**ภาคผนวก ข**

โปรแกรม MATLAB

**การอ่านข้อมูล spreadsheet ของ LabVIEW เข้าสู่ workspace ของ MATLAB**

เนื่องจาก spreadsheet ของ LabVIEW ไม่สามารถอ่านโดยคำสั่ง xlsread ได้โดยตรง จึงจำเป็นต้องใช้ Microsoft Excel ในการแปลงไฟล์ให้เป็น Excel 2007 (.xlsx) ก่อนจึงสามารถใช้คำสั่ง xlsread ในการรับค่าข้อมูลได้

```
%Save data from excel to matlab workspace

time=xlsread('(ชื่อไฟล์).xlsx',1,'A1:A14002');
PV=xlsread('(ชื่อไฟล์).xlsx',1,'B1:B14002');
SP=xlsread(ชื่อไฟล์).xlsx',1,'C1:C14002');
Flow=xlsread('(ชื่อไฟล์).xlsx',1,'E1:E14002');
Current=xlsread('(ชื่อไฟล์).xlsx',1,'F1:F14002');

Current_MA=filter(ones(150,1)/150,1,Current);
PV_MA=filter(ones(150,1)/150,1,PV);
SP_MA=filter(ones(150,1)/150,1,SP);
Flow_MA=filter(ones(150,1)/150,1,Flow);
```

**การสร้างสัญญาณตกค้าง**

```
% clear,clc
% load data

j=0;
while(j<170)
i=0;
j=j+1;

if(j==1)
while(i<1500)
    i=i+1;

%initial approximate output
    if(i<9)
      y_MPE{1,j}(i,1)=yprbs{1,j}(i,1);
    end

%find approximate output
    if(i>8)
      y_MPE{1,j}(i,1)=0.00066208*(uprbs{1,j}(i-3,1)^2) +
(1.1092*y_MPE{1,j}(i-2,1)) - ((0.015799)*y_MPE{1,j}(i-
2,1)*uprbs{1,j}(i-3,1)) + (0.0011043*uprbs{1,j}(i-8,1)^2) -
(0.15999*y_MPE{1,j}(i-4,1)) - (0.038492*uprbs{1,j}(i-8,1)) -
(0.00085339*uprbs{1,j}(i-3,1)*uprbs{1,j}(i-8,1)) +
(0.015954*y_MPE{1,j}(i-1,1)*uprbs{1,j}(i-3,1));
    end
end
end

if(j>1)
while(i<460)
    i=i+1;

%initial approximate output
    if(i<9)
      y_MPE{1,j}(i,1)=yprbs{1,j}(i,1);
    end

%find approximate output
    if(i>8)
      y_MPE{1,j}(i,1)=0.00066208*(uprbs{1,j}(i-3,1)^2) +
(1.1092*y_MPE{1,j}(i-2,1)) - ((0.015799)*y_MPE{1,j}(i-
2,1)*uprbs{1,j}(i-3,1)) + (0.0011043*uprbs{1,j}(i-8,1)^2) -
(0.15999*y_MPE{1,j}(i-4,1)) - (0.038492*uprbs{1,j}(i-8,1)) -
(0.00085339*uprbs{1,j}(i-3,1)*uprbs{1,j}(i-8,1)) +
(0.015954*y_MPE{1,j}(i-1,1)*uprbs{1,j}(i-3,1));
    end
end
end
r_MPE{1,j}=(y_MPE{1,j})-(yprbs{1,j});
end
```

```
j=0;
while(j<170)
i=0;
j=j+1;

if(j==1)
while(i<1500)
    i=i+1;

%initial approximate output
    if(i<9)
      y_OSA{1,j}(i,1)=yprbs{1,j}(i,1);
    end

%find approximate output
    if(i>8)
      y_OSA{1,j}(i,1)=0.00066208*(uprbs{1,j}(i-3,1)^2) +
(1.1092*yprbs{1,j}(i-2,1)) - ((0.015799)*yprbs{1,j}(i-
2,1)*uprbs{1,j}(i-3,1)) + (0.0011043*uprbs{1,j}(i-8,1)^2) -
(0.15999*yprbs{1,j}(i-4,1)) - (0.038492*uprbs{1,j}(i-8,1)) -
(0.00085339*uprbs{1,j}(i-3,1)*uprbs{1,j}(i-8,1)) +
(0.015954*yprbs{1,j}(i-1,1)*uprbs{1,j}(i-3,1));
    end
end
end

if(j>1)
while(i<460)
    i=i+1;

%initial approximate output
    if(i<9)
      y_OSA{1,j}(i,1)=yprbs{1,j}(i,1);
    end

%find approximate output
    if(i>8)
      y_OSA{1,j}(i,1)=0.00066208*(uprbs{1,j}(i-3,1)^2) +
(1.1092*yprbs{1,j}(i-2,1)) - ((0.015799)*yprbs{1,j}(i-
2,1)*uprbs{1,j}(i-3,1)) + (0.0011043*uprbs{1,j}(i-8,1)^2) -
(0.15999*yprbs{1,j}(i-4,1)) - (0.038492*uprbs{1,j}(i-8,1)) -
(0.00085339*uprbs{1,j}(i-3,1)*uprbs{1,j}(i-8,1)) +
(0.015954*yprbs{1,j}(i-1,1)*uprbs{1,j}(i-3,1));
    end
end
end
r_OSA{1,j}=(y_OSA{1,j})-(yprbs{1,j});
end
```

**ประมาณค่าเฉลี่ยและค่าความแปรปรวนของสัญญาณตกค้างในแต่ละชุดสัญญาณทดสอบแบบ PRBS**

```
i=1;m_OSA=0;v_OSA=0;m_MPE=0;v_MPE=0;

while(i<170)
    if(i==1)
        m_OSA(i)=mean(r_OSA{1,i}(1:1500));
        v_OSA(i)=var(r_OSA{1,i}(1:1500));

    end

    if(i>1)
        m_OSA(i)=mean(r_OSA{1,i}(1:460));
        v_OSA(i)=var(r_OSA{1,i}(1:460));
    end
    i=i+1;
end

i=1;
while(i<170)
    if(i==1)
        m_MPE(i)=mean(r_MPE{1,i}(1:1500));
        v_MPE(i)=var(r_MPE{1,i}(1:1500));

    end

    if(i>1)
        m_MPE(i)=mean(r_MPE{1,i}(1:460));
        v_MPE(i)=var(r_MPE{1,i}(1:460));
    end
i=i+1;
end

figure;
subplot(2,2,1);plot(m_OSA);title('a)')
xlabel('Number of Cycle')
ylabel('Mean OSA')
subplot(2,2,2);plot(v_OSA);title('b)')
xlabel('Number of Cycle')
ylabel('Variance OSA')
subplot(2,2,3);plot(m_MPE);title('c)')
xlabel('Number of Cycle')
ylabel('Mean MPE')
subplot(2,2,4);plot(v_MPE);title('d)')
xlabel('Number of Cycle')
ylabel('Variance MPE')

plot(uprbs{1,1}(200:400),'--');hold on;plot(yprbs{1,1}(200:400),'r');
xlabel('Time (0.5 sec)')
ylabel('Percent (%)')
legend('SP','PV')
```

**การเรียบเรียงสัญญาณตกค้างเพื่อให้ใช้งานในโปรแกรมอัลกอริทึม** CUSUM **ได้สะดวก**

```
i=0;r_m=[];r_o=[];
while(i<170)
i=i+1;
r_m=[r_m;r_MPE{1,i}];
r_o=[r_o;r_OSA{1,i}];
end
```

**อัลกอริทึม** CUSUM **แบบ** two-sided mean change detection **และ** one-sided variance change detection **ด้วยสัญญาณที่สร้างขึ้นมาเอง**

```
u0=1;
u1=4;
delta=0.5;
variance0=1;
variance1=1.5;
v=10;

%generate gaussian noise in mean change + case.
gn1=randn(1000,1)*sqrt(v)+u0;   %psuedorandoat at standard normal
with mean=0, variacne =1
gn2=(randn(1000,1)*0.1+u1); %Put mean value into standard normal
%gnm=[gn1;gn2];
gnm=gn1;
%plot(gnm)

%generate gaussian noise in variance change case.
gn1=randn(1000,1)*variance0;  %psuedorandoat at standard normal with
mean=0, variacne =1
gn2=randn(1000,1)*variance1; %Put mean variance into standard normal
gnv=[gn1;gn2];
%plot(gnv)

l=length(gnm);
%Two sided CUSUM algorithm

%Mean change detection
i=0;j=0;jj=0;s_m1=0;s_m2=0;sm1=0;sm2=0;smean1=0;smean2=0;svar=0;a=0;s
v=0;s_v=0;m=0;tam=0;tav=0;
k=0;
while(i<l)
        i=i+1;
        %mean change detection
        delta=abs(u1-u0);
        s_m1(i+1)=s_m1(i)+(delta/v)*(gnm(i)- u0 - delta/2);
        if (s_m1(i+1)<0)
           s_m1(i+1)=0;
        end

        s_m2(i+1)=s_m2(i)-(delta/v)*(gnm(i)- u0 + delta/2);
        if (s_m2(i+1)<0)
           s_m2(i+1)=0;
        end

%         smean1=smean1+s_m1(i+1);
%         sm1(i)=smean1;
%         smean2=smean2+s_m2(i+1);
%         sm2(i)=smean2;
 sm1(i)=s_m1(i);
 sm2(i)=s_m2(i);
        %variance change detection
        a(i)=mean(gnv(1:i));
```

```
        s_v(i+1)=s_v(i)+log(variance0/variance1)+((1/(variance0^2)-
1/(variance1^2))*((gnv(i)-a(i))^2)/2);
        if (s_v(i+1)<0)
           s_v(i+1)=0;
        end
        svar=svar+s_v(i+1);
        sv(i)=svar;

        % time alarm for mean detection
        if(i<l/2)
            j=i;
        end
        if(i>=l/2)
            j=l/2;
        end

        if (sm2(i)>(max(sm2(1:j)))||sm1(i)>(max(sm1(1:j))))
            k=k+1;
            tam(k)=i;
        end

        %time alarm for mean detection
        if(i<l/2)
            jj=i;
        end
        if(i>=l/2)
            jj=l/2;
        end

        if (sm2(i)>1.2*(max(sm2(1:jj)))||sm1(i)>(max(sm1(1:jj))))
            k=k+1;
            tam(k)=i;
        end

        %time alarm for variance detection
        if(sv(i)>1.2*(0.2+1)*(max(sv(1:jj))))
        m=m+1;
        tav(m)=i;
        end
end
time_alarm_mean=min(tam)
time_alarm_variance=min(tav)
figure;
subplot(3,1,1);plot(sm1);title('Positive mean change detection')
subplot(3,1,2);plot(sm2);title('Negative mean change detection')
subplot(3,1,3);plot(sv);title('Variance change detection')
%plot(s2)
% subplot(2,2,1);plot(sm1);title('When input gaussian noise change
mean in negative')
% subplot(2,2,3);plot(sm2)
% subplot(2,2,2);plot(sm1);title('When input gaussian noise change
mean in negative')
% subplot(2,2,4);plot(sm2)
%variance change detection
```

**อัลกอริทึม CUSUM แบบตรวจจับความแปลี่ยนแปลงของค่าความแปรปรวนด้วยสัญญาณตกค้าง**

```
% clear,clc
% load data
i=0;e=2.5;
n=length(r_m);
s_v=0;s_v(1501)=1;
z=1;v0=0;v1=0;u0=0;u1=0;m=0;tav=0;k=0;threshole_v=0;

%p=36
v0=sqrt(var(r_MPE{1,1})),v0_m=v0;
v1=e*v0,v1_m=v1; %var(r_predicted(1500+460*(p-1):1500+460*p))
u0=mean(r_MPE{1,1}),u0_m=u0;%disp('normally variance when use
predited residual is'),disp(v0)

while(i<n)
    i=i+1;

    if(i>2) %1500)
%Variance change detection

        s_v(i)=s_v(i-1)+log(v0/v1)+(((1/(v0^2)-1/(v1^2))*(r_m(i)-
u0)^2)/2);
        if (s_v(i)<0)
           s_v(i)=0;
        end

    end

end
s_v_predicted=s_v;

%%%%%%%
%
%  One step ahead
%
%%%%%%%


s_v=0;s_v(1501)=1;i=0;
z=1;v0=0;v1=0;u0=0;u1=0;m=0;tav=0;k=0;threshole_v=0;
v0=sqrt(var(r_OSA{1,1})),v0_o=v0;
v1=e*v0,v1_o=v1;
u0=mean(r_MPE{1,1}),u0_o=u0;

while(i<n)
    i=i+1;

    if(i>2) %1500)
%Variance change detection
        s_v(i)=s_v(i-1)+log(v0/v1)+((1/(v0^2)-1/(v1^2))*((r_o(i)-
u0)^2)/2);
        if (s_v(i)<0)
           s_v(i)=0;
        end
    end
```

```
end
s_v_onestep=s_v;

figure;
subplot(2,1,1);plot(s_v_onestep);xlabel('Number');ylabel('Variance
OSA'); title(['S.D. x',num2str(e),':
v0=',num2str(v0_o),'v1=',num2str(v1_o),'u=',num2str(u0_o)])
subplot(2,1,2);plot(s_v_predicted);xlabel('Number');ylabel('Variance
MPE');title(['S.D. x',num2str(e),':
v0=',num2str(v0_m),'v1=',num2str(v1_m),'u=',num2str(u0_m)])
```

**การหาสัมประสิทธิ์ของเทอมในโมเดลแบบ NARX**

```
clear, clc
load Data

for j=1:170
% j=j+1;
    %initial reseting
    parameter=[];y=0;

for i=9:length(yprbs{1,j}) %i is index of Vector

    %Using parameter of NARX model to calculate coefficient
    parameter(i,1:8)=[uprbs{1,j}(i-3,1)^2, yprbs{1,j}(i-2,1),
yprbs{1,j}(i-2,1)*uprbs{1,j}(i-3,1), uprbs{1,j}(i-8,1)^2,
yprbs{1,j}(i-4,1), uprbs{1,j}(i-8,1), uprbs{1,j}(i-3,1)*uprbs{1,j}(i-
8,1), yprbs{1,j}(i-1,1)*uprbs{1,j}(i-3,1)];
                      %u(t-3)^2            %y(t-2)           %y(t-
2)u(t-3)                        %u(t-8)^2            %y(t-4)
%u(t-8)            %u(t-3)u(t-8)                        %y(t-1)*u(t-3)
    y(i)=yprbs{1,j}(i,1);

end
%     parameter=parameter';
    [U,S,V]=svd(parameter,0); %Find pseudoinverse

    cof(j,1:8)=y*U;       %Coefficient of NARX model

end

figure;
subplot(2,4,1);plot(cof(1:end,1));
xlabel('Cycle');ylabel('Amplitude');title('1st parameter')
subplot(2,4,2);plot(cof(1:end,2));
xlabel('Cycle');ylabel('Amplitude');title('2nd parameter')
subplot(2,4,3);plot(cof(1:end,3));
xlabel('Cycle');ylabel('Amplitude');title('3rd parameter')
subplot(2,4,4);plot(cof(1:end,4));
xlabel('Cycle');ylabel('Amplitude');title('4th parameter')
subplot(2,4,5);plot(cof(1:end,5));
xlabel('Cycle');ylabel('Amplitude');title('5th parameter')
subplot(2,4,6);plot(cof(1:end,6));
xlabel('Cycle');ylabel('Amplitude');title('6th parameter')
subplot(2,4,7);plot(cof(1:end,7));
xlabel('Cycle');ylabel('Amplitude');title('7th parameter')
subplot(2,4,8);plot(cof(1:end,8));
xlabel('Cycle');ylabel('Amplitude');title('8th parameter')
```

**การแบ่งกลุ่มข้อมูลด้วยอัลกอริทึม** K-mean

```
X=[cof(1:end,1),cof(1:end,4),cof(1:end,6),u,K,sd];
opts = statset('Display','final');
[idx,ctrs] =
kmeans(X,3,'Distance','city','Replicates',5,'Options',opts);

figure;subplot(2,1,1);plot(X(idx==1,1),X(idx==1,2),'g.','MarkerSize',
12)
hold on
plot(X(idx==2,1),X(idx==2,2),'b.','MarkerSize',12)
plot(X(idx==3,1),X(idx==3,2),'r.','MarkerSize',12)
% plot(ctrs(:,1),ctrs(:,2),'kx',...
%      'MarkerSize',12,'LineWidth',2)
% plot(ctrs(:,1),ctrs(:,2),'ko',...
%      'MarkerSize',12,'LineWidth',2)
legend('Cluster 1','Cluster 2','Cluster 3',...
       'Location','NW')

subplot(2,1,2);plot(idx);axis([-5,175,0.5,3.5])
```

**อัลกอริทึม Adaptive CUSUM แบบตรวจจับความแปลี่ยนแปลงของค่าความแปรปรวนด้วยสัญญาณตกค้าง**

```
clear,clc
load data

b=700;
n=1500;

    %Clear condition
u=[];sd0=[];sd1=[];umin=[];umax=[];
s_v=0;s_v(1:2)=0;
    % Initial condition
        input=r_m;%r_o;

    i=0;j=0;k=0;l=0;s_v=0;s_v(1)=0;Ta=0;T=0;dv=0;
    while (i<length(input)-1)%40000)
        i=i+1;

        if(s_v(i)>T||i==1)

            s_v(i)=0;s_v(i+1)=0;

            k=k+1;
            TT(k)=T;  %Threshold
            Ta(k)=i-1;  %Time alarm
            d(k)=dv;  %Derivative of standard deviation

                B=b;
                N=460;
                if (i==1)
                    N=n;
                end

            %initial
                u=mean(input(1+Ta(k):N+Ta(k))); %find new initial
mean when change variance already
                sd0=std(input(1+Ta(k):N+Ta(k))); %find new initial
standard deviation when hypothesis is normal and change variance
already
                sd1= sqrt(sqrt(N*(sd0^2)^2/chi2inv(0.01/2,N)));
%find initial standard deviation when hypothesis is change variance
at probability at 99%

            %Find threshold from N sample
            j=0;
            while (j<N-1)
                j=j+1;
                [s_v] = CUSUM(j+Ta(k),input,sd0,sd1,u,s_v);
            end
            T=B*max(s_v(1+Ta(k):N+Ta(k)));
        end

        %Variance change detection
        [s_v] = CUSUM(i,input,sd0,sd1,u,s_v);
```

```
    end

%     disp('Round = '),disp((Ta-1500)/460),disp('Ta =
'),disp(Ta),disp('Dv = '),disp(d),disp('Threshold = '),disp(TT)
%     figure;plot(s_v);
%     Ta
Ta(1)=[];
Ta, (Ta-1500)/460
figure;plot(s_v)
xlabel('Sample');ylabel('Amplitude of Adaptive CUSUM')
```

**ฟังก์ชันสำหรับอัลกอริทึม** CUSUM **ในการใช้งานร่วมกับโปรแกรมอัลกอริทึม** Adaptive CUSUM

```
function [s_v] = CUSUM(i,input,sd0,sd1,u,s_v) %i is number of sample,
v is variance, u0 is mean0, u1 is mean1, s_m1 is positive CUSUM, s_m2
is negative CUSUM

s_v(i+1)=s_v(i) + (log(sd0/sd1)+(1/(sd0)^2-1/sd1^2)*(input(i+1)-
u)^2/2);
    if (s_v(i+1)<0)
        s_v(i+1)=0;
    end
```