% Virtual Spring Stiffness

kvc = 0.1; %[N/m] Dynamic Range 0.001 to 1 epsilon of 0.1 or 0.001

% Modulator for Virtual Spring Compression Command

ymod = 0.75e-3;%.0075; Dynamic Range of .75e-4 to .75e-2? Epsilon of 0.00001

Mar = 100\*0.075; % Effective Mass of actuator & roller in Kg 7.5e-2 to 7.5 epsilon 1e-3

% Physical Damping (reducing model jitter)

bp = 1000; % N/(m/s) = Ns/m = Kg/s

% Computed Torque Command Feedforward Gains

bp\_hat = bp; % Estimated physical damping (Same range/res as above)

Mar\_hat = Mar; % Estimated physical equivalent mass (Same range/res as above)

% Mass of the Actuator for decoupling purposes

Mar\_hat = Mar;

% Controller Active Damping calculation (N-sec/m)

% ba is depends on actual Mass estimate so ba will increase as Mass is

% increased Range for Mass will set range of ba

ba = (-(z\_1\*z\_2\*z\_3)\*Mar\_hat + Mar\_hat)/Ts;

% Controller Active Stiffness gain calculation (N/m)

% ksa depends on Mass estimate and active damping so see those bounds

% to boud ksa

ksa = (-(z\_1\*z\_2 + z\_1\*z\_3 + z\_2\*z\_3)\*Mar\_hat + 3\*Mar\_hat - 2\*Ts\*ba)/(Ts^2);

% Controller Integrated Stiffness gain calculation (N/(m-sec))

% kisa depends on Mar\_hat, ba, ksa so bound those to see this bound

kisa = (-(z\_1 + z\_2 + z\_3)\*Mar\_hat + 3\*Mar\_hat - Ts\*ba - (Ts^2)\*ksa)/(Ts^3);

>> ba

ba =

1.2450e+03

>> ksa

ksa =

5.1871e+04

>> kisa

kisa =

5.0377e+05

% Compute Back-EMF Constant

voltage\_applied = [1.5 4.5 7.5];

speed\_at\_voltage\_applied = [0.6 2.16 3.7]; % mm/sec

i = [0.006 0.0085 0.011];

Vbackemf = (voltage\_applied-(i\*Ra\_L12));

K\_backemf\_array = Vbackemf./speed\_at\_voltage\_applied; % Units ov Volts-s/mm Volts/(mm/s}

K\_backemf = 1000\*mean( K\_backemf\_array ); %[Volt/(m/s)]

K\_backemf =

1.8653e+03

% Calculate Gains for Observer

% Estimate of Mass of the Actuator for decoupling purposes

Mar\_hat; % from above

% Observer Active Stiffness gain calculation (N-sec/m)

% bo is depends on actual Mass estimate so bo will increase as Mass is

% increased Range for Mass will set range of bo (Probably same dynamic

% range as ba up in the controller)

bo = (-(z\_1\*z\_2\*z\_3)\*Mar\_hat + Mar\_hat)/Ts;

% Observer Active Stiffness gain calculation (N/m)

% kso depends on Mass estimate and active damping so see those bounds

% to bound kso (Probably same dynamic range as ba up in the controller)

kso = (-(z\_1\*z\_2 + z\_1\*z\_3 + z\_2\*z\_3)\*Mar\_hat + 3\*Mar\_hat - 2\*Ts\*bo)/(Ts^2);

% Observer Integrated Stiffness gain calculation (N/(m-sec))

% kiso depends on Mar\_hat, bo, kso so bound those to see this bound

%(Probably same dynamic range as ba up in the controller)

kiso = (-(z\_1 + z\_2 + z\_3)\*Mar\_hat + 3\*Mar\_hat - Ts\*bo - (Ts^2)\*kso)/(Ts^3);

>> bo

bo =

1.7876e+03

>> kso

kso =

1.1138e+05

>> kiso

kiso =

1.6263e+06

***Command State Filter***

%% Command State Filter

SF = 1;

if SF

%Desired Poles

f\_1 = 1; % Desired Corner Freq (Hz) of 1st pole

s\_1 = 2\*pi\*f\_1;

z\_1 = exp(-s\_1\*Ts);

f\_2 = 1; % Desired Corner Freq (Hz) of 2nd pole

s\_2 = 2\*pi\*f\_2;

z\_2 = exp(-s\_2\*Ts);

k\_2 = (2-(z\_1+z\_2))/Ts;

k\_1 = (z\_1\*z\_2-1+k\_2\*Ts)/(k\_2\*Ts^2);

% Velocity Limit

v\_lim = 0.003;

% Acceleration Limit

a\_lim = 0.008;%F\_max/Mar\_hat;

end

>> k\_1

k\_1 =

3.1317

>> k\_2

k\_2 =

12.5270

Force Ramp Rate K\_fr\*Ts/(1-z^-1) (N/sec)\*Ts

Position Ramp Rate K\_pr\*Ts/(1-z^-1) (m/sec)\*Ts Probably set somewhere in slightly less than the max velocity of the actuator

**Feedback parameters to upload**

Feed Forward Control Force Command (N) +-150(N)(backdrive force) 0.1

Feedback Control Force Command (N) +-150(N)(backdrive force) 0.1 epsilon

Net Command Force (N) +-150(N)(backdrive force) 0.1 epsilon

Position,

Velocity Estimate vicinity of +-5mm/s probabaly go factor of 2 above/below