# EEE 342 – Feedback Control Systems

## Lab #2 Assignment

The goal of the second lab is to perform system identification studies on a physical DC motor in frequency domain. You will perform a frequency domain system identification by using sinusoidal signals on hardware.

**NOTE:** Please save all the result **data and figures** you obtained during the lab, since you will not be able to repeat the experiments and there will be too many figures to deal with. Put all the figures and results to your lab report including necessary explanations.

**Q.1. Use the following code sample to generate the bode plot of the estimated transfer function from the first lab. Please do not forget to update A and B parameters of the transfer function, G, in the following code sample.**

```
w = logspace(-1,2,100);
for k = 1:100
        s = 1i * w(k);
        G(k) = A / (s+B);
end
subplot(2,1,1)
semilogx(w,20*log10(abs(G)));
grid on
subplot(2,1,2)
semilogx(w,angle(G)*180/pi)
grid on
```

**Q.2. Obtain the bode plot by applying sinusoidal inputs and compare with theoretical computations.**

Step 1. Download lab2_sinusoidal_input.slx and lab2_velocity_read.slx files to your working directory. ***Please don't forget to repeat hardware initialization and connection steps from the first lab by doing necessary modifications in order to configure files for hardware experiments.***

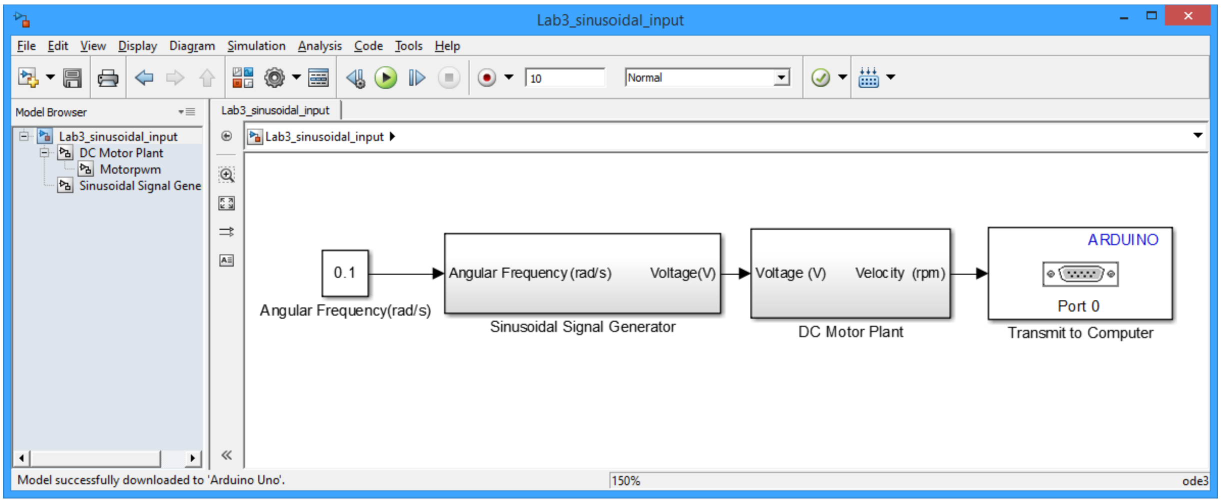
- lab2_sinusoidal_input.slx file can be used to give sinusoidal inputs to the DC motor plant. This file requires an input angular frequency of the desired sinusoidal.

- lab2_velocity_read.slx file can be used to read the velocity output of the DC motor. Please DO NOT use the lab1_velocity_read.slx file in this part.

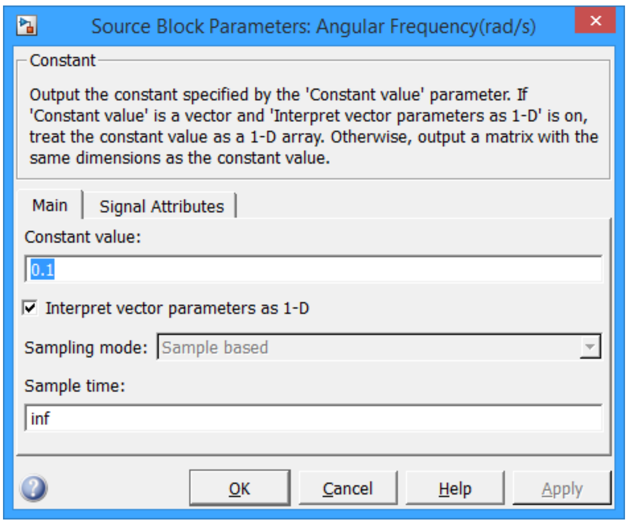
**SYSTEM ID PROCEDURE:** Repeat the following Step 2 to Step 7 for each of the angular frequency inputs in the following table for the corresponding time durations.

| Angular Frequency | Simulation Duration (s) |
|---|---|
| **0.1** | 70 |
| **0.2** | 70 |
| **0.3** | 70 |
| **0.6** | 70 |
| **1** | 25 |
| **2** | 25 |
| **3** | 25 |
| **6** | 25 |
| **10** | 10 |
| **20** | 10 |
| **30** | 10 |
| **60** | 10 |
| **100** | 10 |

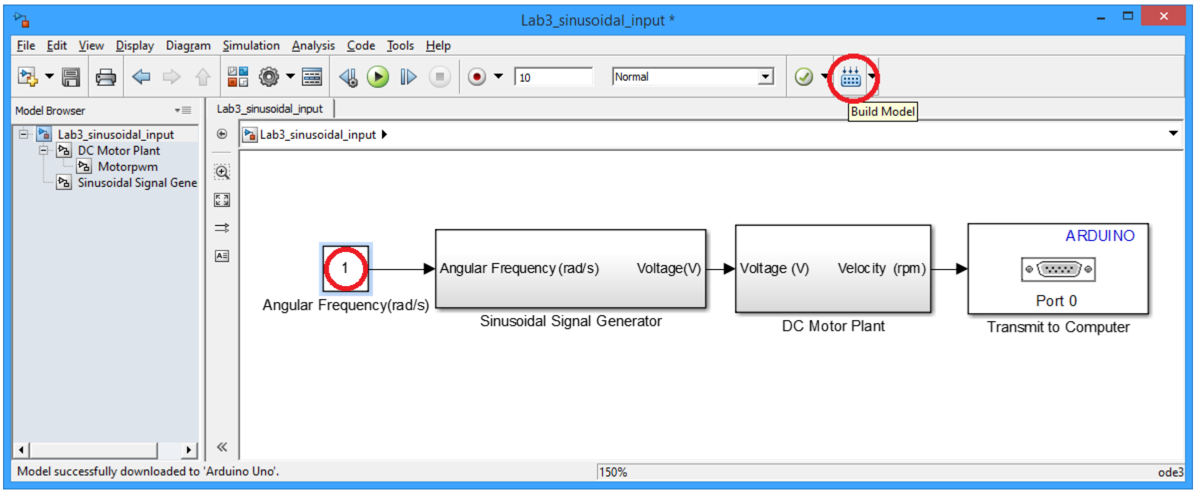
Step 2. Open lab3_sinusoidal_input.slx and open Angular Frequency input block



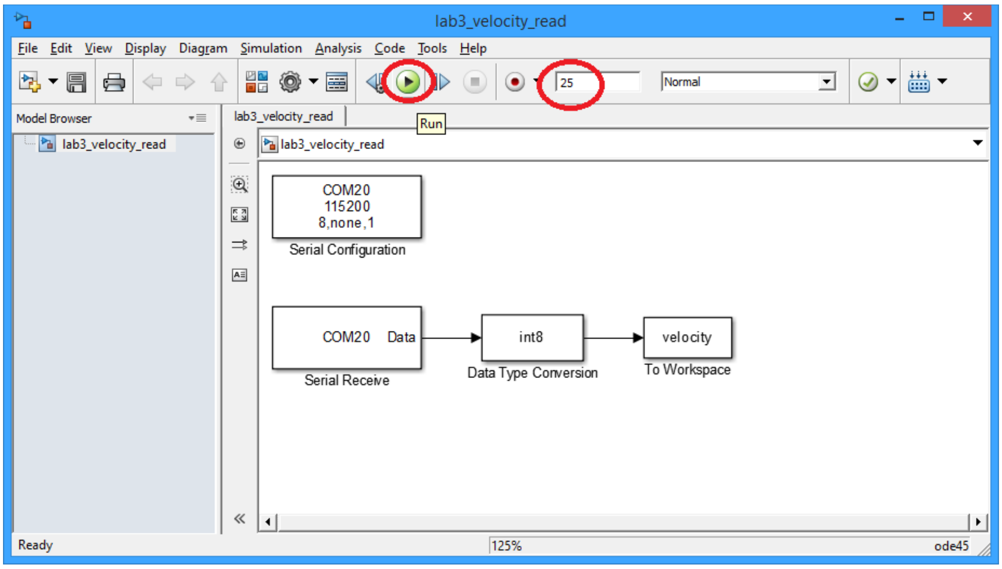

Step 3. Set the angular frequency to your desired angular frequency value from the table



Step 4. Build lab3_sinusoidal_input.slx file to embed the software into Arduino. Note that Angular frequency block shows the given angular frequency value.

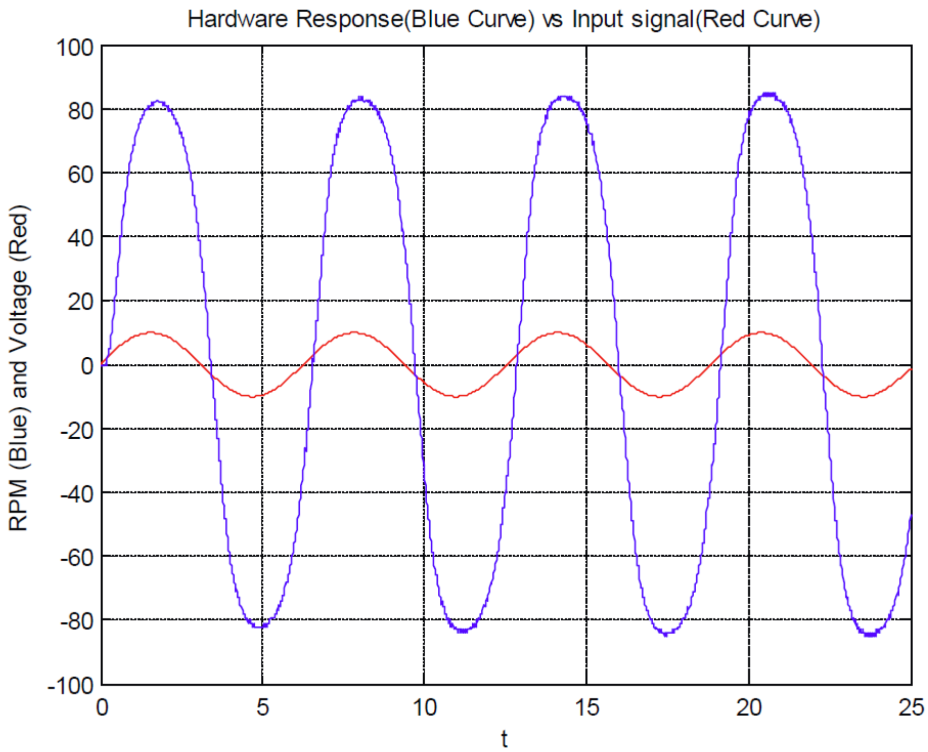



Step 5. Open lab3_velocity_read.slx file and set the communication ports as you did in Q.1. Step 14. Then set the simulation time duration corresponding to your given angular frequency in the table. Then press run button to read data from the Arduino.




Step 6. Plot both input sinusoid and the output sinusoid coming from the Arduino on the same graph. Note that the following script generates the input sinusoid. You need to plot input generated by the following code and the output velocity coming from Arduino on the same graph with different colors. Your result should look like below.

```
angular_frequency = 0.1;
duration = 70;
t = 0:0.01:duration;
input = 10*sin(angular_frequency * t);
```




Step 7. Take Fast Fourier Transform (FFT) of both input and output signals by using Matlab's 'fft' command. Note that the resulting values are complex numbers.

- Magnitude: Find magnitude of the signals found in Step 7 by using Matlab's 'abs' command. The index of the maximum value of the absolute fft signal corresponds to the frequency value you are working on. Therefore, you can find the gain of the DC motor plant at this specific frequency by dividing the maximum value of the output to the maximum value of the input.

  [maxVel, maxVelIndex] = max(abs(fft(velocity.signals.values)))

$$K = \frac{\max\left(abs(fft(\text{velocity.signals.values}))\right)}{\max\left(abs(fft(input))\right)}$$

  Note that maxVelIndex represents the index of the maximum value in the absolute velocity signal.
- Phase: Find the phase of the both fft signals and find the phase difference by subtracting the phase of the input signal in maxVelIndex from the phase of the velocity signal in the same index as shown below.

$$\phi_{velocity} = angle(fft(\text{velocity.signals.values}))$$

$$\phi_{input} = angle(fft(input))$$

$$\phi = \phi_{velocity}(maxVelIndex) - \phi_{input}(maxVelIndex)$$

- Save both 20*log10(K) and Ø values in an array to use them later for bode plot generation

Step 8. Plot both the magnitude and phase graphs using 20*log10(K) and Ø values on the figures you generated in Q.2 for comparison.

**Q.3. The hardware system has a time delay of 10 ms due to the processing requirements. Therefore you may observe a phase difference in the high frequencies. The main reason for this problem is that you did not consider the effect of time delay in your estimated transfer function.**

Step 1. Make a first order Pade approximation for the 10 ms time delay and update your transfer function as

$$G_{delayed}(s) = G(s)\frac{1 - 0.005s}{1 + 0.005s}$$

Step 2. Please draw the bode plot of the new estimated transfer function on the Bode plot comparison you obtained in Step 8 of Q.2. Note that you **will have three Bode plots** on the same graph; estimated transfer function in Q.1., estimated transfer function in Q.3 and experimental Bode obtained in Q.2. for both magnitude and phase.