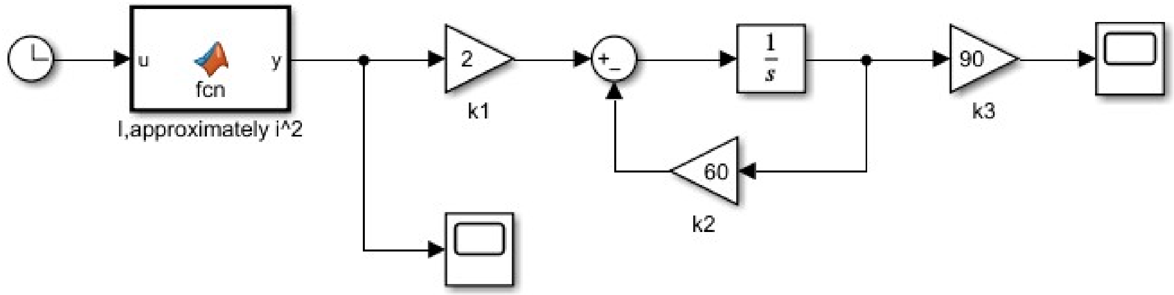
Simulink model for the take-home final exam, question 3

Sub-question (1):



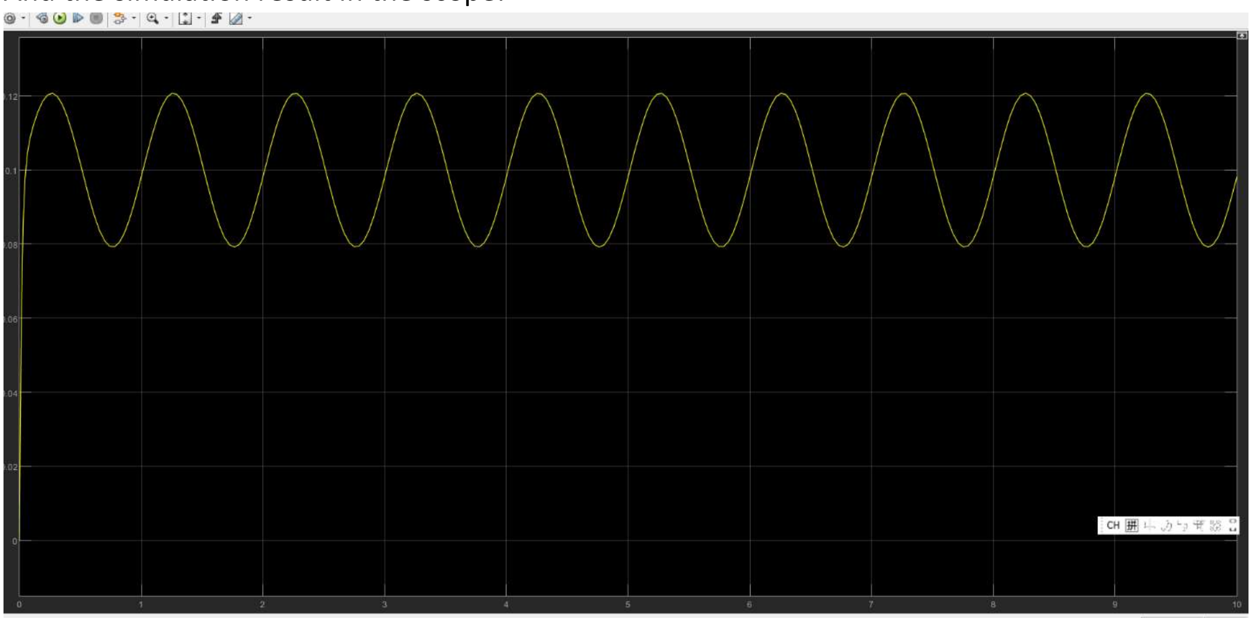
function y = fcn(u)

y = 1/450*(0.0513*cos(2*pi*u)+pi*sin(2*pi*u)+15);

end

The ratio of k2/k3 should be around 2:3, and k2/k1 should be no smaller than 10.

And the simulation result in the scope:



The rate of convergence seems fast.

Question 3, subproblem 3:

Parameters:

A = [0 1;-60 -200];

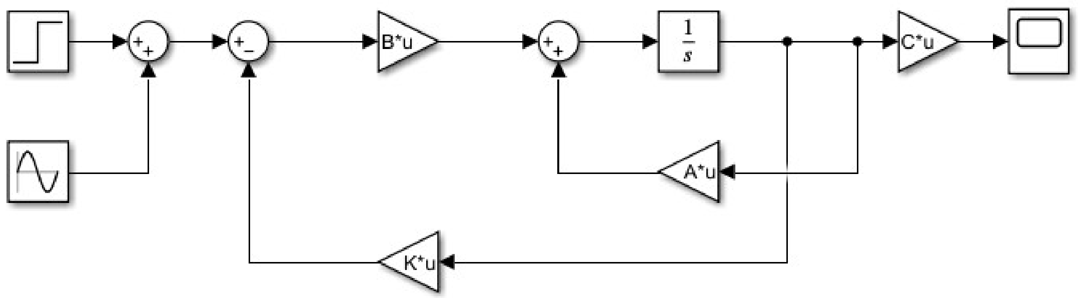
B=[0; 8];

C=[1 0];

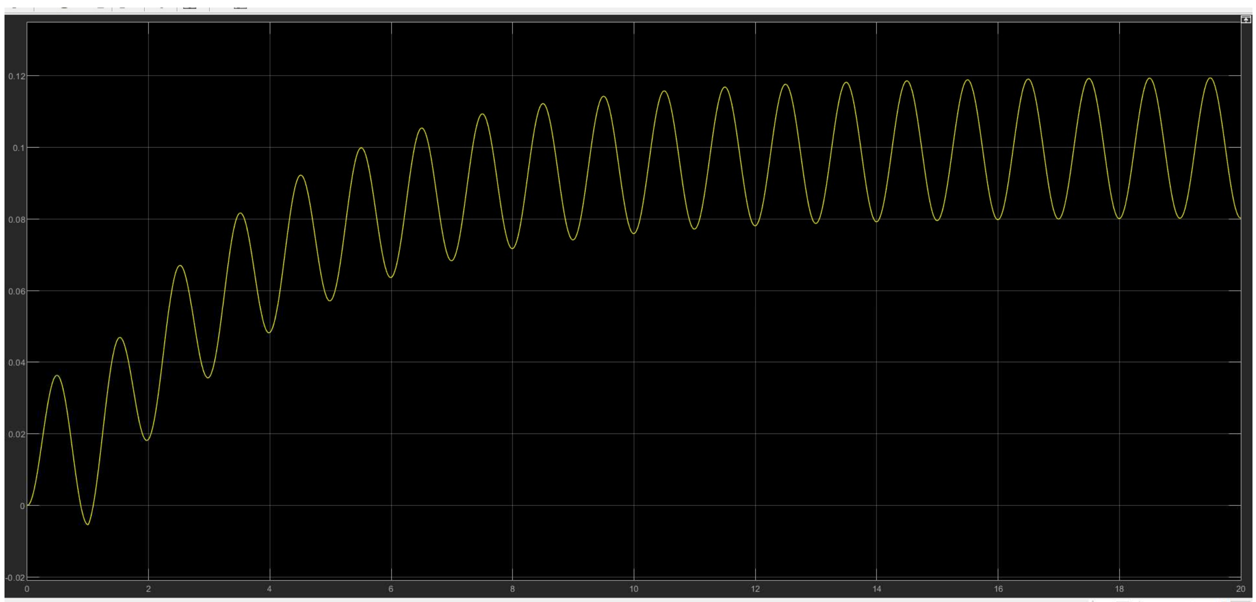
K=[1 1];

Input: step = 0.85, Amplitude of sin = 3.2

Simulink model:



And inside the scope:



State x reaches the designed xdes a bit slower than that in subproblem 1. The system should be asymptotically stable.

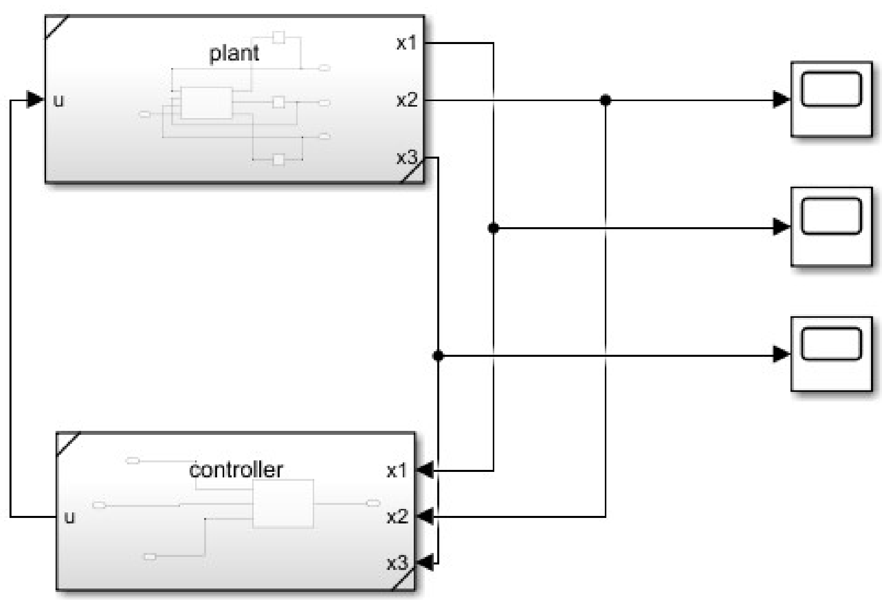
Matlab code for the final project

```
function feedbackLinearize()
numpoints = 10000;
a = 10;
b = 28;
c = 8/3;
tf = 100;
gamma = [-1 -3 -3];
ttot = linspace(0, tf, numpoints)';
nt = numel(ttot);
ic = [10 10 30 0];
odeopts = odeset('RelTol', 1e-9);
[t x] = ode45(@wings, ttot, ic, odeopts);
```

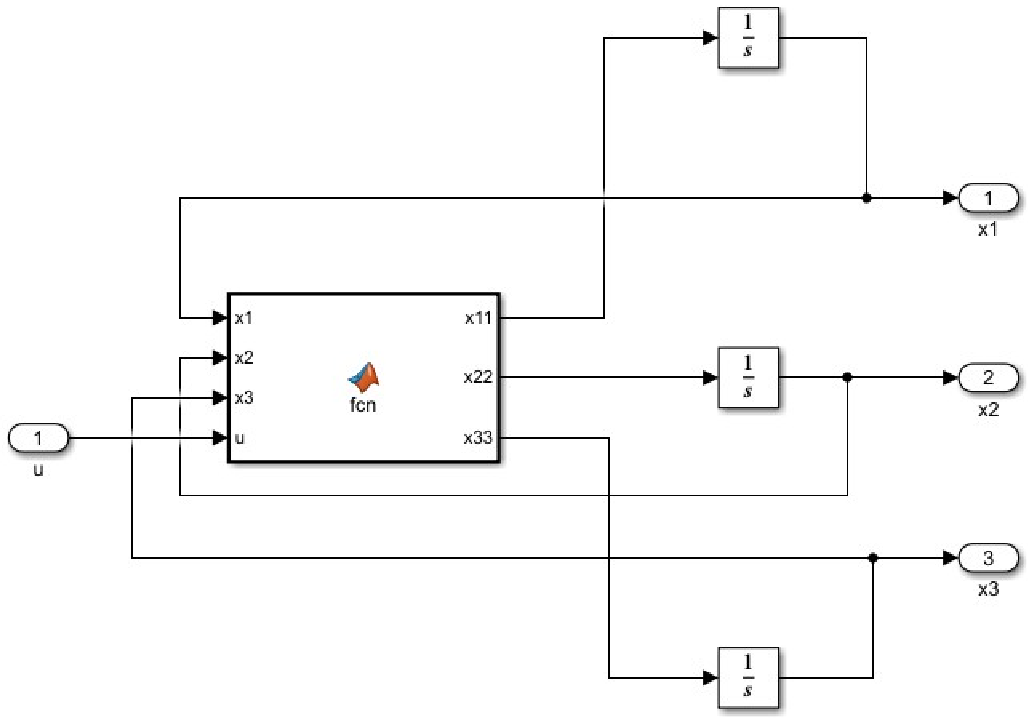
```
figure(1)
subplot(2,2,1);
plot3(x(:,1), x(:,2), x(:,3));
xlabel('x');
ylabel('y');
zlabel('z');
grid on
subplot(2,2,2);
plot(t, x(:,1));
xlabel('t');
ylabel('x');
subplot(2,2,3)
plot(t, x(:,2));
xlabel('t')
ylabel('y')
subplot(2,2,4)
plot(t, x(:,3));
xlabel('t');
ylabel('z');
saveas(gca, 'all.png')

figure(2)
plot(t(1:(end-1)), diff(x(:,4))./diff(t));
xlabel('t');
ylabel('u');
saveas(gca, 'control.png');

function dx=wings(t,x)
   dx=zeros(4,1);
   u=zeros(3,1);
   F = [a*(x(2)-x(1)); b*x(1)-x(2)-x(1)*x(3); x(1)*x(2)-c*x(3)];
   if (t>20)
      f = 2*pi;
      ampl = 50;
      x1d = ampl*sin(f*t);
      x2d = ampl*f*cos(f*t);
      x3d = -ampl*f^2*sin(f*t);
      x4d = -ampl*f^3*cos(f*t);
      xd = [x1d; x2d; x3d];
      d = (2*a-c)/(c*a);
      miu = d*x1d^2;

      z(1,1) = 1/a*x(1)^2-2*x(3)+miu;
      z(2,1) = -2*x(1)^2+2*c*x(3);
```

```
        z(3,1) = -4*a*x(1)*(x(2)-x(1))+2*c*(x(1)*x(2)-c*x(3));
        v = gamma*z;
        const = c-2*a;
        D3Y = [4*a*(x(2)-2*x(1))+2*c*x(2) -4*a*x(2)+2*c*x(1) -2*c^2]
        numer = D3Y*F;
        denom = x(1)^2*(-4*a+2*c);
        u(2) = 1/denom*(v-numer);
    else
        u = zeros(3,1);
    end
    dx = zeros(3,1);
    dx(1:3) = F+u;
    dx(4) = u(2);
end
end
```

Simulink Model (this is to test if the feedback system works)



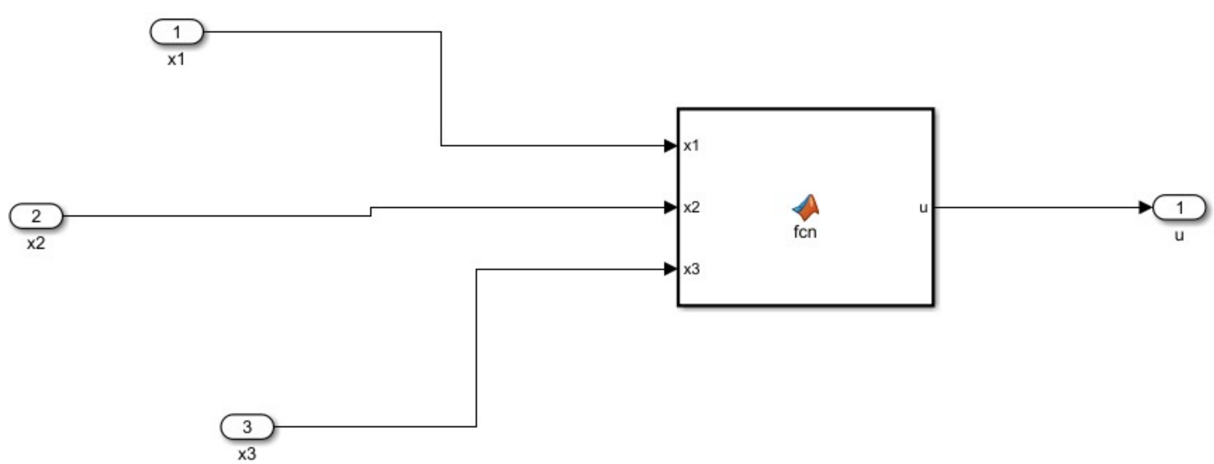
Inside the plant subsystem:



The initial condition for integrator is set to be [8 8 27] for x1, x2, x3.
Code inside the plant block:

```
function [x11, x22, x33] = fcn(x1, x2, x3, u)
x11 = 10*(x2-x1);
x22 = 28*x1-x2-x1*x3-x2*u;
x33 = -8/3*x3+x1*x2;
end
```

Inside the controller subsystem:

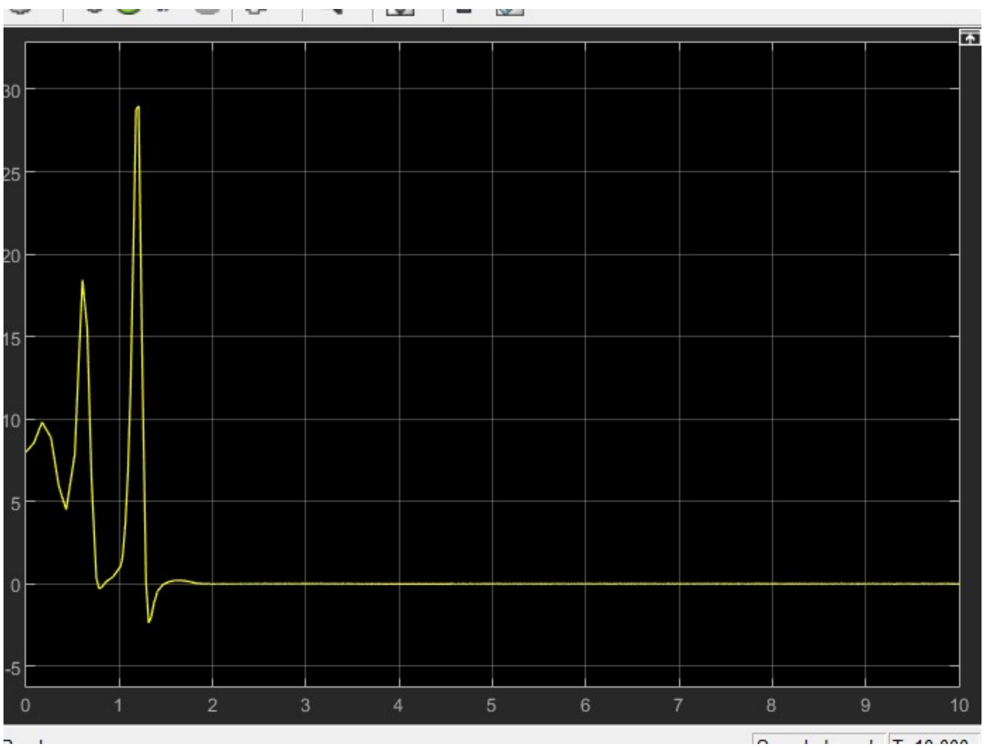


And the fcn inside the block:

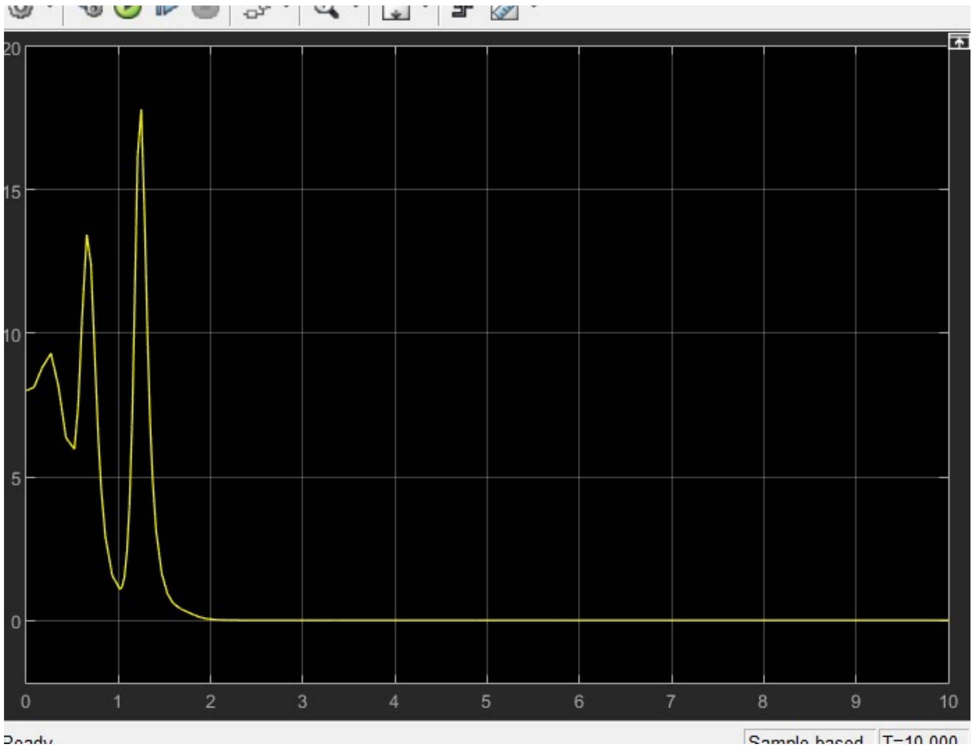
```
function u = fcn(x1,x2,x3)
```

```
miu = 8.5;
gamma = [-1 -3 -3];
z = [1/10*x1^2-2*x3+miu; -2*x1^2+16/3*x3; -40*x1*(x2-x1)+16/3*(x1*x2-8/3*x3)];
v = gamma*z;
F = [10*(x2-x1); 28*x1-x2-x1*x3; x1*x2-8/3*x3];
D3Y = [40*(x2-2*x1)+16/3*x2 -40*x2+16/3*x1 -2*64/9];
numer = D3Y*F;
denom = x1^2*(-40+16/3);
u = 1/denom*(v-numer);
end
```

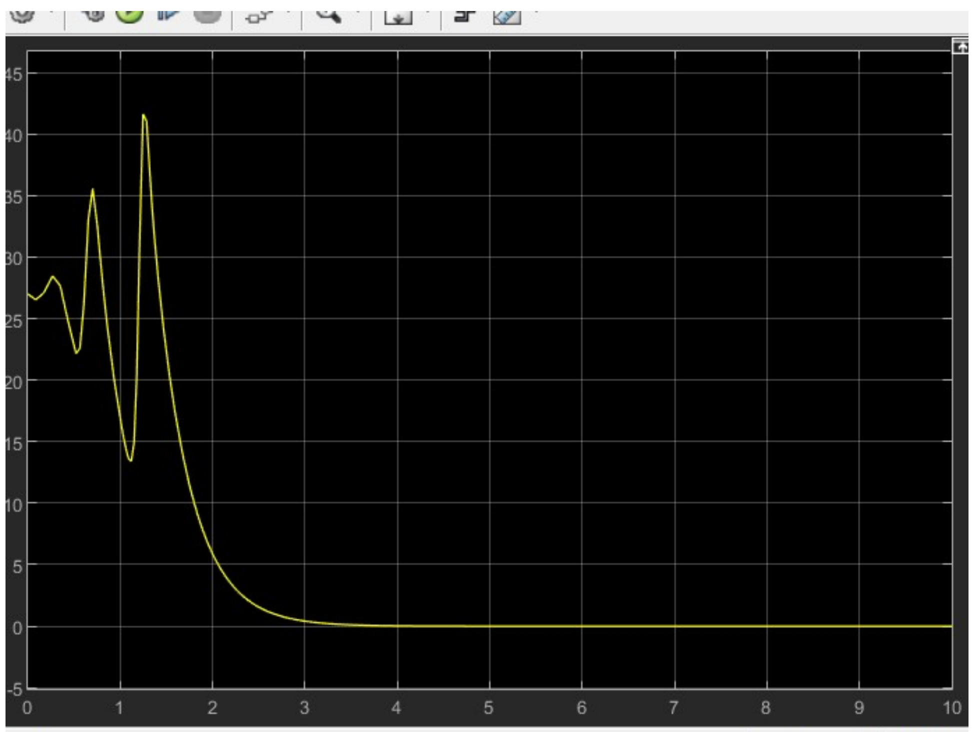
The output of such a system is stabilized. Inside the scope for x1, x2, x3 seperately:



X1 vs. time



X2 vs. time



X3 vs. time