MECH-3221 Control Theory

Homework 10

Instructions

- Make sure the name and student number for this homework are yours, if not contact your course instructor immediately.
- This evaluation covers **material from the tenth week of classes**.
- Note that each student has a different version, so do not try to copy from one another as it would cost you both mark and risk of plagiarism.
- If asked, write all the steps involved and all the equations used. Final answer ≠ full mark!
- This evaluation **is not** strictly multiple-choice
- Be cautious of the **time cues**.
- If this is not a strictly multiple-choice evaluation
    a) For qualitative questions, write down the key points, illustrate key concepts, and be concise.
    b) Make sure to sectionalize your answers referring to question elements and put your final answer for each section in a box.
    c) You need to either print this document, complete writing your solution and scan the material back to PDF and upload it or use a tablet or any other device that allows you to write on PDF files, save it and upload it. If neither is possible, you can only scan your solution pages and upload. For multiple choice questions, on your answer sheet, mention the question number and your choice for the question.
    d) The filename to upload must follow the "Lastname_firstname_XX.pdf" where XX is the last 2 digits of your student number and your name as shown on top of this page.
- All submissions must be electronic, no other submission format is accepted.
- Late submission is not accepted and will get a mark of ZERO.

Evaluation

Questions are graded based on the rubrics

Question 1 [4 marks] [60 minutes] [LO. 5]

Consider the simple mass-spring-damper system shown in Figure 1 with $m = 1\ kg$, $b = 10\ Ns/m$, $k = 20$ N/m, and $F = 1\ N$. Note that the submission for this evaluation must be in computer generated document format. No handwritten or screenshot code, plot, formula, solution is accepted for this evaluation. You must use Microsoft equation or Mathtype/Latex to insert your equations for this evaluations.

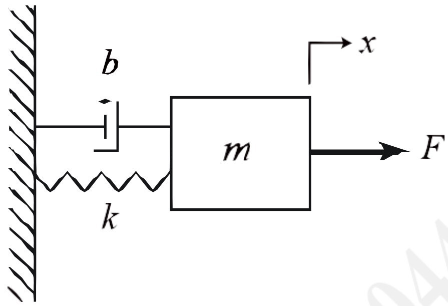


Figure 1 Schematics of mass-spring-damper system

a) Derive a mathematical model for this system with displacemenet (x) as the only dynamic variable. Note that you must provide a FBD as part of your solution. **[$0.4\ marks$]**

b) Derive the TF for the model in part (a) where output is displacement (x) and input is force (F). **[$0.2\ marks$]**

for all the simulation below the input is a step input $F(t) = 1U(t)$ which is a step with magnitude of 1 at $t = 0$. The initial states are $x(0) = 0$ and $\dot{x}(0) = 0$. The simulation time should be $t = 0$ to $t = 3$ sec with the step size of $dt = 0.001\ sec$. Plot the output vs. time, $y(t)$. You have to use $tf(\cdot)$ function and define all required matrices and vectors. Your code and output plot must be in vector format inserted in the document. All your code lines must be properly commented and explain what each line does. Your plot must have proper labels and titles. You must use the $pid(\cdot)$ function to construct the PID transfer function and $feedback(\cdot)$ function to construnct the closed-loop transfer function (CLTF). When constructing feedback controllers, follow the diagram and blocks shown in Figure 2 with $G_C(s)$ for controller, $G_P(s)$ for plant and $H(s)$ for sensor, which for this problem is unity ($H(s) = 1$).

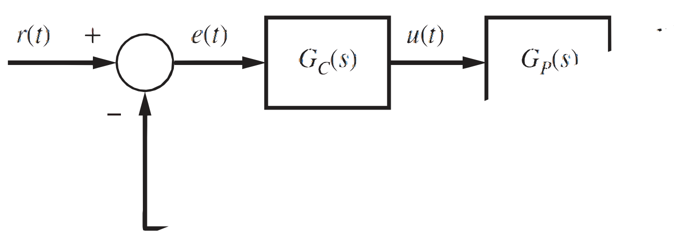


Figure 2 schematic of the general feedback control system

c) Use MATLAB commands to obtain the dynamic response the system. **[$0.8\ marks$]**

d) Add a proportional (P-controller) to the system with K_P **=** *300* in (b) and use MATLAB commands to obtain its dynamic response. **[**$0.8\ marks$**]**

e) Add a proportional-derivative (PD-controller) to the system with K_P **=** *300* and K_D **=** *10* in (b) and use MATLAB commands to obtain its dynamic response. **[**$0.8\ marks$**]**

f) Add a proportional-integral-derivative (PID-controller) to the system with K_P **=** *300* , K_I **=** *300* and K_D **=** *10* in (b) and use MATLAB commands to obtain its dynamic response. **[**$0.8\ marks$**]**

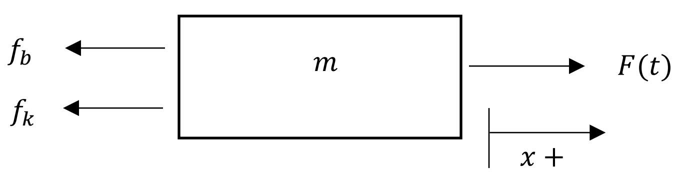
g) Provide a discussion on what you observe from the responses for the system under P, PD, and PID controllers. In particular, what similarities and differences they have, which one works better for this system, what else can be tuned to further improve the system response? **[**$0.2\ marks$**]**

You are required to show all the steps involved in finding your final answer, even the smallest details. Make sure in your final answer, all numerical values are substituted and equations are simplified to the simplest form.

Solution

Provide your step by step solution here. Note that only providing the correct final answer does not guarantee a full mark for the question!

a)



$$\overset{+}{\rightarrow}\sum F_x = m\ddot{x} = -f_b - f_k + F(t)$$

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

Let: $x_1 = x$ and $x_2 = \dot{x}_1 = \dot{x}$

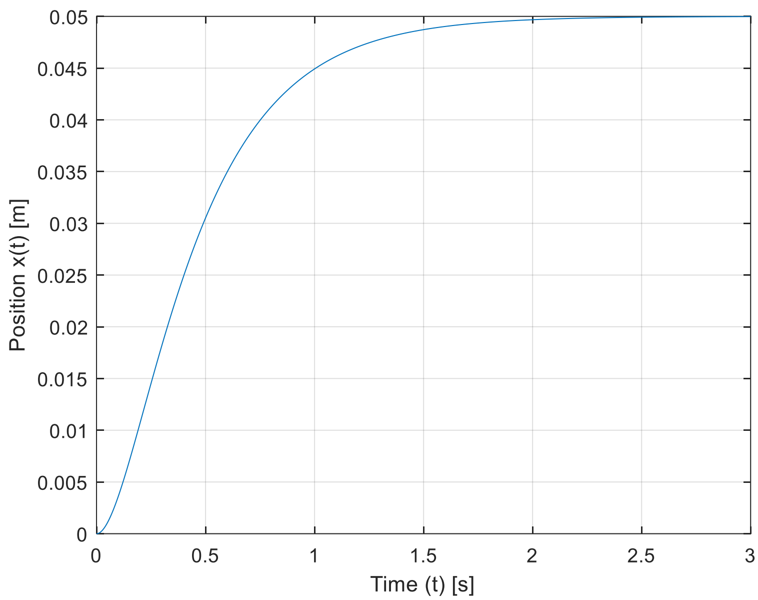
$$f(t) = m\dot{x}_2 + bx_2 + kx_1 = F(t)$$

b)

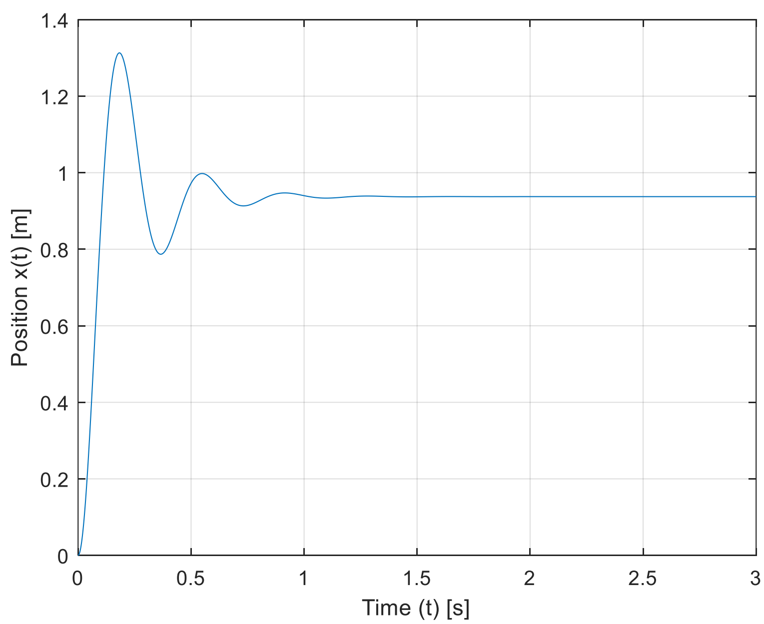
$$\mathcal{L}(f(t)) = \left(ms^2X(s) - sX(0) - \dot{X}(0)\right) + b\left(sX(s) - X(0)\right) + kX(s) = F(s)$$
$$\mathcal{L}(f(t)) = ms^2X(s) + bsX(s) + kX(s) = F(s)$$

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{ms^2 + bs + k}$$

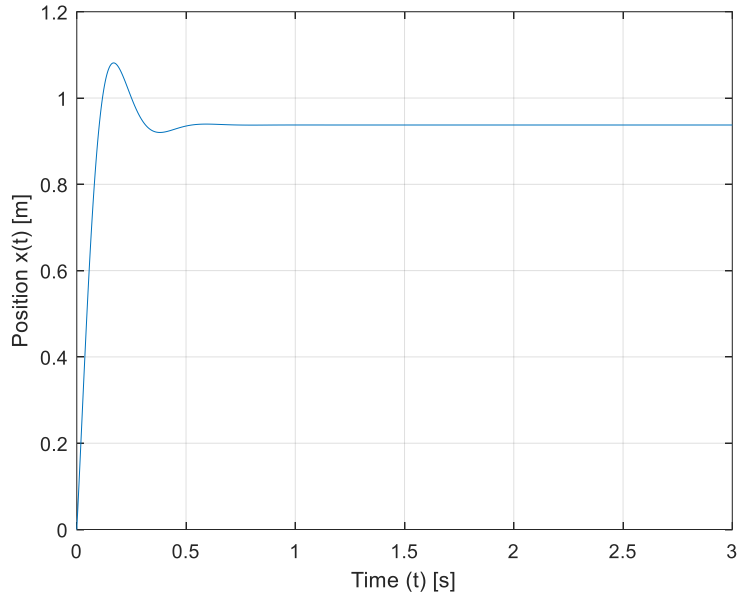
c) Using the code in Appendix 1, the open-loop system response is visualized:



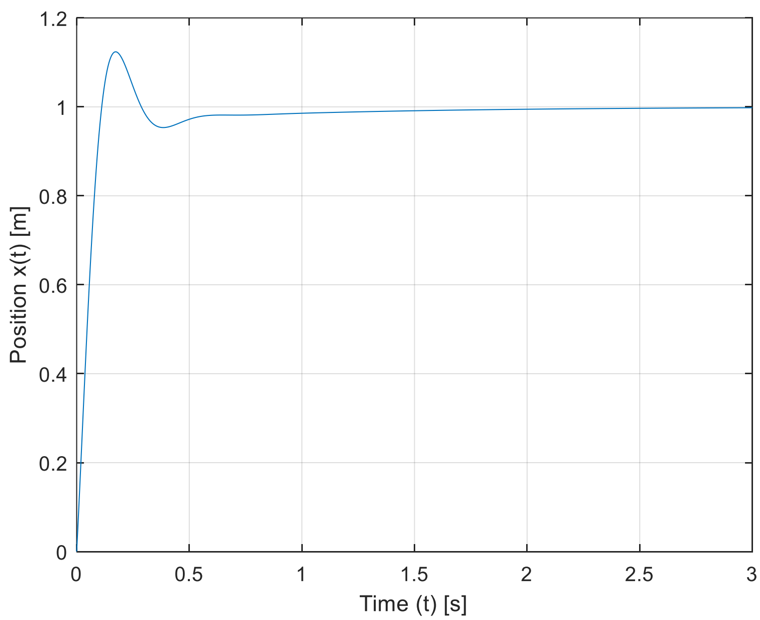
d) Using the code in Appendix 1, the closed-loop system response with a PID controller ($K_P = 300$) is visualized:



e) Using the code in Appendix 1, the closed-loop system response with a PID controller ($K_P = 300, K_D = 10$) is visualized:



f) Using the code in Appendix 1, the closed-loop system response with a PID controller ($K_P = 300, K_I = 300, K_D = 10$) is visualized:



g) Using the standard open-loop feedback, the system took ~2.5 seconds to stabilize at steady state of 0.05. After the integration of the PID controller, the system took less time to stabilize at the unit step input of 1 at ~1.5 seconds. As we tune the PID gains, we get closer to the intended steady state values with an ideal response (K_P increase system response time, K_D introduced higher frequencies to the system allowing "damping" to higher harmonics, and K_I ensure the final steady-state is zeroed out to unit step). For further tuning, K_D can be increased to reduce the harmonic oscillations seen in part f, as well as following PID tuning procedures, such as Ziegler-Nichols Tuning Rules, to find ideal gain values.

APPENDIX 1 – MATLAB CODE

```
%HW10
%Atilla Saadat - 104411786
%August 6, 2020
%----------------------------
%initialize system parameters and constants
m = 1;
b = 10;
k = 20;

dt = 0.001; %define step size for simulation time
t = 0:dt:3.0; %define simulation time domain
u = 1.0*ones(size(t)); %define step input wih init amplitude for all time steps
x0 = 0; %define initial states for x

sysG = tf(1,[m b k]); %define transfer function for SISO problem
[y,t] = lsim(sysG,u,t,x0); %define open-loop system response (as per
announcement)
figure %create new figure
plot(t,y); %create plot of y vs. t
grid on %include grid in plot
ylabel('Position x(t) [m]')  %label plot axis
xlabel('Time (t) [s]')
%%
C = pid(300); %create PID TF with p-gain = 300
sysT = feedback(sysG*C,1,-1); %create negative feedback look with PID
controller and mass system TF
[y,t] = lsim(sysT,u,t,x0); %Simulate closed-loop time response of dynamic
system
figure %create new figure
plot(t,y); %create plot of y vs. t
grid on %include grid in plot
ylabel('Position x(t) [m]')
xlabel('Time (t) [s]')
%%
C = pid(300,0,10);%create PID TF with p-gain = 300 and d-gain = 10
sysT = feedback(sysG*C,1,-1); %create negative feedback look with PID
controller and mass system TF
[y,t] = lsim(sysT,u,t,x0); %Simulate closed-loop time response of dynamic
system
```

```
figure %create new figure
plot(t,y); %create plot of y vs. t
grid on %include grid in plot
ylabel('Position x(t) [m]')
xlabel('Time (t) [s]')
%%
C = pid(300,300,10);%create PID TF with p-gain = 300, i-gain = 300, and d-gain
= 10
sysT = feedback(sysG*C,1,-1); %create negative feedback look with PID
controller and mass system TF
[y,t] = lsim(sysT,u,t,x0); %Simulate closed-loop time response of dynamic
system
figure %create new figure
plot(t,y); %create plot of y vs. t
grid on %include grid in plot
ylabel('Position x(t) [m]')
xlabel('Time (t) [s]')
```