```matlab
%% System description

% Ball & Beam
mball = 0.5; Rball = 1; mbeam = 4.8990; L = 2.4746;
[A,B] = linearizedBeamBall(mball,Rball,mbeam,L);
C = [1 0 0 0; 0 1 0 0]; D = [0;0];
Cz = [1 0 0 0 0]; % Servo-comp will only use x(t), not q(t)

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

%% Noise character
Bnu = [0;0;0;0.4;0];
nu_mu = 0; nu_sig = 0.02;
nr_mu = 0; nr_sig = 0.01;
nq_mu = 0; nq_sig = 0.03;


%% Observer
% Check observability
% OO = [Caug; Caug*Aaug; Caug*Aaug^2; Caug*Aaug^3; Caug*Aaug^4];
% rankOO = rank(OO)
% return

% Kalman filter
F = Bnu; V = nu_sig;
Q = F*V^2*F'; R = diag([nr_sig^2, nq_sig^2]);
Q = Q + 1e-5*eye(5);    % Because Q is too small
H = transpose(lqr(Aaug',Caug',Q,R));

% Check stability
eig(Aaug - H*Caug);

% return


%% Feedback Control Law
Kx = [-49.2998, 84.6513, -30.8701, 20.5731, 0]; % 0 for the dummy disturbance
state
Kz = -20.5669;

%% Linear simulation
% Afull = [Aaug, -Baug*Kx, zeros(5,2); H*Caug, Aaug-H*Caug, zeros(5,2);
zeros(2,5), Caug, zeros(2,2)];
% Bfull = [Baug, Bnu, zeros(5,4); Baug, zeros(5,1), H, zeros(5,2);
zeros(2,4), diag([-1,-1])];
Afull = [Aaug, -Baug*Kx, -Baug*Kz; H*Caug, Aaug-Baug*Kx-H*Caug, -Baug*Kz;
zeros(1,5), Cz 0];
Bfull = [Bnu, zeros(5,3); zeros(5,1), H, zeros(5,1); zeros(1,3), -1];
Cfull = [1, zeros(1,10); ...
         0,1,zeros(1,9); ...
         zeros(1,5),1,0,0,0,0,0; ...
         zeros(1,6),1,0,0,0,0; ...
```

```
         zeros(1,4),1,zeros(1,6); ...
         zeros(1,9),1,0];
Dfull = zeros(6,4);
% return
% Dfull(1,3) = 1; Dfull(2,4) = 1; % To see measured outputs instead...

d = 0.1;    % 0.1 rad offset for angle measurement
% X0 = zeros(11,1);
X0 = zeros(11,1); X0(5) = d;
Tend = 60; t = transpose(linspace(0,Tend,10001)); N = size(t);
% U = [zeros(N), zeros(N), zeros(N), ones(N)];
U = [normrnd(nu_mu,nu_sig,N), normrnd(nr_mu,nr_sig,N),
normrnd(nq_mu,nq_sig,N), ones(N)];
Y = step3(Afull, Bfull, Cfull, Dfull, t, X0, U);

subplot(1,2,1);
plot(t,Y(:,[1:4]));
legend('x(t)','\theta(t)','x_e(t)','\theta_e(t)');
title('States');

subplot(1,2,2);
plot(t,Y(:,[5:6]));
legend('Angle Offset','Observer Estimate of Angle Offset');
title('Accounting for Angle Offset');
```

```
% Ball & Beam System

% System
mball = 0.5; Rball = 1; mbeam = 4.8990; L = 2.4746;
[A,B] = linearizedBeamBall(mball,Rball,mbeam,L);
C = [1 0 0 0; 0 1 0 0];
Aaug = [A, zeros(4,1); zeros(1,4), 0]; Baug = [B;0];
Caug = [C, zeros(2,1)];

% Feedback gains
Kx = [-49.2998   84.6513  -30.8701   20.5731];
Kz = -20.5669;

% Observer gains
H = [4.2513,-0.2425; -2.3155,0.2082; 9.2513,-0.6711; -3.9025,0.2997;
0.1328,0.0957];

% Noise
nu_mu = 0; nu_sig = 0.02;
nr_mu = 0; nr_sig = 0.01;
nq_mu = 0; nq_sig = 0.03;

% Initial conditions
X = zeros(5,1); Xe = zeros(5,1); Z = 0;
d = 0.1;
dX = zeros(5,1); dXe = zeros(5,1);
X(5) = d; % Disturbance

% Simulation setup
Ref = 1;
t = 0; dt = 100e-6; Tend = 60;
N = (Tend / dt) + 1;
DATA = zeros(N,7); % x, th, xe, the, T, d, de

i=1;
tic
while(t < Tend)
    if t < 55
%         U = -Kz*Z - Kx*X([1:4]);
        U = -Kz*Z - Kx*X([1:4]) + normrnd(nu_mu,nu_sig);
        dZ = X(1) + normrnd(nr_mu,nr_sig) - Ref;
    else
%         U = -Kz*Z - Kx*Xe([1:4]);
        U = -Kz*Z - Kx*Xe([1:4]) + normrnd(nu_mu,nu_sig);
%         dZ = Xe(1) - Ref;
        dZ = Xe(1) + normrnd(nr_mu,nr_sig) - Ref;
    end

    dX([1:4]) = BeamDynamics(X([1:4]), U, mball, Rball, mbeam, L);
    dXe = Aaug*Xe + Baug*U + H*Caug*(X +
[normrnd(nr_mu,nr_sig);normrnd(nq_mu,nq_sig);0;0;0] - Xe);

    X = X + dX * dt;
    Xe = Xe + dXe * dt;
    Z = Z + dZ * dt;
```

```
    t = t + dt;

    DATA(i,:) = [X(1), X(2), Xe(1), Xe(2), U, X(5), Xe(5)];
    i = i+1;

end
toc

kk = 1e3;
t = [1:length(DATA)]' * dt;
DATAds = downsample(DATA,kk);
tds = downsample(t,kk);

subplot(1,2,1);
plot(tds,DATAds(:,[1:4]));
grid on;
legend('r(t)','\theta(t)','r_e(t)','\theta_e(t)');
title('Simulated Step Response of Beam-Ball System');

subplot(1,2,2);
plot(tds,DATAds(:,5), tds,DATAds(:,6));
grid on;
legend('d_{est}','d');
title('Disturbance vs Disturbance Estimate');
```