MATLAB Assignment DP8.1

Table of Contents

Show Bode Plots for K = 1

```
K = 1;

% a) Setup loop transfer function
numerator = [K, 2*K];
denominator = [1, 12, 0, 0];
titleText = sprintf('DP8.1 a) Bode Plot of Loop Transfer Function with
 K = %f', K);
bodePlot(numerator, denominator, titleText);

% b) Setup closed loop transfer function
numerator = [K, 2*K];
denominator = [1, 12, K, 2*K];
titleText = sprintf('DP8.1 b) Bode Plot of Closed Loop Transfer
 Function with K = %f', K);
bodePlot(numerator, denominator, titleText);
```

Show Bode Plots for K = 50

```
K = 50;

% c) Setup loop transfer function
numerator = [K, 2*K];
denominator = [1, 12, 0, 0];
titleText = sprintf('DP8.1 c) Bode Plot of Loop Transfer Function with
 K = %f', K);
bodePlot(numerator, denominator, titleText);

% c) Setup closed loop transfer function
numerator = [K, 2*K];
denominator = [1, 12, K, 2*K];
titleText = sprintf('DP8.1 c) Bode Plot of Closed Loop Transfer
 Function with K = %f', K);
bodePlot(numerator, denominator, titleText);
```

d) Find gain so that Mp <= 2 and bandwidth is the maximum attainable for the closed loop system

Find gains where Mp <= 2

```
Mp_max = 2;
KList = 0:1000;
MpList = zeros(1, numel(KList));
for i = 1:numel(KList)
    K = KList(i);
    num = [K, 2*K];
    den = [1, 12, K, 2*K];
    Mp = getPeakGain(tf(num, den));
    MpList(i) = Mp;
end

% Plot Mp vs. K
plt = figure;
plot(KList, MpList);
hold on;
grid on;
maxMpList = zeros(1, numel(KList)) + Mp_max;
plot(KList, maxMpList);
title('DP8.1 Mp vs. K');
xlabel('K');
ylabel('Mp');
uiwait(plt);

% Print range of K that are valid
validKList = [];
for i = 1:numel(KList)
    K = KList(i);
    Mp = MpList(i);
    if and(Mp <= 2, K ~= 0)
        validKList = [validKList, K];
    end
end
disp('Displaying valid values of K');
disp(validKList);

% Plot Bandwidth vs. K
bandwidthList = zeros(1, numel(validKList));
for i = 1:numel(validKList)
    K = validKList(i);
    num = [K, 2*K];
    den = [1, 12, K, 2*K];
    bandwidthList(i) = bandwidth(tf(num, den));
end
plt = figure;
plot(validKList, bandwidthList);
```

```
hold on;
grid on;
title('DP8.1 Bandwidth vs. K');
xlabel('K');
ylabel('Bandwidth');
uiwait(plt);

% Search for K with highest bandwidth
disp('Finding the K with the highest bandwidth such that Mp <= 2');
Kmin = validKList(1);
Kmax = validKList(end);
desiredBandwidth = 1000000;
[K_best, bandwidth_best] = findBest(desiredBandwidth, Kmin, Kmax,
 0.1);

for K = K_best-1:0.01:K_best+1
    num = [K, 2*K];
    den = [1, 12, K, 2*K];
    Mp = getPeakGain(tf(num, den));
    thisBandwidth = bandwidth(tf(num, den));
    if and(Mp <= 2, thisBandwidth > bandwidth_best)
        K_best = K;
        bandwidth_best = thisBandwidth;
    end
end

disp(sprintf('Best K: %f. Best Bandwidth: %f', K_best,
 bandwidth_best));

function [K_best, bandwidth_best] = findBest(desiredBandwidth, Kmin,
 Kmax, Kdiff)
    K_best = Kmin;
    bandwidth_best = 0.0000000001;
    for K = Kmin:Kdiff:Kmax
        % Calculate bandwidth with given K
        num = [K, 2*K];
        den = [1, 12, K, 2*K];
        thisBandwidth = bandwidth(tf(num, den));

        % Store K and bandwidth that is closest to desired
        if abs(bandwidth_best - desiredBandwidth) > abs(thisBandwidth
 - desiredBandwidth)
            K_best = K;
            bandwidth_best = thisBandwidth;
        end
    end
end

function bodePlot(num, den, titleText)
    plot = figure;
    bode(tf(num, den));
    title(titleText);
    uiwait(plot);
end
```

```
Index exceeds the number of array elements (42).

Error in MATLAB_Assignment_DP8_1 (line 53)
title('DP8.1 Mp vs. K');
```

Published with MATLAB® R2019a