Lab #2 Servo Motor Modeling

Objectives

The objectives of this experiment are:

- To derive a linear mathematical model of the servo motor system
- To create a SIMULINK model that simulates the dynamics of a servo motor
- To carry out tests on our actual servo motor and then use the results to guide us in making nonlinear adjustments to our SIMULINK model of the motor.
- To yield a nonlinear SIMULINK model that adequately represents its dynamics of the actual motor and can be used for controller design and simulation.

Completion of a pre-lab[1], checked by the lab instructor, will be required at the beginning of this Lab #2. See instructions contained in Lab #1.

Background

All real life systems display some non-linear behaviour. In some cases, nonlinearities do not significantly affect the system behaviour within an operating range, and the linear time-invariant model can be safely assumed. In other cases, nonlinearities cannot be ignored. Nonlinearities in the Servo Module are associated with:

- Resolution error caused by quantization
- Saturation of the controller output
- Deadband in gears
- static friction (stiction).

The first two are negligible, but the latter two affect the system response in a significant manner, and will be discussed here.

Effect of Saturation on the Servo Module Response

Saturation is caused by the limited range of the D/A converter in the DSP board. In our case, our analog output is limited to +/- 10 volts.

[1] Pre-lab is worth 25% of the total mark for this lab.

In Simulink, saturation can be modeled by a nonlinear **saturation block**. The block and its parameters are shown in Figure 7.   This will be used to limit the voltage signal coming from the controller.

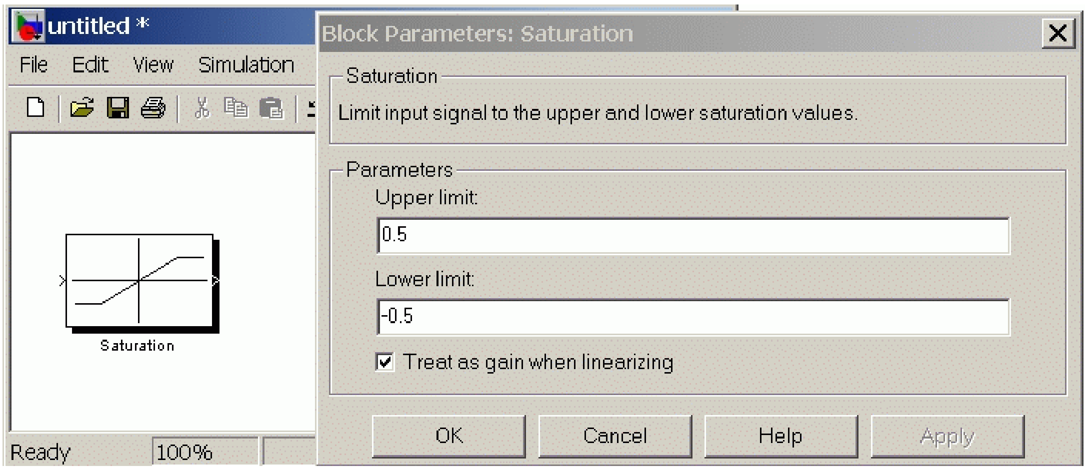


Figure 7: Saturation Block and its Parameters.

Effect of Stiction and Gearlash on the Servo Module Response

Effects of stiction on the servo motor response become visible for low values of the input signal. I.e., if a signal amplitude is very small, it may not be enough to overcome static friction effects so that the system output doesn't move; it 'sticks'.

Gearlash occurs when the input signal switches polarity, where any loose coupling in the gears will result in a delayed response until the gears re-engage.

Deadband is the maximum amount that the input signal can change, including direction reversal, that produces no change in output, essentially a combination of gearlash and stiction.

These combined effects are perhaps best illustrated by Figure 8, which shows a response of the real motor and the ideal motor (no stiction) to a small magnitude slowly varying triangle wave applied on the armature voltage, positive and negative.   Notice that the real motor stops, sticks for a bit, and experiences a bit of gearlash, before beginning to move in the opposite direction.

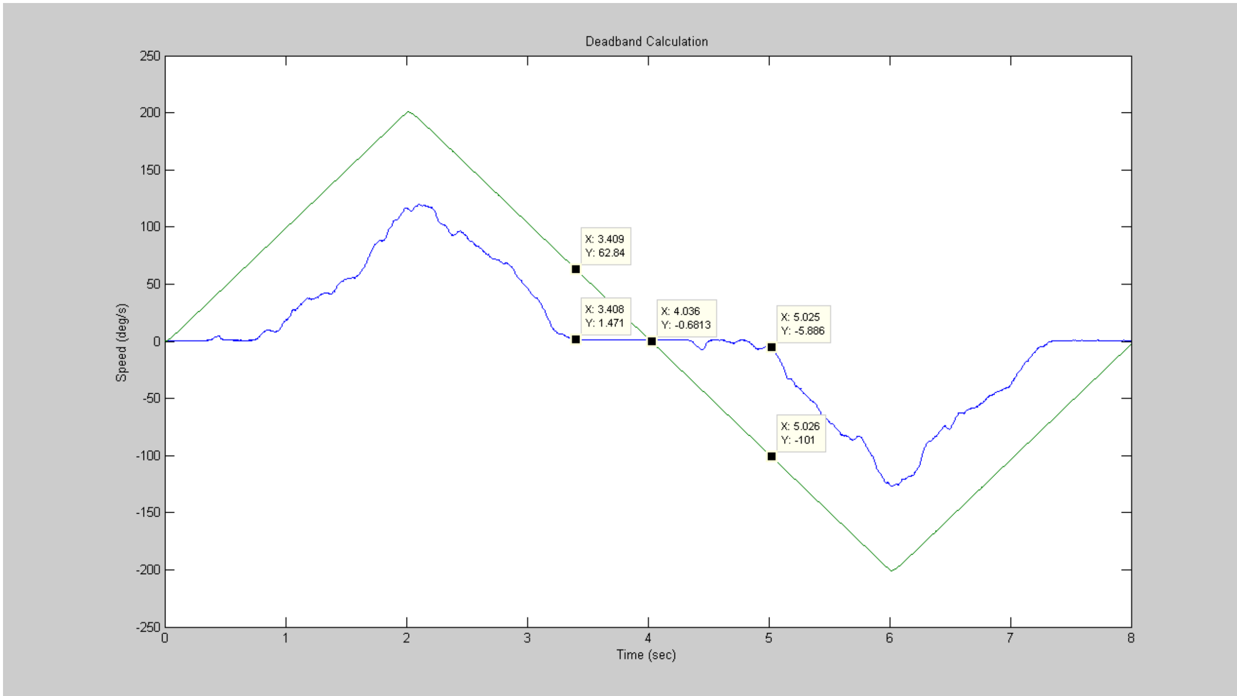


Figure 8: Illustration of gearlash and stiction in the servo motor

In Simulink, these effects can be approximated by including a nonlinear (and non-symmetrical) **deadzone block**. The block is shown in Figure 9, and is parameterized by a measure of the deadband.

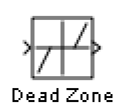


Figure 9: Deadband

To calculate the thresholds of the deadband, consider again the diagram of Figure 8. If we measure the time elapsed from the point at which we enter the deadband (i.e., stick, and go no lower) until the point at which we exit the deadband (i.e., we start to decrease, and carry on decreasing), then the total time elapsed is 5.025-3.408 = 1.617 seconds. During this elapsed time, the linear model speed has dropped 62.84+101= 162.84 deg/second. Dividing the deadband equally between the forward/reverse motion, we end up with 162.84/2 = 81.42 deg.. To back-compute to armature voltage, we need to compute

$$db = 81.42 \times \pi/180 \times K_g/K_m$$
$$= 81.42 \times \pi/180 \times 14/123.8 = 0.1638 volts$$

The deadband threshold is therefore calculated as d_1 = -0.1638 volts, d_2 = 0.1638 volts. For this calculation, the resulting graph will be

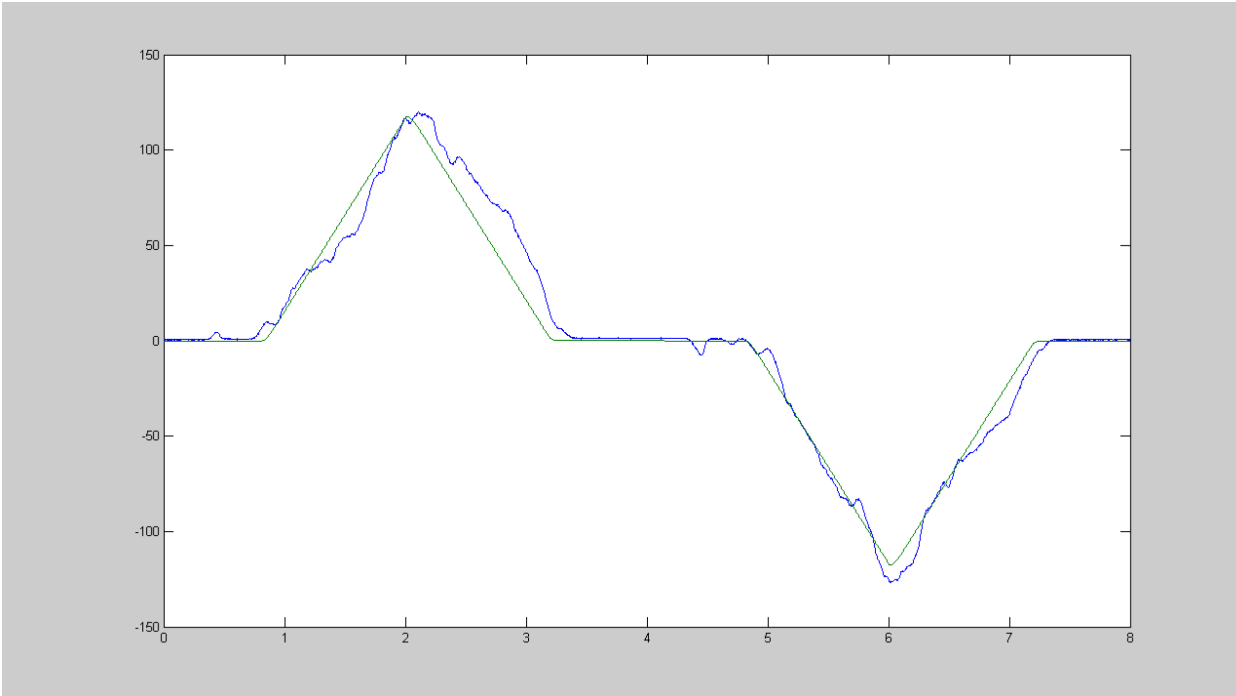


Figure 10: Illustration of theoretically adjusted gearlash and stiction in the servo motor, modeled as deadband

A bit of trial-and-error adjustment should end up with the correct split between d_1 and d_2 Plot the angle to obtain some additional insight into how to make appropriate adjustments:

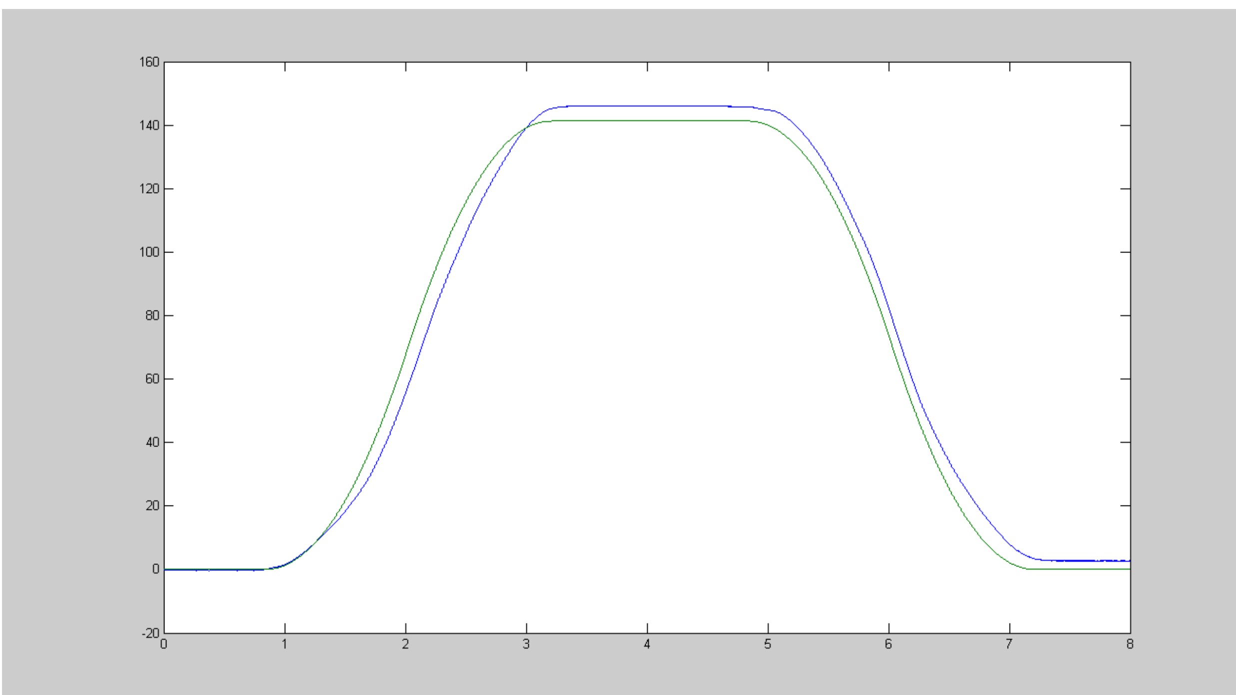


Figure 11: Illustration of effect of the theoretically adjusted deadband in the servo motor

After some minor adjustments, for the example shown, d_1 = -0.162 volts, d_2 = 0.16 volts to produce the results shown in Figure 12. If need be, very minor adjustment to the motor gain K_m may be made, but no more than 1%!

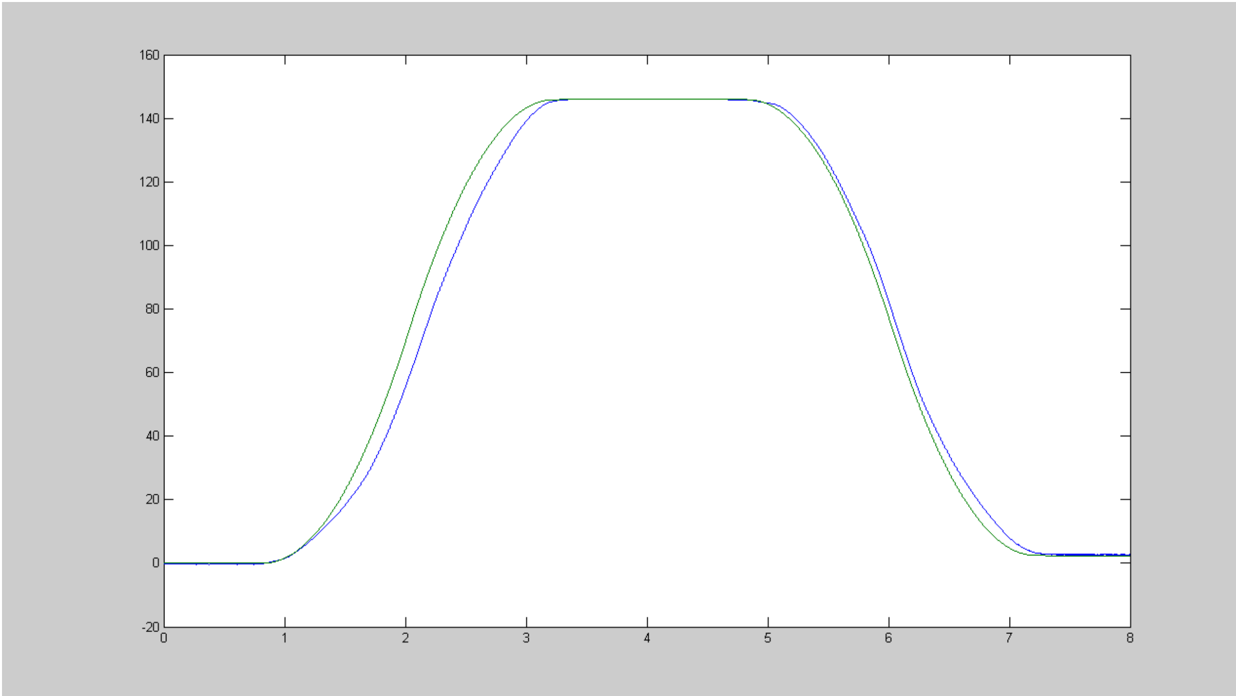


Figure 12: Illustration of effect of the final adjusted deadband in the servo motor

Procedure

Now you are ready to start building the simulation model for the servomotor system. Note that you will be using variables for parameters in this model, and that these variables have to be declared first. To do that, create an m-file, **lab_servo.m**, in which all the values of parameters will be listed, and always run this file at the MATLAB command line prior to executing the simulation. In this m-file, put in all of your tabulated pre-lab data stored as parameters.

The screen captures below show a basic servo positioning model. Create a SIMULINK model reflecting this system. As you set up the different blocks, you will also have to add some additional parameters to **lab_servo.m**.

- Amplifier and armature voltage: In the data file parametrize the amplifier gain as $K_a = 1$, the deadband of the **Deadzone** block as zero, the limits of the **saturation** block as $L_{\min} = -10, L_{\max} = 10$. The block 'Motor1' is a Simulink subsystem, i.e., a collapse of what is shown in the following Figure 11.

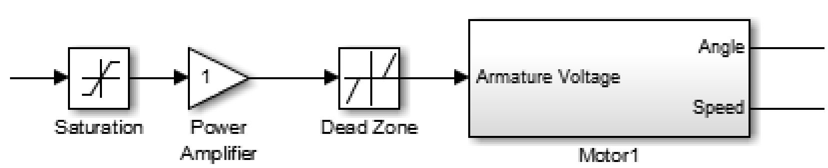


Figure 10: Servo Motor Model for Simulink Lab Experiments

Motor subsystem: In the data file declare parameters N, D, and n to be your pre-lab values for **first order**[2] motor numerator, denominator, and total gear ratio, and calibration gain 2 as $180/\pi$. In the Integrator block, there is a field labelled 'Absolute Tolerance'. Set it to 10e-6.

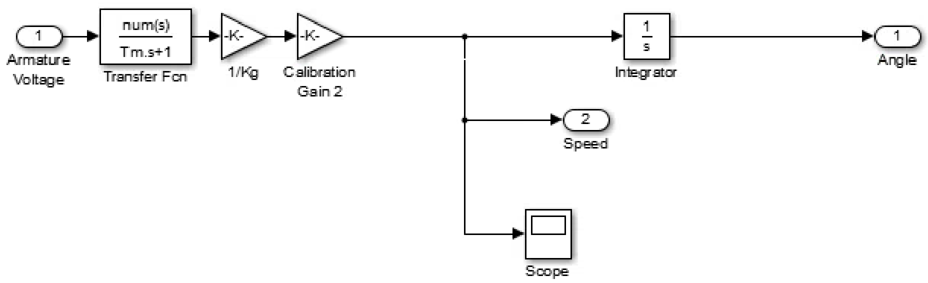


Figure 11: Motor Subsystem in the Servo Model

Now modify the model so that it can accommodate a variety of testing inputs, as shown in Figure 12.

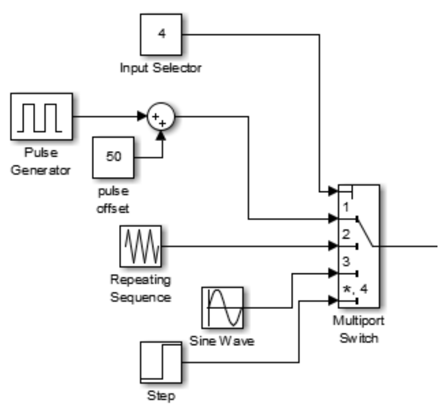


Figure 12: Inputs for the Simulation

- Choose a fixed-time step solver for the integration routine, i.e., from the simulation -> configuration parameters menu, and select a fixed step size of 0.002 seconds. Select the ode3 (Euler) method of integration.

[2] The dynamics of the 2nd order are too fast for our fixed step integration.

- Run the simulation for a square wave with amplitude of +/-0.5 volts[3]. Adjust the period of the square wave for a good view of the output angle (i.e, try not to let it go past +/- 180 degrees). Observe the plot of the armature voltage signal, and the resulting motor speed and position.

Data Acquisition

Since the objective of this part of the experiment is to build a realistic model of the servo, the model responses have to be verified against real responses of the servo, and the model parameters have to be adjusted, if necessary, to obtain a "good match" with the measurements. In this segment of the experiment you will collect data from the Servo, which will allow you to adjust parameters of your model.

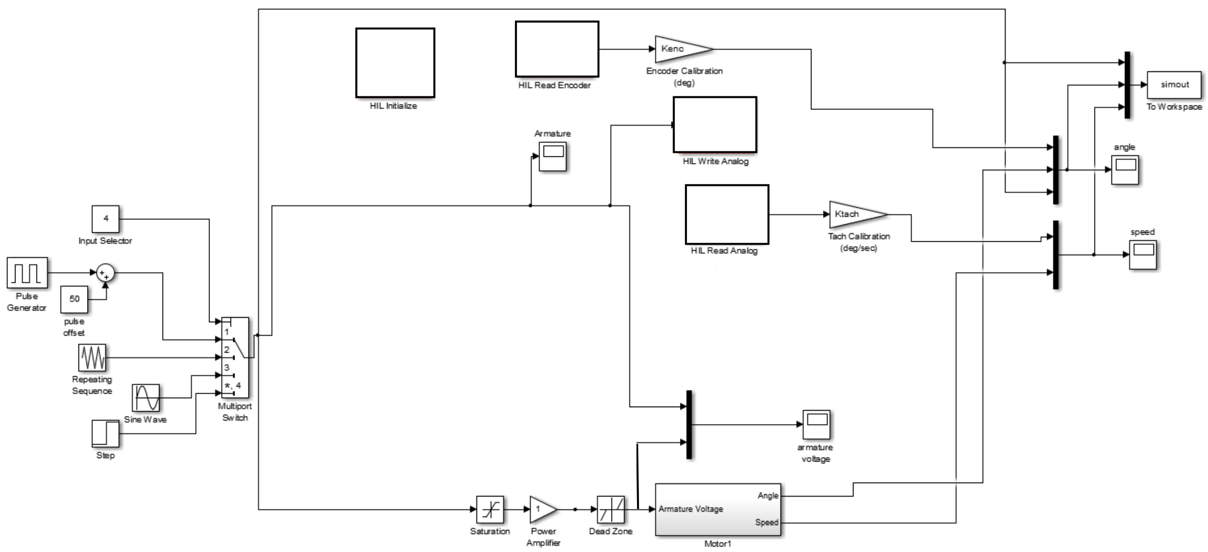


Figure 13: Simulink Model and connection to Servo Motor

Expand your Simulink model as shown in Figure 13, and make physical connections to the DAQ, servo motor and interface as done in Lab #1.

NOTE 1: The 'to workspace' block needs to be configured to save 'structure with time'. For each of the tests performed, the data should be saved to a file, i.e.,

>> save *filename* simout

Such files can be loaded into MATLAB workspace by typing: load *filename* at the MATLAB command line. You can then later plot data from MATLAB. More importantly, the data can also be loaded back into the simulation at a later time so that you can do parameter adjustments.

Configure the input as described below, run a test for an appropriate amount of time, and capture the data output for the specified input. This input/output data will be used to compare with your model of the system

[3] Some motors will be stickier than others. You may need a larger armature voltage. The idea is to have a low enough voltage to emphasize the effects of stiction.

Time Response (Pulse) Measurements

This set of measurements will help you directly in adjusting the system model parameters, so that the "good match" model can be found. The following time response measurements will be needed:

- Use the selector to switch to take measurements for a ramp input with small slope, starting in one direction, and the coming back in the other direction, positive and negative voltage. **(I.e, make a slow moving triangle wave, with max/min voltage of that used previously for the low voltage square wave volts, so that 2-3 cycles are captured over an elapsed time of 10 seconds, and a load position of about +/- 50 deg is maintained .)** Store the results in a .mat file.
- Take measurements for a square wave input with period selected to keep the load positioned to within +/- 50 deg.. Adjust the period of the square wave to capture 2 to 3 cycles over 10 seconds. Store the results in a .mat file. These responses should not saturate the controller output and will be closest to an ideal LTI representation of the system.
- If time permits, try a variety of other signals, making sure to always start from the zero reference point, to not saturate the output, and save the results for comparison later on

Using the saved data files

1. At this point, you can continue to stay connected to the actual motor, or you can modify your SIMULINK model of the Servo so that it can read from your saved data files. I.e.,

   >> testinput = load (‘*filename’)*

   should load your saved .mat file into the workspace variable *testinput*. Modify your manual ‘multiport’ switch to be able to select data from testinput. Now you can work on adjustment of your model parameters without having to work with the actual motor.

2. The appropriate SIMULINK model configuration is shown in Figure 14. In order to be able to switch to using your saved data, you will have to set your Simulation Mode to *‘normal’* mode instead of *‘external’* mode. Test one of your data files to make sure this is working. You are now able to use your Simulink model as a decent approximation of the system dynamics for the purpose of controller design and simulation.

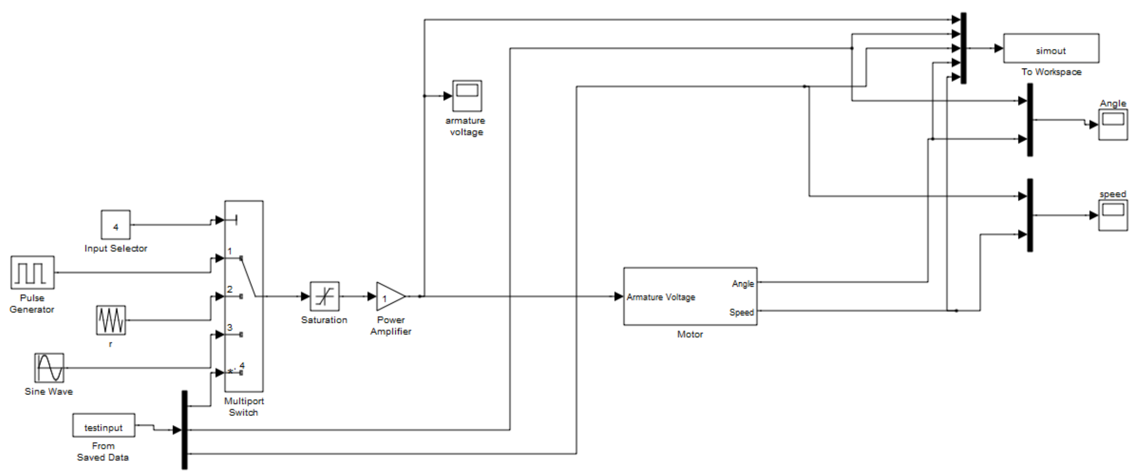
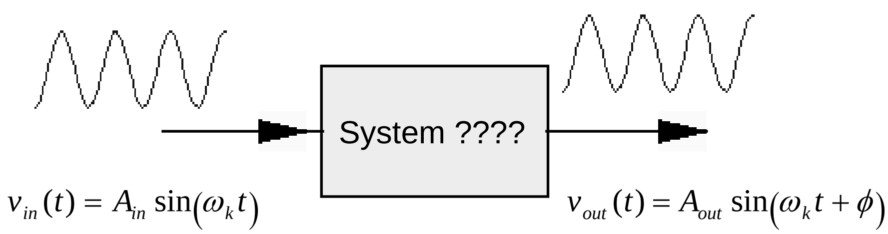


Figure 14 Simulink Configuration for Model vs. Data Comparisons.

Frequency Response (Sinewave) Measurements (optional – for future lab)

This set of measurements is not needed to complete this lab and to write the lab report. However, you will need them to complete a future controller design lab using a frequency domain based approach. If all goes well, analysis done in that experiment will be consistent with the motor model estimate you're working on in this experiment, based on the parameter adjustments you will make for time pulse/step responses.

Since access to the servomotor is limited to lab hours, you should perform these measurements now and store them for future use. This set of measurements will allow you to obtain an open-loop frequency response of the servo. For these measurements, a sinusoidal signal of a specific frequency ω_k (in rad/sec) is input into the tested system: $v_{in}(t) = A_{in}\sin(\omega_k t)$. A record of a sinusoidal steady state of the resulting output signal is taken: $v_{out}(t) = A_{out}\sin(\omega_k t + \phi)$.



The frequency response $G(j\omega)$ of a system described by open loop transfer function G(s) can be obtained by substituting $s = j\omega$. $G(j\omega)$ is a complex-valued function of a complex variable and can be represented in polar coordinates as:

$$G(j\omega) = M(\omega)\cdot e^{j\Phi(\omega)}$$

Scalar functions of magnitude $M(\omega)$ and phase $\Phi(\omega)$, respectively, can be calculated for each required frequency spectrum point ω_k as follows:

$$M(\omega_k) = \frac{A_{out}}{A_{in}} \qquad \Phi(\omega_k) = -\phi$$

In order to obtain a reasonably smooth frequency response curve at least 10-15 frequency measurements should be taken. The frequency range appropriate for the servomotor is approximately 1 to 200 rad/sec. You will need to try to spread your measurement points evenly over this range. To take the measurements do the following:

- Choose a sinusoidal input option for the armature voltage, scale it to +/- 10 volts, and set the frequency of the input to be within the recommended range.
- Take a record of your real servo sinusoidal tachometer response, store it in a data file for processing later. In a future experiment, you will be required to calculate the magnitude ratio and the phase shift for each of the measurements taken, and plot the resulting frequency response.
- Repeat the procedure for a reasonable number of frequency points over the specified range.

The sinusoidal analysis of the system response is very illuminating, and leads to good estimates of the system structure and model, especially in cases where little is known about the physics of the system, or in the so-called "black-box" system modeling, where only I/O data is available. Modern analysis methods yield the direct frequency response through analysis of a power spectrum of the Input-Output signals, or through a Fast Fourier Transform (FFT) of these signals. Thus, the whole tedious procedure can be reduced to taking a single shot of the system response, and numerical processing of the resulting information, provided that an input of a sufficient frequency content is used.

NOTE: you should take care in obtaining and storing your data from these measurements, since the procedure is quite tedious. Should your data sets be corrupted, lost, or otherwise inappropriate, you may have to redo the whole procedure in the lab at some later date.

Parameter Adjustment

1. Load the first data file, the slow moving triangle wave. Compare the simulated response with the real load speed response. This will allow you to estimate the deadband, as described previously, for the Simulink 'deadzone' block. Adjust this parameter accordingly in your Simulink model.

2. Load the low amplitude square wave data file. Compare and adjust, if required, the gain and time constant of the motor. (This will be very minimal adjustments, if any. Speak with your instructor otherwise**.** The final adjusted model for the DC motor of the servo will be a single transfer function $G_m(s)$, with a parametrized deadzone block, which best matches all collected data.

Results

Please refer to the Lab 2 detailed marking rubric.