COURSEWORK INSTRUCTIONS AND GUIDANCE FOR LAB WORK
Course Instructor: Prof. Tim Gordon
Module Code & Title: EGR2006 Control Systems
Contribution to Final Module Mark: 50%
Coursework Title: Modelling and Control of a DC Motor

Objectives
The objective is to develop your understanding of modelling and controller design and using a DC Motor Control Trainer (DCMCT). In particular you will:

- Practice first principles of modelling.
- Determine system parameters by experimental tests.
- Design a controller and test it using simulation
- Test control performance experimentally
- Build understanding of real-time control and how it relates to modelling and simulation.

Equipment – this will be used in the lab during the final part of the assignment.

The Quanser DC Motor Control Trainer (DCMCT) in this lab is used in conjunction with QICii software running on a PC. The setup was demonstrated in class and here is a short reminder. The layout of the DCMCT is shown in Figure 1. Power and USB connectors may already be in place; if not connect them to locations 15 (power) and 7 (USB) shown in the figure.

To run the DCMCT:

1. Switch on the power.
2. Press the white reset switch (11) - LED2 and LED3 will briefly light, then LED2 goes off and LED3 will flash (see next to User Switch, 12). If not, repeat this step, check cables etc.
3. Press the User Switch (next to the flashing LED) and then both LED's will be off.
4. In QICii, select Modelling (etc.) in the drop-down menu.
5. Connect to data source (see Figure 2). LED2 should light and the motor should start. In case of any problem go back to step 2
6. When changing parameter you must press RETURN, otherwise the parameter change will not be recognised.

More detail is available in [2]. You should **bring a notebook and a USB drive** to the lab. Data left on the local PC may be erased, and you will need to take notes as to which file relates to which test.

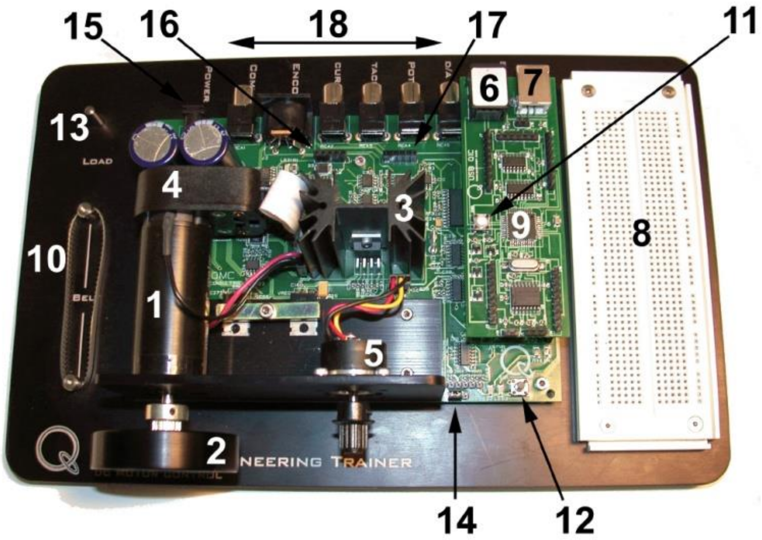

Figure 1. DCMCT General Layout.

ID #	Description
1	Maxon DC Motor
2	Removable Inertial Load
3	Linear Power Amplifier
4	High Resolution Optical Encoder
5	Ball Bearing Servo Potentiometer
6	RJ11 Port on QIC: for downloading firmware using a compatible programming device
7	USB Port on QIC: for online tuning and plotting (used by QICii)
8	Breadboard Option: to implement controllers with your own circuits
9	Embedded/Portable Option: QIC installs in this socket to perform embedded control in place of PC-based control
10	Removable Belt: to drive the potentiometer
11	PIC Reset Switch
12	User Switch: Momentary Action Pushbutton Switch For Manual Interaction
13	Inertial Load Storage Pin
14	Jumper J6: to switch between DAQ and QIC use
15	6-mm Power Jack
16	Power Supply Header: J4
17	Analog Signals Header: J11
18	i. PC Interface Option: this is implemented by using one of Quanser's HIL boards, NI's E-Series or M-Series boards, or the dSPACE DS1104 board ii. Analog Controller Option: to implement controllers using Quanser's Analog Plant Simulator

Table 1. DCCT Component Nomenclature for Figure 1.

PART 1. INTRODUCTION, BACKGROUND, ENGINEERING PRINCIPLES: 20%

In this part, include a review of the application of DC motors in consumer products or in industry. In what circumstances are DC motors preferred to other types? In this coursework the motor under consideration is a brushed DC motor. How does this differ from a brushless DC motor?

How is the motor modelled? Using your own words, summarise and explain the physical principls behind a brushed DC motor. You may refer to the DC motor modelling section of the University of Michigan Control Tutorials [3]:

https://ctms.engin.umich.edu/CTMS/index.php?example=MotorPosition&section=SystemModeling

A transfer function is given in this reference together with a set of parameters. It is suggested that you familiarize yourself with this transfer function, as it will be used (Model 1) in Part 2. There is no need to present any simulation results in this part of the coursework, but you may want to comment on the effect different physical parameters have on the open-loop step response.

PART 2. MODELLING AND SIMULATION: 20%

Use the transfer function from Part 1, together with the data below in Table 2 to create "Model 1" of the DC motor plant. Where parameters are missing you should estimate them (e.g., no viscous damping constant is given) – stating clearly what you have done.

Symbol	*Description*	*Value*	*Unit*
	Motor:		
k_m	Motor Torque Constant	0.0502	N.m/A
R_m	Motor Armature Resistance	10.6	Ù
L_m	Motor Armature Inductance	0.82	mH
	Motor Maximum Continuous Torque	0.035	N.m
	Motor Power Rating	18	W
J_m	Moment Of Inertia Of Motor Rotor	1.16E-006	kg.m^2
τ_m	Motor Mechanical Time Constant	0.005	s
M_l	Inertial Load Disc Mass	0.068	kg
r_l	Inertial Load Disc Radius	0.0248	m
	Linear Amplifier:		
V_{max}	Linear Amplifier Maximum Output Voltage	15	V
	Linear Amplifier Maximum Output Current	1.5	A
	Linear Amplifier Maximum Output Power	22	W
	Linear Amplifier Maximum Dissipated Power With Heat Sink, R_{load} = 4 Ù	8	W
	Linear Amplifier Gain	3	V/V

Table 2. DCMCT specifications.

Model 1 is for you to verify the general physical principles of a DC motor operation. Model 2 will be simpler and based directly on measurements of the motor and should be used in controller design.

Creating Model 2 takes more explanation, but it is very simple and based on recorded data. Open Matlab-Simulink and load the data from the file **LabData.mat**. It was saved from the test run of Figure 2, where there was no added load on the rotor. Saving in QICii captures the previous 10 seconds of data from when save is selected; the saved data was captured a little later than that shown in the figure, but the results are equivalent. Plot the data and check that it agrees with the plot shown. Note, here the model parameters were set poorly – the blue line (model) is slow to rise and has too large amplitude.

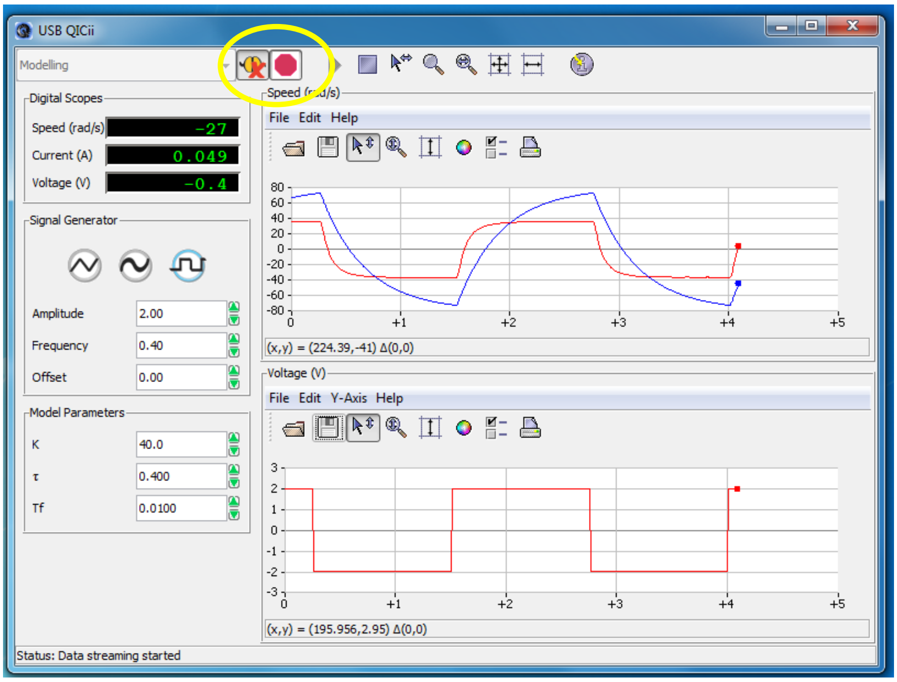

Figure 2. QICii software running in Modelling mode. The yellow highlight shows the buttons for connect/disconnect to data-source (left) and run/stop (right).

Plot the loaded data – your plot should look like Figure 3. [Note, any plots you put in your report should have axis labels and caption, just like this one!]

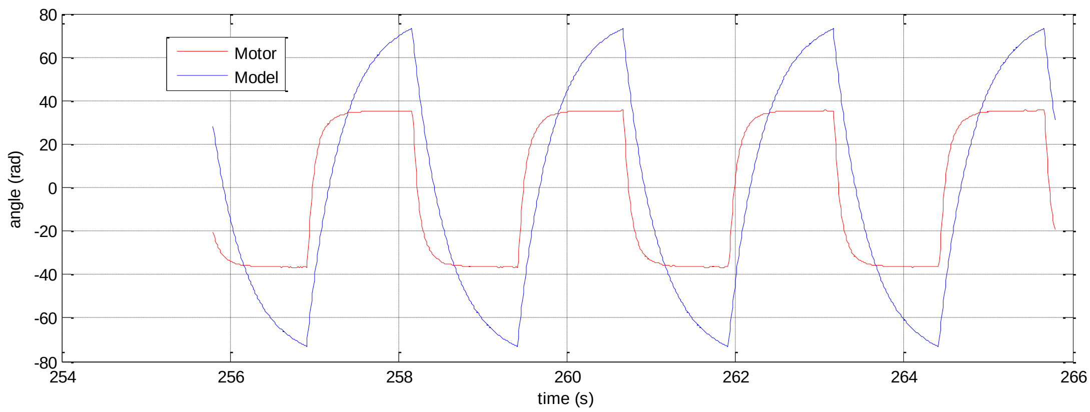

Figure 3. Matlab plot with axis labels and legend included.

Now create a Simulink version of the simple motor model using the signal generator and transfer function blocks. Figure 4 shows how this may look. The plant transfer function is $G(s) = \frac{K}{\tau s+1}$ and the parameters K and τ need to be adjusted.

Confirm the model gives the same or similar results as the loaded 'Model' data. *Show a plot comparing the two signals*.

Tune the parameters to give a good representation of the actual motor result. *Present your results and note the values of K and τ, as well as including a brief summary of your method.*

Note: It is a good idea to add 'To Workspace' blocks to help with plotting, rather than copy-paste the Scope output.

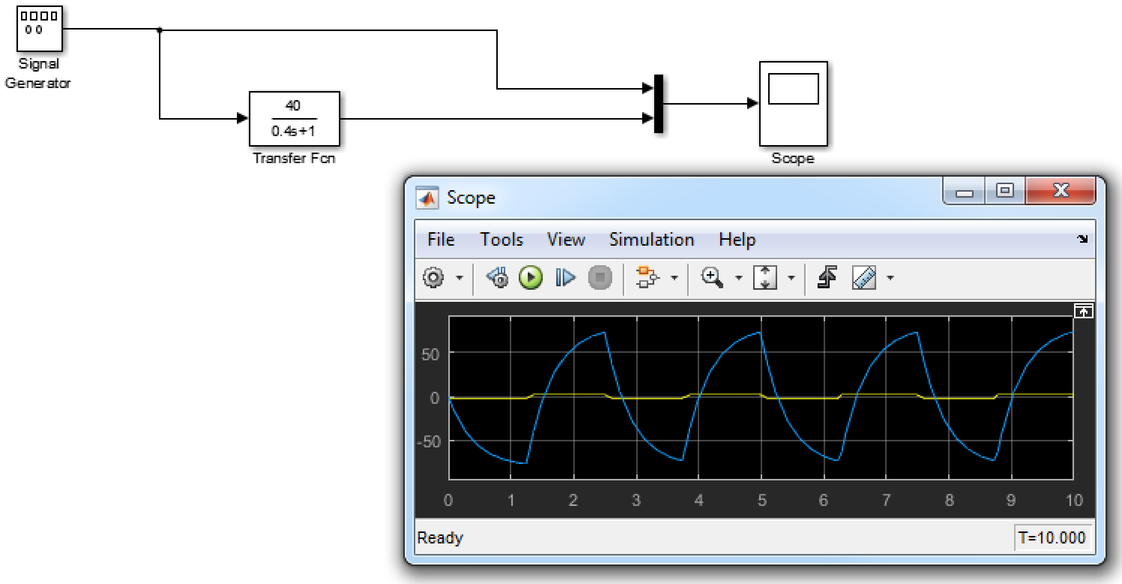

Figure 4. Simulink version of the model test of Figs. 2, 3.

Now add the transfer function of Model 1 to the Simulink diagram, so you can compare the two, using the same input voltage.

Compare angular velocity and angle responses. You will need to make some adjustments since Model 1 predicts angle and Model 2 predicts speed!

Also compare pole locations. Comment on similarities and differences. In any case you will use Model 2 for the remainder of the assignment, since we have confidence that it represents the actual plant dynamics.

PART 3. PID CONTROLLER DESIGN AND SIMULATION: 20%

Expand your Simulink model (Model 2) to include a PID controller for rotor angle, using the signal generator to create a square-wave reference signal, with amplitude = 1 radian and frequency 0.4 Hz. Selecting the PID gains $k_p = 1,\ k_d = 0, k_i = 0,$ you should obtain the result shown in Figure 6. To give consistent results it is recommended to set the Simulink solver to Runge-Kutta (ode4), fixed-step = 0.001 (use 'Model Configuration Parameters' option in the 'Simulation' menu).

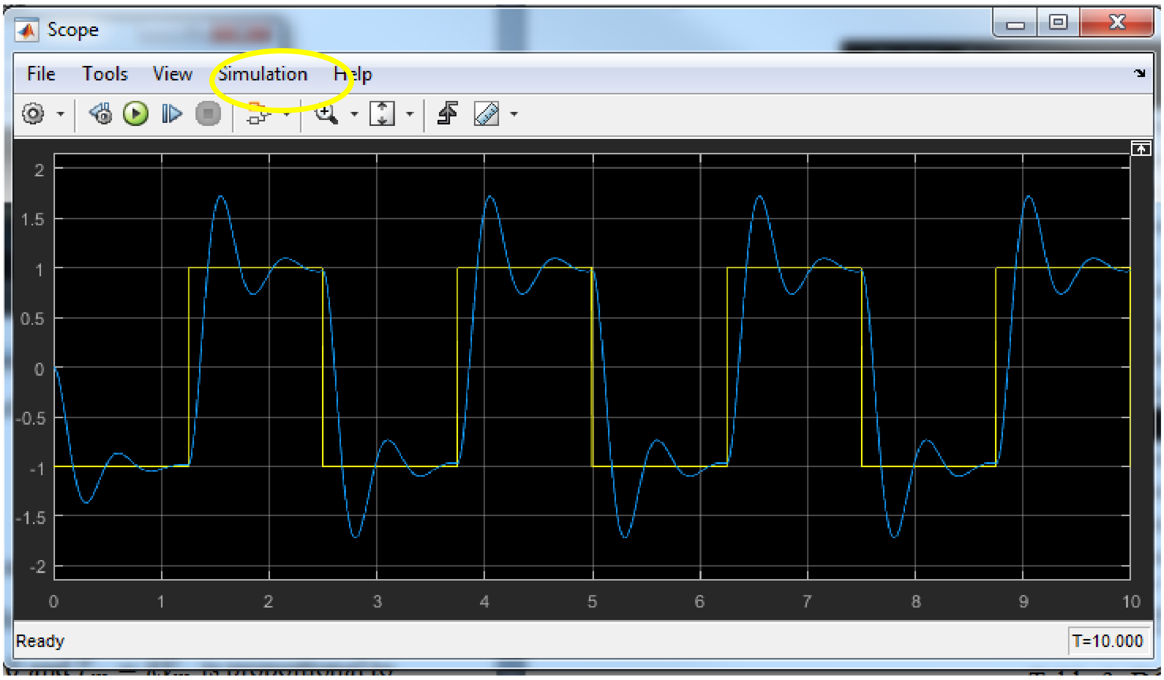

Figure 6. Scope screen-shot for the PID initial test.

Now manually tune the PID controller to improve tracking of the square-wave. Try to work systematically in your approach, tuning one gain at a time.

- *Describe the method you use to tune the three PID gains, and how each affects control performance.*
- *After you obtain your best result, compare with the initial result similar to Figure 6 and point out which performance metrics have improved and by how much.*
- *The actual motor has a limited input voltage of 15V. Use the PID Advanced tab to mimic this (set the upper and lower saturation limits to 15 and -15). How does this affect your results?*
- *Test your controller with another type of reference signal, e.g. a sine or sawtooth. What limitations are there, if any, when you operate the reference at (i) high frequency (5 Hz say) (ii) low frequency (0.2 Hz say)?*
- *Include plots and brief descriptions to explain your results.*

Note: there are many choices for how you carry out this part of the lab, there is no single 'correct answer'. Taking account of the report length, you may need to be selective about the plots you show.

PART 4. PID CONTROL EXPERIMENT (LAB TESTING): 20%

Here you (i) test the PID controller that was designed in simulation, (ii) see how real-world performance matches simulation for step changes in the reference, (iii) test robustness when weights are added to the rotor. Your exact approach is up to you. Testing may be done in groups, but results captured should be specific to the controller gains obtained by the individual student.

Note

The PID controller in the DCMCT takes the expanded form:

$$U(s) = k_p\left(b_{sp}R(s) - Y(s)\right) + k_{ds}s\left(b_{sd}R(s) - Y(s)\right) + k_i\frac{1}{s}(R(s) - Y(s))$$

where b_{sp} and b_{sd} are defined as 'reference weights' to include additional direct response to the reference signal. The term b_{sp} can potentially be used to offset any steady-state error, but where integral control is used this is not required, so please always select $b_{sp} = 1$. The term b_{sd} would also normally also be set to 1, but there may be an advantage of setting it to zero – test this out and try to explain any difference you see. You should perform step and sinusoid tests when looking at this part.

<u>**REFERENCES:**</u>

1. Ogata, K. "Modern Control Engineering".

2. USB_QICii_QET-DCMCT Student WorkBook.

3. University of Michigan Control Tutorials (online – use search engine).