**%% Problem 1 Design servo-comp**

**% Input System**

**A = [0 0 1 0; 0 0 0 1; 0 39.2 0 0; 0 -49 0 0];**

**B = [0;0;1;-1];**

**C = [1 0 0 0]; D = 0;**

**% Get Az, Bz**

**Az = [0 1.5 0; -1.5 0 0; 0 0 0]; Bz = ones(3,1);**

**Aaug = [A zeros(4,3); Bz\*C Az]; Baug = [B; 0;0;0]; Caug = [C 0 0 0];**

**des\_poles = [-1 -2 -3 -4 -5 -6 -7];**

**KK = placePoles(Aaug, Baug, Caug, des\_poles);**

**Kx = KK(1:4)**

**Kz = KK(5:7)**

**% Check computed gains**

**Acl = [A-B\*Kx, -B\*Kz; Bz\*C, Az]; Bcl = [0;0;0;0 ;-Bz];**

**Ccl = Caug; Dcl = D;**

**eig(Acl)**

**%% Problem 2 Linear Simulation**

**% % Plot using step3**

**% % R = sin(t), d = 0**

**% X = zeros(7,1); t = transpose(linspace(0,15,1001));**

**% R = sin(t); d = 0;**

**% BclR = Bcl;**

**% Y = step3(Acl,BclR,Ccl,Dcl,t,X,R);**

**% plot(t,Y,t,R,'LineWidth',2);**

**% legend('r(t)','Ref(t)');**

**% ylim([-2,2]); grid on; title('Step3 Response Check: R = sin(t), d = 0');**

**% pause**

**% % R = 0, d = 10**

**% clf;**

**% BclD = [B; 0\*Bz];**

**% d = 0\*t + 10;**

**% Y = step3(Acl,BclD,Ccl,Dcl,t,X,d);**

**% plot(t,Y\*100,t,d,'LineWidth',2);**

**% legend('r(t)\*100','d(t)');**

**% ylim([-1,11]); grid on; title('Step3 Response Check: R = 0, d = 10');**

**% % pause**

**% % R = sin(t), d = 10**

**% clf;**

**% BclR = Bcl; BclD = [B; 0\*Bz];**

**% BclRD = [BclR, BclD];**

**% DclRD = [0, 0];**

**% Ref = sin(1.5\*t);**

**% d = 0\*t + 10;**

**% Y = step3(Acl,BclRD,Ccl,DclRD,t,X,[Ref,d]);**

**% plot(t,Y, t,Ref, t,d, 'LineWidth',2);**

**% legend('r(t)','Ref(t)','d(t)');**

**% ylim([-2,11]); grid on; title('Step3 Response Check: R = sin(1.5\*t), d = 10');**

**% % pause**

**%% Problem 3 Nonlinear Simulation**

**% m1 = 1.0kg**

**% m2 = 4.0kg**

**% L = 1.0m**

**X = zeros(4,1); % [x q dx dq]**

**Z = zeros(3,1);**

**dt = 100e-6; T\_end = 15;**

**t = 0;**

**d = 10;**

**N = (T\_end / dt) + 1;**

**DATA = zeros(N,4);**

**i = 1;**

**tic**

**while(t < T\_end)**

**Ref = sin(1.5\*t);**

**U = -Kz\*Z - Kx\*X;**

**dX = GantryDynamics(X, U + d);**

**dZ = Bz\*(C\*X - Ref) + Az\*Z;**

**X = X + dX \* dt;**

**Z = Z + dZ \* dt;**

**t = t + dt;**

**% if(mod(i,100)==0)**

**% GantryDisplay(X, Ref);**

**% end**

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

**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);**

**% ylim([-2,11]);**

**% grid on;**

**% legend('x(t)','\theta(t)','Ref(t)','d(t)');**

**% title('Simulated Response of Servo-Comp Gantry System -- Ref = sin(1.5t), d = 10'); xlabel('Time (s)');**

**%% Problem 4 Design full-order observer**

**% Since we have input disturbance,**

**A5 = [A, B; zeros(1,5)]; B5 = [B;0]; C5 = [C 0];**

**des\_poles = [-3 -5 -5 -5 -5];**

**H = transpose(placePoles(A5',C5',C5,des\_poles))**

**% Open loop system**

**Aol = [A, zeros(4,5), zeros(4,3); H\*C, A5-H\*C5, zeros(5,3); Bz\*C, zeros(3,5), Az];**

**Bol = [B;B;zeros(3,1)];**

**Col = [C, zeros(1,5), zeros(1,3)];**

**Dol = 0;**

**%% Problem 5 Simulate**

**Kxe = [Kx, 0];**

**Acl = [A, -B\*Kxe, -B\*Kz; H\*C, A5-H\*C5-B5\*Kxe, -B5\*Kz; Bz\*C, zeros(3,5), Az];**

**Bcl = [zeros(4,1); zeros(5,1); -Bz]; Ccl = Col; Dcl = Dol;**

**% % R = sin(1.5t), d = 0**

**% X = zeros(12,1); t = transpose(linspace(0,15,1001));**

**% R = sin(1.5\*t); d = 0;**

**% BclR = Bcl;**

**% Y = step3(Acl,BclR,Ccl,Dcl,t,X,R);**

**% plot(t,Y,t,R,'LineWidth',2);**

**% legend('r(t)','Ref(t)');**

**% ylim([-2,2]); grid on; title('Step3 Response Using Full-State Observer Check: R = sin(t), d = 0');**

**% R = 0, d = 10**

**% X = zeros(12,1); t = transpose(linspace(0,15,1001));**

**% R = 0; d = 0\*t + 10;**

**% BclD = [B;B5;zeros(3,1)];**

**% Y = step3(Acl,BclD,Ccl,Dcl,t,X,d);**

**% plot(t,10\*Y,t,d,'LineWidth',2);**

**% legend('10\*r(t)','Ref(t)');**

**% ylim([-2,11]); grid on; title('Step3 Response Using Full-State Observer Check: R = 0, d = 10');**

**%% Problem 6 Nonlinear sim**

**% % First using actual states**

**% X = zeros(4,1); % [x q dx dq]**

**% Xe = zeros(5,1);**

**% Z = zeros(3,1);**

**% dt = 100e-6; T\_end = 15;**

**% t = 0;**

**% d = 10;**

**% N = (T\_end / dt) + 1;**

**% DATA = zeros(N,4);**

**%**

**% i = 1;**

**% tic**

**% while(t < T\_end)**

**%**

**% Ref = sin(1.5\*t);**

**%**

**% U = -Kz\*Z - Kx\*X;**

**%**

**% dX = GantryDynamics(X, U + d);**

**% dXe = A5\*Xe + B5\*U + H\*(C\*X - C5\*Xe);**

**% dZ = Bz\*(C\*X - Ref) + Az\*Z;**

**%**

**% X = X + dX \* dt;**

**% Z = Z + dZ \* dt;**

**% t = t + dt;**

**%**

**% % if(mod(i,100)==0)**

**% % GantryDisplay(X, Ref);**

**% % end**

**%**

**% DATA(i,:) = [X(1), X(2), Ref, d];**

**%**

**% 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);**

**% ylim([-2,11]);**

**% grid on;**

**% legend('x(t)','\theta(t)','Ref(t)','d(t)');**

**% title('Simulated Response of Servo-Comp Gantry System Using Actual States -- Ref = sin(1.5t), d = 10'); xlabel('Time (s)');**

**% Now using estimated states**

**X = zeros(4,1); % [x q dx dq]**

**Xe = zeros(5,1);**

**Z = zeros(3,1);**

**dt = 100e-6; T\_end = 15;**

**t = 0;**

**d = 10;**

**N = (T\_end / dt) + 1;**

**DATA = zeros(N,4);**

**i = 1;**

**tic**

**while(t < T\_end)**

**Ref = sin(1.5\*t);**

**% if t<1.5**

**% U = -Kz\*Z - Kx\*X;**

**% else**

**% U = -Kz\*Z - Kxe\*Xe;**

**% end**

**U = -Kz\*Z - Kxe\*Xe;**

**dX = GantryDynamics(X, U + d);**

**dXe = A5\*Xe + B5\*U + H\*(C\*X - C5\*Xe);**

**dZ = Bz\*(C\*X - Ref) + Az\*Z;**

**X = X + dX \* dt;**

**Xe = Xe + dXe \* dt;**

**Z = Z + dZ \* dt;**

**t = t + dt;**

**% if(mod(i,100)==0)**

**% GantryDisplay(X, Ref);**

**% end**

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

**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);**

**ylim([-2,11]);**

**grid on;**

**legend('x(t)','\theta(t)','Ref(t)','d(t)');**

**title('Simulated Response of Servo-Comp Gantry System Using Estimated States -- Ref = sin(1.5t), d = 10'); xlabel('Time (s)');**