ELE 301, Fall 2011
Laboratory No. 8

1 Background

The goal of this lab is to control the position or speed of a motor. In the process you will become familiar with Simulink and how to interface with the Quanser SRV-02 system.

Simulink is a graphical program primarily used to simulate systems. The Quanser system is designed to work with Simulink in order to control the motor.

1.1 Running the Quanser system with Simulink

For this lab you will need to sign in to the lab computers using the username `.\QUANSER` and password `feedback`. We will supply you with the Simulink models that came with the Quanser system, so you won't need to become an expert at Simulink, but you will want to poke around and get a basic understanding of what it is doing.

The 'models' that you will open and run will have filenames beginning with `q` and ending with `.mdl`. The files that have `q3` in the name are for a different motor system, and the filenames beginning with `s` are subsystems used in the main models.

To run a model you must first open it in Matlab (Simulink) and select the correct board type: `q8_usb`. This is done by double clicking QuaRC icon and making the change (this icon may be inside one of the orange blocks). Then, with the Quanser system powered on, you build the model (`CTRL->B`), connect to target (`CTRL->T`), and run.

2 Lab Procedures

2.1 Motor System

First take the time to get to know Simulink and the Quanser motor control by running each of the models contained in the Experiment 0 folder in the following order:

- `q_srv02_volt.mdl`
- `q_srv02_pot_raw.mdl`
- `q_srv02_pot.mdl`
- `q_srv02_tach.mdl`
- `q_srv02_enc.mdl`
- `q_srv02_pot_addl.mdl`

Please consider having a different partner open, compile, and run the different models so that you each have experience with the nuances of this. Take time to understand how each model works as well as how you drive the Quanser motor system and what sensor measurements are available to you.

If a TA is present, please show the TA when you have each of the models running and have them initial your lab report.

Questions:

1. Explain what you've learned about driving the system.

2. What sensor measurements do you have available? What are the differences between them?

2.2 Modeling the motor drive

You've learned that the voltage applied to the Quanser system causes a directly proportional rotational velocity in the motor. The input to the motor is a voltage v which results in an angular speed of rotation of the motor $\nu \propto v$. The files in Experiment 1 are to assist you in modeling the system a little more accurately.

In the instructor manual you will see details about the circuit and the motor drive system. Using some simple physics and circuit analysis it is shown that the system (where the voltage is the input and the angular velocity of the motor is the output) is a one-pole system of the form:

$$H(s) = \frac{K}{\tau s + 1} \tag{1}$$

What this means is that the angular velocity doesn't change immediately when a voltage is applied to the system. There is a reaction time caused by the pole of the system. The pole comes from the fact that both momentum and viscous friction are resisting the torque that the motor applies. The momentum term causes the pole. If we were being really careful, we would consider this to be a two-pole system because there is a small inductance in the circuit that has been ignored. That small inductance would cause a second pole at a much higher cutoff frequency. This can be safely ignored for our purposes.

The manual shows how to calculate the transfer function from the differential equations imposed by the physics and the known parameters of the system, including masses, radii, resistance, etc. From this, one can obtain the "nominal" parameters K and τ.

Experimentally, you can calculate the parameters quite easily by measuring the gain of the system and the -3dB cutoff frequency of the frequency response, or instead measuring the time constant from the unit step response. You don't have time to do such a careful modeling, but please run the 'setup' file and the 'model' to see how well the model matches the actual system. Try tweaking the parameters K and τ (using the Matlab command line) to see if you can make the model match the actual system better (try using both square waves and sine waves as input).

Questions:

1. Briefly describe your observations from this experiment.

2. How does the -3dB cutoff frequency relate to the pole and to τ?

3. What effect does K have on behavior of the system?

4. Plot the poles and zeros for this system.

5. What is the impulse response?

6. (Optional) Estimate the second pole of the system that was ignored in the modeling.

2.3 Speed Control

Now we will create a feedback control system so that our input voltage directly controls the speed of the motor. This may not sound very exciting since we already control the speed of the motor without any feedback. That's correct. But by creating a feedback loop in the common "tracking" or "servo" configuration, we can create a system that depends very little on the physical characteristics of the motor drive. In other words, k and τ which we measured in the previous experiment will have little effect on the behavior of the system once we've implemented our feedback. We'll cause the response time to improve and normalize the gain of the system to unity.

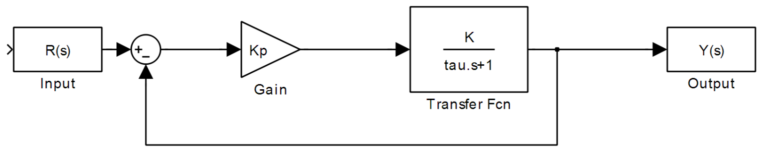

Figure 1: Speed Control

The model provided in Experiment 3 looks a bit confusing. They have added a lot of things that we don't care to use at this point. No worries though. Just leave the manual switch set to "PI Compensator." We won't care about the "set-point weight" (leave it set to 1). The PI Compensator is set up to provide two kinds of feedback, which have each been initially set to zero gain. We will not use the integral feedback. So the PI Compensator is actually very simple. It simply amplifies the difference between the two inputs. Adjust the gain `kp` (using the Matlab command line), and see what effect it has. Try both positive and negative gains.

Using proportional error feedback with gain K_p for the open loop system (1) (see Figure 1) results in the closed loop transfer function:

$$H(s) \quad = \quad \frac{K_p \cdot K}{\tau s + K_p K + 1}$$

Questions:

1. Briefly describe your observations from this experiment.
2. Plot the poles and zeros of the feedback system. Indicate how they depend on K_p.
3. Is there a problem with making K_p too large?
4. (Optional) Play around with the integral feedback gain `ki`.

2.4 Position Control

Suppose that instead of controlling the motor speed $\nu = \frac{\delta}{\delta t}\theta$ we would like to control the angular position of the motor shaft θ.

Since θ is the integral of ν, the open loop transfer function from input voltage v to output angle θ has two poles:

$$P(s) = \frac{K}{s(\tau s + 1)} \tag{2}$$

Let's add feedback like we did for speed control. For starters, we'll do the exact same thing as speed control, except that we'll use the position sensor readings in the feedback loop rather than the speed sensor readings. Use the model provided in Experiment 2. The setup file will set the gains to zero. This model is set up to use three types of feedback, but we won't use the integral feedback, only the proportional and derivative ("velocity") feedback.

To begin with, let's only use proportional feedback. What is the closed loop transfer function that results by using proportional control with this open loop system? Experiment with different values of `kp`. You'll notice that at some point increases in `kp` will not lead to quicker response time but will have another effect instead.

We can give ourselves more degrees of freedom for designing a good tracking system by including not only proportional feedback but also the derivative of the feedback, each with a separate gain. Notice that this just means we are using the velocity of the motor in our feedback loop as well as the position. The transfer function for this system will be:

$$H(s) = \frac{K_p \cdot K}{\tau s^2 + (1 + K_d K)s + K_p K}$$

Now try using both proportional and derivative feedback by adjust both `kp` and `kv`. (Start with small values of `kv` compared to `kp`)

Questions:

1. What is the closed loop transfer function that results by using proportional control?
2. Show how the pole and zero locations change as the gain is adjusted.
3. What does the derivative feedback (velocity) do to the location of the poles?
4. Describe your observations from conducting this experiment. Which gain values worked well, etc.?