ELE 301, Fall 2011
Laboratory No. 9

1 Background

The purpose of this lab is to use feedback to stabilize and control a rotary inverted pendulum. The modern example of an inverted pendulum is a segway. A simpler example is trying to balance a pen vertically on the tip of your finger. For today's lab, we will be dealing with a more restricted case. The pen will only be allowed to rotate in a plane perpendicular to your finger and in order to control the system, you must hold your hand in the same place and can only move your finger right and left. The measurements taken from the system are: the angle that the pendulum makes with the vertical axis, called α, and the angle that the controlling arm makes with a reference point, called θ. See Figure 1.

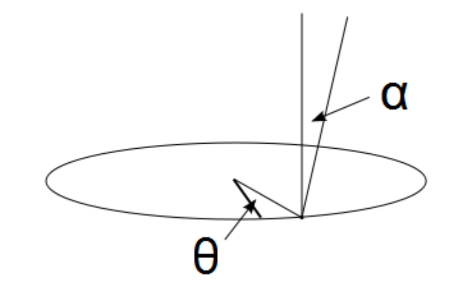

Figure 1: An Inverted Pendulum.

2 Lab Procedures

2.1 Files and Models

Among the lab files provided, you have the instructor's manual from Quanser, a setup file that assigns a handful of variables, and two Simulink models of interest:

- `q_sip.mdl`
- `q_sesip.mdl`

We will only use `q_sip.mdl`, but you are welcome to play with the other model when you've completed the balancing procedure. The difference between the two models is simply that `q_sesip.mdl` includes an ability to swing the pendulum up and then balance it. The model we'll use, `q_sip.mdl`, requires that you manually lift the pendulum until it senses that it is vertical. Since the block diagram appears a bit less cluttered and confusing, it's a good place to start.

2.2 Understanding the Configuration

Start by taking a good look at the model provided. Notice that there is a control block that drives the motor. Take a look at the "high-gain observer" block and see what its purpose is.

The "balance control" block is very simple, although it may not look so at first glance. It has two purposes. One is to detect when the pendulum has reached the vertical position, and the other is to balance the pendulum.

Question: *If you wanted to create a system that swung the pendulum up from rest, how would you incorporate that into this Simulink model? What is the model currently designed to do before it detects that the pendulum is vertical?*

Notice that the green block which is supposed to represent the physical system (motor and sensors) has a feedback path that goes directly to it, not passing through the control block. In general this is not a very clean and organized way to design a feedback control system. We'd do ourselves a favor by having all feedback go through our control block. However, the reason this is done in this case is to simplify the physics of the system. If you open up the block and take a look at what's happening, you'll see that positive feedback is being applied. Additional voltage is being sent to the motor to cancel the effect of the friction and back-emf of the motor, so that the system acts almost frictionless if no signal is intentionally applied.

If you build and run the system (remember to run the setup file and change the 'board type' to `q8_usb`), but don't lift the pendulum up, you will notice that there seems to be less friction on the motor as you slide the arm back and forth. Compare this to when the system is off. If you'd like to adjust the feedback gain, you can do so by adjusting either `Kg` or `km`. I found that increasing the gain (for example, I changed `Kg` to 85 instead of 70) made if feel even more frictionless. If you increase the gain too much, it will accelerate in an unstable manner. Feel free to change this gain and leave it that way for the rest of the lab. It will make the results match the theory a little better.

Recall that viscous friction and back-emf in the motor play a first-order role (first derivative of position) while the moment of inertia plays a second-order role (second derivative of position) in the differential equations that govern how the motor moves. Together, this gave us a one pole system that we investigated in the previous lab. In this lab we have a much higher moment of inertia, and it would be convenient to think of the motor drive signal as controlling the torque on the motor rather than the speed. This would be consistent with your homework assignment, where the input to the inverted pendulum was the acceleration of the cart.

Task: *Draw a simplified block diagram of the system that shows the feedback going directly to the green block (ignoring the "balance control" block). Use the one-pole transfer function from the previous lab to represent the system:*

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

What is the closed-loop transfer function for this feedback system? If the gain on the feedback is calibrated just right, can the system be made into an integrator?

From now on we will assume that the input to the system drives the torque of the motor and our main focus will be on the control block.

Task: *Draw a simple block diagram that shows the workings of the control block (ignoring the "mode-switching" component). You can treat the physical system as a single block. A torque signal goes in, and two signals come out, θ and α.*

2.3 Stabilize the Pendulum

The best method for stabilizing this system is two use a state-space model, which is beyond the scope of this course, even though it's not terribly difficult to understand. The reason to use a state-space model is because there are two types of sensor measurement that we're dealing with (this can be thought of as a vector). We have only dealt with scalar signals. So we will design this system in two steps, using the theory that we've learned in this class, and end up with the same result.

The first step is that we will stabilize the pendulum without using the feedback from the position sensor, θ. Therefore, in this step we cannot attempt to move the stable pendulum to a particular position—it will simply drift around.

We also don't care right now to have an input to the system. We just want it to balance the pendulum. So go ahead and set the amplitude of the signal generator to 0.

According to the analysis in your homework, proportional feedback (α) should not be enough to stabilize the system. It should only be enough to cause the pendulum to oscillate. Go ahead and try applying proportional feedback and see what happens. You may find that a little bit of friction serves to your advantage (if you didn't increase `Kg` to cancel the friction).

Task: *Apply proportional feedback and observe the result. Increment the gain in steps of size .2 and see at which point you start observing oscillations.*

In your homework you also learned that adding derivative feedback can allow you to stabilize this system. If we knew the pole locations, we could use them to design good coefficients for the feedback. Instead, we will just try some values to create a stable system.

Task: *Apply proportional and derivative feedback and observe the result. You can start from a good value for the proportional feedback, from the previous task, but as you increase the derivative feedback from zero it can be useful to increase the proportional feedback as well.*

Now you have a system that stabilizes the pendulum. There is no input to this system. Let's change that. Why don't we design the pendulum to balance the pendulum at a non-zero angle with respect to vertical? If our signal is positive it will lean one way, and if it is negative it will lean the other. (Obviously, if it is leaning then it will be accelerating as well because it is unbalanced.)

Task: *Adjust the model so that it is in the tracking configuration similar to the position tracking system from Lab 8. Draw a block diagram of this. Also, use one of the scopes to compare the input signal to the actual angle α.*

2.4 Controlling the Position

Now we will do a little trick to allow us to control the position (θ) of the pendulum. We will treat the entire system we've built as a black-box. It has an input (which we designed to control the angle α), but now we won't worry about α anymore. We'll just assume that it's balanced. Now we'll consider the output of the system to be θ.

This system behavior is complex and depends on the design of the balancing controller and the physics of the system. If the black-box functioned perfectly as we designed it to (meaning that the angle α perfectly tracked the input signal), then the input would directly influence the acceleration of the position θ. Thus, the transfer function from the input of the black-box to θ would be similar to

$$H(s) = \frac{K}{s^2}.$$

Of course this is an oversimplification, and you will see that the using this assumption can easily lead to instability. But we can also get lucky and stabilize the system by designing according to this assumption, if we don't get too aggressive with our feedback parameters.

Task: *Adjust the Simulink model to track the position θ specified by an input signal (while at the same time balancing the pendulum), first using only proportional feedback and then adding derivative feedback. Start by drawing a block diagram that includes a black-box with an input signal and θ as the output. Then plug in the diagram for the inside of the black-box (note that the actual system has two sensor outputs). This block diagram will have a number of gain components that are cascaded. It can be simplified to look like the block diagram of the Simulink model, which means you can implement this entire system without making any major changes to the Simulink model aside from adjusting gain parameters. One more thing to be warned of is that the position sensor (θ) reads in the opposite direction than you would like, so you will need to make the gains negative for the position feedback.*