Lab 6: Closed-Loop System Identification

Objectives

- Become familiar with the XY Stage, its sensor feedback, and the controller software.
- Identify the open-loop transfer function $\frac{K_{sys}}{s(\tau_{sys}s+1)}$ for both the X and Y axis by closing the loop with a proportional control and estimating the closed-loop transfer function. Parameters k_p, ξ, and ω_n from the closed-loop step response are used to determine K_{sys} and τ_{sys} of the open-loop transfer function.
- Verify the estimated transfer functions by generating a Bode plot of the plant in MATLAB. Then choose one or two frequencies to implement on the actual system and compare gain and phase values.

Procedure

Open-Loop Control:

1. Test out the X and Y axes and determine which direction in positive from the encoder values. Copy the file 'N:\HydraulicsLab\ME460\Lab6_XYstage.slx' to your directory: 'C:\ME460_SPxx\ABx\Lab6'. Run MATLAB and set your workspace to that directory.
2. Open the .slx file to start Simulink. The default block diagram has a switch that chooses between open-loop control and closed-loop control. You will find this useful throughout the experiments. For instance, you will want to center the X and Y axes in the middle of their tracks when zeroing the encoder values. It is possible to move the Y axis by hand, but the X axis is very difficult to move by hand so you will want to use the open-loop control to center the X axis.
3. *Build* your Simulink model. Wait for the code generation to complete and click *OK*. Then *Connect to Target*.

   a. You will need to click *Run* after each time you *Build* to enable the control outputs on the B&R X20 Controller.

   b. Each time you *Build*, the encoder values are reset to zero. Additionally, if your model is already built, you can simply press *Halt* and then *Run* again to reset the encoder values. Try this out.
4. Change to the open-loop control for one of the axes in your model. In additional to pressing *Run*, you will need to hold down the handheld enable switch for the electric motor drives. Experiment driving the axes back and forth with an open-loop command value of 0.7 and -0.7. It will be best to drive one axis at a time and keep the other axis' input zero. The X axis is the top track and travels left to right. The Y axis is the bottom track and travels front to back. It would be helpful to label your scopes and inputs accordingly instead of '1' and '2'.
5. The output of the gain conversion blocks are measured in millimeters. When collecting data for the remainder of the lab, be sure to start each axis in the middle of the track with a corresponding encoder reading of zero millimeters.

Closed-Loop Time Domain Identification:

1. Identify the open-loop transfer function for each axis one at a time. The transfer function has the form $\frac{K_{sys}}{s(\tau_{sys}s+1)}$. This model is only an approximation of the real system. For instance, static friction (which is nonlinear) is not considered in this model. This model will allow you to design feedback controllers in

future labs that can achieve performance close to desired specifications. When implemented on the real system, the controllers can be tweaked further to achieve the exact specifications.

2. As mentioned in the Prelab, an open-loop input to the system simply drives the stage until it hits the end of the track. Alternatively, closing the loop with a simply proportional control will generate a more familiar step response with identifiable parameters. Set k_p, identify the parameters of the closed-loop step response, estimate the closed-loop transfer function, then finally determine the open-loop transfer function.

3. Construct the closed-loop unity feedback control law in your Simulink model for both axes. Use a signal generator block with a 5 mm amplitude square wave at a frequency of 0.25 Hz as the reference. Add the reference as a second signals to the encoder's scope. *Build*, *Run,* and *Connect* your Simulink model. One axis at a time using the manual switches, start with a small $k_p = 0.15$, and slowly increase it just until you see the closed-loop step response achieve two distinct peaks. $k_p = 0.5$ is too large as it is maxing out the speed of the electric motors.

4. The data from the encoder scopes is saved to the MATLAB workspace by default. After *Disconnecting from Target*, plot the data in MATLAB. Use the data point cursors to find the time and value of the first two peaks in your step response. Copy the file 'N:\HydraulicsLab\ME460\figureCursorCallback.m' to your Lab 6 workspace. For more precision, right-click on the data point and choose 'Select cursor callback'. Select the .m file you just added to your current directory. This will now give you three decimals of precision in your measurements for time and value. Apply the logarithmic decrement formula as in the Prelab.

5. The closed-loop transfer function has the form of a standard second order system $\frac{\omega_n^2}{s^2+2\xi\omega_n+\omega_n^2}$. Find ξ and ω_d from the logarithmic decrement formulas and then find ω_n. Once these values are known, you can use your solution to the Prelab question to solve for K_{sys} and τ_{sys} of the open-loop transfer function.

6. Repeat the steps above for the other axis.

Verify the Estimated Transfer Functions:

i. Create a Simulink simulation of the estimated plant transfer function. Apply the same unity feedback closed-loop controller with proportional gain. Observe the simulated response and compare with the response on the actual system. Keep in mind the static friction is not considered within the simulation.

ii. Generate a Bode plot of your identified plant transfer function. Locate the point on the magnitude plot where the gain is equal to 1 (0 dB). Record the input frequency and phase at this point. You will perform a verification of this frequency on the actual system. Go back to your Simulink model with the B&R X20 Controller. Change the signal generator reference to a sine wave at the recorded frequency. Leave the magnitude at 5 mm. Note the units for frequency.

For this verification, we would like to compare the gain and phase shift found when exciting the system at the 0 dB frequency. What is the input and output we are interested in? Normally, we would think the sine wave coming out of the signal generator as the input. Looking at your Simulink model you will see the input to your open-loop transfer function (i.e. the plant) is the signal coming out of your k_p gain block. This is the input signal we are interested in for measuring the gain and phase shift of the actual system. The output is the encoder reading in millimeters after the conversion block. Be sure to plot the input sinusoid (signal coming from the k_p gain block) and the output (encoder value in millimeters) on the same scope block and log the data to the MATLAB workspace.

Build, *Connect to Target*, and *Run*, your Simulink model. Beware this make shave the entire XY Stage structure so only run the controller for about ten seconds. Measure the gain and phase of the actual system for this input frequency. Plot the data in MATLAB and use the trick above to increase the precision of the data point cursor. How well did these parameters match with the original Bode plot? This is just one frequency, so we could input several more frequencies to obtain an experimental Bode plot, but this would cause unnecessary wear on the system.

iii. Repeat the steps above for the other axis.

Report Questions

1. How did the closed-loop step response simulation compare to the step response of the actual system?
2. How did the frequency domain verification of the actual system compare with the generated Bode plot?