Realistic DC Motor Step Response Simulation

Objectives:

- Compare 1st and 2nd order DC motor models, with and without friction
- Examine the hardware effects of discretization, delay
- Compare the simulated response with the experimental response

This is a simulation only lab – no hardware is needed. Simulink is used to solve equations of motion (Ordinary Differential Equations or ODEs) and simulate/predict the response of the system.

Part 1: 1st and 2nd order Step Response

Objectives:

- Compare 1st and 2nd order DC motor models, with and without friction
- Determine what model is a "good enough" approximation to use
- Evaluate if the simulation matches the experiment step response

Background Information:

The 2nd order transfer function for a DC motor is:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{LJs^2 + (JR + Lb)s + (Rb + K_tK_b)}$$

If the inductance is "small" it can be neglected (L=0) and the first order transfer function is obtained:

$$G(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{JRs + (Rb + K_tK_b)}$$

Where the motor parameters are:

- K_t – torque constant
- K_b – back emf constant
- b – viscous damping coefficient

- R – armature resistance
- J – armature resistance (or combined motor and load inertia if load is attached)
- L – inductance

This lab will examine the effects of the inductance and damping terms. It will also examine the effect of nonlinear coulomb friction (which cannot be included in a transfer function)

$$F_c = sign(\omega) * T_f$$

Where F_c is the total coulomb friction torque – it is a constant magnitude T_f that depends on the direction of rotation $sign(\omega)$. This nonlinear effect will be included in the simulation.

Define Important Motor parameters:

Create an m-file that will have all the important motor parameters – these values may need to be changed for your particular DC motor:

```
% Motor_Parmaeters.m
Rm = 10;     % ohms
Kb = .7;     % Vs/rad
Kt = .7;     % Nm/A
Bm = 0.01;   % Nms/rad (viscous friction)
Lm = 0.004;  % H
Jm = 0.002;  % kgm^2 (combined J)
Tf = 0.02;   % Nm (coulomb friction)
encoder_counts=720;    % number of counts (if using quad encoding)
Vsupply=4.5;           % driver supply voltage (max saturation voltage)
```

Figure 1: Motor parameters stored in script "Motor_Parameters.m"

Simulation Diagram

Create the following simulation diagram that has 3 different DC motor models, a 2^{nd} order, a 1^{st} order and a 1^{st} order with Coulomb/Dry friction.

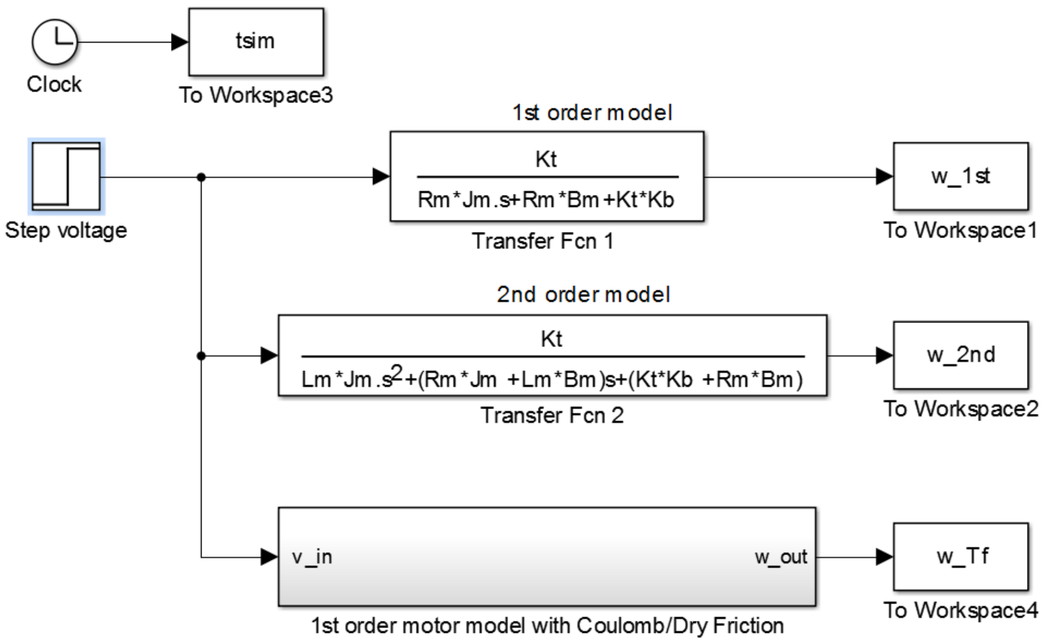


Figure 2: Simulink diagram for 3 different motor models

The first two blocks use transfer function blocks, the third is a subsystem you will create as shown below.

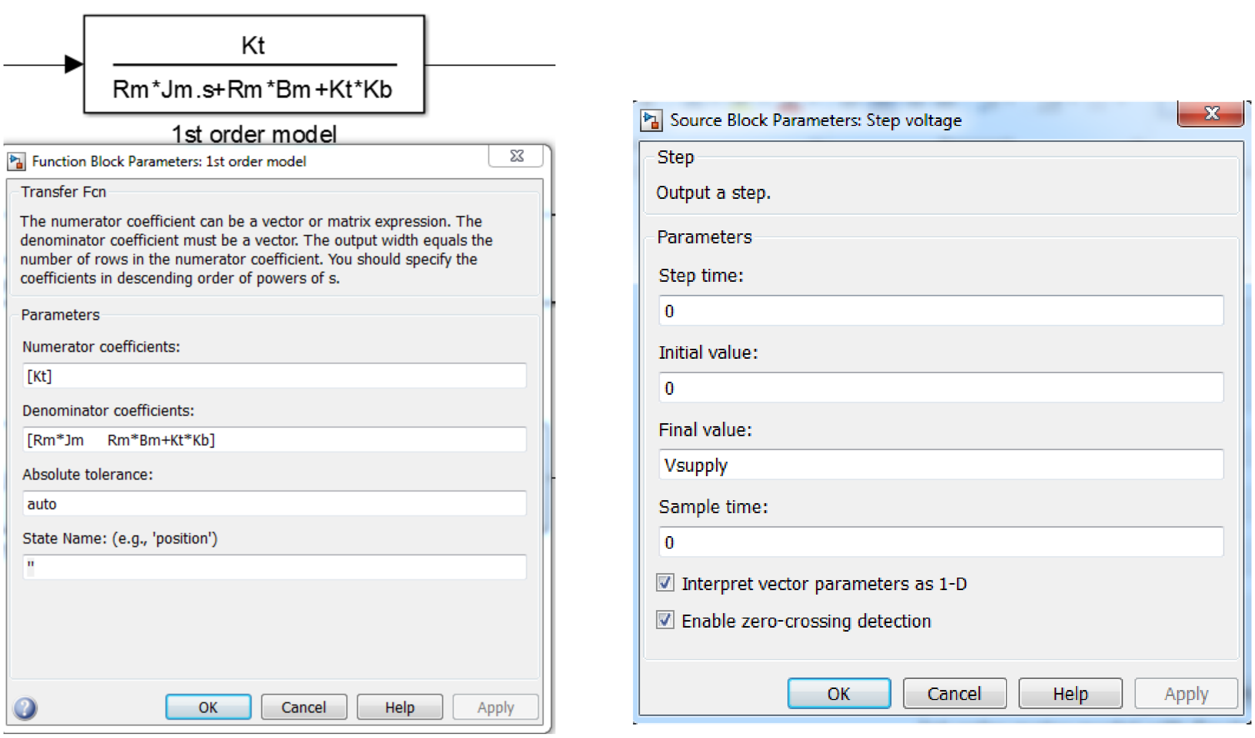


Figure 3: Transfer function and Step block parameters

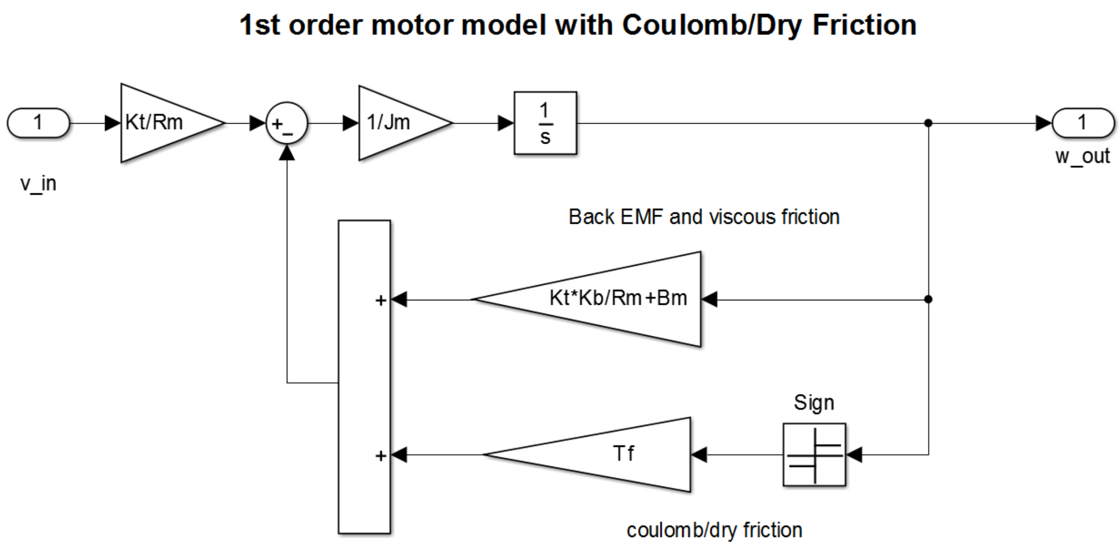


Figure 4: The subsystem contents of the previous figure

Simulation Script

Create an m-file that will contain all the important simulation settings, run the simulations and plot the results:

```
clc, clear all, close all

% load motor parameters:
Motor_Parameters;

% simulation parameters
Tsim_final=.3;          % final simulation time in seconds
sim_step_size = .0001; % how often we want data printed from the simulation

% Simulate the model
sim('Motor_Step_Response_Linear_1st_2nd_Coulomb')

% plot the response in RPM
figure(1), hold on
h1=plot(tsim, w_1st*RADSEC2RPM, 'b');
h2=plot(tsim, w_2nd*RADSEC2RPM, 'r');
htf=plot(tsim, w_Tf*RADSEC2RPM, 'g');

% Simulate with no Friction and plot
Bm=0; Tf=0;
sim('Motor_Step_Response_Linear_1st_2nd_Coulomb')
h1_nf=plot(tsim, w_2nd*RADSEC2RPM, 'c--');
h2_nf=plot(tsim, w_Tf*RADSEC2RPM, 'm--');

xlabel('Time (seconds)')
ylabel('angular velocity (RPM) - input 4.5v step')

legend([h1 h2 htf h1_nf h2_nf],...
    {'1st order - viscous only',...
    '2nd order - viscous only',...
    '1st order - coulomb','1st order - no coulomb no viscous',...
    '2st order no viscous'})
```

Figure 5: m-file “Sim_Motor_Response.m” used to simulate and plot motor model simulations

Use the following simulation solver settings:

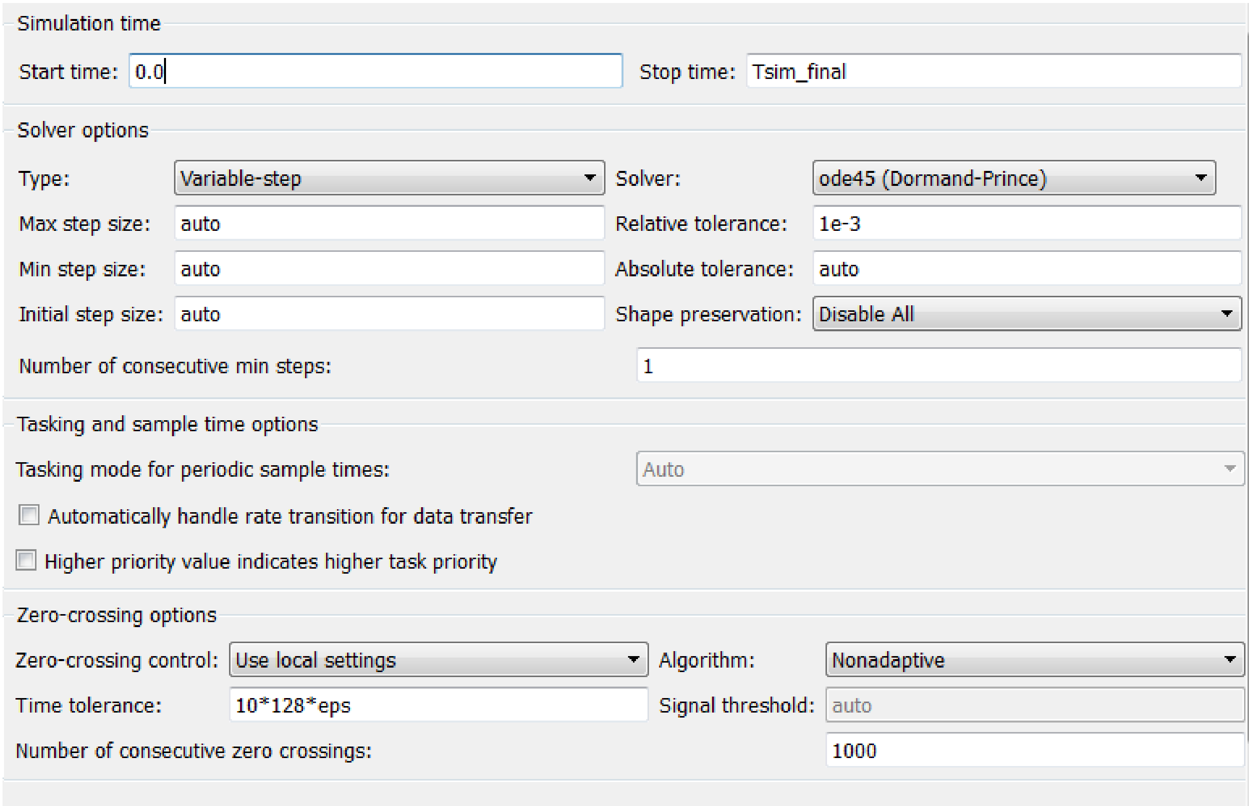


Figure 6: Simulation settings used to solve system response

- Notice the different settings here are used to ***solve*** equations of motion, ***not*** to program hardware
- Notice the data recorded by the simulation is specified in the "To Workspace" Blocks
- The **parameter `sim_step_size=.0001` sec is how often the *simulation* will provide you data points as it solves the system response.** Notice it is much smaller than the step size used to program the hardware (typically .005-.03 sec). This is to ensure your calculated predicted solution from Simulink has enough resolution to capture all dynamics that may occur in the system.
- You will need to calculate the correct conversion factors and add these to you m-file
- The size of the voltage step is `Vsupply`

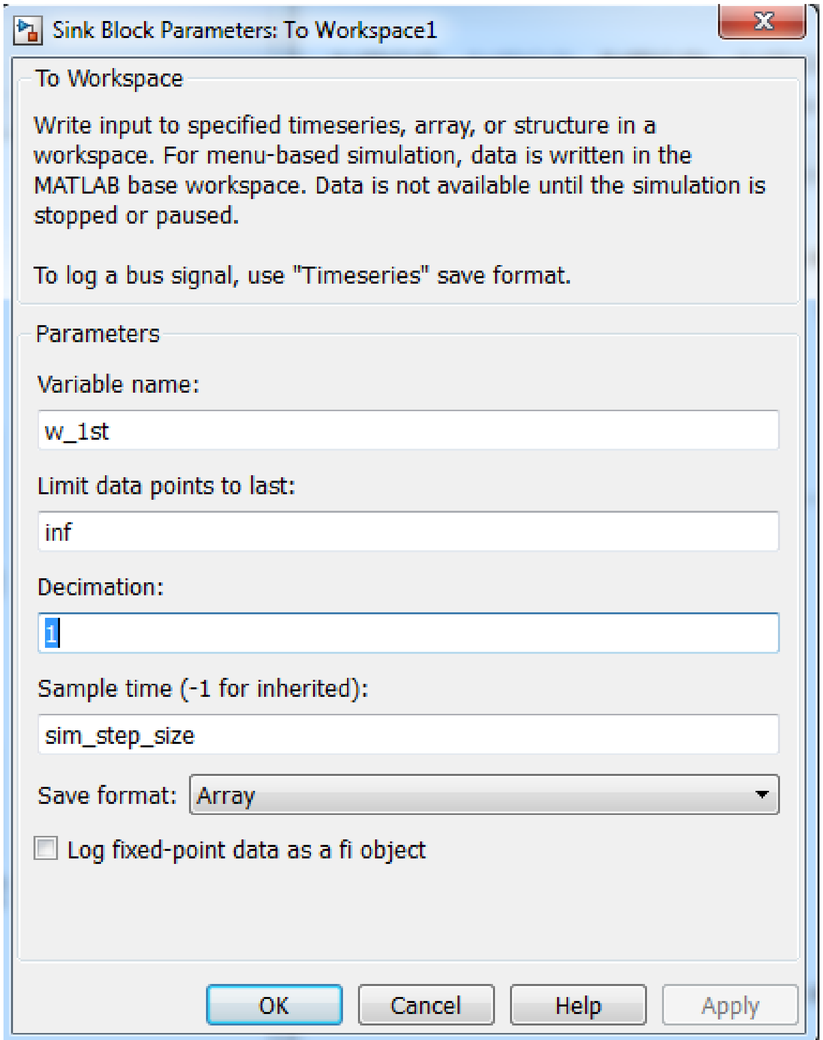


Figure 7: settings for "To Workspace" blocks

Run the simulation with and without friction as specified in the m-file and plot the results.

Questions:

- For the specified set of parameters
    - What can you say about using a 1^{st} order model?
    - What is the dominant “damping” in the system? What “damping” terms can be ignored (if any)?
    - Does this step response match the experimental step response from previous labs?

Part 2: 1st Order System with Discretization and Delay

Objectives:

- Add the hardware implementation effects to the simulation
    - Discretization - from sensors and data type
    - Delay - from sampling data

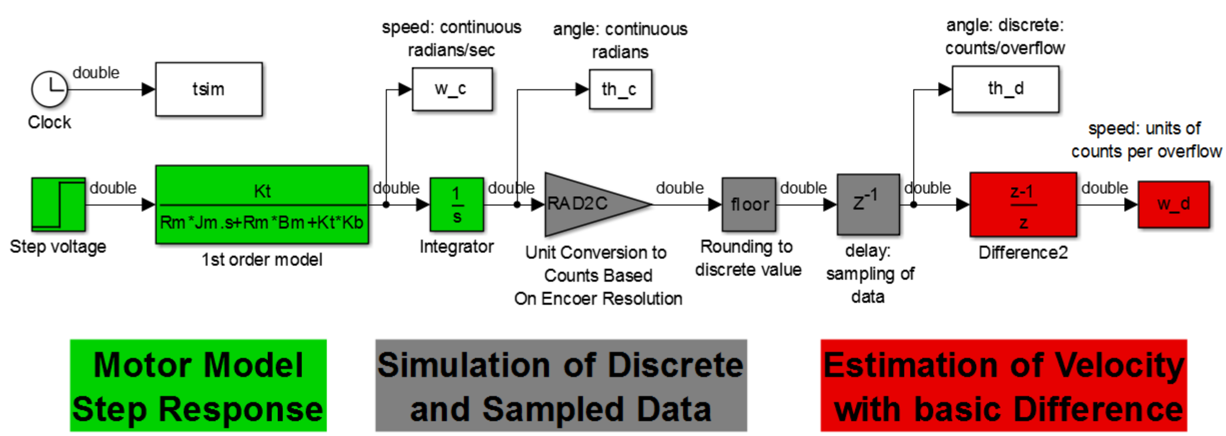
Background Information:

The implementation on the hardware introduces several effects:

- Encoder provides only discrete data points – 720 counts per revolution
- The hardware only samples this data at specific time interval `TS` – a delay introduced from measurement to measurement
- The velocity is calculated from the difference between two discrete position measurements

Simulation Diagram

Create the following simulation diagram:



Notes:
- This simulates the discrete nature of the encoder and the sampling of the data.
- All the data types are doubles & all the math is floating point: this does not take into account datat types and fixed point math!

The delay block has a sample time specified by "TS" our microcontroller sample time,

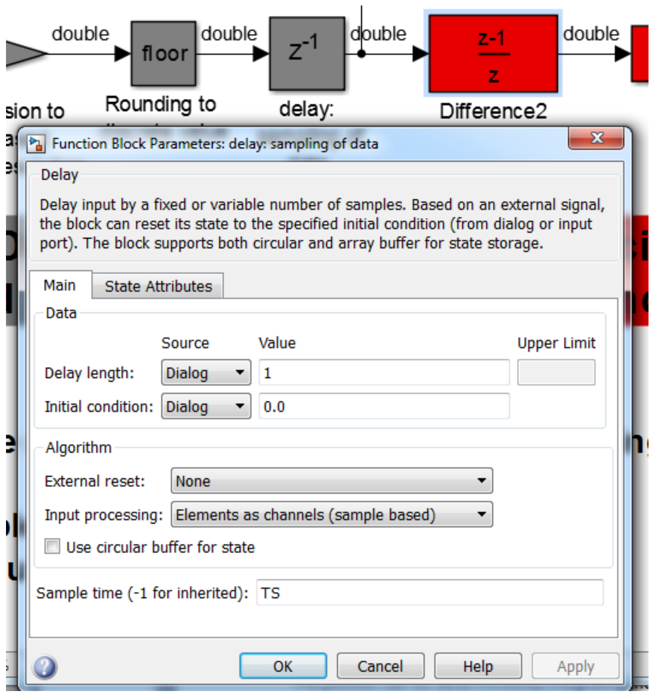


Figure 8: Delay block setting

Diagram Overview:

- Step response is simulated with the first order model
- Integrator computes the position (what is actually measured in the system)
- Position is converted to counts by the number of encoder counts available in quadrature mode – this represents the discrete resolution of your measurement system
- Counts are rounded to integers to represent the discrete data type of the encoder (int16)
- Delay represents the delay time in measuring the signal with the microcontroller
- And the difference is the calculation of the speed as implemented on the microcontroller

The m-file can be run to simulate and plot the data:

```
%% 1st order system with discretization and delay and velocity calculation
Tsim_final=.5;          % final simulation time in seconds
TS=.03;sim_step_size = TS;  %#ok<*NASGU>
sim('Motor_Step_Response_Linear_With_Encoder_And_Delay')
plot(tsim, w_c*RADSEC2RPM, 'b'), hold on
plot(tsim, w_d*C2RAD/TS*RADSEC2RPM, 'g')

TS=.003;sim_step_size = TS;
sim('Motor_Step_Response_Linear_With_Encoder_And_Delay')
plot(tsim, w_d*C2RAD/TS*RADSEC2RPM, 'm')
```

Notice that the “sim_step_size” is set to “TS”. This is forcing the simulated solution to provide data points at the same frequency that the hardware obtains data points.

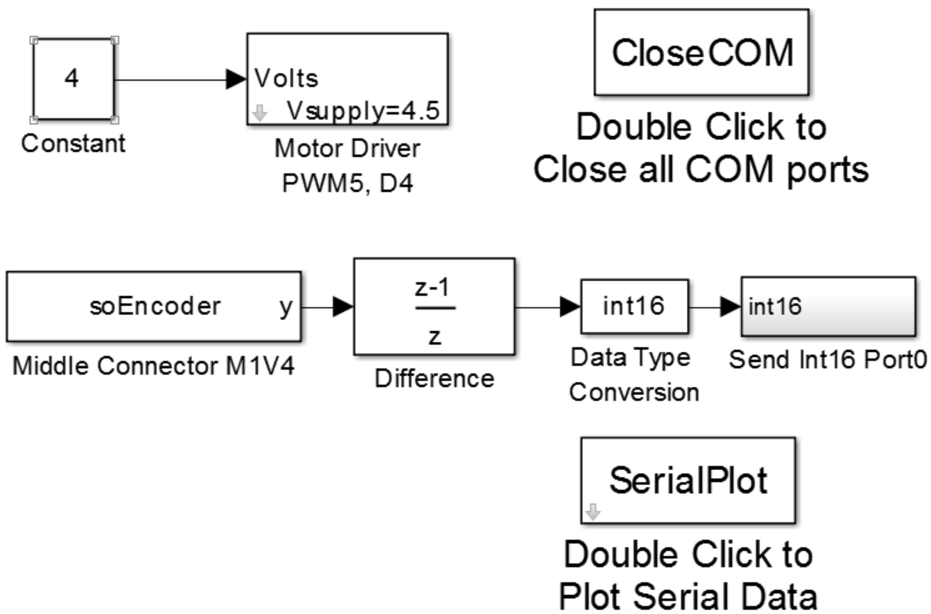
Questions:

- Provide a plot of the 1st order system simulated response of the discretized system at TS=0.03 seconds and the discretized system at TS=0.003 seconds

Part 3: Experimental Response

Experimental Response

Use the following Simulink diagram (used previously) to capture the step response in serial mode at a time step of 0.03 seconds and 0.003 seconds. (you may have this already from previous labs)



- Use the Encoder block for your system
- Be sure “enable override detection” is enabled on pin 13 to ensure your program does not overflow

Questions:

- Provide a plot the of the experimental step response versus the simulated discretized step response for both 0.03 seconds and 0.003 seconds (be sure to convert your units correctly!)
- How well does your simulated response match the experimental response now?