Lecture Power Electronics Devices and Circuits - Control Part
University of Freiburg – Summerterm 2017

Exercise 1: State Space Control in MATLAB

Prof. Dr. Moritz Diehl, Dr. Gianluca Frison and Benjamin Stickan

For questions on the exercise please contact Benjamin Stickan (`benjamin.stickan@imtek.uni-freiburg.de`)

Within the control part of the "Power Electronic Devices and Circuits" course there will be two exercise sheets that should be worked on at home. Each exercise will be handed out and explained on Fridays. At the beginning of the next Lecture, the solutions will be discussed (apprx. 45 min.). The exercises require a **MATLAB** installation including the Control System Toolbox and a **PLECS** installation (Standalone or Blockset).

Getting started

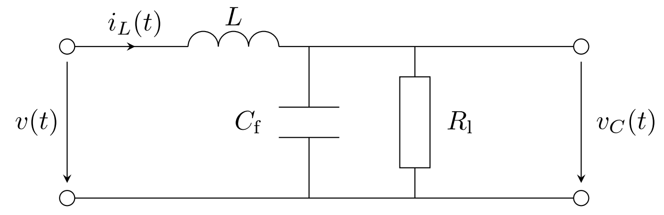
1. As **PLECS** should be already installed on your computer, the first thing you need to do is to install **MATLAB**. Detailed installation and licensing instructions can be found at
   `https://www.rz.uni-freiburg.de/services-en/beschaffung-em/software-en/matlab-license`
   Remember that the Control System Toolbox is required.
2. If you are new to **MATLAB**, the first thing you will appreciate is the extensive help system. You can simply type `doc` into the console and the documentation opens. If you type `doc plot`, you will find a detailed description of function `plot`.
3. Here are some useful commands for the exercises:
```
hold on/off
figure
close all
clear
clc
```

Tasks

1. The electrical circuit sketched below shows a simplified buck-converter with a constant load at the output. The system can be described in state-space representation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \quad y = \mathbf{C}\mathbf{x}, \quad \mathbf{D} = [0],$$

with the state vector given as $\mathbf{x} := \begin{bmatrix} i_L & v_C \end{bmatrix}^\intercal$, input $u := v$ and output $y := v_C$.



(a) Derive matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ using equations

$$i_C = C_\mathrm{f}\frac{dv_C}{dt}, \quad v_L = L\frac{di_L}{dt} \quad \text{and} \quad i_R = \frac{v_C}{R_\mathrm{l}}$$

(Hint: Use Kirchhoff's voltage law for inductors and current law for capacitors)

(b) Derive the characteristic polynomial. Evaluate the eigenvalues of the system for $L = 4.7$ mH, $C_\mathrm{f} = 100\ \mu$F and

i. $R_\mathrm{l} = \infty\ \Omega$
ii. $R_\mathrm{l} = 100\ \Omega$

Is the system BIBO-stable in both cases?

(c) Write down the time constant τ in seconds and the resulting oscillating frequency in Hz for both values of R_l.

(d) Create a new MATLAB script and define variables L $= 4.7$ mH, Cf $= 100\ \mu$F and Rl $= \infty$. Also define matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D} = 0$ according to task (1a).

(e) Use the `ss(A,B,C,D)` command to create a state-space model `sys_ol` and evaluate the systems step response with the `step(sys,Tfinal)` function (Tfinal = 0.05 s) for

  i. $R_\mathrm{l} = \infty\,\Omega$

  ii. $R_\mathrm{l} = 100\,\Omega$

2. The aim of this task is to design an LQR controller for the buck-converter to **track** a voltage reference y_ref at the capacitor. To track a reference with the controller, it is at first useful to calculate an equilibrium point (steady-state) of the system when the desired output resides at the reference. In this case, the following equations hold:

$$\dot{\mathbf{x}}_\mathrm{ss} = \mathbf{A}\mathbf{x}_\mathrm{ss} + \mathbf{B}\mathbf{u}_\mathrm{ss} = 0, \quad y = \mathbf{C}\mathbf{x}_\mathrm{ss} = y_\mathrm{ref}, \quad \mathbf{D} = [0],$$

(Remark: This is the alternative way of deriving the prefilter gain $\mathbf{N}$ described in the script in section 3.3)

(a) Calculate the steady-state input u_ss and state vector $\mathbf{x}_\mathrm{ss}$ on paper as functions of y_ref.

(b) Define weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ in MATLAB. Set the penalty on the voltage state to 1, current state to 0.001 and penalize the control by 1. Calculate the feedback-gain $\mathbf{K}$ using the MATLAB function `lqr(A,B,Q,R,[])`.

(c) The closed loop system including reference tracking is now defined as:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u && (1)\\ y &= \mathbf{C}\mathbf{x}, \quad \mathbf{D} = [0], && (2)\\ \text{with} \quad u &= u_\mathrm{ss} - \mathbf{K}(\mathbf{x} - \mathbf{x}_\mathrm{ss}). && (3) \end{aligned}$$

Derive the matrices $\mathbf{A}_\mathrm{cl}$, $\mathbf{B}_\mathrm{cl}$ and $\mathbf{N}$ that describe equation (1) in the form

$$\dot{\mathbf{x}} = \mathbf{A}_\mathrm{cl}\mathbf{x} + \mathbf{B}_\mathrm{cl}y_\mathrm{ref}$$

with $\mathbf{B}_\mathrm{cl} = \mathbf{B}\mathbf{N}$ and implement them in MATLAB.

(d) To plot the system states as well as the control, create a new output matrix

$$\mathbf{C}_\mathrm{cl} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \multicolumn{2}{c}{-\mathbf{K}} \end{bmatrix}$$

Derive the corresponding matrix $\mathbf{D}_\mathrm{cl}$ from eqn. (3) and set up a new model `sys_cl` for the closed-loop system in MATLAB using the `ss(A,B,C,D)` command. Use the `step(sys,Tfinal)` function to evaluate the step-response of the closed-loop system (Tfinal = 0.05 s, $R_\mathrm{l} = 100\,\Omega$).

3. The developed controller can be implemented in a PLECS model with little effort.

(a) Open the PLECS model `LQR_buck.plecs` and insert the numeric values for the prefilter $\mathbf{N}$ and feedback gain $\mathbf{K}$ you calculated in task (2).
(If you could not finish task (2), you can use $\mathbf{N} = 1.118$ and $\mathbf{K} = \begin{bmatrix}2.8940, & 0.0891\end{bmatrix}$.)

(b) Compare the <u>shape</u> of the state trajectories you obtained from the `step()` command and the PLECS simulation results.

(c) Explain why the trajectories are are not exactly identical. (What happens to the main inductor current?)