Kalman Filter

Jan Švancar

December 8, 2021

1 Assignment

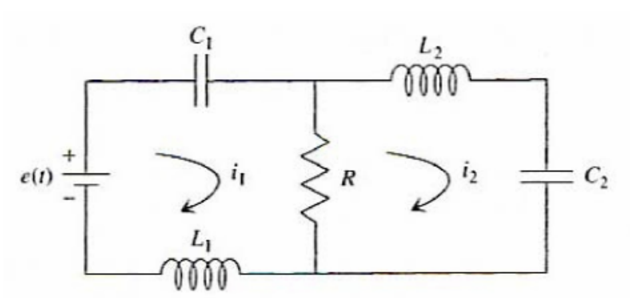

Figure 1: Circuit used for estimation

$$L_1\ddot{q}_1(t) + R(\dot{q}_1(t) - \dot{q}_2(t)) + \frac{1}{C_1}q_1(t) = e(t)$$
$$L_2\ddot{q}_2(t) + R(\dot{q}_2(t) - \dot{q}_1(t)) + \frac{1}{C_2}q_2(t) = 0$$

Figure 2: Equations describing estimated circuit

It was assigned to implement Kalman filter to estimate states of charges and their rates.
Resulting filter was then tested with three innovations tests and RMSE metric.

Model mismatch and noise statistics mismatch were then implemented to ideal Kalman filter to test its behaviour under unideal conditions.

2 Discretization

2.1 Deterministic part

Discretization of deterministic part was done with use of zero-order hold method. System was discretized with sample time $T_c = 0.2\,\mathrm{s}$. Output of the system was measured with sample time $T_s = 1\,\mathrm{s}$.

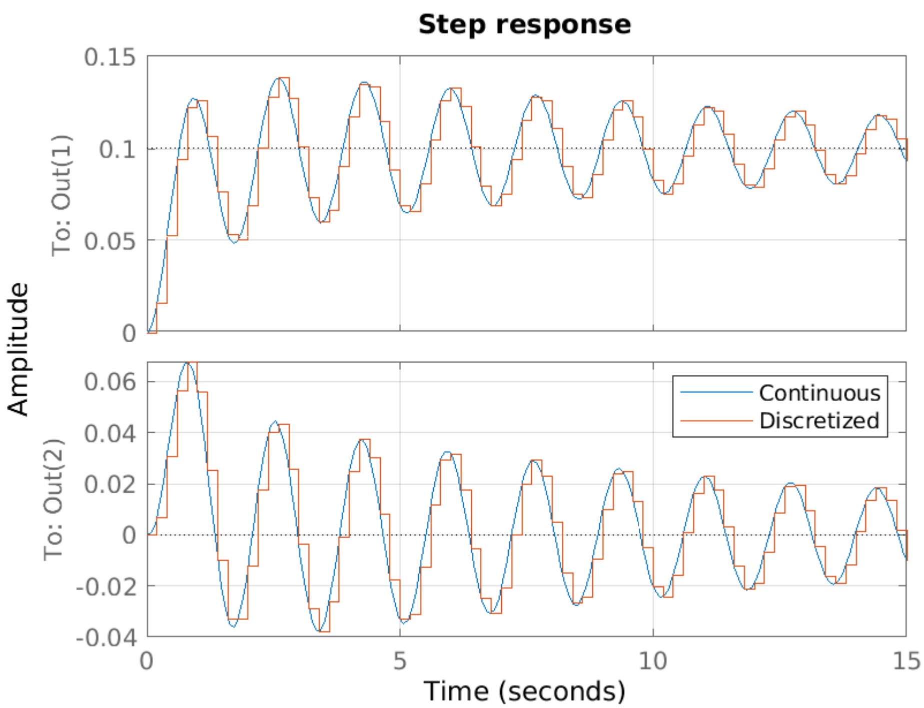

Figure 3: Step response of continuous and discretized system

2.2 Stochastic part

In case of measurement error $n(k)$ its covariance matrix R_d is already in discrete state.

$$R_d = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix} \tag{1}$$

In case of process noise $v(t)$ its covariance matrix has to be discretized Q has to be discretized. Matrix G_c tell us how process nois $v(t)$ enters system so covariance matrix Q can be written as

$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

Discretized covariance matrix Q_d is then obtained with following calculation

$$Q_d = \int_0^{T_c} e^{At} Q e^{A^T t} dt, \tag{3}$$

where T_c is system sample time, Q is contiuous covariance matrix and A is continuous system matrix.
In our case covariance matrix Q_d is

$$Q_d = \begin{pmatrix} 0.01592 & 0.007483 & 0.1036 & 0.07461 \\ 0.007483 & 0.00365 & 0.04296 & 0.03336 \\ 0.1036 & 0.04296 & 1.068 & 0.5561 \\ 0.07461 & 0.03336 & 0.5561 & 0.3714 \end{pmatrix}. \tag{4}$$

3 Data Generation

Experimental data were created by simulating discretized system and sampling it with sample time $T_s = 0.2\,\mathrm{s}$. Restriction that output is measured five times slower than system is discretized was involved only in Kalman algorithm.

4 Kalman Algorithm

Kalman algorithm was implemented so that it performs time-step update with period $T_c = 0.2\,\mathrm{s}$ and data-update step with period $T_s = 1\,\mathrm{s}$. Data-update step has longer period due output measurements that are available only every second.
Kalman algorithm is implemented in MATLAB script.

5 Kalman algorithm with incorrect parameters

5.1 Model mismatch

To simulate model mismatch value of several parameters was changed.

- Resistor Resistance - $R = 15\,\Omega$
- Coil Inductance - $L_1 = 5\,\mathrm{H}$
- Capacitor Capacitance $C_1 = 0.25\,\mathrm{F}$

5.2 Noise statistics mismatch

To simulate noise statistics mismatch noise covariance matrices were slightly changed.

$$Q_d = \begin{pmatrix} 0.01592 & 0.007483 & 0.1036 & 0.07461 \\ 0.007483 & 0.5 & 0.04296 & 0.03336 \\ 0.1036 & 0.04296 & 1.068 & 0.5561 \\ 0.07461 & 0.03336 & 0.5561 & 0.3714 \end{pmatrix} \tag{5}$$

$$R_d = \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.75 \end{pmatrix} \tag{6}$$

6 Performance Tests

To evaluate performance of implemented Kalman filter following tests were used:

- Test 1 - Innovation magnitude bound test

- Test 2 - Normalised innovations squared χ^2 test
- Test 3 - Innovation whiteness (autocorrelation) test
- Predicted value RMSE
- Filtered value RMSE

Innovation tests are described in study material provided with source code.

6.1 Predicted value RMSE

Error for this metric is defined as

$$e(k) = y(k+1) - C \cdot \hat{x}(k+1), \tag{7}$$

where y is output measurement, C is output matrix and $\hat{x}(k+1)$ is state value predicted by Kalman algorithm.
Basically it's difference between measured output and output estimated with Kalman algorithm.
From these errors vector is then calculated RMSE.

6.2 Filtered value RMSE

Error for this metric is defined as

$$e(k) = y(k) - C \cdot \hat{x}(k), \tag{8}$$

where y is output measurement, C is output matrix and $\hat{x}(k)$ is state value after data-update step of Kalman algorithm.
From the error vector defined above RMSE is calculated.

7 Results

	Test 1		Test 2	Test3		Prediction RMSE		Filtered RMSE	
	Valid	Percentage	Valid	Valid	Percentage	q_1	$\dot{q}_1$	q_1	$\dot{q}_1$
Ideal	Yes	96.5%	Yes	Yes	99.75%	0.8023	0.7252	0.3719	0.3844
Model Mismatch	No	82.5%	No	No	61.5%	2.0488	1.1256	0.3556	0.4686
Noise Mismatch	Yes	100%	No	Yes	99.75%	0.8052	0.7426	0.2493	0.2351

7.1 Estimation

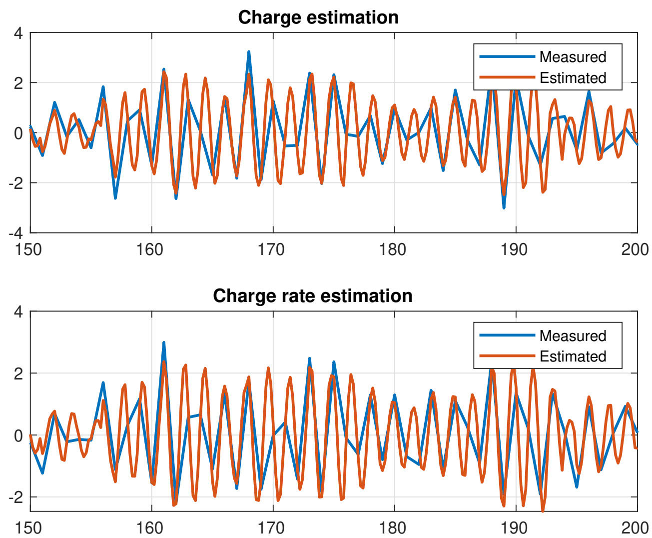

Figure 4: Measured and estimated values of idealy modeled case

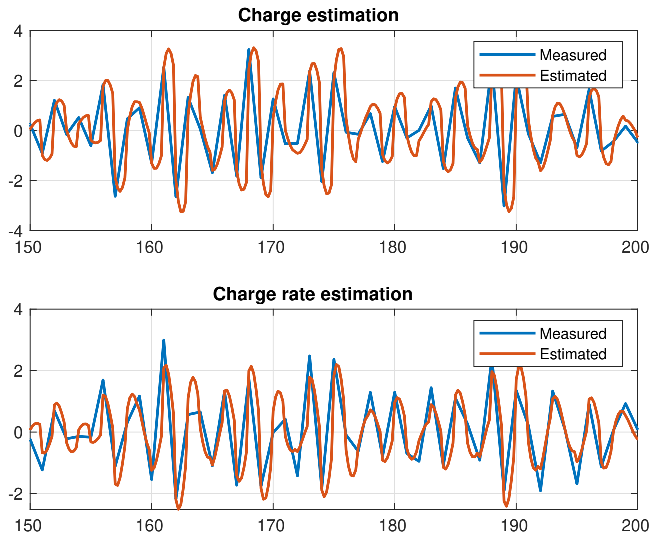

Figure 5: Measured and estimated values of model mismatched case

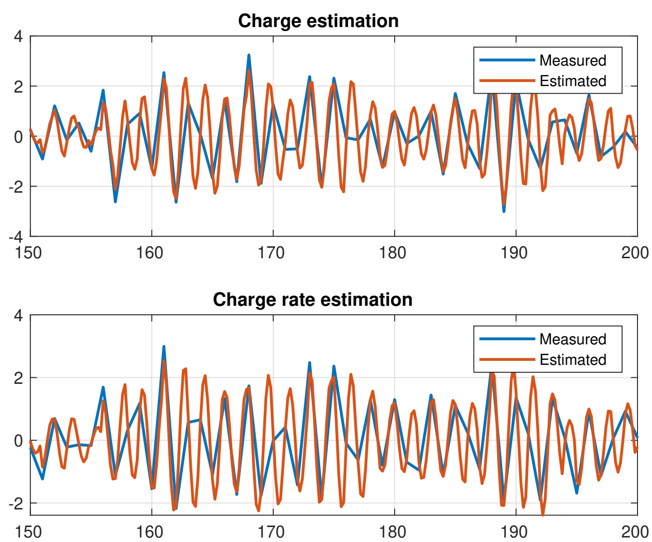

Figure 6: Measured and estimated values of noise statistics mismatched case

7.2 Innovation boundedness test

Figure 7: Innovation boundedness of idealy modeled case

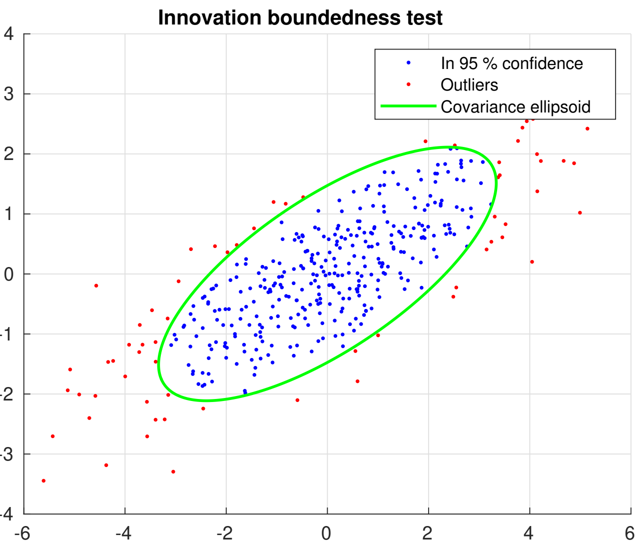

Figure 8: Innovation boundedness of model mismatched case

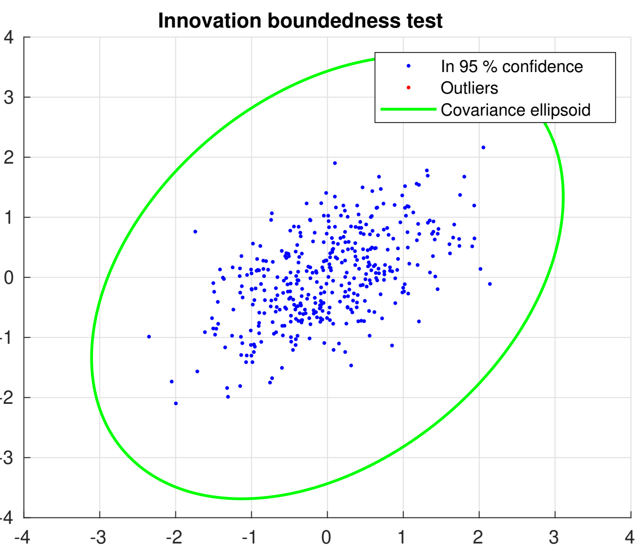

Figure 9: Innovation boundedness of noise statistics mismatched case

7.3 Innovation whiteness (autocorrelation) test

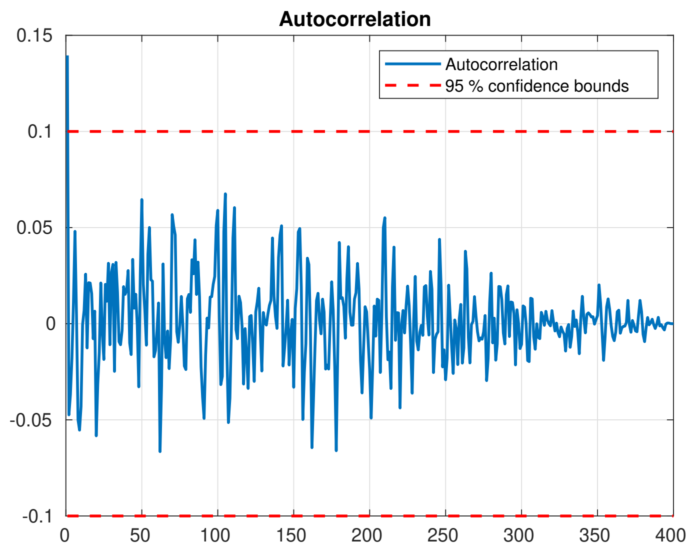

Figure 10: Autocorrelation of innovation of idealy modeled case

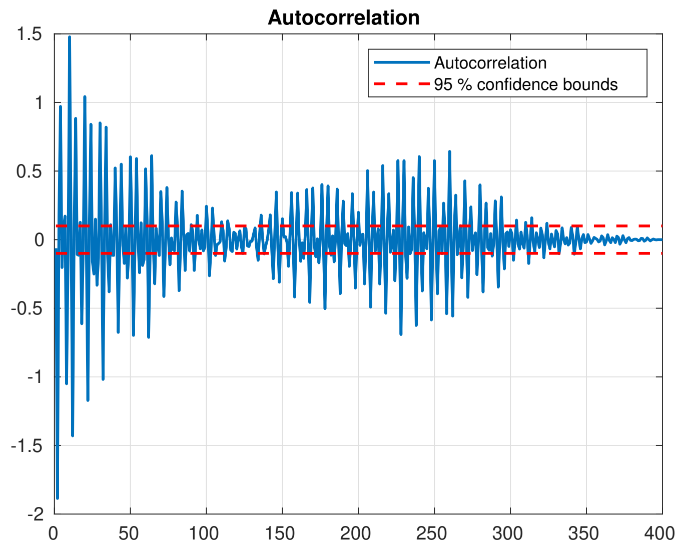

Figure 11: Autocorrelation of innovation of model mismatched case

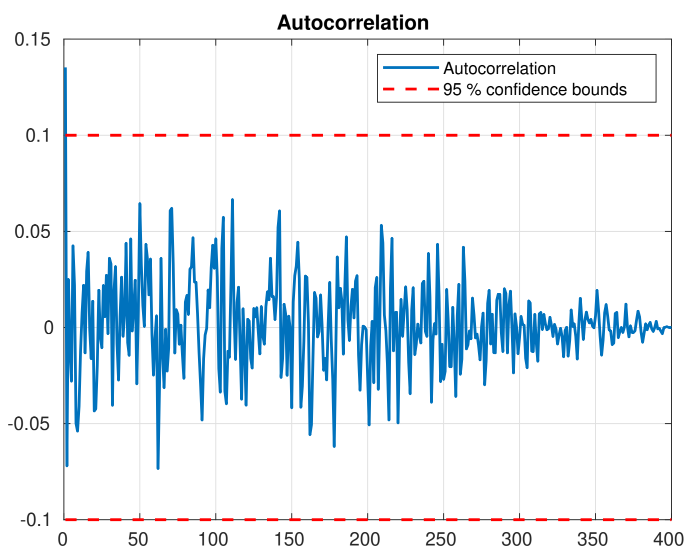

Figure 12: Autocorrelation of innovation of noise statistics mismatched case

8 Conclusion

As expected Kalman filter with model mismatch performed worst of all algorithms and didn't pass any test. Example of this could be seen in figure 5. Kalman filter with noise statistics mismatch passed two out of three innovation tests. This might happen fue to the fact that all changed parameters of covariance matrices were slightly scaled up and because of that test boundaries were slightly extended as well. I think that if changed covariance matrices paramteres were scaled down instead of scaled up this algorithm would perform a lot worse in innovation tests.

9 Appendix

In file *inputs.mat* are input vectors used to get presented results.