ELEC4631 Lab 1
Linear Algebra on Matlab and Real-Time Control using Simulink Coder

Term 2, 2020

1 General information

1. This lab is conducted as a **remote lab**, please follow the Remote Lab Instructions for ELEC4631 to attend it (can be found on Moodle).

2. The lab tasks must be carried out and completed during the lab time to be marked by demonstrators.

3. Lab tasks are divided into well defined time checkpoints: tasks that are scheduled to be completed in first **50 minutes**, in the first **1 hour 40 minutes**, and in **2 hours 30 mins.** Students must pace themselves to complete the tasks and have them assessed at the defined time checkpoints.

4. Students must not arrive late at the lab by more than 10 minutes. Demonstrators can refuse to allow late students into the lab.

2 Preparatory tasks

Learn the function of, and how to use, the following Matlab commands:

1. `det`, `trace`, `eig`, `rand`, `randi`, `rank`

2. `orth`, `null`, `qr`

   For information about a certain command, e.g., det, you can go to the Matlab command line and type, e.g., `help det`.

3 Laboratories

3.1 Laboratory task 1

(15 marks in total, to be completed and marked by 50 mins)

Write a single Matlab m-file to answer the following exercises. Explain your m-file and execute it for the lab demonstrator.

1. (10 marks) Generate five random real 10×10 matrices. Then for each matrix generated, say denoted by A, using Matlab verify the equalities

$$\begin{aligned} \det(A) &= \textstyle\prod_i \lambda_i(A), \\ \mathrm{trace}(A) &= \textstyle\sum_i \lambda_i(A), \end{aligned}$$

   Here $\lambda_i(A)$ denotes an eigenvalue of A and the product and sum above have indices ranging over all eigenvalues of A.

2. (5 marks total) Generate a random real 10×10 *symmetric* matrix A. Express A in the form

$$A = \sum_{j=1}^{10} \lambda_j(A) v_j v_j^T,$$

   for some real vectors v_j, $j = 1, 2, \ldots, 10$. On Matlab, display the quantities:

   (a) (1 mark) $\lambda_j(A)$ for $j = 1, 2, \ldots, 10$. (1 mark)

   (b) (1 mark) v_j for $j = 1, 2, \ldots, 10$.

   (c) (2 marks) $\lambda_j(A) v_j v_j^T$ for $j = 1, 2, \ldots, 10$.

   (d) (1 mark) $\sum_{j=1}^{10} \lambda_j(A) v_j v_j^T$. Show that this sum is the same as A.

3.2 Laboratory task 2

(10 marks in total, to be completed and marked by 1 hr 40 mins)

For a matrix $A \in \mathbb{R}^{m \times n}$ let $\mathrm{im}(A)$ and $\ker(A)$ denote the image and null space of A, i.e.

$$\begin{aligned} \mathrm{im}(A) &= \{Ax \mid x \in \mathbb{R}^n\} \subseteq \mathbb{R}^m \\ \mathrm{null}(A^T) &= \{x \in \mathbb{R}^m \mid A^T x = 0\}, \end{aligned}$$

A fundamental result in linear algebra states that $\ker(A^T)$ and $\mathrm{im}(A)$ are complementary subspaces:

$$\mathrm{null}(A^T) \perp \mathrm{im}(A) \quad \text{(they are orthogonal)} \tag{1}$$

$$\mathrm{null}(A^T) \oplus \mathrm{im}(A) = \mathbb{R}^m, \tag{2}$$

That is,
(1) means that $x^T y = 0$ for all $x \in \mathrm{im}(A)$ and $y \in \mathrm{null}(A^T)$.
(2) means that every $y \in \mathbb{R}^m$ can be written uniquely as $y = z + w$ with $z \in \mathrm{im}(A)$ and $w \in \mathrm{null}(A^T)$.

In Matlab, create the matrix

$$A = \begin{bmatrix} 7 & -8 & -7 & -55 \\ 9 & -5 & 10 & 19 \\ -8 & 1 & 10 & 81 \\ 9 & 10 & 0 & -46 \\ 3 & 10 & 6 & 8 \end{bmatrix}.$$

Using the appropriate Matlab commands when required, determine the following:

(a) (2 marks) What should be the dimension of $\mathrm{im}(A)$? Determine an orthogonal basis for $\mathrm{im}(A)$ and then verify that your basis is indeed orthogonal.

(b) (2 marks) Based on the answer to part (a), and without doing any calculations or using Matlab, what would be the dimension of $\mathrm{null}(A^T)$? Determine an orthogonal basis for $\mathrm{null}(A^T)$ and then verify that your basis is indeed orthogonal.

(c) (3 marks) Take $x = (1, 2, -1, 0)^T$. Determine the coordinates of Ax with respect to the orthogonal basis you computed in part (a).

(d) (3 marks) The matrix B given below is a representation of a linear transformation T in terms of the standard basis vectors of $\mathbb{R}^4$,

$$B = \begin{bmatrix} 7 & -6 & -38 & 1 \\ -5 & -5 & -10 & -10 \\ 9 & 2 & -10 & 11 \\ -3 & -1 & 2 & -4 \end{bmatrix}$$

You will now switch to a basis which consists of the combination of an orthogonal basis of $\mathrm{im}(B)$ and an orthogonal basis of $\mathrm{null}(B^T)$. Compute the representation of the same transformation T in terms of the new basis vectors.

3.3 Laboratory task 3

(5 marks, to be completed and marked by 2 hrs 30 mins)

3.3.1 Simulink Coder

Simulink Coder^{TM} complements Simulink by providing automatic C code generation directly from Simulink models and Matlab functions. Simulink Coder supports the execution of dynamic system models on a wide range of computer platforms, including real-time hardware, to allow real-time simulation, rapid prototyping, and hardware-in-the-loop simulation of embedded real-time systems. Using Simulink Coder, one can quickly generate C or C++ code for discrete-time and hybrid systems, including systems containing triggered and enabled subsystems. You can also generate code for finite state machines modeled in Stateflow using Simulink Coder.

For more detailed description of Simulink Coder please go to:
https://au.mathworks.com/products/simulink-coder.html

One can see that Simulink Coder has a number of advanced features, but we will be using what is called "single user mode" where you can configure all code generation settings from a single user interface in Simulink model. Also note in terms of real-time control structure, the Host (the machine on which code is generated from Simulink models) and the Target (the machine on which the generated code runs and connected to the real plant) will be on the same computer.

Thus only the following features are used:

- Simulink Coder Simulink Development Environment (SDE) generates/builds code from MATLAB and Simulink models.
- Simulink Coder Windows Target runs the generated code from Matlab/Simulink models on a real-time Windows target (local).

Simulink Coder supports the built-in blocks for Simulink. The user can design and configure the model in the Simulink model environment using blocks from Simulink. If you click **Deploy to Hardware** to compile the real-time code and load to the target; If you click **Run**, it will rebuild the code and load the the real-time code to the real-time target and starts it; all icons are on Simulink model menu bar.

Important note: there are two **Simulation modes**, **Normal** and **External**. You will need to set it to **External** when running Simulink Coder in real-time mode.

3.3.2 Interface with a real plant

For this lab task you will need remote access to a Matlab instance running on a remote lab computer through your demonstrator.

Each computer in Lab 109 has intalled a National Instruments NI PCIe-6351 Multifunction Input/Output (I/O) card to connect a robot arm. Conventionally called DAQ (Data Acquisition Card) it does much more that just data acquisition. This device supports 16 bit, 16 analogue inputs, 4 analogue outputs, 24 bits digital inputs and 4 quadrature encoders through digital inputs. It connects the "the real world" through a Quanser

M&X series terminal board. The Quanser M&X series terminal board is a small black box connected by a thick cable on the bench. You could identify it via a Webcam view on the desktop.

3.3.3 Analogue input/output experiment

You are required to create a Simulink model to implement a simple analogue input/output experiment as show in Figure 1 to learn how to configure DAQ, as well as use Simulink functions to implement a real-time control.

Figure 1: Simulink model for Analogue I/O Experiment

(5 marks) Create a simple Simulink model to implement an I/O function. Set a 5 V signal for the analogue output port 0 and read it back from the analogue input port 1, both on the Quanser terminal board. Show the results on a scope, save to a file as a data array with time. Then plot the responses of voltage over time from the file saved, adding x, y labels and legends, on the Matlab command line (or using a Matlab script) using the Matlab command **plot**. Set the input and output gains to different values to see the effects. (Warning: the resulting output voltage should not exceed ± 10 V). Discuss with the demonstrator why there is a need for the input and output gains.

Here are some hints:

- To create a Simulink model: In Matlab window click **Home** $\rightarrow$ **New** $\rightarrow$ **Simulink Model**
- To insert Simulink function modules: In Simulink model window go to **Library Browser** and drag one module you intend to use to your Simulink model. Then click it to setup its parameters.
- For example to insert a analogue output module: **Library Browser** $\rightarrow$ **Simulink Desktop Real-Time** and find it to drag to your model. Then double click it for configuration: choose **National Instruments PCIe-6351 [auto]** in **Data acquisition** section, set 0.002 (e.g. 2ms) **Sample time** in **Timing** section, set Output channel number in **Input Output** section, and more importantly set both **Initial value and Final value** to 0 in the same section.

Similarly you can make the analogue input module.

- Other items needed for the program are all in **Sources** and **Sinks** in **Simulink library**. Actually it would be very beneficial if you use **Library Browser** to explore more function modules and find out how to use them.

- Note that there are input gains in the signal generator module. Now insert a gain into the point before the scope. These are the scaling factors. The issue of scaling is very important when dealing with a real plant for the units of physical measurements, such as speed, position, flow rate, weight, just to name a few. You need to always keep this in mind.

(End of lab 1)