ANU ENGN3223/6223 Control Systems S2/2022

Lab 1: Introduction to Simulink & Open Loop Response

Instructions regarding the Lab Report (worth 10% of the course):

- In lab sessions, you follow the lead of your tutor to setup Simulink for experiments, collect data, and understand the lab report questions.
- You will hand in your answers for Labs 1-4 in a single lab report (per group) in Week 12. (Submission through wattle. Deadline 27/10/22, 17:00pm.)
- Submission needs to be a single pdf-file written in LaTeX or Word. For your solutions, full sentences in your answers are expected, your answers need to be well justified and formatting will be taken into account in the marking process.
- Clearly indicate the names and the SIDs on the first page of the report.
- Clearly indicate if your solutions are based on 'hardware experiments' (in person labs) or 'hardware simulations' (online labs). For students using 'hardware simulations' additional information is given on the last page.

In this lab we will look at:

- learning to use Simulink with block diagrams and first order systems,
- performing *parameter identification* for a simulated servo motor — finding physical properties of the hardware from experiment.

Background

Modeling

In this laboratory we will use the Quanser Controls Board (with inertia disc) mounted on an NI Elvis III device (see Figure 1) to identify parameters of the DC motor of the Controls board. Here, the DC motor represents a plant with unknown dynamics which we intend to identify.

Figure 1: Quanser Controls Board (with inertia disc) mounted on an NI Elvis III device.

The motor armature circuit schematic is shown in Figure 2. The DC motor shaft is connected to the load hub. The hub is a metal disk used to mount the red disk (see Figure 1) or a rotary pendulum and has a moment

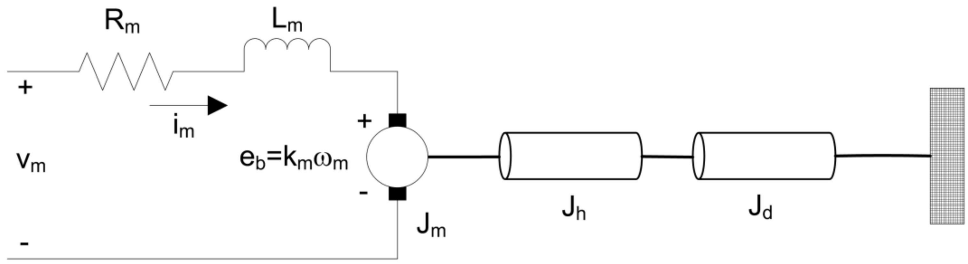

Figure 2: DC motor and load diagram.

of inertia of J_h (load hub inertia [$kg \cdot m^2$]). In Figure 1 a disk load is attached to the output shaft with a moment of inertia of J_d (disc inertia [$kg \cdot m^2$]).

The back-emf (electromotive) voltage $e_b(t)$ depends on the speed of the motor shaft, ω_m, and the back-emf constant ([$V/(rad/s)$]) of the motor, k_m. It opposes the current flow. The back-emf voltage is given by:

$$e_b(t) = k_m \omega_m(t).$$

Kirchoff's Voltage Law leads to the equation

$$v_m(t) - R_m i(t) - L_m \tfrac{d}{dt} i_m(t) - k_m \omega_m(t) = 0$$

where v_m is the input voltage (in $[V = kg \cdot m^2/(As^3)]$), R_m is the terminal resistance (in $[\Omega]$), i_m is the current (in $[A]$) and L_m is the rotor inductance (in $[H]$).

We assume that inductance is very small when compared to the resistance and hence we simplify the last expression to

$$v_m(t) - R_m i(t) - k_m \omega_m(t) = 0$$

or equivalently,

$$i_m(t) = \frac{1}{R_m} v_m(t) - \frac{k_m}{R_m} \omega_m(t). \tag{1}$$

Finally, the motor shaft equation is expressed as

$$J_{eq} \dot{\omega}_m(t) = \tau_m(t) \qquad \Longleftrightarrow \qquad \dot{w}_m(t) = \frac{1}{J_{eq}} \tau_m(t), \tag{2}$$

where J_{eq} is the total moment of inertia acting on the motor shaft and τ_m (in $[N \cdot m]$) is the applied torque from the motor. Additionally, the torque satisfies the equation

$$\tau_m(t) = k_t i_m(t) \tag{3}$$

with torque constant k_t (in $[N \cdot m/A]$).

Combining equations (1)-(3), the first order dynamics

$$\dot{w}_m(t) = \frac{1}{J_{eq}} \left(k_t i_m(t) \right) = \frac{k_t}{J_{eq}} \left(\frac{1}{R_m} v_m(t) - \frac{k_m}{R_m} \omega_m(t) \right)$$

with state/output ω_m and input v_m are obtained. With $x(t) = \omega_m(t)$ and $u(t) = v_m(t)$ the first order system is given by

$$\dot{x}(t) + \frac{k_t k_m}{J_{eq} R_m} x(t) = \frac{k_t}{J_{eq} R_m} u(t). \tag{4}$$

With the additional definitions

$$a = \frac{k_t k_m}{J_{eq} R_m} \qquad \text{and} \qquad b = \frac{k_t}{J_{eq} R_m}$$

the standard form $\dot{x} + ax = bu$ of the lecture is recovered.

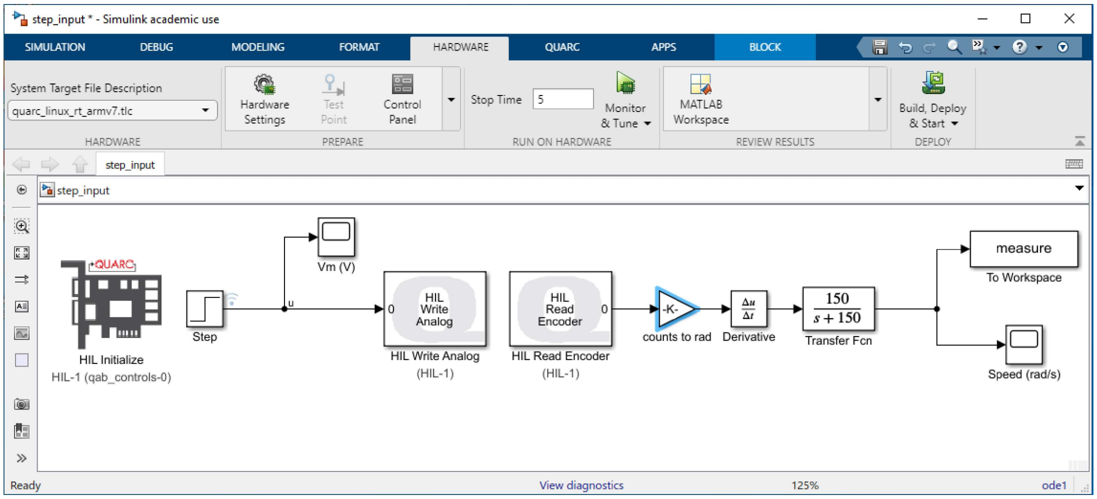

Figure 3: Simulink/QUARC interface between Matlab and the Quanser Controls Board in Figure 1.

Hardware-in-the-loop (HIL) simulation using Matlab/Simulink and QUARC

To identify the parameters a and b we perform experiments on the hardware in Figure 1 by applying input signals $v_m(t)$ to the motor and by using a sensor to measure the velocity $\omega_m(t)$.

The software package QUARC builds an interface between Matlab/Simulink and the hardware. A simple example of the graphical user interface of QUARC embedded in Simulink is shown in Figure 3. Here, from left to right,

- the 'Step' block generates a step input (i.e., voltage in the case of the DC motor),
- the 'Scope' block labeled Vm (V) visualizes the input signal (see Figure 4, left),

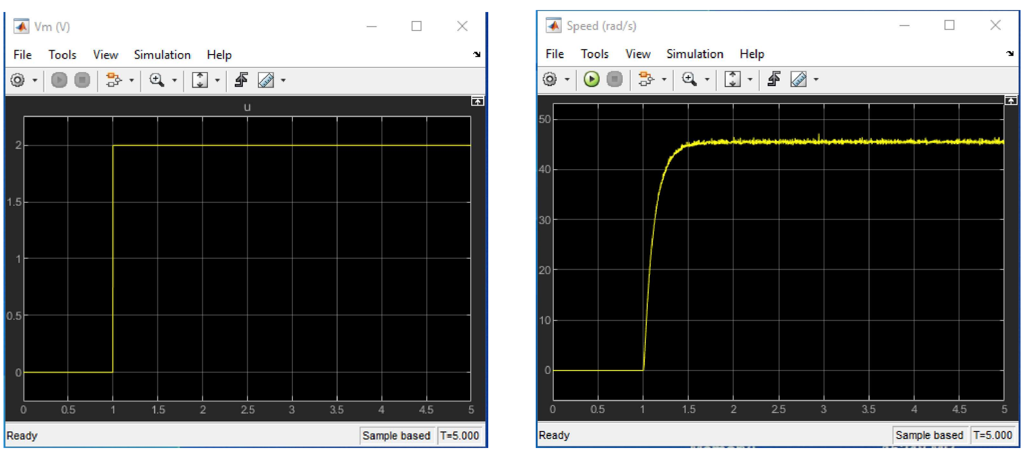

Figure 4: Input to (left) and processed measured output of (right) the Quanser Controls Board corresponding to simulation in Figure 3.

- the block 'HIL Write Analog' represents the interface between the computer and the actuator,
- the 'HIL Read Encoder' represents the interface between the sensor and the computer. **Note:** The Controls Board has a position sensor, not a velocity sensor, i.e., the block 'HIL Read Encoder' returns the position (in counts) not the velocity.
- the 'Gain' represents a multiplication which translates the position information from counts to radians,
- the 'Derivative' block takes the derivative of the signal with respect to time, i.e., the derivative of the position is the velocity/speed. **Note:** It is in general not a good idea to take the time derivative of a noisy measurement obtained through a sensor and the result needs to be taken with care.

- the 'Transfer Fcn' represents a Filter, which removes high frequencies (i.e., rapid changes) from the noisy signal. (The definition of the block 'Transfer Fcn' will become clear later in the lecture and is not important here.)
- the block 'To Workspace' saves the signal (i.e., discrete time signals and the processed sensor data) to the Matlab workspace from where it can be plotted or manipulated, for example, and
- the 'Scope' block Speed (rad/s) visualizes the processed senor data (see Figure 4, right).

Starting the Quanser Controls Board and Matlab/Simulink (for the first time)

To test if the hardware/software is working properly perform the following steps in the Lab.

1. Sign in to your Windows account in the Lab.
2. Switch on NI Elvis III board:
   - Turn the switch on the back
   - Press the button in the top left corner
3. (Only needs to be done once.) If you connect the hardware for the first time change the Target to Remote according to Figure 5.

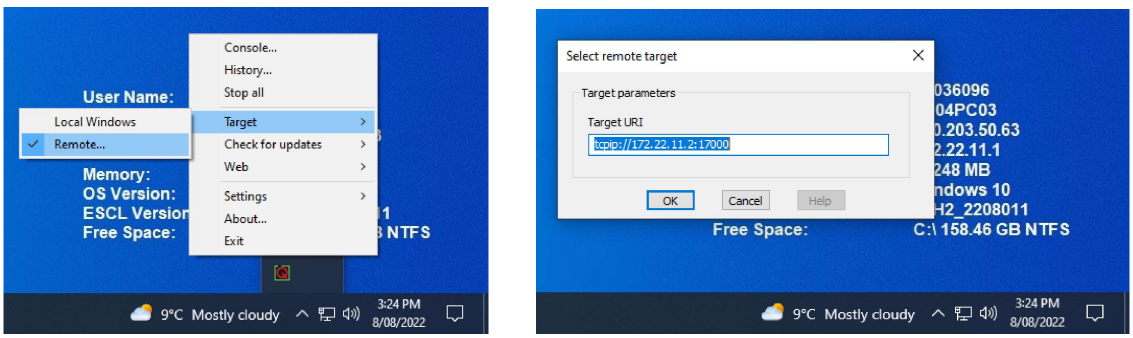

Figure 5: Change the target of QUARC (according to the screenshot on the left and the right) to be able to communicate with the hardware.

4. Open Matlab from the desktop.
5. Download the Simulink file shown in Figure 3 from Wattle, save it and open it through Matlab.
6. Run the file by clicking on 'Monitor & Tune' (see Figure 3).
7. (After a couple of seconds) The red inertia disc should spin for 5 seconds.
8. If you open the scopes, you should see something similar to Figure 4.
9. Start adapting/changing the Simulink file according to the instructions in this file and according to the instructions given in the lab. Additional Simulink blocks to be added in the Simulink file can be found in the 'Library Browser' (see Figures 6 and 7).

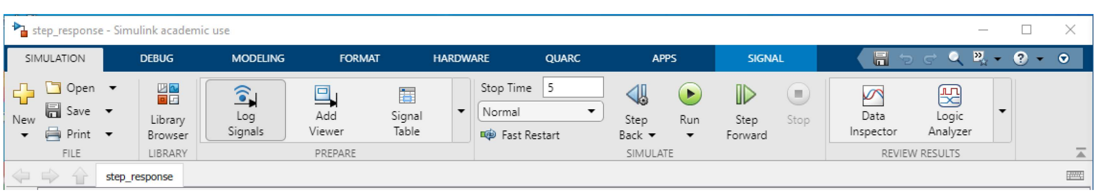

Figure 6: Additional Simulink blocks can be added to the file by opening the 'Library Browser' in the panel.

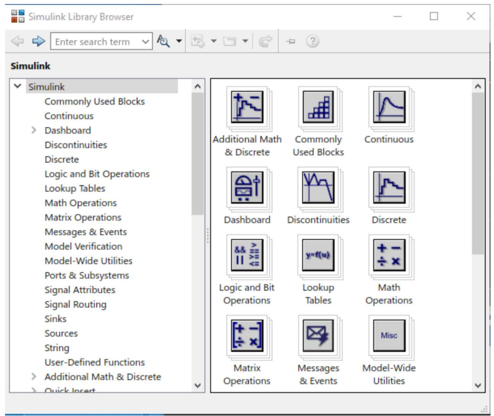

Figure 7: In the 'Library Browser' you can search particular blocks and drag and drop them to the Simulink file.

Lab questions

Pre-Work

1. Recall the first order dynamics

$$\dot{x}(t) + \frac{k_t k_m}{J_{eq} R_m} x(t) = \frac{k_t}{J_{eq} R_m} u(t)$$

motivated in (4).

(a) How does the solution of (4) with respect to the initial condition $x(0) = 0$ for a step input

$$u(t) = \begin{cases} 0, & t < 0, \\ r, & t \geq 0, \end{cases}$$

and a number $r \in \mathbb{R}$ look like?

(b) It turns out that the solution can be written in the form

$$x(t) = x(t;0) = Kr(1 - e^{-\frac{1}{\tau}t}) \qquad t \geq 0$$

where K is the steady-state gain and τ is called the time constant. In particular, for a stable system, τ is the time it takes to reach $(1 - 1/e)\%$ (about 63%) of the final (steady-state) value.
Using your answer from part (a), find K and τ in terms of the constants in (4) and r.

2. If you have not used Simulink before, watch the 'Simulink Introduction' video contained in the content of week 3 on Wattle.

Hardware experiment: System identification

3. We want to identify τ and K based on the information in Figure 4. To this end, run the file in Figure 3 and have a look at the graphs in the Scopes.

(a) What is the final change in the input value of your system (in [V])? (In particular, how large is the step between the initial input and the final input?) 2

(b) What is the corresponding change in the output value of your system (in [rad/s])? 45

(c) So what is K for your system (in [rad/(Vs)])? 22.5

(d) When (in terms of [s]) did the step begin? 1s

(e) Figure out when (in [s]) 63% of the final speed is reached. 1.117

(f) So, what is the time constant τ (in [s])? 0.117

4. Follow the instructions in the lab to extend the Simulink file shown in Figure 3. In particular, add the differential equation

$$\dot{x} = -\frac{1}{\tau}x + \frac{K}{\tau}u \tag{5}$$

in the Simulink file with τ and K defined in item 3. to compare the input-output relation in simulation and the input-output relation using the hardware.

A screenshot of the overall Simulink file needs to be included in the Lab report.

5. Save the data from the simulation and the hardware experiment to the Matlab workspace. Use `plot.m` to visualize the input (step function) and the output/state from the simulation and from the hardware experiment. (Add labels to the axes and use legends to indicate what can be seen in the plots.)

Additional Questions

6. In question 3. we have identified τ and K which we can use to identify characteristics of the DC motor. Assume that we know the terminal resistance $R_m = 8.4[\Omega]$ and assume that $k_t \cdot [A/(Nm)] = k_m \cdot [(rad/s)/V]$ (i.e., the values of k_t and k_m coincide if their units are ignored).

   Use this information to compute the overall torque J_{eq} as well as the torque and the Motor back-emf constants k_t and k_m.

7. Think about the following things (and perhaps discuss with your tutor). Write down your answers.

   (a) Now that you have a model of your system, how can you use it to determine the required control input u for a desired state x? For example, if you want the final velocity to be $x = 20$ rad/s (i.e. $\lim_{t\to\infty} x(t) = 20$, what should the input voltage be?

   (b) Compare the output of your model and the actual plant output (the encoder output). Why might you prefer to use the model rather than the raw sensor reading?

   Simpler, elegant, more responsive, and easier to adapt

Additional Instructions for Online Labs

Those students who cannot attend the Labs in person and thus do not have access to the hardware, can use the Simulink file simulating the hardware component (see Figure 8). The corresponding file does not depend

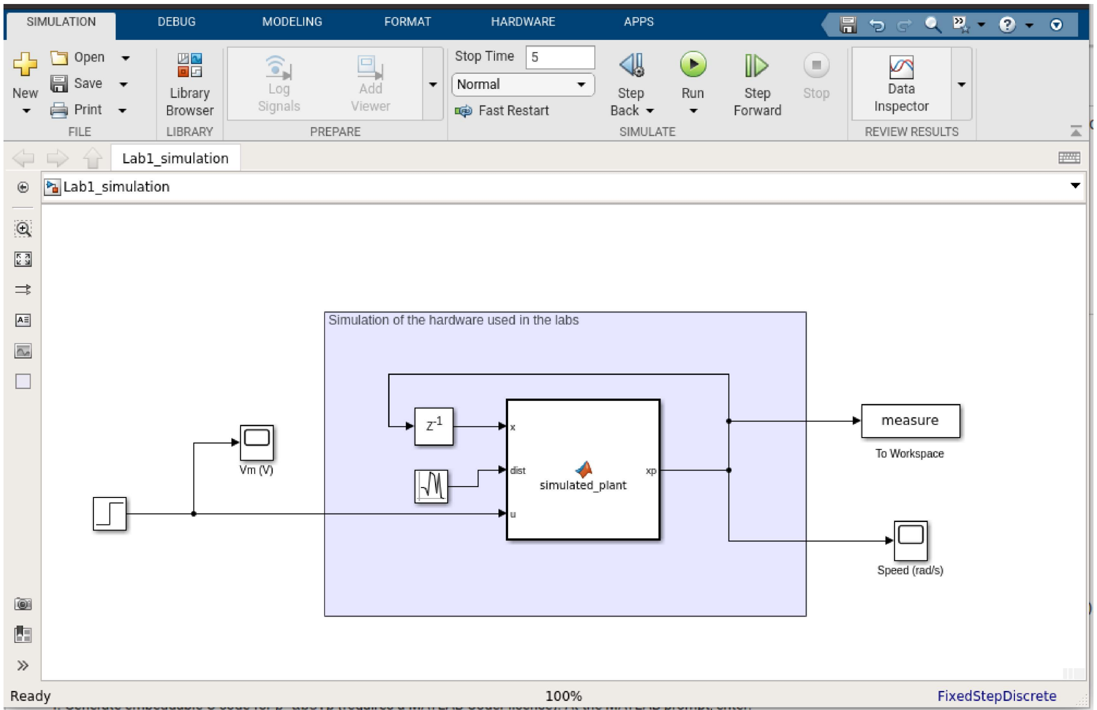

Figure 8: Simulink file simulating the hardware component in Figure 1. Press the 'Run' button to start the simulation. To run the simulation, additionally the function `plant_simulation.p` is necessary.

on the QUARC software and thus can be used to run experiments at home. To run the simulations press the 'Run' button. Additionally the file `plant_simulation.p` (on Wattle) is necessary to run the simulation.

The tasks for Lab 1 (based on the hardware or the simulation) remain the same.

Note: Your Matlab version must be at least Matlab2020a and might need to have additional Matlab toolboxes installed to be able to run the Simulink file provided on wattle and shown in Figure 8.