Modeling and System Identification for a DC Servo

Nicanor Quijano and Kevin M. Passino
The Ohio State University,
Department of Electrical Engineering,
2015 Neil Avenue, Columbus Ohio, 43210, USA

March 29, 2002

Abstract

In this lab, you are going to perform simple system identification for the motor that is the main element for the Quanser SRV-02ET series of experiments in the lab. For that, you are going to follow two different methods: the time domain method, and the frequency domain, and at the end, you are going to compare the results of both approaches.

Contents

1 Introduction

In the pre-lab, you studied some basic ideas in time and frequency domain based system identification. Now, we are going to apply those methods in the lab. For that, we are going to use the DC Servo motor SRV02ET that is provided by Quanser. The motor has a transfer function that we found in the Appendix A of the pre-lab. Here we will see how accurate the transfer function is. To acquire the data, we are going to use the dSPACE software; therefore we are going to apply all the knowledge that you acquired in the first laboratory, and in the pre-lab 2.

2 Laboratory Procedures

2.1 Necessary Equipment

- 1 dSPACE software.
- 1 DS1104 interface card.
- 1 Universal Power Module, UPM-2405.
- 1 SRV 02ET servo DC motor.
- 1 Analog sensor cable.
- 1 To load Cable.
- 1 To A/D Cable.
- 1 From D/A Cable.
- 1 Inertia.
- 3 72 teeth gears (low gear ratio).

2.2 Connections

We are going to connect the motor using the tachometer as it was described in the document "Interfacing dSPACE to the Quanser rotary series of experiments (SRV02-ET)." For that, we are going to follow the following steps:

1. Take the "Analog sensor cable" and connect one side to the S3 (in the Universal Power Module), and the other side in the tachometer.
2. Take the "To Load Cable" and connect one side to the Universal Power Module, and the other in the motor.
3. Take the "From D/A" cable and connect the RCA termination to the analog output 0 in the DS1104 Interface Board (remember that this analog output is # 1 in the software). The other connector goes to the UPM-2405.
4. Take the "To A/D cable" and connect the red RCA connector to the analog input 2 in the DS1104 card (but remember that this one belongs to the analog input # 3). The other connector goes to the UPM-2405.
5. Remember that you have to put the motor in the low gear ratio.

2.3 Procedure

Now we are ready to start the laboratory. We are going to divide the laboratory into two parts: first, we are going to do system identification in the time domain, and after that, in the frequency domain.

2.3.1 Time Domain Identification

1. Open dSPACE and Matlab. Remember that you have to change the path where you are working (you will work under C:\EE758\LAB2\).
2. Build a Simulink model that has a step input of amplitude 2, and "step time" of 4. This is going to be the input of our system, and should go to two points:
   - A transfer function block that is going to show the theorical transfer function
   $$G(s) = \frac{1}{0.0026s^2 + 0.1081} \tag{1}$$
   - The analog output that is going to go directly to the motor (do not forget the scale factor due to dSPACE).

   The analog input (channel # 2 in the card, # 3 in the software), goes directly to a gain of the tachometer to make the units consistent. The first gain should be the gain that converts from volts to rpm, whose value is 1000/1.5. Then you have to convert to degrees using a gain of 360/60, and finally you have to take into account that you are working with a 14:1 gear ratio, so you have to put in a gain of 1/14. Remember that besides these gains, you will need a gain of 10 due to the dSPACE card.
3. This model should have a block named "Safety Stop Time." The safety block is NOT a Simulink block, so you have to think how to build it. The general idea of this block is to take the simulation time and compare it to some value (that is going to be your variable "end simulation" when you mask the sub-system), and when the clock arrives to this value, stop the simulation (Hint: See the subsystems clock, rational operator, absolute value, constant, and stop simulation, and how to mask a subsystem in Simulink). After you build the block, mask it, and allow the user to enter the "end simulation" value. How did you mask the subsystem? Explain.
4. Define your sampling time (we suggest 1 ms), fix the Stop time of the system at "inf" and the end simulation time of the block that you built before at 8 seconds.
5. Save the model as model1.mdl in the same directory.
6. Compile the model, and start a new experiment in dSAPCE called LAB2A (it should be stored in the same directory as above, i.e. C:\EE758\LAB2\). The experiment must have a plotter that shows in the same plot the input and output of the model, and the real output. The x-axis of the plotter should be fixed at 8 seconds (to be consistent with the stop time simulation constant). Besides, you have to acquire these signals. Check the boxes that you will find under Settings/Acquisition, as was explained in the tutorial.
7. Change the decimation time to 10, and the length to 8.
8. When the simulation stops (always under the animation mode in dSPACE), you will acquire the data. If the window that was described in the document "A Tutorial Introduction to Control Systems Development and Implementation with dSPACE " to acquire data doesn't appear, press # 1. Under this window you will find the Settings button. Press it, and under "Capture Variables" select all of them. Then, press Save, and save the data as step1.mat.
9. Repeat all of these steps two or three times, so that you will have more data.
10. The data that you just acquired characterizes the transfer function
   $$\frac{\Omega}{V} = \frac{K_{tach}K_{motor}}{\tau s + 1} \tag{2}$$
   The problem is that we want to characterize just the transfer function
   $$\frac{\Omega}{V} = \frac{K_{motor}}{\tau s + 1} \tag{3}$$

Therefore, we have to find one way to find the value of K_{tach}, and then we may find the transfer function of the motor. The following steps are the ones that allow you to find this constant[1].

11. Align the screw of the center gear at zero degrees.
12. In your model set the step time at 8.
13. Change the end simulation time in your Safety Stop Time Block, and set it at 9 seconds.
14. Compile and run the experiment.
15. Reduce the end simulation time until the motor moves only one revolution.
16. Record the stop time, and save it as t_{stop}.
17. Assume that the signal reaches the steady state in this small amount of time, and then compute the mean of the signal as v_{ss}.

2.3.2 Frequency Domain Identification

1. Build a model similar to the one that you built before, but instead of having a step input, you should have a sine wave.
2. Fix the sampling time in 1 ms, and try a sine wave with amplitude equal to 1, and a frequency $w = 4$ *rad/sec*. The end simulation time should be fixed at 20 seconds.
3. Save the model as model2.mdl in the same directory.
4. Compile the model, and start a new experiment in dSAPCE called LAB2B (it should be stored in the same directory as above, i.e. C:\EE758\LAB2\). The experiment must have a plotter that shows in the same plot the input and output of the model, and the real output. The x-axis of the plotter should be fixed at 20 seconds (consistent with the stop time simulation constant). Besides, you have to acquire these signals. Check the boxes that you will find under Settings/Acquisition, as was explained in the tutorial.
5. Save the data as bode4.mat under the same directory, as it was described before.
6. Now, you are going to repeat this procedure with frequencies $w = 40$ *rad/sec*, and $w = 400$ *rad/sec*. For the last frequency, change the sampling time to 0.1 ms.
7. Save the data using the same procedure as above, but now name the data bode40.mat and bode400.mat respectively.
8. Take data for three or four more different frequencies. Save them.

3 Post-Laboratory Exercises

1. Explain briefly how you built the subsystem, and how you masked it.
2. How would you change the name of the inputs/outputs pins?
3. Take the data that you stored in the step*.mat files, and give the transfer function from the data by estimating the transfer function parameters from the responses. For that you may the command *getfield* in Matlab useful to acquire the data in a specific field. One example of that is the following:

```
load step1
m = step1;
yy = getfield(m.Y(1),'Data');
```

---

[1] This procedure due to Prof. John Watkins, Systems Engineering Department, U.S. Naval Academy

What are the values of τ and the gain K if you have a transfer function of the form

$$\frac{\Omega}{V} = \frac{K}{\tau s + 1} \tag{4}$$

4. With the values t_{stop} and v_{ss} compute the value of K_{tach}. First, compute the average velocity during one revolution w_0 as

   $$w_0 = \frac{2\pi}{t_{stop}} \tag{5}$$

   Now, compute the K_{tach} using the previous value and v_{ss} as

   $$K_{tach} = \frac{v_{ss}}{w_0} \tag{6}$$

5. Find the value of K_{motor} in Equation(2), using the values of K and K_{tach} that you just found.

6. Take the data that you stored in the frequency domain procedure, and plot the data on the same plot where you plot the Bode plot (both magnitude and phase) for the model that was derived using physics. On the same plot, sketch the Bode plot of the transfer function that you found in the previous item. Compare, discuss.