ELEC 341 – Project

Project P-1

6 Marks

<u>Learning Objectives</u>

- Simulink System
  - Lay-out Control System
  - Create Sub-systems
  - Generate M-file
  - Set Callback Routines

Simulink is not just a mathematical simulation tool. It's also a handy drafting tool for laying out a control system. Even if you aren't sure what belongs in a box, you can always set it to a gain of "1" and it will act like it's not even there. Once everything is laid out, you can identify all inputs, outputs, and physical units. Once you have a basic frame-work, you can update it as you go.

*I like to colour-code my blocks to indicate which family they belong to (u-controller, system, etc). I used the **LTI System** block from the **Control System Toolbox** for my transfer functions, but you should use "**Continuous/Transfer Fcn**" so you can specify your numerator and denominator by a vector.*

Simulink System

The "Electro-Mechanical System" shown is a position control system including:

- P-Controller
- Voltage Amplifier
- Motor & Mechanism
- Position sensor

Generate a Simulink system with all of the indicated components. Use a script to set all of the specified values.

The specified values are all bogus, but valid, values for now. You will change these to more meaningful values as the course goes on. But you should see something similar to the example curves if you implemented the system correctly.

You can assign almost any parameter in a Simulink System to a Matlab workspace variable. This allows you to keep all of your specifications in a Matlab script. You can use any Matlab functions to calculate them. You can add comments. You can even generate Matlab plots using the same values without having to port the code.

Simulink Callback functions can be used to automatically run a script so your system runs seamlessly. The PreLoadFcn is run when a Simulink system is first opened. This makes sure your variable names all have values. If they don't, Simulink displays "???'" The InitFcn is the first thing that is run when you press Play. This allows you to edit the values in the script and have them automatically applied, any time you run the simulation.

System Values

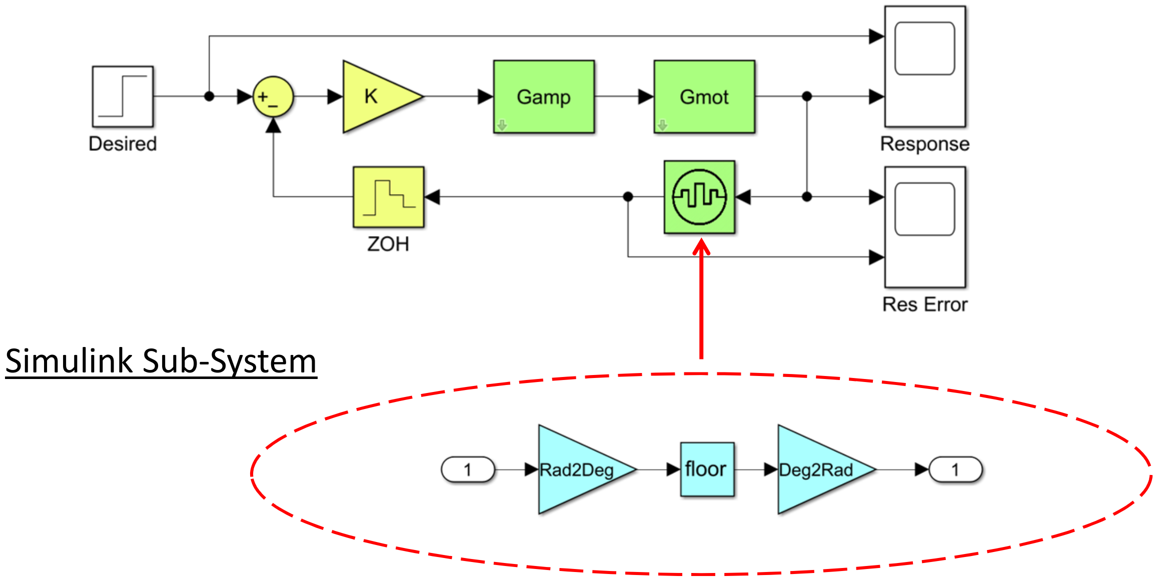
Generate a Matlab script that sets the following parameters.

Organize the script and add **COMMENTS** so it is clear and easy to follow.

- simTime = simulation time (sec)
- ipMag = magnitude of input step (rad)
- numXXX = numerator of each transfer function (use any naming convention you like)
- denXXX = denominator of each transfer function (use any naming convention you like)
- Rad2Deg = physical unit conversion (deg/rad)
- Deg2Rad = physical unit conversion (rad/deg)
- K = controller gain (V/rad)

Call the script in the **PreLoadFcn** and **InitFcn** Callback Functions of the Simulink system.

Simulink System



Simulink Sub-System

Parameter Values

$$K = \frac{\#C}{5}$$

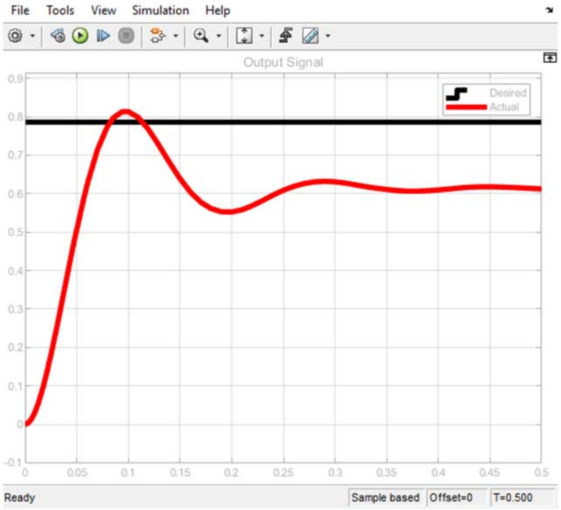
$$Gamp = \frac{\#A}{s + \#A}$$

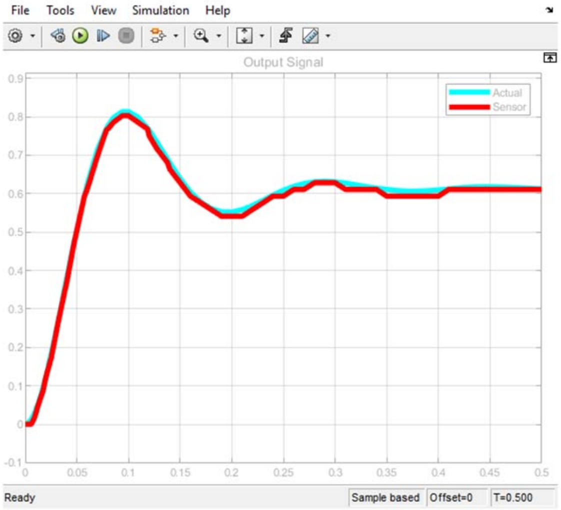
$$Gmot = \frac{\#B}{s + \#B}$$

$$ZOH = \frac{1}{50}$$

ipMag = pi/4

simTime = Just enough to let system settle





Deliverables

Calculations

- n/a

Figures

- System (2 marks)
- Sub-System (1 marks)
- Response Plot (1 marks)
- Res Error Plot (1 marks)
- Matlab Function (1 marks)
  - printout of code