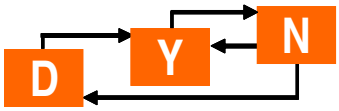
TECHNISCHE UNIVERSITÄT DORTMUND
Heiko Brandt, Yi Yu
Dep. of Biochemical- and Chemical Engineering
Chair of Process Dynamics and Operations
Prof. Dr.-Ing. Sebastian Engell



State Estimation - Extended Kalman Filter

Solutions

Exercise - Discrete extended Kalman filter

1.

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

2. Transform the system to a discrete-time system by applying the finite difference approach: The system can be rewritten as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \begin{pmatrix} \frac{\dot{V}}{V_R} c_{A0} - \frac{\dot{V}}{V_R} x_{k,1} - x_{k,4} x_{k,1} \\ x_{k,4} x_{k,1} - \frac{\dot{V}}{V_R} x_{k,2} - k_{BC} x_{k,2}^2 + k_{CB} x_{k,3} \\ k_{BC} x_{k,2}^2 - \frac{\dot{V}}{V_R} x_{k,3} - k_{CB} x_{k,3} \\ 0 \end{pmatrix}$$

with

$$x_k = [c_A,\ c_B,\ c_C,\ k_{AB}]^T.$$

The implementation can be found in the solution m-file of `model_discret`.

3. Jacobian of the discretized system:

$$A = \begin{pmatrix} 1 - \frac{\dot{V}}{V_R}\Delta t - x_{k,4}\Delta t & 0 & 0 & -x_{k,1}\Delta t \\ x_{k,4}\Delta t & 1 - \frac{\dot{V}}{V_R}\Delta t - 2k_{BC}x_{k,2}\Delta t & k_{CB}x_{k,3}\Delta t & x_{k,1}\Delta t \\ 0 & 2k_{BC}x_{k,2}\Delta t & 1 - \frac{\dot{V}}{V_R}\Delta t - k_{CB}\Delta t & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The implementation can be found in the solution m-file of `model_discret`.

4. Q might be:

$$Q = \begin{pmatrix} 1e-6 & 0 & 0 & 0 \\ 0 & 1e-6 & 0 & 0 \\ 0 & 0 & 1e-6 & 0 \\ 0 & 0 & 0 & 1e-6 \end{pmatrix}$$

To reduce the number of adjustable parameters it is suitable to chose $P0$ to be dependent on Q:

$$P0 = Q * 200.$$

This results in an observer behavior as depicted in Figure 1:

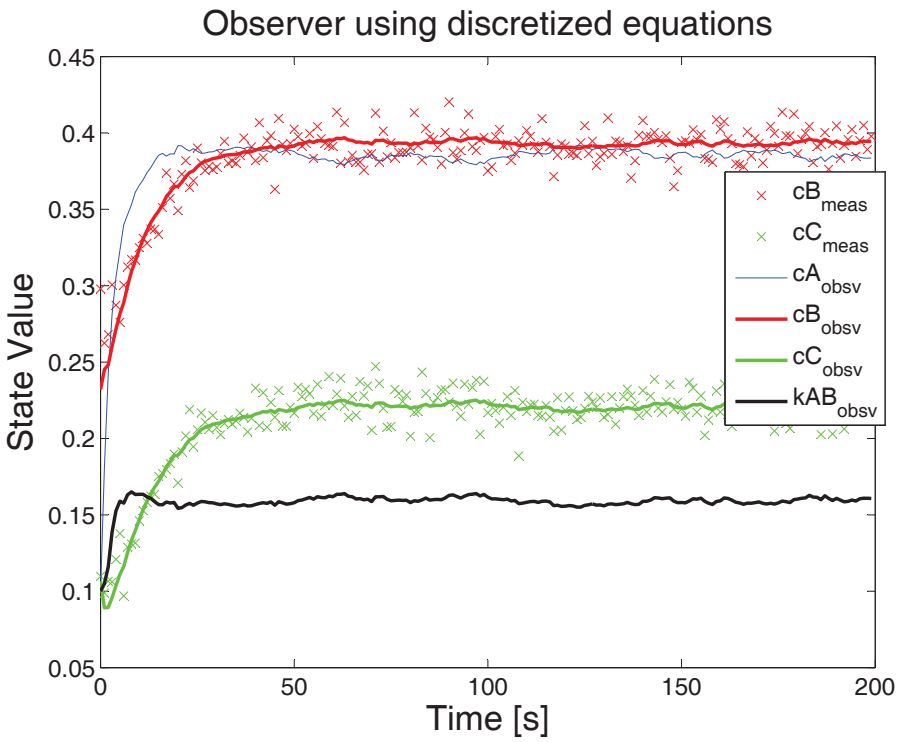


Figure 1: Performance of a discrete Kalman Filter

The tuning is quite good as the observer converges quickly to the real state values. Thereby the convergence might also be achieved by even smaller entries of Q, but the error dynamic would not be that fast. On the other hand, greater entries of Q lead to faster observer reactions by what the measurement noise gets amplified and the state trajectories are fluctuating around their values.

5. The issue in having only discrete measurements (which is in principle the case for all measurements) is, that no continuous measurement update of the observer system is possible. Between two sampling points the state values depend only on the simulation of the model (prediction), therewith plant model mismatch as well as numerical errors have an influence on the observer behavior. The choice of the numerical method which is used to solve the differential equations is therefore an important

point. Here the simple forward difference method (explicit Euler) was applied to the system to discretize between to sample points and to integrate the equations. For such a simple method two things have to be kept in mind:

- The numerical error decreases with decreasing integration step width until the machine accuracy is reached, but the computation time increases as the solution is calculated at more points.
- The numerical error increases with greater prediction periods, i.e. the error between model and plant increases between two sampling points. The greater the sampling time, the greater the numerical error. Therefore a smaller sampling time and therewith a more frequent correction makes tuning much easier. However, in most cases you cannot choose the sampling rate as it depends on the speed of your analytic devices.

That the numerical error plays a role can be observed when the integration step width is changed from 0.1 to 1. The obtained observer performance using the above given tuning is shown in Figure 2. The simulation crash is due to the fact that the

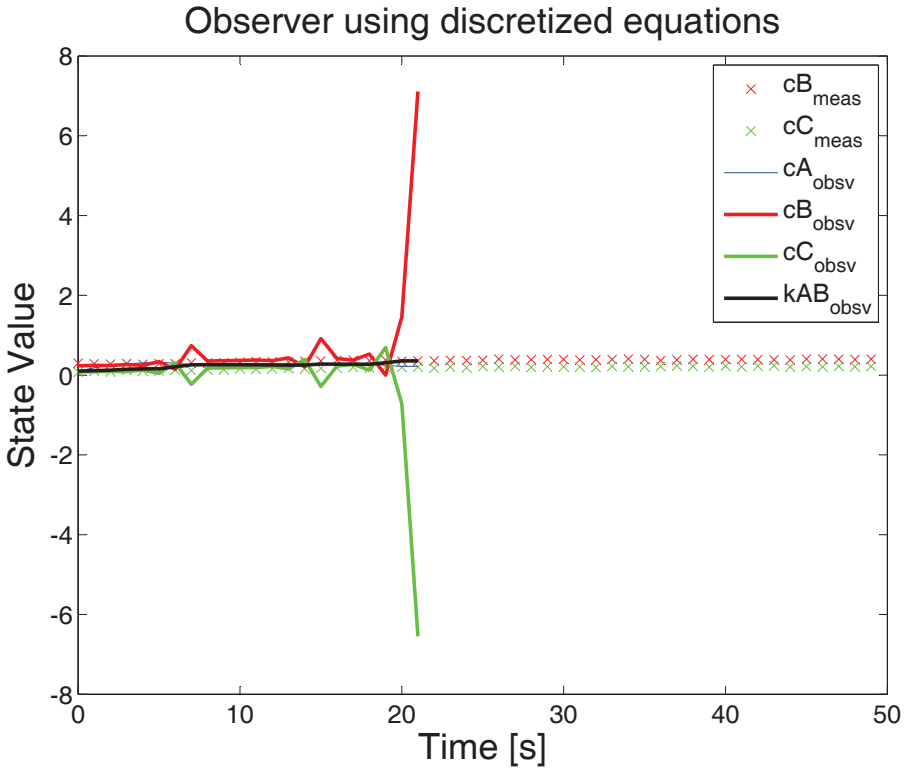


Figure 2: Performance of a discrete Kalman Filter with increased integration step width

explicit Euler uses only the gradient at time point k to calculate the state value at point $k+1$ by multiplying the gradient with the step width. When the step from k to $k+1$ is to large, some of the concentration become negative, leading finally to the simulation crush.
This numerical error is not corrected by the observer with the tuning found above.

The ratios $Q(2,2)/R(1,1)$, $Q(3,3)/R(2,2)$ are less than one, meaning that we trust the model more than the measurements as it is assumed that the model noise is "smaller" than the measurement noise. Therefore the correction done by the observer is too weak to compensate the bad numerical solution at point $k+1$. Changing the tuning to $Q(2,2)/R(1,1)$, $Q(3,3)/R(2,2) > 1$ make the observed states follow the measured values (cf. Figure 3). The observer becomes more noise, but still it's performance is quite good. Q might be:

$$Q = \begin{pmatrix} 1e-5 & 0 & 0 & 0 \\ 0 & 1e-3 & 0 & 0 \\ 0 & 0 & 1e-3 & 0 \\ 0 & 0 & 0 & 1e-5 \end{pmatrix}$$

$$P0 = Q * 200$$

.

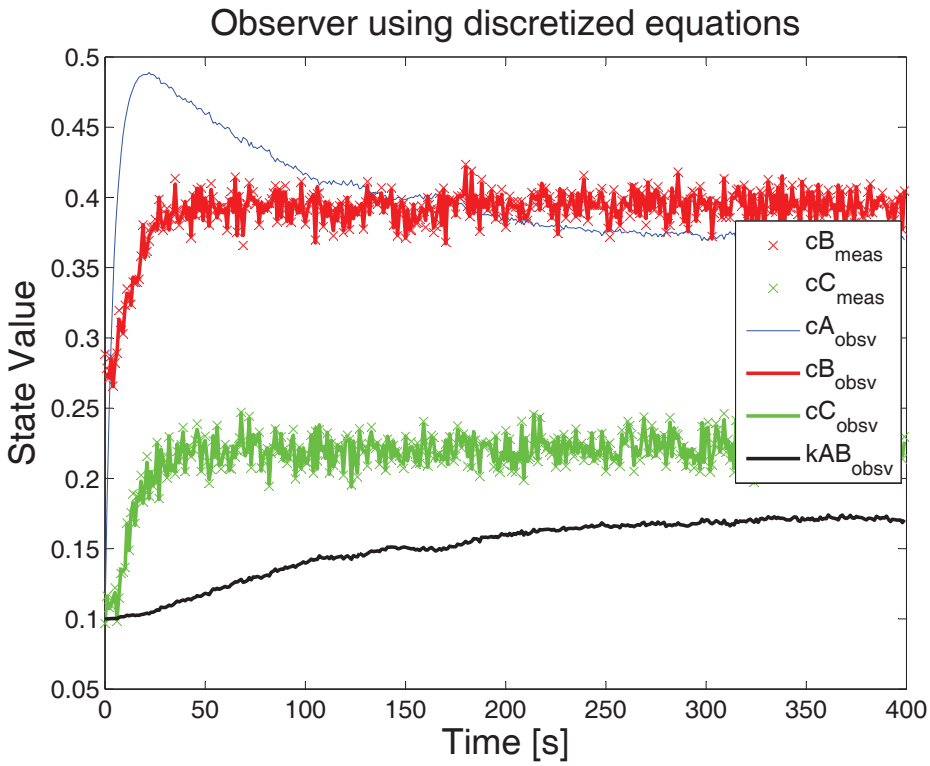


Figure 3: Performance of a discrete Kalman Filter with increased integration step width

Conclusion: The numerical method and its integration step width have a direct influence on the estimates. For this reason it is in theory the best to use the continuous equations between two samples. This is, of course not possible, but one should think about higher order methods to integrate the system.

6. As mentioned above, the numerical method has a strong influence on the observer behavior. Therefore the performance can be improved by using higher order methods for numerical simulation than explicit Euler as they are implemented in MATLAB (e.g. `ode45`, based on Runge-Kutta of 4th order): Instead of discretizing the

system, the continuous equations are implemented and solved between two sample times (they are solved numerically at discrete points by MATLAB). The measurements are used to correct the observer before again the system is simulated till the next sample point. The performance is depicted in Figure 4. Q might be:

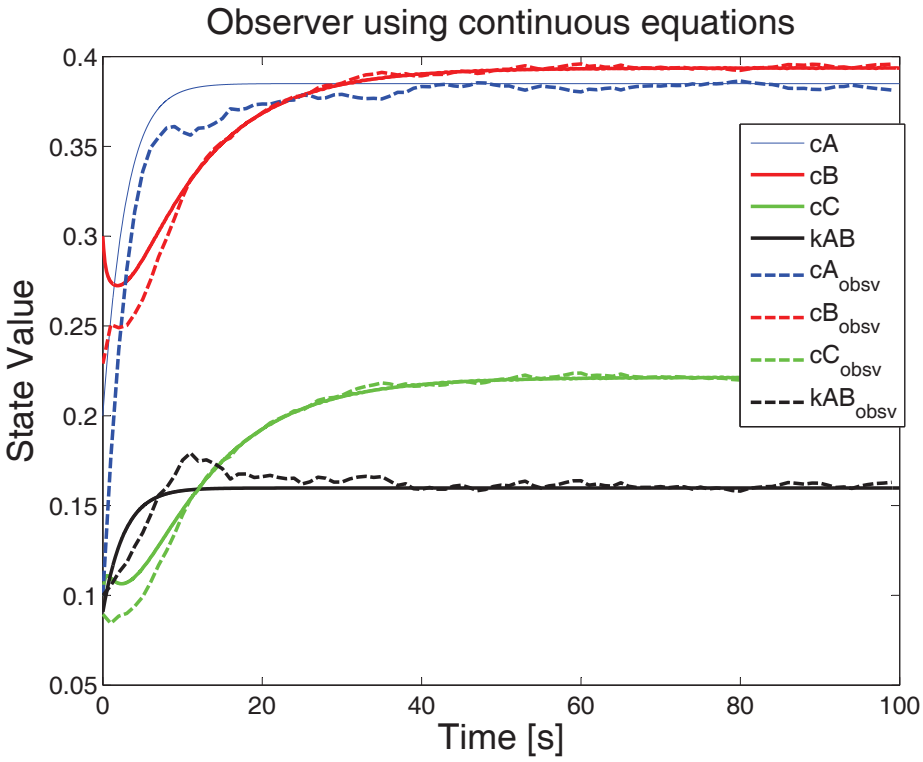


Figure 4: Kalman Filter performance using the continuous equations between two sampling points

$$Q = \begin{pmatrix} 1e-6 & 0 & 0 & 0 \\ 0 & 1e-6 & 0 & 0 \\ 0 & 0 & 1e-6 & 0 \\ 0 & 0 & 0 & 1e-6 \end{pmatrix}$$

$$P0 = Q * 200$$

.