```
%clear
```

```
LF = RBS_out.Left_Flowrate
RF = RBS_out.Right_Flowrate
```

```
LLL = RBS_out.Lower_Left_Level
TLL = RBS_out.Top_Left_Level
LRL = RBS_out.Lower_Right_Level
TRL = RBS_out.Top_Right_Level
```

```
figure
plot(LF)
ylabel('u_1')
xlabel('Time(s)')
title('Left Pump flow rate')
figure (2)
plot(RF)
ylabel('u_2')
xlabel('Time(s)')
title('Right Pump flow rate')
%%
figure (3)
plot(LLL)
ylabel('y_1')
xlabel('Time(s)')
title('Bottom Left Tank Level')
%%
figure (4)
plot(LRL)
ylabel('y_2')
xlabel('Time(s)')
title('Bottom Right Tank Level')
ylim([0.08 0.21])
%%
figure (5)
plot(TLL)
ylabel('y_1')
xlabel('Time(s)')
title('Top Left Tank Level')
%%
figure (6)
plot(TRL)
ylabel('y_2')
xlabel('Time(s)')
title('Top Right Tank Level')
```

```
figure(7)
```

```
plot(RBS_R(:,2))
title('RBS Right pump')
figure(8)
plot(RBS_L(:,2))
title('RBS left pump')
```

```
y1 = LLL
y2 = LRL
u1 = LF
u2 = RF
%Ts = 14  Sample time used in RBS
z_L = iddata(y1,[u1 u2],Ts)
z_R = iddata(y2,[u1 u2],Ts)
```

```
nk1 = delayest(z_L,3,3,1,100)
nk2 = delayest(z_R,3,3,1,100)
```

```
% ARX approxiamtion for outlier detection
ARX1 = arx(z_L,[3 [3 3], nk1])
ARX2 = arx(z_R,[3 [3 3] nk2])

% Defining residual
R1 = predict(ARX1,z_L,1)
es_1 = (y1 - R1.y)
R2 = predict(ARX2,z_R,1)
es_2 = (y2 - R2.y)
N_1 = length(y1); % No. of data
P = 9; % No. of parameters
MSE_1 = es_1'*es_1/(N_1-P)
MSE_2 = es_2'*es_2/(N_1-P)
d1 = es_1./sqrt(MSE_1)
d2 = es_2./sqrt(MSE_2)
```

```
figure
plot(d1,'o')
yline(3,'--')
yline(-3,'--')
title('Outlier from Z_L')
figure(2)
plot(d2,'o')
yline(3,'--')
yline(-3,'--')
title('Outlier from Z_R')
% chnage the outlier with the predicted value
```

```
outliers_y1 = [];
outliers_y2 = [];

for i = 1:length(y1)
    if d1(i) > 3 || d1(i) < -3
        outliers_y1 = [outliers_y1; i, y1(i)];
    end
    if d2(i) > 3 || d2(i) < -3
        outliers_y2 = [outliers_y2; i, y2(i)];
    end
end

disp('Outliers in y1:')
disp(outliers_y1) %7
disp('Outliers in y2:')
disp(outliers_y2) %16
```

```
z_L = iddata(y1,[u1 u2],Ts)
z_R = iddata(y2,[u1 u2],Ts)

delayest(z_L)
delayest(z_R)
```

```
ZT_1 = z_L(1:ceil(0.7*length(y1))); % Extract identification data 1st output
ZT_2 = z_R(1:ceil(0.7*length(y2))); % Extract identification data 2nd output
ZV_1 = z_L(ceil(0.7*length(y1))+1:end); % Valid identification data 1st
output
ZV_2 = z_R(ceil(0.7*length(y2))+1:end); % Valid identification data 1st
output
```

```
ZT_1 = dtrend(ZT_1); % Final testing data for y1
ZT_2 = dtrend(ZT_2); % Final testing data for y2
ZV_1 = dtrend(ZV_1); % Final validation data for y1
ZV_2 = dtrend(ZV_2); % Final validation data for y2
```

```
nk1_Tf = delayest(ZT_1,3,3,1,100)
nk2_Tf = delayest(ZT_2,3,3,1,100)
z_11 = iddata (y1, u1, Ts)
z_12 = iddata (y1, u2, Ts)
z_13 = iddata (y2, u1, Ts)
z_14 = iddata (y2, u2, Ts)
```

```
figure
cra (z_11)
title('Impulse responce of Y1 and U1')
```

```
figure
cra (z_12)
title('Impulse responce of Y1 and U2')

figure
cra (z_13)
title('Impulse responce of Y2 and U1')

figure
cra (z_14)
title('Impulse responce of Y2 and U2')
```

```
clc
bode_11 = spa (z_L)
bode_12 = spa (z_R)
```

```
bode(bode_11,bode_12)
title ('Bode Diagram of model Z-L and Z-R ')
legend('model Z - L','model Z-R')
```

```
nk1_Tf = delayest(ZT_1,3,3,1,100)
nk2_Tf = delayest(ZT_2,3,3,1,100)
```

```
% z_f = iddata ([y1 y2], [u1 u2], Ts)
```

```
%% Model Identification
%%% ARX %%%

ARX_model1= arx(ZT_1,[1 [4 3] nk1_Tf]);
resid(ZV_1,ARX_model1)
compare(ZV_1,ARX_model1)
compare(ZT_1, ARX_model1)
```

```
arx_model2 = arx(ZT_2,[2 [3 4] nk2_Tf]); % 2 3 6
resid(ZV_2,arx_model2)
compare (ZT_2, arx_model2)
```

%%% ARMAX %%%

```
Armax_model1 = armax(ZT_1,[2 [3 3] 1 nk1_Tf]); %2 [2 2] 2
resid(ZV_1,Armax_model1)
compare(ZV_1,Armax_model1)
```

```
Armax_model2 = armax(ZT_2,[2 [3 3] 1 nk2_Tf]); % 2 [3 3] 2
resid(ZV_2,Armax_model2)
compare(ZV_2,Armax_model2)
```

%%% OE %%%

```
OE_model1 = oe(ZT_1,[[2 2] [1 1] nk1_Tf]);
resid(ZV_1,OE_model1)
compare(ZV_1,OE_model1)
```

```
OE_model2 = oe(ZT_2,[[1 1] [1 1] nk2_Tf]);
resid(ZV_2,OE_model2)
compare(ZV_2,OE_model2)
```

%%% BJ %%%

```
BJ_model1 = bj(ZT_1,[[4 2] 2 2 [2 2] nk1_Tf]); %[4 2] 2 2 [4 2]
resid(ZV_1,BJ_model1)
compare(ZV_1,BJ_model1)
```

```
BJ_model2 = bj(ZT_2,[[2 3] 1 1 [2 2] nk2_Tf]); % [1 1] 1 1 [1 1]
resid(ZV_2,BJ_model2)
compare(ZV_2,BJ_model2)
% Uinput = [u1,u2];
```

%%% non-linear ARX %%%

```
Non_linear_model1 = nlarx(ZT_1,[3 [2 2] nk1_Tf]);
resid(ZV_1,Non_linear_model1)
compare(ZV_1,Non_linear_model1)
```

```
figure
Non_linear_model2 = nlarx(ZT_2,[3 [2 2] nk2_Tf]);
resid(ZV_2,Non_linear_model2)
compare(ZV_2,Non_linear_model2)
```

```
Non_linear_model1
```

```
%%% subspace %%%
Subspace_model1=n4sid(ZT_1,4);
Sunspace_model2=n4sid(ZT_2,4);
resid(ZV_1,Subspace_model1)
resid(ZV_2,Sunspace_model2)
compare(ZV_1,Subspace_model1)
figure
```

```
compare(ZV_2,Sunspace_model2)
```

```
Subspace_model1
Sunspace_model2
```

```
figure
compare
(ZV_1,ARX_model1,Armax_model1,BJ_model1,OE_model1,Non_linear_model1,Subspace_
model1)
title ('Model Comparision for Left tank ')
```

```
figure
compare
(ZV_2,arx_model2,Armax_model2,BJ_model2,OE_model2,Non_linear_model2,Sunspace_
model2)
title ('Model Comparision for Right tank')
```

```
bode (bode_11, ARX_model1,Armax_model1,BJ_model1,OE_model1,Subspace_model1)
legend ('left tank ', 'ARX model1','Armax model1','BJ model1','OE
model1','Subspace model1')
```

```
bode (bode_12, arx_model2,Armax_model2,BJ_model2,OE_model2,Sunspace_model2)
legend ('Right tank ', 'ARX model2','Armax model2','BJ model2','OE
model2','Subspace model2')
```

```
AIC_arx1 = aic(ARX_model1)
Aic_Armax1 =  aic(Armax_model1)
Aic_BJ1  = aic(BJ_model1)
Aic_OE1 = aic(OE_model1)
Aic_nonlinear1 = aic(Non_linear_model1)
Aic_SubSpace1 = aic (Subspace_model1)
```

```
aic ( arx_model2,Armax_model2,BJ_model2,OE_model2,Non_linear_model1,
Sunspace_model2)
%legend ('Right tank ', 'ARX model2','Armax model2','BJ model2','OE
model2','Subspace model2')
```

```
ARX_model1
Armax_model1
BJ_model1
OE_model1
Non_linear_model1
Subspace_model1
```

```
arx_model2
Armax_model2
BJ_model2
OE_model2
Non_linear_model2
Sunspace_model2
```