Lehrstuhl für Systemverfahrenstechnik
Prof. Alexander Mitsos, PhD

Institut für Geometrie und Praktische Mathematik
Prof. Dr. A. Reusken

Model-based Estimation Methods, SS 2020

Exercise 11 – Bonus (Due on July 01, 2020, 5 pm)

Programming task

Submit your MATLAB-script until 17:00 on July 01, 2020 via the course room in RWTHMoodle and be prepared for a short Q&A session on your results during the exercise on July 02. Only one person of the team should submit the solution. Please write down the names and matriculation numbers of both team members at the beginning of your file.

Bonus problem 6: Continuous Luenberger observer (2 Bonus points)

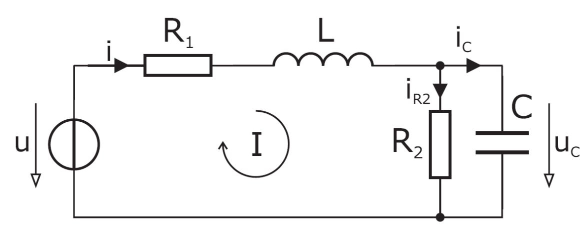

Figure 1: Passive electronic circuit

Consider the passive electronic circuit depicted in Figure **??**. It can be described by the following mathematical equations

$$\begin{aligned}\frac{du_c(t)}{dt} &= -\frac{1}{c \cdot R_2}u_c(t) + \frac{1}{c}i(t)\\ \frac{di(t)}{dt} &= -\frac{1}{L}u_c(t) + \frac{R_1}{L}i(t) + \frac{1}{L}u(t)\ ,\end{aligned} \tag{1}$$

where $c = 2$, $L = 12$, $R_1 = 0.5$, and $R_2 = 4$ are constants and $u(t) = 5 \cdot (1 - \exp(-\frac{t}{2})$ is the input voltage. Estimate the trajectories of both differential states for $t \in [0, 200]$ based on measurements for voltage u_c.

i. Rewrite Model (**??**) as a continuous LTI system of the form

$$\begin{aligned}\dot{x} &= Ax + Bu, \quad x(0) = x_0 \\ y &= Cx + Du\ .\end{aligned} \tag{2}$$

ii. Implement a stable continuous-time Luenberger observer for Model (??). Consider the initial dates to be $x_0 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ and the measurements at equidistant time points $\tilde{t} \in \{0, 1, 2, \ldots, 200\}$ as given in `uc.csv`. Assume that the dynamic systems can be simulated exactly by MATLAB function `ode45`.

Plot the estimated model prediction together with the measurements.

Hints:

- You can use MATLAB function `place` to determine a suitable gain K.
- MATLAB code `uc=csvread('uc.csv');` allows to read the provided measurement data into a row vector `uc`, where `uc(1)`= $u_c(\tilde{t}_1) = u_c(0)$ and so on.
- For the right-hand side of the state equation of the Luenberger observer, you can use 1) a time-dependent measurement `y=@(t) interp1(...);` based on MATLAB function `interp1` and the given measurement data, and 2) a state-dependent output estimation `y_hat=@(x_hat) (...);` based on current estimation `x_hat` and system matrix C.

iii. Tune your Luenberger observer by changing gain K. Does the Luenberger observer give the exact model output, now?