ES 3011 C-2019

Lab #5: Transient Response, Proportional Control, PID Control

INTRODUCTION

In this lab, you and a partner will create controllers for applied system models representing a mass-spring-damper, cruise-control, and DC motor position. Lastly, you will finish the cruise-control Simulink model by creating a PID controller.

Please have the following outline for your report:

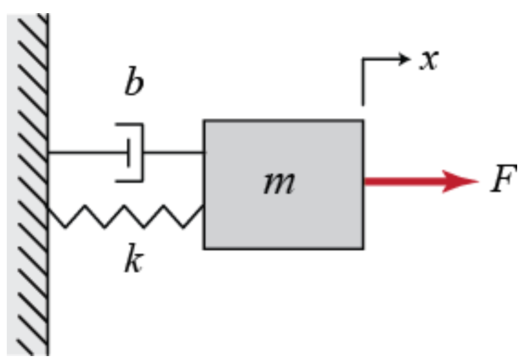
1. A few sentences of introduction of the topic of the lab.
2. Answers to each problem with concise explanations on your process in solving and outcome.
3. A paragraph concluding the report explaining the goals, what you learned, and any other conclusions.

Note: In general, increasing the following parameters has the following effect on the transient response:

CL RESPONSE	RISE TIME	OVERSHOOT	SETTLING TIME	S-S ERROR
Kp	Decrease	Increase	Small Change	Decrease
Ki	Decrease	Increase	Increase	Decrease
Kd	Small Change	Decrease	Decrease	No Change

MATLAB

I) MASS-SPRING-DAMPER SYSTEM



$$F(t) - b\dot{x} - kx = m\ddot{x}$$

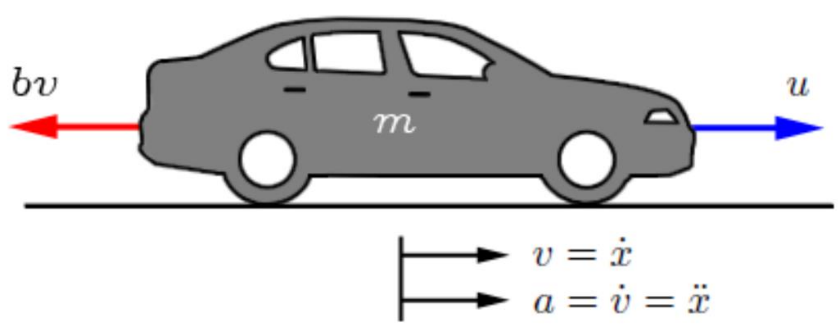
This system has the following properties:

m	Mass	1 kg
k	Spring Constant	10 N/m
b	Damping constant	20 N.s/m
F	Input Force	2 N

1. Plot the step response of the system with the input force. *Note property value changes*
2. Create a Proportional Gain Controller with a value of Kp=350 and plot the step response, time sample from 0 to 2, with 0.01 time step. Assume unity feedback.
3. Create a PID Controller with gains Kp=350, Ki=300, and Kd=50 and plot the step response. Assume Unity Feedback.

HINTS: Useful MATLAB commands to reference { pid(P,I,D), step(Gain*TF, t), feedback(Plant, 1)}

II) Cruise-Control Car



$$m\dot{v} + bv = u$$

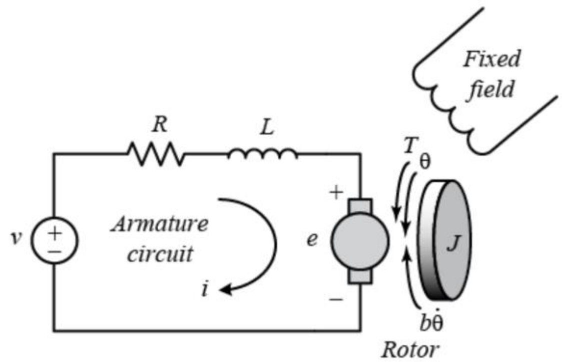
This system has the following properties:

m	Vehicle mass	1500 kg
b	Damping Coefficient	50 N.s/m
u	Reference Speed	10 m/s

1. Create a Proportional Gain Controller with a value of Kp=1, 10, 50 and plot the step responses on one graph. Let the time sample be 0 to 200 seconds, with a step of 0.1. Your input is now the reference speed 10 m/s.
2. Create a PID Controller with gains that you find creates a response with a rise time < 1.5 seconds, settling time < 10 seconds, Overshoot < 10%, 0<Kp<250, 0<Ki<100, 0<Kd<150. What are your Kp, Ki, Kd, rise time, settle time, and overshoot values? Use pidTuner(Plant, 'pid'). Note: Use sliders and check values by clicking 'Show Parameters'. Write down your final values.

HINTS: Useful MATLAB commands to reference {pid(P, I, D), step(Gain*TF, t), feedback(Plant, 1)}. Also, use 'hold on' between step() plots to plot multiple step functions on same graph. End with 'hold off'.

III) MOTOR POSITION



Newton's 2nd Law and Kirchhoff's voltage law gives us these equations:

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

This system has the following properties:

J	Moment of inertia of the Rotor	3.23 E-6 kg.m^2
b	Motor friction constant	3.51 E-6 N.s/m
K	Electric Force and Motor Torque Constant	0.0275 V/rad/sec
R	Electric Resistance	4 Ohm
L	Electric Inductance	2.75 E-6 H

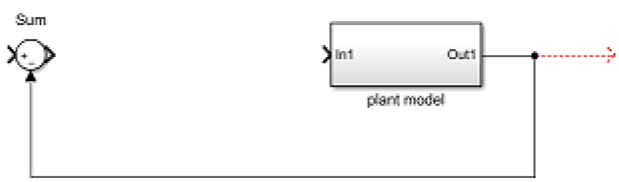
1. Create a Proportional Gain Controller with values of Kp=1, 10, 50 and plot the step responses on one graph. Set time from 0 to 0.25 seconds, with a 0.001 step. Assume unity feedback. Comment on the output. Note: your plant TF should be Theta(s)/V(s).
2. Create a PID Controller with Kp = 20, Ki=500, and Kd = 0.15. Use the time range of 0 to 0.1. Plot the step response and comment on the result. Gather the step response info.

SIMULINK

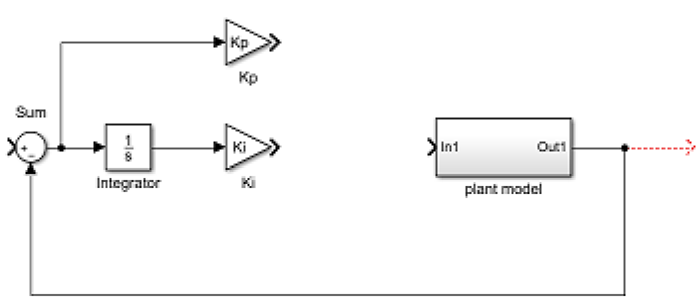
CRUISE-CONTROL: CONTROLLER DESIGN

1) Open a new model window.
2) Drag a Subsystem block from the Ports & Subsystems library into your new model window.
3) Double-click on this block. You will see a blank window representing the contents of the subsystem (which is currently empty).
4) Open your previously saved model of the cruise control system.
5) Select **Select All** from the **Edit** menu (or **Ctrl-A**), and select **Copy** from the **Edit** menu (or **Ctrl-C**).
6) Select the blank subsystem window from your new model and select **Paste** from the **Edit** menu (or **Ctrl-V**). You should see your original system in this new subsystem window. Close this window.
7) You should now see input and output terminals on the Subsystem block. Name this block "plant model".

8) Now, we will build a PI controller around the plant model. First, we will feed back the plant output.
   a. Draw a line extending from the plant output.
   b. Insert a Sum block and assign "+-" to it's inputs.
   c. Tap a line of the output line and draw it to the negative input of the Sum block.



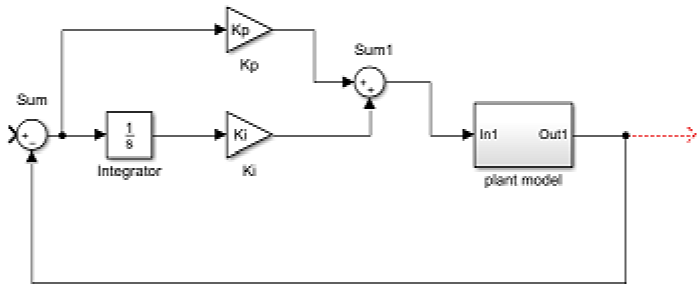
9) The output of the Sum block will provide the error signal. From this, we will generate proportional and integral components.
   a. Insert an Integrator block after the Sum block and connect them with a line.
   b. Insert and connect a Gain block after the Integrator block to provide the integral gain.
   c. Label this Integrator "Ki" and assign it a value of "Ki".
   d. Insert a new Gain block and connect it with a line tapped off the output of the Sum block.
   e. Label this gain "Kp" and assign it a value of "Kp".



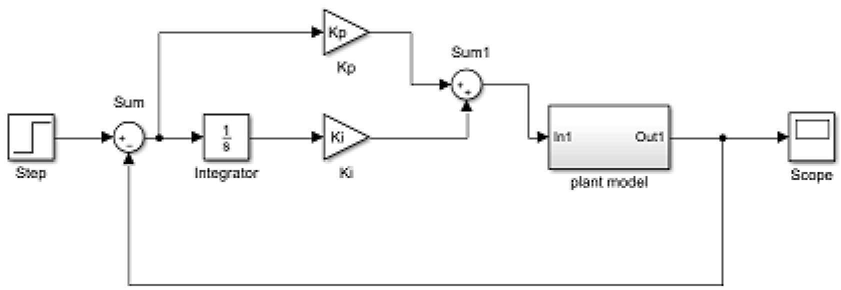
10) Now we will add the proportional and integral components and apply the sum to the plant.

a) Insert a Sum block between the Ki block and the plant model and connect the outputs of the two Gain blocks to the Sum block inputs.

b) Connect the Sum block output to the input of the plant block.



11) Finally, we will apply a step input and view the output with a Scope block.
    a. Attach a Step block to the free input of the feedback Sum block.
    b. Attach a Scope block to the plant output.
    c. Double-click the Step block and set the Step Time to "0" and the Final Value to "u". This allows the input magnitude to be changed outside of Simulink.



12) To simulate this system, first, an appropriate simulation time must be set. Select **Parameters** from the **Simulation** menu and enter "10" in the Stop Time field. The design requirements included a rise time of less than 5 sec, so we simulate for 10 seconds to view the output. The physical parameters must now be set. Run the following commands at the MATLAB prompt:
    a. m=1500
    b. b=50
    c. u=100
    d. Kp=800
    e. Ki=40

13) Run the simulation and save the closed-loop response.