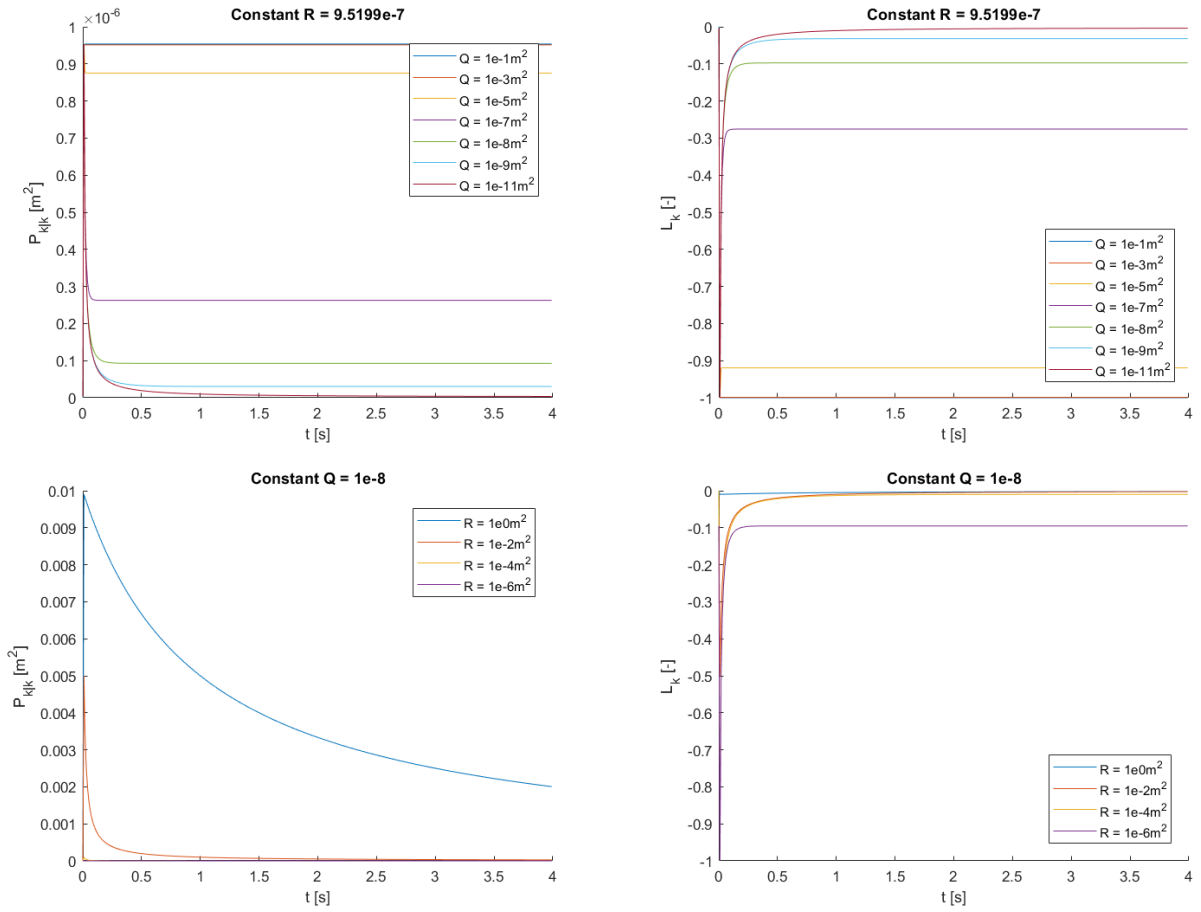


Figure 5: The evolution of $\hat{P}_{k|k}$ and L_k for different values of Q and R

These results are in line with the answers given in question 2(e) and 2(d). In 2(e) the remark was made that an estimator with a large Q/R ratio is faster than one with a low Q/R ratio. The same can be noticed in Figure 5. Both $\hat{P}_{k|k}$ and L_k reach their constant value sooner when Q/R is large.

The results are also consistent with the answers given in 2(d). In Table 3 a few values of L_∞ are listed for some of the Q and R values used in the experiments. The third column shows the value when calculated with the expression obtained in 2(d). The fourth column lists the values of L after it has stabilised in the experiments. The same pattern is followed in column 5 and 6, but with values for P. It is clear these correspond to each other as they are almost always exactly the same.

R $[m^2]$	Q $[m^2]$	L_∞ calculated [-]	L_∞ measured [-]	P_∞ calculated $[m^2]$	P_∞ measured $[m^2]$
9.5199e-7	1e-1	-0.99999	-0.99999	0.95367	0.95198
9.5199e-7	1e-3	-0.99904	-0.99905	0.95111	0.95108
9.5199e-7	1e-5	-0.91950	-0.91950	0.87536	0.87536
9.5199e-7	1e-7	-0.27581	-0.27581	0.26257	0.26256
9.5199e-7	1e-8	-0.09737	-0.09737	0.092698	0.092698
9.5199e-7	1e-9	-0.03188	-0.03188	0.03036	0.03035
9.5199e-7	1e-11	-0.00323	-0.00376	0.00358644	0.00308

Table 3: Comparing calculated and measured values of L_∞

3.d Kalman filter consistency: NIS and SNIS

The NIS and SNIS are calculated as follows (values of A, B and C filled in):

$$NIS_k = \nu_k S_k^{-1} \nu_k = \frac{\left[y_k + \left(\hat{x}_{k-1|k-1} + RT_s u_{k-1}\right)\right]^2}{\left(\hat{P}_{k-1|k-1} + Q_{k-1}\right) + R_k} \tag{26}$$

with y the measured distance and u the desired velocity. The state estimate covariance $\hat{P}$, state estimate $\hat{x}$ and u are essential to the functioning of the controller and are calculated every time step. They can always be recorded in an experiment. The same is true for the measurement y. The controller also calculates ν and S as intermediate steps for the calculations. They can also be recorded. Both formulations of the NIS can thus be used starting from the recorded experimental data.

$$SNIS_k = \sum_{j=k-M+1}^{k} NIS_j \tag{27}$$

The SNIS is calculated out of the calculated NIS values as illustrated in Equation 27. In this section M is chosen to be 5.

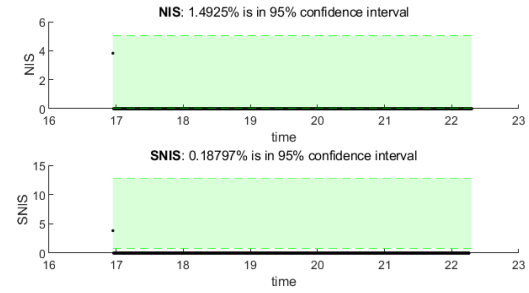
The expected probability density of the NIS is χ^2. In the numerator of equation Equation 26 three gaussian random variables, namely y, $\hat{x}$ and $u = f(\hat{x})$ (due to feedback) are summed together. The sum of these random variables, which is also a gaussian distribution, is then squared, resulting in a χ^2 distribution.

The SNIS is a χ^2 distribution of degree M (degree 5 in this section). This is the result of the summation of M χ^2 distributed NIS values to obtain the SNIS.

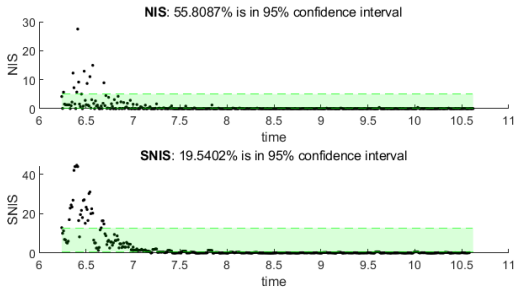
The consistency is checked using confidence intervals. The confidence bounds α_1 and α_2 for e.g. an 95% confidence interval are determined as:

$$0.025 = F_{\chi^2_M}(\alpha_1) \quad \text{and} \quad 0.975 = F_{\chi^2_M}(\alpha_2)$$

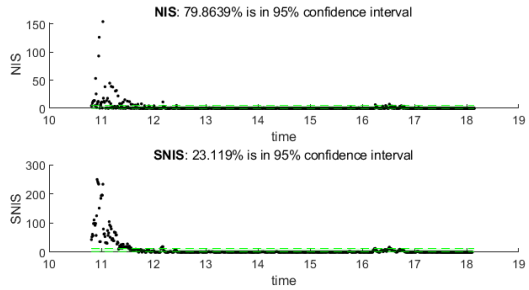
with $F_{\chi^2_M}(\alpha)$ the cumulative distribution corresponding to the χ^2 distribution of degree M. If R is measured and Q chosen correctly and the assumption of gaussian measurement noise and process noise hold, the innovation should be gaussian distributed to. In that case the percentage of NIS or SNIS points between the confidence bounds should exactly match the value for which the bounds are calculated, here 95%. If the statistical model does not match with reality, less (S)NIS points will lie in the interval. A greater discrepancy in the percentage results in lower confidence in the statistical model and thus in a lower consistency between the model and the real world.



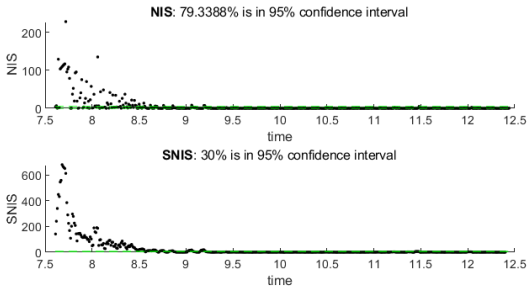
Q = 1e-1m^2, R = 9.52e-7m^2



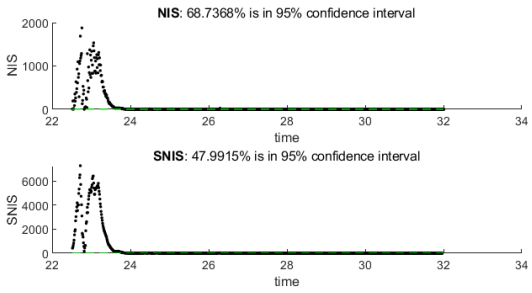
Q = 1e-5m^2, R = 9.52e-7m^2



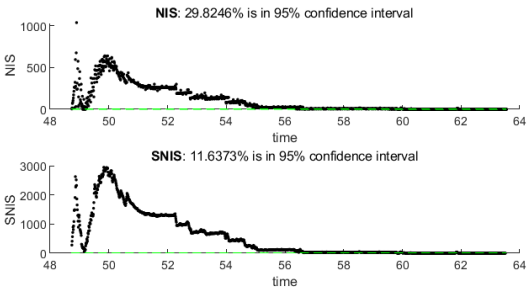
Q = 1e-6m^2, R = 9.52e-7m^2



Q = 1e-7m^2, R = 9.52e-7m^2

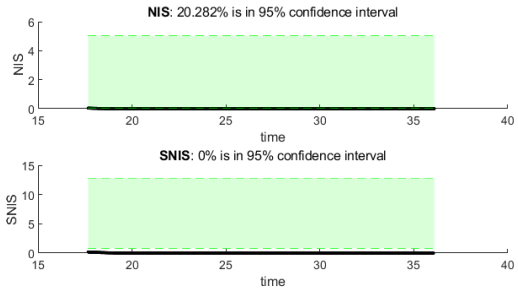


Q = 1e-8m^2, R = 9.52e-7m^2

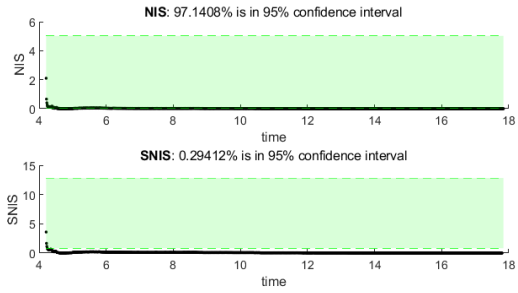


Q = 1e-11m^2, R = 9.52e-7m^2

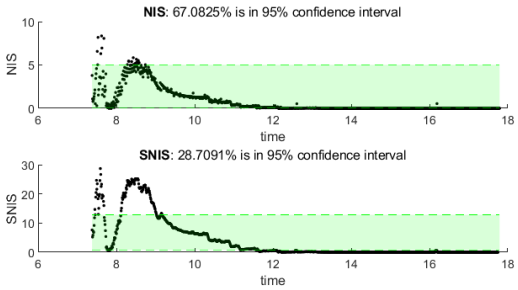
Figure 6: NIS and SNIS for varying values of Q and constant value of R = 9.52e-7m^2.



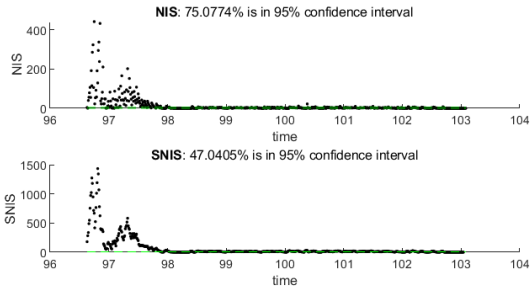
Q = 1e-8m^2, R = 1e-0m^2



Q = 1e-8m^2, R = 1e-2m^2



Q = 1e-8m^2, R = 1e-4m^2



Q = 1e-8m^2, R = 1e-6m^2

Figure 7: NIS and SNIS for varying values of R and constant value of Q = 1e-8m^2.

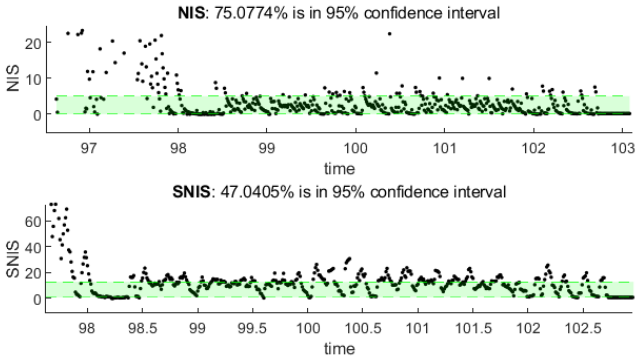


Figure 8: Enlarged view of the Q and R setting which comes closest to consistency: Q=1e-8m^2, R=1e-6m^2

The resulting NIS and SNIS are plotted above in Figure 6 and Figure 7. Figure 6 shows the results for a constant R of $9.52e-7m^2$ and different values of Q, going from a large Q to a to small Q. The overall best result is there found for Q = 1e-8 m^2, but in none of the cases the Kalman filter is consistent. Figure 7 shows the results for a constant Q of 1e-8m^2 and different values of R. Here as well the overall best result is found for $R = 1e-6m^2 \approx 9.52e-7m^2$.

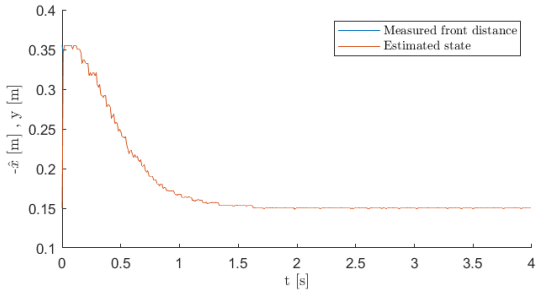
Figure 8 shows an enlarged view of the NIS and SNIS for Q=1e-8m^2 and R=1e-6m^2, cutting away the transient and focusing in the steady state results. Of all the experiments, this one comes closest to consistency. Especially the NIS points are quite good located within the confidence bounds. Their distribution also starts to become less deterministic (more random), as would be expected from a consistent filter. This Q and R tuning is also practically the same as the one given in subsection 3.a.

The inconsistency of the filter is generally due to deviations on the assumed statistical behaviour, like correlated noise sources or deterministic modeling errors. For example non linear system behaviour in the transient will cause a deterministic error between the model and the real world, causing the NIS and SNIS to be highly deterministic and also show a transient. This effect is visible in the plots made above. Related to this, it is virtually impossible to make a good estimate for Q, since it is an artificial quantity to fix random deviations of the system compared to the model. Q can be tuned to let the filter come close to consistency, but it will never be consistent if the modeled statistics do not correspond to the real world.

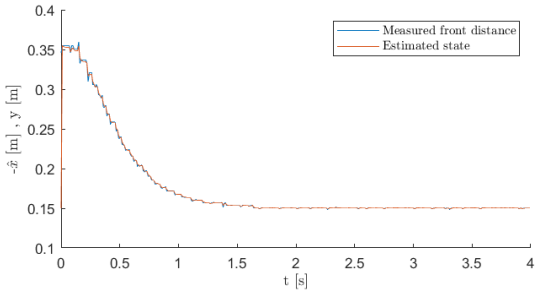
3.e Influence of Q/R ratio on the position estimate

The results of the demanded experiments are visualised in Figure 9 and Figure 10. The plots in the figures are arranged from large to small ρ = Q/R ratio. It is clear that the estimator converges faster to the measurements for larger Q/R ratios. R is a measure for the uncertainty of the measured state. If R is small, the measurements are trusted a lot. Q is a measure for the uncertainty of the calculated state based on the model. When this uncertainty is large, the correctness of the model is doubted.

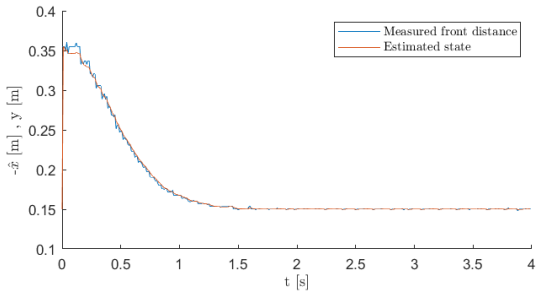
Combining these interpretations explains the results. A large Q/R ratio means little trust is given to the model, and a lot of trust is given to the measurements, thus converging to the measurements faster.



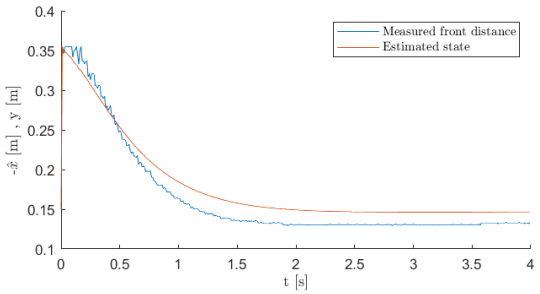
Q = 1e-1m^2, R = 9.52e-7m^2 $\implies$ ρ = 1.1e5



Q = 1e-6m^2, R = 9.52e-7m^2 $\implies$ ρ = 1.1e-0

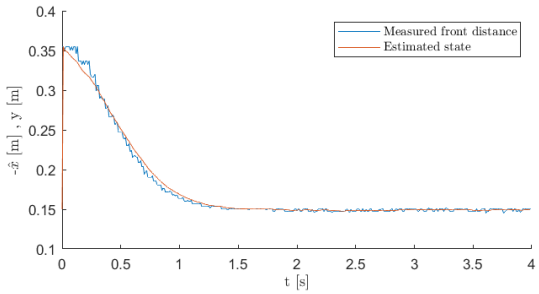


Q = 1e-7m^2, R = 9.52e-7m^2 $\implies$ ρ = 1.1e-1

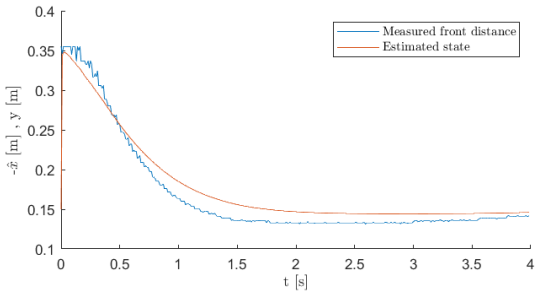


Q = 1e-11m^2, R = 9.52e-7m^2 $\implies$ ρ = 1.1e-5

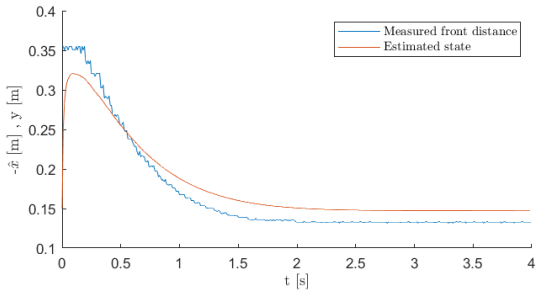
Figure 9: Estimated state evolution and actual measurement for wrong initial position estimate. Effect of varying Q (with constant value of R = 9.52e-7m^2).



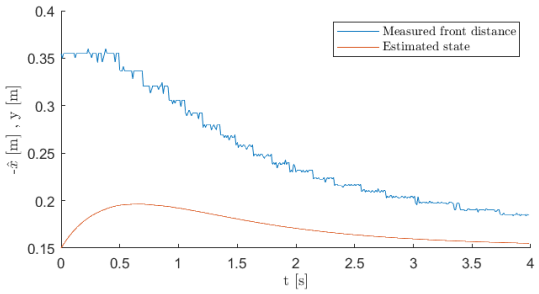
Q = 1e-8m^2, R = 1e-6m^2 $\implies$ ρ = 1e-2



Q = 1e-8m^2, R = 1e-4m^2 $\implies$ ρ = 1e-4



Q = 1e-8m^2, R = 1e-2m^2 $\implies$ ρ = 1e-6



Q = 1e-8m^2, R = 1e-0m^2 $\implies$ ρ = 1e-8

Figure 10: Estimated state evolution and actual measurement for wrong initial position estimate. Effect of varying R (with constant value of Q=1e-8m^2).

3.f System behaviour using slow pole placement estimator

The closed loop continuous time pole of the state feedback controller is $p_{c,c} = -(BK - A)$. The discrete pole of the estimator that is ten times slower then the controller is thus:

$$p_{e,d} = e^{-T_s \frac{(BK-A)}{10}} \tag{28}$$

Using the continuous time A = 0, B = 1 and the above used value for the feedback K = 55, $p_{e,d}$ is calculated. L is calculated using acker in matlab:

$$L = acker\,(A, AC, p_{e,d}) \tag{29}$$

using the discrete time A = 1 and C = -1. The numerical results are:

$p_{e,d}$	0.9982 rad/s
L	-0.0018

Implementing this estimator and giving it a wrong initial estimate results in the following dynamic behaviour:

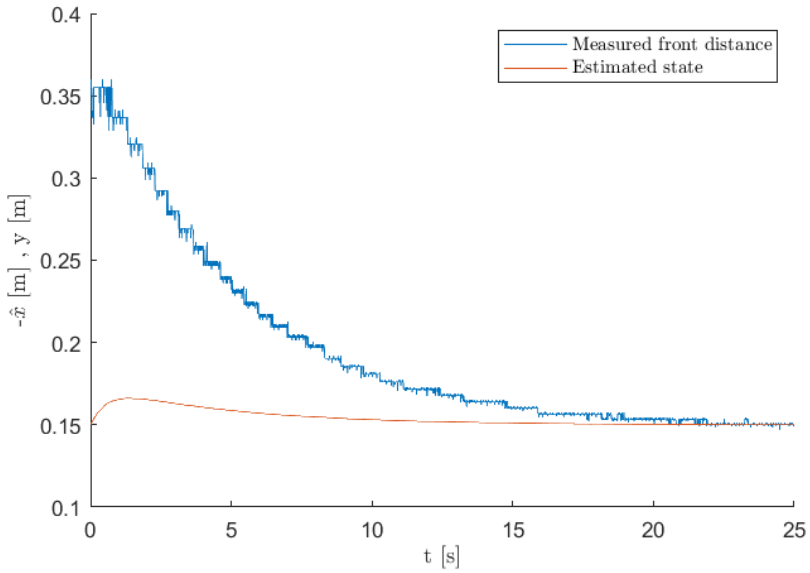


Figure 11: State evolution using pole placement estimator with L=-0.0018. The estimator is ten times slower than the feedback controller.

This control performance is not satisfactory. The estimate updates much to slow. The slow state results in a small error and thus a slow response. Comparing the response in Figure 10 for R = 1e-6m^2 (closest to the actual implemented values in the Kalman filter) and the one in Figure 11, it takes the cart about 18.5 seconds longer to get to the reference position. The system is dominated by the slow pole of the estimator.

To overcome this problem when using pole placement estimators, the estimator pole has to be chosen about ten times faster than that of the feedback controller. This way, the feedback controller gain K is responsible for the response time (the feedback controller pole is the dominant one).