

## BeamServoComp.m

```
% Ball & Beam System
% Lecture #16
% Servo Compensators at DC

%% System
% System setup
mball = 0.5; mbeam = 8/3;
Rball = 5e-2; L = 3;

% Obtain matrices
[A,B] = linearizedBeamBall(mball,Rball,mbeam,L);
C = [1 0 0 0]; % We are looking at the position of the ball
D = 0;

% %% Simple Compensation
% % des_poles = [-0.5+0.5244j, -0.5-0.5244j, -2.5, -3.5];
% des_poles = [-2/3, -2, -3, -4];
% [Kx,Kr] = placePoles(A,B,C,des_poles)

%% Servo-compensation
% % For constant setpoint/disturbance
% Aaug = [A B*0; C 0]; Baug = [B;0]; Caug = [C, 0];
% des_poles = [-2/3,-2,-3,-4,-5];
% [Kx_aug, Kr_aug] = placePoles(Aaug, Baug, Caug, des_poles);
% Kx = Kx_aug(1:4)
% Kz = Kx_aug(5)

% Let's have it track inputs with frequency 1rad/s
Az = [0 1; -1 0]; Bz = [1;1];

% Augmented system
Aaug = [A, zeros(4,2); Bz*C, Az];
Baug = [B; 0; 0];
Caug = [C, 0, 0];
des_poles = [-1 -2 -3 -4 -5 -6];
[Kx_aug, Kr_aug] = placePoles(Aaug,Baug,Caug, des_poles);
Kx = Kx_aug(1:4) %#ok<NOPTS>
Kz = Kx_aug(5:6) %#ok<NOPTS>

%% Check /w approximation
% AA = [A - B*Kx, -B*Kz; Bz*C, Az];
% eg1 = [eig(A); 1.2345; 1.2345];
% fprintf('[Eigenvalues of Aapprox, Eigenvalues of Servo-comp Approx System]:\n');
% [eg1, eig(AA)]

% % Check linearized response
% % Acl = [A - B*Kx, -B*Kz; C, 0];
% Acl = [A - B*Kx, -B*Kz; Bz*C, Az];
% % Bcl = [zeros(4,1); 1];
% Bcl = [zeros(4,1); -Bz];
% Bcl2 = [B;0;0];
% % Ccl1 = Caug; Dcl = D;
% % Ccl2 = [-Kx, -Kz];
% Ccl = Caug; Dcl = D;
% % G1 = ss(Acl,Bcl,Ccl1,Dcl);
% % step(G1);
% % title('Step Response of Linearized Beam&Ball'); xlabel('Time (s)');
% % hold on;
% % G2 = ss(Acl,Bcl,Ccl2,Dcl);
% % step(G2);
```

```

% % legend('r(t)', 'u(t)');
% X = zeros(6,1); t = transpose(linspace(0,10,1001));
% R = sin(t);
% Y = step3(Ac1,Bc12,Cc1,Dc1,t,X,R);
% % plot(t,Y,t,R,'LineWidth',2);
% plot(t,100*Y,t,R,'LineWidth',2);
% % legend('r(t)', 'Ref(t)');
% legend('r(t)*100', 'Ref(t)');
% ylim([-2,2]); grid on; title('Step3 Response Check');
% pause
% clf

%% Simulate
% Setting ICs and simulation config
X = zeros(4,1); % [r q dr dq]
% Z = 0;
Z = zeros(2,1);
dt = 100e-6; T_end = 15;
t = 0;
Ref = sin(t);
N = (T_end / dt) + 1;
DATA = zeros(N,4);

% Change system and see if servo-comp still manages /w disturbance
mball = 0.6;
disturb = 0;

% Simulate
i=1;
tic
while(t < T_end)
%     U = Kr*Ref - Kx*X + disturb;
%     disturb = sin(3*t);
    U = -Kz*Z - Kx*X + disturb; % Here, U is torque T
    Ref = sin(t);

    dX = BeamDynamics(X, U, mball, Rball, mbeam, L);
%     dZ = C*X - Ref;
    dZ = Bz*(C*X - Ref) + Az*Z;

    X = X + dX * dt;
    Z = Z + dZ * dt;
    t = t + dt;

    DATA(i,:) = [Ref, X(1), X(2), U];

%     if(mod(i,5) == 0)
%         BeamDisplay(X, Ref);
%     end

    i = i+1;
end
toc

t = [1:length(DATA)]' * dt; %#ok<NBRAK>
DATAds = downsample(DATA,10); tds = downsample(t,10);
plot(t,DATA, 'LineWidth',2);
grid on;
legend('Ref', 'r(t)', '\theta(t)', 'T(t)');
title('Simulated Step Response of Full Servo-Comp Ball&Beam System'); xlabel('Time (s)');

```

```

function [A,B] = linearizedBeamBall(mball, Rball, mbeam, L)
% mball=0.5kg, Rball=10e-2, mbeam=8/3kg, L=3m
g = 9.8;
Jball = (2/5)*mball*Rball^2;
Jbeam = (1/12)*mbeam*L^2;
A = [0 0 1 0; 0 0 0 1; 0 (-mball*g/(mball+Jball*Rball^2)) 0 0;
     (-mball*g/Jbeam) 0 0 0];
B = [0;0;0;1/Jbeam];
end

```

```

function [dX] = BeamDynamics( X, T, mball, Rball, mbeam, L )
% Ball and Beam: Sp21 Version
% m = 0.5 kg
% J = 2.0 kg m^2

r = X(1);
q = X(2);
dr = X(3);
dq = X(4);
g = 9.8;
% mball = 0.5; Rball = 10e-2;
% mbeam = 1; L = 4;

Jball = (2/5)*mball*Rball^2;
Jbeam = (1/12)*mbeam*L^2;

M = [(mball + Jball*Rball^2), 0;
     0, ((mball + Jball*Rball^2)*r^2 + Jbeam)];
B1 = (mball + Jball*Rball^2)*dq^2 - mball*g*sin(q);
B2 = T - 2*(mball + Jball*Rball^2)*dq*r*dr - mball*g*cos(q);

ddX = inv(M)*[B1; B2];
dX = [dr; dq; ddX];

end

```