%% ECE141 Final Design Project TEMPLATE

% Kambiz Shoarinejad, Feb 26, 2018

% Based on MIT 2.14 Design Project By:

% Darya Amin-Shahidi, May 3., 2011

% DL Trumper April, 27, 2014

%% Important Notes

% Please use this template to present your design.

% - You are required to add your code to perform the different design tasks

% and initialize the transfer functions and variable declared as [] within the

% template.

% - The template will generate the required plots based on your design.

% - The variable names used here match the diagrams and the text of the

% design problem.

% - Do not change any variable names declared here and do not overwrite

% the values given in the problem set.

%% Values and data given in the problem statement

Rc=4; %coil resistance

Lc=2e-4; %coil inductance

Kf=20; %motor constant

R1=10e3; %value of resistance R1

Rs=0.2; %value of sense resistance

Ga=-0.5; %closed loop low-freq gain of the current amplifier

G1=5e5; %capacitive probe sensor's gain

load 'testfreqdata'; % loads the experimental frequency response data on the

% mechanical plant Gp=X/F

% Make sure the data file is stored in the same

% folder

s=tf('s'); % Specify the Laplace variable s as a tf object so you can specify your transfer functions in variable s

%% Electrical Loop Design

% Initialize the following transfer functions and the component values

P\_elec = [5/(2\*10^(-4)\*s +4.2)] %electrical plant transfer function: Vs/Vc

% Designing the current controller

R2 = [5000] %value of resistor R2

R3 = [121287.86] %value of resistor R3

C1 = [1.37415\*10^(-10)] %value of capacitor C1

C2 = [1.388\*10^(-12)] %value of capacitor C2

C\_elec = [(R3\*C1\*s + 1)/(s\*R2\*(C1 + C2 + R3\*C1\*C2\*s))] %electrical controller transfer function -Vc/Vs

%specify the transfer function in terms of the resistor and

%capacitor values given above

H\_vsic = [1] % transfer function: Ic/Vs

H\_vsetvr = [-0.5] % transfer function: Vr/Vset

L\_elec = P\_elec \* C\_elec; %electrical loop transfer function

%% Plot and report Loop-shaping results for the electrical loop

% Find and print the DC gain from Vset to Ic

curr\_cl = H\_vsetvr\*feedback(L\_elec,1,-1)\*H\_vsic; % current closed loop transfer function: Ic/Vset

fprintf('DC gain from Vset to Ic: %4.2f\n',dcgain(curr\_cl));

% Plot the Bode diagrams for the plant, the compensator, and the loop transfer function for the

% electrical loop, and confirm phase margin and gain crossover frequency

figure;

bode(P\_elec);

hold on;

bode(C\_elec);

margin(L\_elec);

grid on;

legend('Plant','Compensator','Loop Transfer Function');

%% Mechanical Loop Design

% Initialize the following transfer functions

Gp = [1e-4 \* exp(-1.21e-5\*s)\*1002.509^2/(s^2 + 100.2509\*s + 1002.509^2)]% fitted mechanical plant Gp\_fit=X/F

figure;

bode(Gp);

C\_mech = [(-0.03)\*(1 + 1.5\*10^(3)/s)\*((1+ (10.06)\*(3.1528\*10^(-5))\*s)/(1+ 3.1528\*10^(-5)\*s))] % mechanical controller V\_set/V\_e

L\_mech = C\_mech\*Ga\*Kf\*Gp\*G1; % mechanical loop transfer function

%% Plot the loop-shaping results for the mechanical loop

[Pm Pp]=bode(Gp,ww); Pm=Pm(:); Pp=Pp(:);

[Cm Cp]=bode(C\_mech,ww); Cm=Cm(:); Cp=Cp(:);

[Lm Lp]=bode(L\_mech,ww); Lm=Lm(:); Lp=Lp(:);

% Plot and compare the Bode plots of the measured freq response data vs the fitted

% model

figure;

subplot(2,1,1)

semilogx(ww,20\*log10(Gpmag),'b',ww,20\*log10(Pm),'g');

grid on;

title('Plant Identification')

ylabel('Magnitude (dB)');

xlabel('\omega [rad/s]');

subplot(2,1,2)

semilogx(ww,Gpphase,'b',ww,Pp,'g');

legend('Gp','Gp\_f\_i\_t')

ylabel('Phase (deg)');

xlabel('\omega [rad/s]');

grid on;

% Plot the Bode diagrams for the compensator and the loop transfer function

figure;

subplot(2,1,1)

semilogx(ww,20\*log10(Cm),'r',ww,20\*log10(Lm),'k');

grid on;

title('Loop Shaping')

ylabel('Magnitude (dB)');

xlabel('\omega [rad/s]');

subplot(2,1,2)

semilogx(ww,Cp,'r',ww,Lp,'k');

legend('C','L')

ylabel('Phase (deg)');

xlabel('\omega [rad/s]');

grid on;

% Plot the Nyquist diagram of the loop transfer function

figure;

nyquist(L\_mech); % Nyquist Plot

L\_mechS=ss(L\_mech); % the transfer function is converted to state space

% to allow arithmetic with the delay term

% Obtain and plot the Sensitivity function, and confirm max sensitivity

S=feedback(1,L\_mechS); %sensitivity transfer function

figure;

[Sm,Sp]=bode(S);

bode(S);

title('sensitivity')

grid on;

maxS = max(Sm);

fprintf('Maximum Sensitivity (dB): %6.2f\n',20\*log10(maxS));

CL=feedback(L\_mechS,1); % closed loop x/x\_ref transfer function

%% Simulate Time Response

%% Simulate and plot the closed-loop step response and report the transient specs as well as the steady-state error

figure;

step\_amp = 1e-6; % 1um step input in position

step\_resp\_params = stepinfo(CL)

ts = step\_resp\_params.SettlingTime;

t=0:ts/100:6\*ts;

opt = stepDataOptions;

opt.StepAmplitude = step\_amp;

[x]=step(CL,t,opt);

e=x'-step\_amp; % SS tracking error

t\_indx = t<4\*ts;

plot(t(t\_indx)\*1e3,x(t\_indx)\*1e6); % plot step response

ss\_indx = t>5\*ts;

title('Step Response')

ylabel('Position [um]')

xlabel('time [msec]')

grid on;

fprintf('RMS SS step tracking error (nm): %6.4f\n',rms(e(ss\_indx))\*1e9);

%% Simulate and plot response to sinusoidal x\_ref and report the steady-state error

figure;

wr=3e3; % frequency of the reference signal [rad/s]

fr=wr/(2\*pi); % frequency of the reference signal [Hz]

t=0:(1/fr/100):(1/fr)\*100; % simulation time = 100 cycles

x\_ref=1e-5\*sin(wr\*t); % position reference signal (x\_ref) [m]

x=lsim(CL,x\_ref,t); % Simulate

e=x'-x\_ref; % tracking error

plot(t\*1e3,x\_ref\*1e6,t\*1e3,x\*1e6,t\*1e3,e\*1e6);

ss\_indx = t>0.95\*t(end);

ss\_time = 1e3\*t(ss\_indx);

xlim([ss\_time(1) ss\_time(end)]);

legend('x\_{ref}','x','e');

title('Response to Sinusoidal Reference')

ylabel('Position [um]')

xlabel('time [msec]')

grid on;

fprintf('RMS SS sinuosoidal tracking error (um): %6.4f\n',rms(e(ss\_indx))\*1e6);

%% Simulate and plot response to measurement noise and report the steady-state output due to noise

figure;

wn=1e5; % frequency of the noise [rad/s]

fn=wn/(2\*pi); % frequency of the noise signal [Hz]

t=0:(1/fn/100):(1/fn)\*100; % simulation time = 100 cycles

Vn=5e-2\*sin(wn\*t); % noise signal (x\_ref) [V]

CLn=(-L\_mechS/G1)/(1+L\_mechS); % x/Vn transfer function

e=lsim(CLn,Vn,t); % Simulate the noise response

plot(t\*1e3,Vn/G1\*1e9,t\*1e3,e\*1e9);

ss\_indx = t>0.95\*t(end);

ss\_time = 1e3\*t(ss\_indx);

xlim([ss\_time(1) ss\_time(end)]);

legend('noise','error');

title('Response to Reference')

ylabel('Position [nm]')

xlabel('time [msec]')

grid on;

fprintf('RMS SS noise error (nm): %6.4f\n',rms(e(ss\_indx))\*1e9);