TA Solutions, Computerpracticum 1 Control Engineering

Modeling and Analysis of a Coreless DC Motor

1 Introduction and objective

The DC (direct-current) motor is one of the first machines devised to convert electrical energy into mechanical energy. The principle of operation originated in 1821 as a result of experiments performed by Michael Faraday (1791-1867). Nowadays, DC motors can be found in a wealth of commercial products including servo-positioning systems, disk drives, robotic manipulators, laser printers and photocopiers, to name but a few.

In this assignment you will investigate the dynamic behavior of a coreless DC motor with help of MATLAB and Simulink. A coreless DC motor is a specialized form of a brush DC motor. Major advantages of coreless motors include very low intertia, low mechanical time constant, and high efficiency. Because the corse is ironless, its low mass allows more rapid acceleration and deceleration than any other class of DC motor. Coreless DC motors are most often used in high-performance servo-controlled systems.

The objective of this assignment is twofold: First, you will get familiarized with MATLAB and Simulink in the context of control systems analysis and design (Section 2-4). Second you will gain insight in the behavior of the DC motor system (Section 2) and use this insight to design a state feedback controller to control the speed and position of the DC motor (Section 3).

- **The assignment consists of six exercises divided into three parts: system modeling and analysis, a brief introduction to Simulink, and state feedback design.**
- **The first exercise concerns the derivation of the differential equations describing the system. It is required to prepare the first exercise at home before the first MATLAB session!**
- **A start-up file including all parameters can be found on Nestor under `Assignments > Practicum`. You are kindly requested to use this file throughout the practicum, to allow the teaching assistants to quickly check your progress.**

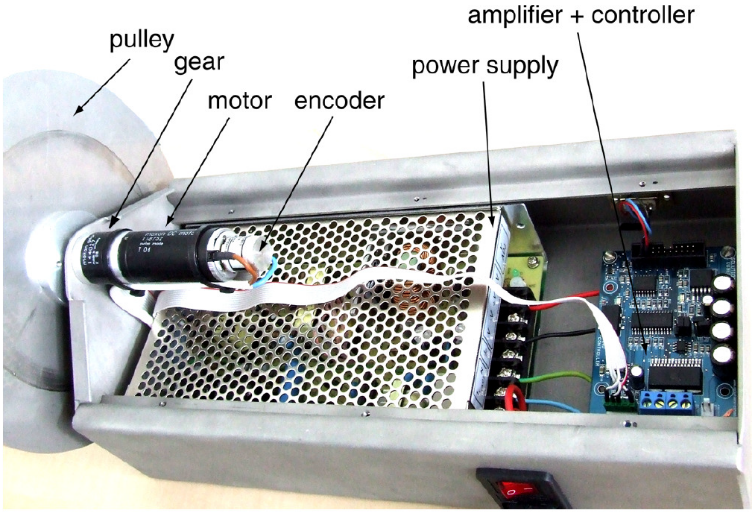

Figure 1: DC motor setup

2 System model

The physical system that we are going to investigate is shown in Figure 1. It consists of a 20W coreless Maxon Gear DC motor (Maxon RE 25, 118752) driving an intertial load (aluminium pulley) through a Maxon gear box (Maxon GB 26 B, 144027). The system is driven by a PWM (pulse-width-modulated) power-amplifier (H-bridge). Motor speed and position are measured using an encoder (Scancon 2MCH).

In order to adequately describe a physical system, it is important to have knowledge of the physical parameters that characterize the system. These parameters can include the properties of the materials that are incorporated in the system, the characteristics of the actuators and sensors of the system, as well as the operating ranges (the limits of the system). The Maxon motor specifications are given in Table 1. The intertia of the aluminium pulley, denoted by J_p, is given by the formula

$$J_p = \frac{1}{2} m_p r_p^2,$$

where $m_p = 0.25kg$ is the mass of the pulley, and $r_p = 0.095m$ is the radius of the pulley. The gear ratio is 4.4 : 1 (i.e., 4.4 shaft revolutions equal 1 pulley revolution). The dynamics of the amplifier and the encoder are neglected in the remainder of this study.

Table 1: Specification Maxon RE 25, 118752

CHARACTERISTIC	VALUE	UNITS	PARAMETER
Assigned power rating	20	W	-
Nominal voltage	24	V	-
No load speed	9550	rpm	-
No load current	37	mA	-
Starting current	1040	mA	-
Terminal resitance	2.32	Ω	R
Terminal inductance	0.24	mH	L
Max. permissible speed	11000	rpm	-
Max. continuous current	1210	mA	-
Max. continuous torque	26.1	mNm	-
Max. power output at nominal voltage	58300	mW	-
Max. efficiency	85	$\%$	-
Torque constant	23.4	$\frac{mNm}{A}$	k
Mechanical time constant	4	ms	τ_m
Rotor intertia	$10.3 \cdot 10^{-7}$	$kg\,m^2$	J_m

$*$ Preparation exercises

Exercise 1: Physical modeling of the system

Figure 2 shows a schematic representation of the DC motor system. The input of the system is the armature voltage $u[V]$. The corresponding current is denoted by $i[A]$. Measured variables are the angular displacement $\theta[rad]$ and the angular velocity $\dot{\theta}[rad/s]$ of the shaft.

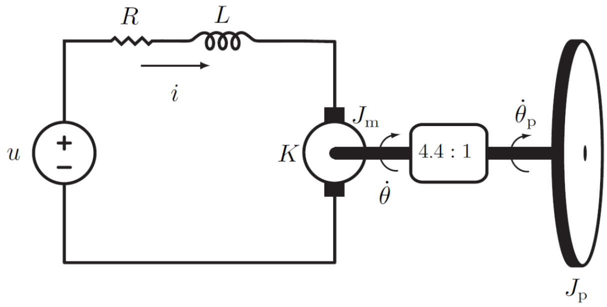

Figure 2: Schematic representation of a geared DC motor

Let $\theta = q_m$ and $\dot{\theta} = \dot{q}_m$ denote the generalized mechanical displacement and velocity, respectively. Furthermore, let the electrical part of the system be defined by the armature loop charge $q = q_e$ and the armature current $i = \dot{q}_e$. The Langragian function for the system is given by

$$\mathcal{L}(q_m, \dot{q}_m, q_e, \dot{q}_e) = \frac{1}{2}L\dot{q}_e^2 + \frac{1}{2}J_t\dot{q}_m^2 - \frac{1}{2}k\dot{q}_m q_e + \frac{1}{2}k\dot{q}_e q_m.$$

Determine

- The equations of motion.
- Two state space descriptions: One state space description with output $y = \dot{\theta}$, and one with output $y = \theta$.

Exercise 2: MATLAB modeling

An easy to use Control tutorial for MATLAB and Simulink (by Michigan Engineering) can be found at http://www.engin.umich.edu/class/ctms/. MATLAB also provides an extensive Product Help which can be opened by selecting `Help>Product Help (F1)` from the menubar or directly by pressing F1 on your keyboard. Command specific help is easily accessible by typing `help command` in the command window (e.g. `help ss` will provide you some basic information about the `ss()` command for defining state space models).

Please present your results in one of the following ways

1. **Showing the MATLAB command line output directly in a structered way**
2. **Copy-pasting the command line output into Microsoft Word**
3. **Writing down the command line output in the boxes below the questions in this manual**

Plots should be stored in separate figure-windows, such that the teaching assistant can check your results easily

- Determine the parameters needed for your model ($R, L, k, J_t = J_p + J_m$)
- Download the MATLAB script `cp1_startup.m` from Nestor, and declare your system parameters within this file. End all lines with a “`;`”.
- Declare your state space models in the script (i.e., matrices `A1`,`B1`,`C1`,`D1` for the system with output $y = \dot{\theta}$, and matrices `A2`,`B2`,`C2`,`D2` for the system with output $y = \theta$), using “`ss(A,B,C,D)`”
- Save your file and press the “Run script”-button or select from the menubar `Debug>Run cp1_startup.m (F5)` to run. Make sure no error messages are displayed in the command window.
- Let MATLAB display your state space systems by typing in their names in the command window. Write down the result.

Exercise 3: Stability

- Calculate the eigenvalues of the A-matrices found in the previous exercise.
- Is the system stable? Explain.

Exercise 4: System response

- Plot the impulse response of both $\dot{\theta}(t)$ and $\theta(t)$.

  Show MATLAB figure to teaching assistant

- Plot the step response of both $\dot{\theta}(t)$ and $\theta(t)$ in a new figure window.

  Shown MATLAB figure to teaching assistant

3 State feedback design

The goal of this exercise is to design a state feedback controller that assigns the eigenvalues of the system to a desired location. In this exercise we only consider the second order state space model with output $y = \dot{\theta}$.

Exercise 5:

The MATLAB command `poly([`$x_1, \ldots, x_n$`])` returns the coefficients $y_1, \ldots, y_n$ of the polynomial $y(s) = (s - x_1)(s - x_2) \ldots (s - x_n) = y_1 s^n + y_2 s^{n-1} + \ldots + y_{n-1} s + y_n$.

- Use the `poly()` command to determine the coefficients a_1, a_2 of the open loop characteristic equation

$$\lambda(s) = s^2 + a_1 s + a_2.$$

- Verify that the system is reachable using the reachability matrix W_r.

- Assume the desired location of the closed loop eigenvalues is $-100 \pm 100i$. Determine the coefficients p_1, p_2 of the closed loop characteristic polynomial

$$p(s) = s^2 + p_1 s + p_2.$$

- Using Theorem 6.3 calculate the feedback gain matrix K_{sf} and reference gain k_r.

- Verify that using your state feedback design, the closed loop eigenvalues are assigned at $100 - 100i$.

- Compare the open and closed loop step response. Explain the differences.

Show MATLAB figure to teaching assistant Differences:

4 Introduction to Simulink

This last exercise is to get you familiar with Simulink. Simulink is an extension to MATLAB which uses an icon-driven interface for the construction of block diagram representations of a system. Simulink is started by typing `simulink` on the MATLAB command line or by pressing the Simulink-button on the toolbar.

Exercise 6: Block diagrams

- Enter the second order system dynamics $\dot{x} = Ax + Bu$ (see Exercise 1) in Simulink. A short overview of the blocks you might want to use can be found on Nestor at `Assignments>Practicum>Useful MATLAB and Simulink commands`.
  Note that you should not use the standard State-Space block from the Simulink Library. Using this block makes implementing the state feedback controller (see last step of this exercise) unnecessary complicated.

  Show Simulink model to teaching assistant

- Now extend your Simulink model with the output equation $y = Cx + Du$ with output $y = \dot{\theta}$. Note that you obtain only one Simulink model, to model two sets of equations! Add a step input to your model and check the response.

  Show Simulink model to teaching assistant

- Lastly, implement the state feedback controller you designed in Exercise 5. Do not use A_{cl} and others, but rather the values for K_{sf} and k_r you calculated. Apply a step function to the closed loop system, and compare the response to the open loop. What is your conclusion?

  Show Simulink model to teaching assistant

  Conclusion: