PID Position Control of a Spring-Mass-Damper

Table of Contents

Clear Variables and turn on pause feature ................................................................. 1
Define parameters ....................................................................................................... 1
Define the open-loop transfer function and plot the root locus diagram ..................... 1
Find a suitable "Kd" for the closed loop system ............................................................. 2
Define the closed-loop transfer function with the selected gain "Kd" and plot the step response ................................................................. 3
Calculate and display the performance characteristics (performanceCharacteristics) in the Command Window ............................................................. 4
Plot the Bode diagram of the closed loop system and display characteristics in Command Window ........................................................................................................... 5

This file produces root locus diagram and step responses for PID position control of a spring-mass-damper system.

It also plots a Bode diagram of the closed loop system.

Clear Variables and turn on pause feature

clear m c kspring a b Kd poles numP denP sysOL sysCL performanceCharacteristics
clear titlestringRL titlestring
disp('PID Position Control of a Spring-Mass-Damper');

% turn on pause feature
pause on;

Define parameters

m = 1.0; % mass in slugs
c = 8.8; % damping coefficient in lb-s/ft
kspring = 40.0; % spring stiffness in lb/ft
a = 15.0; % zero of the PID controller = Kp/Kd
b = 50.0; % second zero of the PID controller = Ki/Kd

Define the open-loop transfer function and plot the root locus diagram

% define the open-loop transfer function (sysOL)
numP = [1,a,b]; denP = conv([1,0],[m,c,kspring]); sysOL = tf(numP,denP);
PID Position Control of a Spring-Mass-Damper

% plot the root locus diagram in figure window #1
figure(1); clf; hold on;
$rlocus(sysOL); grid on;
titlestringRL = ['Root Locus Diagram for Kd. (Kp/Kd = ',num2str(a),',)'; ... 
' (Ki/Kd = ',num2str(b),')'];
title(titlestringRL);

Find a suitable "Kd" for the closed loop system

% pause to allow user to zoom into areas of interest
pause;

% call rlocfind to allow user to pick the pole locations
[Kd,poles] = rlocfind(sysOL);

% display the derivative gain and associated poles in Command Window
disp('Derivative Gain:'); disp(Kd);
disp('Poles Associated with this Gain:'); disp(poles);

Select a point in the graphics window

selected_point =

-17.1933560477002 + 6.41078838174274i

Derivative Gain:
Poles Associated with this Gain:
-17.2259157729046 + 6.44065769502465i
-17.2259157729046 - 6.44065769502465i
-4.45013248359659

Define the closed-loop transfer function with the selected gain "Kd" and plot the step response

```matlab
% define the closed-loop transfer function (sysCL) with selected gain (Kd) and negative unity feedback
sysCL = feedback(Kd*sysOL,1,-1);

% plot the closed loop step response in figure #2
figure(2); clf; hold on;
step(sysCL); grid on;
ylabel('Position Change (ft)');
titlestring = ['Step Response of SMD with PID Control. (Kd = ', num2str(Kd), '), (Kp/Kd = ', num2str(a), '), (Ki/Kd = ', num2str(b), ')'];
title(titlestring);
```

% pause to allow user to annotate the plot
Calculate and display the performance characteristics (performanceCharacteristics) in the Command Window

```
performanceCharacteristics = stepinfo(sysCL);
disp('Performance Data for the Compensated System');
disp('-----------------------------------------------');
disp('Rise Time (sec) ='); disp(performanceCharacteristics.RiseTime);
disp('Settling Time (sec) ='); disp(performanceCharacteristics.SettlingTime);
disp('Percent Overshoot ='); disp(performanceCharacteristics.Overshoot);
disp('Peak Value ='); disp(performanceCharacteristics.Peak);
disp('Peak Time (sec) ='); disp(performanceCharacteristics.PeakTime);
```

**Performance Data for the Compensated System**

- **Rise Time (sec)** = 0.0530858787571838
- **Settling Time (sec)** = 0.196241487017048
- **Percent Overshoot** = 5.39098350305367
Plot the Bode diagram of the closed loop system and display characteristics in Command Window

% plot the Bode diagram of the closed-loop system in figure #3
figure(3); clf;
bode(sysCL); grid on;
titlestring = ['Bode Diagram of Closed Loop SMD with PID Control. (Kd = ', ...'
               num2str(Kd),'), (Kp/Kd = ',num2str(a),') (Ki/Kd = ',num2str(b),')]';
title(titlestring);
% pause to allow user to annotate the plot
pause;
% calculate the bandwidth and DC gain of the closed-loop system
BW = bandwidth(sysCL); DC_gain = 20*log10(dcgain(sysCL));
disp('DC Gain and Bandwidth of the Compensated, Closed-loop System');
disp('============================================================');
disp('DC Gain (dB) ='); disp(DC_gain);
disp('Bandwidth (rad/s) ='); disp(BW);
% turn pause feature off
pause off;

DC Gain and Bandwidth of the Compensated, Closed-loop System
============================================================
DC Gain (dB) =
0
Bandwidth (rad/s) =
36.7836992162764
PID Position Control of a Spring-Mass-Damper

Published with MATLAB® 7.11.1