PI 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 proportional gain "Kp" for the closed loop system ....................... 2
Find the closed-loop transfer function and plot the step response with the selected "Kp" ......... 3
Calculate and display the closed-loop performance characteristics to Command Window ...... 4

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

Clear Variables and turn on pause feature

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

% turn on pause feature
pause on;

PI Position Control of a Spring-Mass-Damper

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 = 3.0; % zero for PI controller = Ki/Kp

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

% define the open-loop transfer function (sysOL)
numP = [1,a]; denP = conv([1,0],[m,c,kspring]); sysOL = tf(numP,denP);

% plot the root locus diagram in figure window #1
figure(1); clf; hold on;
rlocus(sysOL); grid on;
titlestringRL = ['Root Locus Diagram for Kp. (Ki/Kp = ',num2str(a),')'];
title(titlestringRL);
Find a suitable proportional gain "Kp" for the closed loop system

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

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

% display the proportional gain and associated poles in Command Window
disp('Proportional Gain = '); disp(Kp);
disp('Poles Associated with this Gain = '); disp(poles);

Select a point in the graphics window

selected_point =
    -3.43100511073254 + 8.24688796680498i

Proportional Gain =
    53.11364399916

Poles Associated with this Gain =
    -3.39775651601785 + 8.24301991649123i
    -3.39775651601785 - 8.24301991649123i
    -2.00448696796431
```
Find the closed-loop transfer function and plot the step response with the selected "Kp"

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

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

% pause to allow user to annotate the plot
pause;
```
Calculate and display the closed-loop performance characteristics to Command Window

```matlab
% calculate performance characteristics; stored in structure performanceCharacteristics
performanceCharacteristics = stepinfo(sysCL);

% display results to Command Window
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);

% turn pause feature off
pause off;
```

Performance Data for the Compensated System
==============================================
Rise Time (sec) =
0.238474323769754

Settling Time (sec) =
1.56052041877085
Percent Overshoot = 0.549646728234299

Peak Value = 1.00549646728234

Peak Time (sec) = 0.424497920812817

Published with MATLAB® 7.11.1