```
clear; close all;
%% Get data
% open_system('Q2_FCC_Block_Diagram');
out = sim('Q2_FCC_Block_Diagram');
% load('out_original.mat');
tout = squeeze(out.tout);
Y1 = squeeze(out.Y1.signals.values);
Y1 = rescale(Y1);
plot(Y1);
plot(tout,Y1,'LineWidth',2);
hold on;
lol = smoothdata(Y1);
lol(1:13300)=Y1(1:13300);
Y1 = lol;
plot(tout,Y1,'LineWidth',2);
title('Y_1 original'); xlabel('time'); ylabel('Y_1');
hold off;
Y2 = squeeze(out.Y2.signals.values);
Y2 = rescale(Y2);
% lol = smoothdata(Y2);
% lol(1:31868)=Y2(1:31868);
% Y2 = lol;
figure;
plot(tout,Y2,'LineWidth',2);
title('Y_2 original'); xlabel('time'); ylabel('Y_2');
Y3 = squeeze(out.Y3.signals.values);
Y3 = rescale(Y3);
% lol = smoothdata(Y3);
% lol(1:14743)=Y3(1:14743);
% Y3 = lol;
figure;
plot(tout,Y3,'LineWidth',2);
title('Y_3 original'); xlabel('time'); ylabel('Y_3');
U1 = squeeze(out.U1.signals.values);
U2 = squeeze(out.U2.signals.values);
% For SIMULINK
U1_struct = struct();
U1_struct.time = out.tout;
U1_struct.signals = out.U1.signals;

U2_struct = U1_struct;
U2_struct.signals = out.U2.signals;

Ts = tout(2) - tout(1);
```

```
% Use iddata
Y = [Y1 Y2 Y3];
U = [U1 U2];
data = iddata(Y,U,Ts);
U1_struct.signals.values = data.InputData(:,1);
U2_struct.signals.values = data.InputData(:,2);
%% Part a): n4sid, estimate state space
% number of states as given
Nx = 5;
Nu = 2;
Ny = 3;
% Estimate a continous state space model
[sys,x0] =
n4sid(data,Nx,'DisturbanceModel','none','Feedthrough',true(
1,Nu),'Ts',0);
%% Part b): Check for OL stability
% Finding poles
poles = eig(sys.A)
% We realise that all poles are within the unit disk and 1
is at zero so
% marginally stable
% confirm it MATLAB
stability = isstable(sys)
%% Intermission: Need to account for initial state
Bnew = zeros(Nx,Nu+Nx);
Bnew(:,1:2) = sys.B;
Bnew(:,3:end) = sys.A;
Dnew = zeros(Ny,Nx+Nu);
Dnew(:,1:2) = sys.D;
Dnew(:,3:end) = sys.C;
ss_again = ss(sys.A,Bnew,sys.C,Dnew);
tfnew = tf(ss_again);
%% Part c) Obtain transfer functions
tfs = tf(sys);
G11 = minreal(tf(tfs.NUM(1,1),tfs.DEN(1,1)));
G12 = minreal(tf(tfs.NUM(2,1),tfs.DEN(2,1)));
G13 = minreal(tf(tfs.NUM(3,1),tfs.DEN(3,1)));
G21 = minreal(tf(tfs.NUM(1,2),tfs.DEN(1,2)));
G22 = minreal(tf(tfs.NUM(2,2),tfs.DEN(2,2)));
G23 = minreal(tf(tfs.NUM(3,2),tfs.DEN(3,2)));
% open_system('q2_part_c');
figure;
compare(data,sys,5e4);
```

```
close all;
%% Initialisation
Kc = 0.0669137468092512;
KI = 5.31692810242195e-08;
Gm = G11;
Gc = tf([Kc KI],[1 0]);
s = tf('s');
%% RH criterion
Dr = 1 + G11*Gc;
[num,~] = tfdata(Dr,'v');
%% Root locus
L = minreal(Gm/(s*(1+Kc*Gm)));
rlocusplot(L,KI);
grid on;
figure;
rlocusplot(L)
%rltool(L);
%% Bode plot
% CL_sys = minreal((Gm*Gc)/(1+Gm*Gc));
Lg = Gc*Gm;
figure;
margin(Lg);
%% Nyquist Plot
figure;
nyquist(Lg);
hold on;
n = 500;
theta = linspace(0,2*pi,n);
x = cos(theta); y = sin(theta);
plot(x,y,'r-.');
hold off;
legend('Nyquist','Unit Circle');
```