Lab 6: Closed Loop System Identification

Experimental Objectives

Completion of the laboratory exercise will require you to:

- Take some time to get to know the XY stage, its feedback and controller software.
- Identify the open-loop transfer function $\frac{K_{sys}}{s(\tau_{sys}s+1)}$ for the X and Y axes individually by first closing the loop with a proportional control and identifying the closed loop transfer function. Then knowing Kp, ξ, and ω_n of the closed loop system solve for K_{sys} and τ_{sys}.
- Verify the identified transfer functions by creating a Bode plot of the transfer functions in Matlab and then choosing two frequencies to verify the gain and phase with the actual system.

Experimental Procedure

1. First, experiment with the X and Y axes of the stage and figure out which direction is positive. In Matlab open the Simulink file XYstage_start.slx located in the N:\Hydraulicslab\ME460 directory. Like you have for all previous labs, save this file with a new name to your folder located on the C:\ drive. Then make sure to change Matlab's current folder to the folder where you just saved the model. Looking at the XYstage_start.slx block diagram you see a manual switch. Throughout this lab we are going to use those switches to switch between closed loop control of the stage and open command. For the Y axis, you can move it by hand pretty easily when the motor's amplifier is disabled so you may not need the open loop mode very often. The X axis is very difficult to move by hand so the open loop mode will help you move the stage back to its middle starting point.

   Build your Simulink/Real-Time Windows Target model file by simply pressing the play button. This generates C code for your model file and then compiles this code into a real-time controller running at the Windows device driver level. Once your controller is running, experiment by driving the axes back and forth with an open loop value of 0.7 and -0.7. Take some time to document for yourself the positive and negative directions of each axis. The X axis/direction is the top stage and travels right/left. The Y axis/direction is the bottom stage and travels forward/back. Probably the best thing to do is just drive one axis at a time keeping the other input at zero. As you are experimenting here also change the labels of the gain blocks and scope blocks in your Simulink file indicating which blocks are for the X axis and which are for the Y axis. Note that the output of the gain blocks attached to Enc0 and Enc1 is in millimeters. You will want to always start each axis in the middle of its travel to ensure enough distance to travel in either the positive or negative direction. Each time you restart the controller (stop then run again), the encoder values reset to zero wherever the current positions of the axes are. When collecting data for all the steps in the remainder of this lab assignment, you will want each axis to be zeroed in the middle position.

2. Again using just one axis at a time, identify the open loop transfer function for each stage. The transfer function has the form of $\frac{K_{sys}}{s(\tau_{sys}s+1)}$. This of course is an approximation for the model of the system because for example static friction, which is nonlinear, is not taken into account in this model. But this model will allow you in future labs to design controllers for each stage that are close to the given specification. Then once the controller is implemented you will be able to tweak the controller gains to achieve the exact design specifications.

   As discussed in the prelab, if we apply an open-loop input to this system it simply drives until it hits one of its end-stops. For that reason a nice way to identify these types of systems is to close the loop with a simple proportional controller and generate a step response of the closed loop system. Set Kp, identify the parameters of the closed loop transfer function, then solve for the open-loop transfer function parameters.

   Working with one axis at a time in your Simulink model, implement a simple unity feedback proportional controller that controls the position of the stage. Use a signal generator block to generate a square wave with .25 Hz frequency. Make the amplitude of this square wave 5mm so that our step response steps between -5mm and 5mm. Make sure to modify your scope parameters to show 3 signals (position, reference, and control) and set the time span to 5 sec. Start with a Kp gain of 0.25 and try not to go higher than Kp = 0.5 to avoid the maximum speed limits of the motors. Build and run your Simulink controller. While pressing the enable switch, watch the step response in the scope blocks. Adjust the Kp gain until the step response has two distinct peaks in order to apply the logarithmic decrement formula as in the prelab.

   As you saw in the prelab, the closed loop transfer function has the form of the standard second order system $\frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$. Using one step response from the data saved, calculate ξ and ω_n using logarithmic decrement and the measured period of oscillation ω_d. Once you have these parameters use the equations you found in the prelab to solve for K_{sys} and τ_{sys}.

3. Perform two verifications to see if the open loop transfer function that you just identified makes sense.
   i. Create a Simulink simulation of the system. Use unity feedback with the same proportional control and the identified open loop transfer function. How does the simulated step response compare to the actual closed loop step response? Keep in mind that static friction is not taking into account in the simulation.

   ii. Plot the Bode plot of the identified open loop transfer function. Find the frequency where the gain is equal to 1 or 0dB. Perform a verification of this frequency using the actual system. In your Simulink real-time controller, change the signal generator to output a sine wave with same amplitude of 5mm. Also change the frequency of the sine wave to be the 0dB frequency. Note that the Bode plot frequencies are in radians/second so change the signal generator parameters appropriately.

   For this verification we would like to compare the gain and phase shift found when exciting the system at the 0dB frequency. But what is the input and output we are interested in? Normally we would think of the sine wave coming out of the signal generator as the input and the output being the position of the stage. Well the position of the stage is the output of our open loop transfer function but what is the input? If you look at your simulation you just created you will see that the input to the open loop transfer function is the control effort or the output of the Kp controller. So plot in the same scope block (using a Mux) the control effort (output of the Kp gain after satuation) and the position output in millimeters.

Build and run this verification. Beware this run will probably shake the whole XY stage structure quite a bit, but go ahead and run it for 10 seconds or so and collect enough data to make gain and phase shift measurements. How does the gain found compare to the gain of 1 (0dB) found in the Bode plot? How well did the phase shift match? This is of course just one frequency of the Bode plot but if the values are close we know at least we are on the right track. Of course we could have had you create a full bode plot by varying the sine wave's input frequency, but this analysis would shake the XY stage terribly and cause unneeded wear and tear on the system.

4. Repeat the above steps to identify and verify the open loop transfer function for the Y axis.

Lab Report

1. Include calculations, Bode plots of data, and specific points on those plots used to identified the transfer functions for both the X axis and Y axis.
2. Using the verification steps given, how well do the transfer functions model the X and Y axes?

** *Last Updated: Apr 2018, R2017a*