**lqr\_test\_script.m**

**%% Desired response**

**des\_dom\_poles = [-2/3+0.6991j, -2/3-0.6991j];**

**num = abs(des\_dom\_poles(1))^2;**

**den = poly(des\_dom\_poles);**

**Gd = tf(num,den);**

**%% Current system**

**% Kr example**

**% [A,B] = linearizedCartPend2(1,1,1);**

**% C = [1 0 0 0;**

**% 0 1 0 0];**

**%**

**% Q = 100\*C'\*C;**

**% R = eye(2);**

**% [K,S,CLP] = lqr(A,B,Q,R)**

**% Kr = 1 ./ (-C\*inv(A-B\*K)\*B)**

**% Servo-comp example**

**% mc = 1; ml = 4; L = 1;**

**% [A,B] = linearizedCartPend(mc,ml,L);**

**mball = 0.5; Rball=1; mbeam = 4.8990; L = 2.4746;**

**[A,B] = linearizedBeamBall(mball,Rball,mbeam,L);**

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

**Aaug = [A, zeros(4,1); C, 0]; Baug = [B;0]; Caug = [C,0];**

**Cz = [0 0 0 0 1];**

**Qx = Caug'\*Caug;**

**Qz = Cz'\*Cz;**

**a = 6.0988e3; b = 6.8267e3;**

**% a = 10e3; b = 50e3;**

**Q = a\*Qx + b\*Qz**

**R = 1;**

**[K,S,CLP] = lqr(Aaug,Baug,Q,R);**

**Kx = K(1:4); Kz = K(5);**

**Acl = [A-B\*Kx, -B\*Kz; C, 0]; Bcl = [zeros(4,1); -1]; Ccl = Caug; Dcl = 0;**

**%% Test**

**G = ss(Acl, Bcl, Ccl, Dcl);**

**step(G)**

**hold on;**

**step(Gd)**

**legend('G','G\_{desired}');**

**grid on;**

**resp\_err(G,Gd,10)**

**Functions:**

**function [err] = err\_from\_Q\_a\_b(X)**

**a = X(1); b = X(2);**

**if a<0 || b<0**

**err = 1e3;**

**return**

**end**

**% mc = 1; ml = 4; L = 1;**

**% [A,B] = linearizedCartPend(mc,ml,L);**

**mball = 0.5; Rball=1; mbeam = 4.8990; L = 2.4746;**

**[A,B] = linearizedBeamBall(mball,Rball,mbeam,L);**

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

**Aaug = [A, zeros(4,1); C, 0]; Baug = [B;0]; Caug = [C,0];**

**Cz = [0 0 0 0 1];**

**Qx = Caug'\*Caug;**

**Qz = Cz'\*Cz;**

**Q = a\*Qx + b\*Qz;**

**R = 1;**

**[K,S,CLP] = lqr(Aaug,Baug,Q,R);**

**Kx = K(1:4); Kz = K(5);**

**Acl = [A-B\*Kx, -B\*Kz; C, 0]; Bcl = [zeros(4,1); -1]; Ccl = Caug; Dcl = 0;**

**G = ss(Acl, Bcl, Ccl, Dcl);**

**des\_dom\_poles = [-2/3+0.6991j, -2/3-0.6991j];**

**num = abs(des\_dom\_poles(1))^2;**

**den = poly(des\_dom\_poles);**

**Gd = tf(num,den);**

**Tend = 10;**

**err = resp\_err(G,Gd,Tend);**

**% dom\_poles = eig(G);**

**% dom\_poles = dom\_poles([1,2]);**

**% err = norm(dom\_poles - des\_dom\_poles)^2;**

**end**

**function [err] = resp\_err(G,Gd,Tend)**

**t = linspace(0,Tend,1001);**

**y = step(G,t);**

**yd = step(Gd,t);**

**err = norm(y-yd)^2;**

**end**