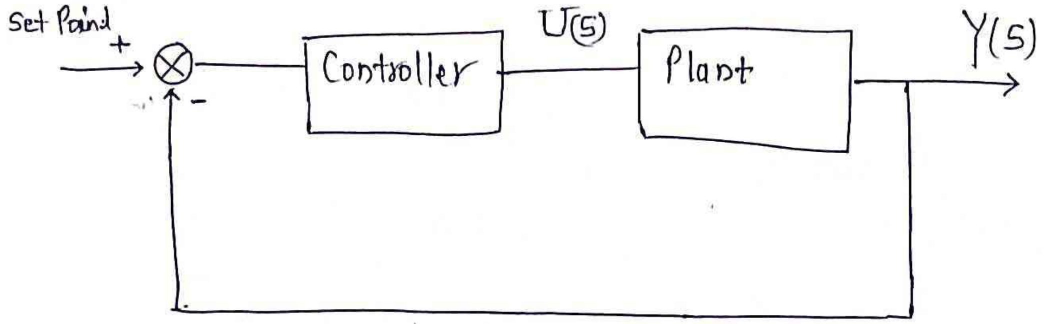
Chapter -4

MATLAB CODES



$$\frac{Y(s)}{U(s)} = \frac{e^{-0.2s}}{(s+1)^2}$$

$$Y(s)\,(s+1)^2 = e^{-0.2s} \quad U(s)$$

$$s^2Y(s) + 2s\,Y(s) + Y(s) = e^{-0.2}\,U(s)$$

$\Downarrow$ Time domain

$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + y(t) = u(t-0.2)$$

Let $\frac{dy}{dt} = x$ $\quad \therefore \frac{d^2y}{dt^2} = \frac{dx}{dt}$

$$\frac{dx}{dt} + 2x + y = u(t-0.2)$$

$$\frac{dx}{dt} = u(t-0.2) - y - 2x$$

$$\boxed{Fxy = @(u,y,x)\ \ u - y - 2*x}$$

(Way to write in Matlab)

Note : $\frac{dy}{dt} = x$

$$\frac{y(i+1) - y(i)}{h} = x(i+1)$$

(h = step size)

$$\boxed{y(i+1) = y(i) + h * x(i+1)}$$

We have written the code for following process model.

- $G_p(s) = \dfrac{e^{-Ls}}{(1+s)^2}$, L=0.2 and 0.3;
- $G_p(s) = \dfrac{e^{-Ls}}{s(s+1)}$, L=0.2 and 0.3;

SN.	PROCESS MODEL	DELAY	ULTIMATE GAIN (calculated at first delay)	ULTIMATE PERIOD
1-	$G_p(s) = \dfrac{e^{-Ls}}{(1+s)^2}$	L=0.2 and 0.3	10.5	2.0333
2	$G_p(s) = \dfrac{e^{-Ls}}{s(s+1)}$	L=0.2 and 0.3	5.09	2.9

Matlab code for ultimate gain and ultimate time period calculation for the process model

$$G_p(s) = \frac{e^{-Ls}}{(1+s)^2}$$

```
%%%% calculation of ultimate gain and time period
%%%%tf=exp(-0.2s)/(s2+2s+1)

clc;
clear all;

h=0.1;
t = 0:h:30;
y = zeros(1,length(t));
u = zeros(1,length(t));
```

```
e = zeros(1,length(t));
x = zeros(1,length(t));
i = zeros(1,length(t));
input = 1;
y(1) = 0;
x(1)=0 ;
e(1)=input - y(1);

kp=10.5;

u(1)=kp*e(1);

F_xy = @(x) -2*x;

for i = 1:2

%y and u taken as time input and x as output

k_1 = F_xy(x(i));
k_2 = F_xy(x(i)+0.5*h*k_1);
k_3 = F_xy((x(i)+0.5*h*k_2));
k_4 = F_xy((x(i)+k_3*h));


x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);

e(i+1)=1-y(i+1);

u(i+1)=kp*e(i+1);

end

F_xy = @(u,y,x) u-y-2*x;


for i = 3:length(t)

k_1 = F_xy(u(i-2),y(i),x(i));

k_2 = F_xy(u(i-2)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-2)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-2)+h ) ,(y(i)+h) ,(x(i)+k_3*h));


x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);
```

```
e(i+1)=1-y(i+1);

u(i+1)=kp*e(i+1);

end


z=y(1:301)
plot(t,z)
xlabel('value of time t')
ylabel('value of y')
title(' calculation of ultimate gain,k =10.5 ultimate period=2.033sec, second order system tf=exp(-0.2s)/(s2+2s+1) ')
grid on
```

matlab code for CPID of TF1

```
clc;
 clear all;

h=0.1;
t = 0:h:16;

tf=16/h;
y = zeros(1,length(t));
u = zeros(1,length(t));
e = zeros(1,length(t));
x = zeros(1,length(t));
y(1) = 0;
x(1)=0 ;
r=1
e(1)=r - y(1);

ku=10.5;
tu=2.033;

kc=0.6*ku
ti=tu/2
td=tu/8

 u(1)=kc*(e(1) +(0.1/ti)*sum(e));

 F_xy = @(x) -2*x;

 for i = 1:2
          %y and u taken as time input and x as output
           k_1 = F_xy(x(i));
```

```
            k_2 = F_xy(x(i)+0.5*h*k_1);
            k_3 = F_xy((x(i)+0.5*h*k_2));
            k_4 = F_xy((x(i)+k_3*h));

            x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

            y(i+1)=y(i)+h*x(i+1);

            e(i+1)=r-y(i+1);

            er=sum(e);

            ed=e(i+1)-e(i);

            u(i+1)=kc*(e(i+1) +(0.1/ti)*sum(e)+(td/0.1)*ed);

end

F_xy = @(u,y,x) u-y-2*x;

for i = 3:(tf/2)
        k_1 = F_xy(u(i-2),y(i),x(i));

        k_2 = F_xy(u(i-2)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

        k_3 = F_xy((u(i-2)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

        k_4 = F_xy((u(i-2)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

        x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

       y(i+1)=y(i)+h*x(i+1);
       e(i+1)=r-y(i+1);
       er=sum(e);
       ed=e(i+1)-e(i);

      u(i+1)=kc*(e(i+1) +(0.1/ti)*sum(e)+(td/0.1)*ed);

end

% effect of load on the process(25%)

u((tf/2)+1)=-14

F_xy = @(u,y,x) u-y-2*x;


for i = (tf/2)+1:length(t)

    k_1 = F_xy(u(i-2),y(i),x(i));

    k_2 = F_xy(u(i-2)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

    k_3 = F_xy((u(i-2)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

    k_4 = F_xy((u(i-2)+h ) ,(y(i)+h) ,(x(i)+k_3*h));
    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;
```

```
    y(i+1)=y(i)+h*x(i+1);
    e(i+1)=r-y(i+1);
    er=sum(e);
    ed=e(i+1)-e(i);
     u(i+1)=kc*(e(i+1) +(0.1/ti)*sum(e)+(td/0.1)*ed);
end

z=y(1:tf+1);
plot(t,z,'--')
xlabel('Time t ')
ylabel('Response y')
title(' CPID(- - -) Response of second order system TF=exp(-0.2s)/s2+2s+1)')
grid on
```

Matlab code for APID of TF1

```
%%%%APID  adaptive PID CONTROLLER  Response of second order system TF=exp(-
0.2s)/s2+2s+1

clc;
clear all;

h=0.1;
t = 0:h:16;
tf=16/h;

y = zeros(1,length(t));
u = zeros(1,length(t));
e = zeros(1,length(t));
x = zeros(1,length(t));

kpp = zeros(1,length(t));
kii = zeros(1,length(t));
kdd = zeros(1,length(t));

v = zeros(1,length(t));
x = zeros(1,length(t));

r= 1;

setpoint=r;
y(1) = 0;
x(1)=0 ;
e(1)=r - y(1);


ku=10.5;
tu=2.033;

kp=0.6*ku
ti=tu/2
td=tu/8

ki=kp*(0.1/ti)
kd=kp*(td/0.1)

k1=1
```

```
k2=1
k3=12

u(1)=kp*e(1)+ki*sum(e);

 kpp(1)=0;
 kii(1)=0;
 kdd(1)=0;
 v(1)=0;


F_xy = @(x) -2*x;

for i = 1:2


k_1 = F_xy(x(i));
k_2 = F_xy(x(i)+0.5*h*k_1);
k_3 = F_xy((x(i)+0.5*h*k_2));
k_4 = F_xy((x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);

e(i+1)=r-y(i+1);

er=sum(e);

ed=e(i+1)-e(i);

v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));
kii(i+1)=ki*(0.3+k2*v(i+1));
kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end


F_xy = @(u,y,x) u-y-2*x;

for i = 3:(tf/2)
k_1 = F_xy(u(i-2),y(i),x(i));

k_2 = F_xy(u(i-2)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-2)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-2)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);
```

```
e(i+1)=r-y(i+1);
ed=e(i+1)-e(i);
er=sum(e);
v(i+1)=(e(i+1)/r)*(ed/r);
kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(0.3+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end


u((tf/2)+1)= -14;

F_xy = @(u,y,x) u-y-2*x;


for i = (tf/2)+1:length(t)
    k_1 = F_xy(u(i-2),y(i),x(i));

    k_2 = F_xy(u(i-2)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

    k_3 = F_xy((u(i-2)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

    k_4 = F_xy((u(i-2)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

   y(i+1)=y(i)+h*x(i+1);

e(i+1)=r-y(i+1);
ed=e(i+1)-e(i);
er=sum(e);
v(i+1)=(e(i+1)/r)*(ed/r);
kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(0.3+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end


z=y(1:(tf+1));

plot(t,z)
hold on
xlabel('Time t ')
ylabel('Response y')
```

```
title('APID  Response of second order system TF=exp(-0.2s)/s2+2s+1) ')
grid on
```

Matlab code for PSO based PID controller, PSO-CPID of TF1

```
%%%%%%%%%%%%%%%genetic algorithm based pid controller
%%%% exp(-0.2s) /(s+1)^2

%%%first we will clculate kp ki kd from zeigler nichols method.
%By varying plus minus 20% of all the tuned values(kp ki kd)
%We will calculate best tuned valuve of kp ki kd with the help of particle swarm
optimization algorithms




%%%%Transfer func=  TF = exp(-0.2s)/(s+1)^2
%%%%%%%znpid with pso-algorithm

clc ;
clear all;


popsize=10;

npar=3;


c1=2;
c2=2;


no_of_variable=3;




par=rand(popsize,npar);

vel=rand(popsize,npar);



%%%%%%%%%%%%%%%%%initialization %%%%%%


ku=10.5;
tu=2.0333;
```

```
kp=0.6*ku;

ti=tu/2;
td=tu/8;



ki=kp*(0.1/ti);

kd=kp*(td/0.1);


%%%%%%%%%%%%%%%%5


xh1=kp+0.20*kp;
xl1=kp-0.20*kp;


xh2=ki+0.20*ki;
xl2=ki-0.20*ki;

xh3=kd+0.20*kd;
xl3=kd-0.20*kd;


h=0.1;
t = 0:h:16;
tf=16/h


delay=0.2;
de=2;



w1=1;
w2=1;


%%%% bring position vector in range

for i=1:10

x1(i,1)=xl1+((xh1-xl1)/(0.9999-0.0001))*par(i,1);

x2(i,1)=xl2+((xh2-xl2)/(0.9999-0.0001))*par(i,2);

x3(i,1)=xl3+((xh3-xl3)/(0.9999-0.0001))*par(i,3);



end

p=[x1 x2 x3 ] ;
```

```
%%%define the rande of velocity

vh1=(xh1-xl1);
vl1= -(xh1-xl1);


vh2=(xh2-xl2);
vl2= -(xh2-xl2);


vh3=(xh3-xl3);
vl3= -(xh3-xl3);



%%% Bring velocity vector in range


for i=1:10

v1(i,1)=vl1+((vh1-vl1)/(0.9999-0.0001))*vel(i,1);

v2(i,1)=vl2+((vh2-vl2)/(0.9999-0.0001))*vel(i,2);

v3(i,1)=vl3+((vh3-vl3)/(0.9999-0.0001))*vel(i,3);



end



v=[v1 v2 v3 ] ;

   %%%%%%%objective fun..


for j=1:10


    kp=p(j,1);
    ki=p(j,2);
    kd=p(j,3);





y = zeros(1,length(t)+100);
x = zeros(1,length(t)+100);
u = zeros(1,length(t)+100);

e = zeros(1,length(t)+100);
er = zeros(1,length(t)+100);
ed = zeros(1,length(t)+100);
```

```
g = zeros(1,length(t)+100);

input = 1;
y(1) = 0;
x(1)=0 ;
e(1)=input - y(1);

 u(1)=kp*e(1)+ki*sum(e);



F_xy = @(x) -2*x;



for i = 1:de


    k_1 = F_xy(x(i));
    k_2 = F_xy(x(i)+0.5*h*k_1);
    k_3 = F_xy((x(i)+0.5*h*k_2));
    k_4 = F_xy((x(i)+k_3*h));

    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

    y(i+1)=y(i)+h*x(i+1);

    e(i+1)=1-y(i+1);

    ed=e(i+1)-e(i);

    u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;



end


F_xy = @(u,y,x) u-y-2*x;


for i = (de+1):(tf/2)

    k_1 = F_xy(u(i-de),y(i),x(i));

    k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

    k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

    k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));


    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

    y(i+1)=y(i)+h*x(i+1);
```

```
    e(i+1)=1-y(i+1);

    ed=e(i+1)-e(i);

    u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;

end


u((tf/2)+1)=-20;

F_xy = @(u,y,x) u-y-2*x;

for i = (tf/2)+1:length(t)

    k_1 = F_xy(u(i-de),y(i),x(i));

    k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

    k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

    k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

    y(i+1)=y(i)+h*x(i+1);

    e(i+1)=1-y(i+1);

    ed=e(i+1)-e(i);

   u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;

end

z=y(1:161);

figure(j+1)

plot(t,z)

xlabel('Time t ')

ylabel('Response y')

title(' exp(-0.2s) /(s+1)^2  ')

grid on

iaeiae(j,1)=0.1*sum(abs(e));
```

```
h=0.1;

for i= 1:length(t)

g(i)=0.01*i*e(i);

end

itaeitae(j,1)=sum(abs(g));



end




iae=w1*iaeiae+w2*itaeitae;


[iae ind]=sort(iae);


p=p(ind,:);
v=v(ind,:);

localpar=p(1,:);
localminima(1,1)=iae(1,1);

globalpar=p(1,:);
globalminima(1,1)=iae(1,1);


display('start of while loop')

iter=0;

m=1;
n=10;

maxit=30;




while iter < maxit


iter=iter+1;

r1=rand(popsize,npar);

r2=rand(popsize,npar);

wt=(maxit-iter)/maxit;
```

```
for i=1:10

v(i,:)= wt*v(i,:) + c1*r1(i,:).*(localpar-p(i,:))  +  c2*r2(i,:).*(globalpar-
p(i,:));


end


v;

v1=v(:,1);

v2=v(:,2);

v3=v(:,3);


%%In main equation of PSO there are three term of addition so sometimes it
%%may cross the boundary limit so we will do with the following method


for i=1:10

if v1(i,1)  > vh1

v1(i,1)= vh1;

elseif v1(i,1)  < vl1
v1(i,1)= vl1;


end


end




for i=1:10

if v2(i,1)  > vh2
v2(i,1)= vh2;

elseif v2(i,1)  < vl2
v2(i,1)= vl2;


end

end
```

```
for i=1:10

if v3(i,1)  > vh3
v3(i,1)= vh3;

elseif v3(i,1)  < vl3

v3(i,1)= vl3;


end




end




v1;
v2;
v3;

v=[v1 v2 v3];




%%%%%new position of partice will be


    p=p+v;

%%% bring p in the range

x1=p(:,1);

x2=p(:,2);

x3=p(:,3);




%%%bring kp ki kd in range....it means x1 x2 x3

for i=1:10

if x1(i,1)  > xh1

x1(i,1)= xh1;

elseif x1(i,1)  < xl1
x1(i,1)= xl1;


end
```

```
end


for i=1:10

if x2(i,1)  > xh2

x2(i,1)= xh2;


elseif x2(i,1)  < xl2

x2(i,1)= xl2;


end


end


for i=1:10

if x3(i,1)  > xh3

x3(i,1)= xh3;


elseif x3(i,1)  < xl3

x3(i,1)= xl3;


end


end


x1;
x2;
x3;

v1;
v2;
v3;

v=[v1 v2 v3];

p=[x1 x2 x3];

p;
```

```
%%%objective function



for j=1:10


    kp=p(j,1);
    ki=p(j,2);
    kd=p(j,3);




y = zeros(1,length(t)+100);
x = zeros(1,length(t)+100);
u = zeros(1,length(t)+100);




e = zeros(1,length(t)+100);
er = zeros(1,length(t)+100);
ed = zeros(1,length(t)+100);
g = zeros(1,length(t)+100);

input = 1;
y(1) = 0;
x(1)=0 ;
e(1)=input - y(1);

 u(1)=kp*e(1)+ki*sum(e);

F_xy = @(x) -2*x;

for i = 1:de



    k_1 = F_xy(x(i));
    k_2 = F_xy(x(i)+0.5*h*k_1);
    k_3 = F_xy((x(i)+0.5*h*k_2));
    k_4 = F_xy((x(i)+k_3*h));


    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

    y(i+1)=y(i)+h*x(i+1);

    e(i+1)=1-y(i+1);

    ed=e(i+1)-e(i);

    u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;


end
```

```
F_xy = @(u,y,x) u-y-2*x;

    for i = (de+1):(tf/2)

    k_1 = F_xy(u(i-de),y(i),x(i));

    k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

    k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

    k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

    y(i+1)=y(i)+h*x(i+1);

    e(i+1)=1-y(i+1);

    ed=e(i+1)-e(i);

    u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;

   end

u((tf/2)+1)=-20;


F_xy = @(u,y,x) u-y-2*x;

for i = (tf/2)+1:length(t)

    k_1 = F_xy(u(i-de),y(i),x(i));

    k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

    k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

    k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

   y(i+1)=y(i)+h*x(i+1);

   e(i+1)=1-y(i+1);
```

```
   ed=e(i+1)-e(i);

   u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;


end

z=y(1:161);

figure(j+1)

plot(t,z)


xlabel('Time t ')

ylabel('Response y')

title(' exp(-0.2s) /(s+1)^2  ')

grid on


iaeiae(j,1)=0.1*sum(abs(e));


h=0.1;

for i= 1:length(t)

g(i)=0.01*i*e(i);

end

itaeitae(j,1)=sum(abs(g));



end




iae=w1*iaeiae+w2*itaeitae;


[iae ind]=sort(iae);


p=p(ind,:);

v=v(ind,:);


localpar=p(1,:);
```

```
localminima(iter+1,1)=iae(1,1);


ss=length(localminima);

tt=length(globalminima);


if  (localminima(ss,1) < globalminima(tt,1))

  globalpar= p(1,:);

  globalminima(tt+1,1)= iae(1,1);


else


   globalminima(tt+1,1)= globalminima(tt,1);

end


vv(m:n,:)=v(1:10,:);
pp(m:n,:)=p(1:10,:);

m=n+1;
n=n+10;


end


display('please note the kp ki kd -this is the result of pso algorithm')

globalpar
%%%%%%%%

for j=1


kp=globalpar(j,1);
ki=globalpar(j,2);
kd=globalpar(j,3);




y = zeros(1,length(t)+100);
x = zeros(1,length(t)+100);
u = zeros(1,length(t)+100);




e = zeros(1,length(t)+100);
er = zeros(1,length(t)+100);
ed = zeros(1,length(t)+100);
```

```
g = zeros(1,length(t)+100);




input = 1;
y(1) = 0;
x(1)=0 ;
e(1)=input - y(1);

u(1)=kp*e(1)+ki*sum(e);

F_xy = @(x) -2*x;

for i = 1:de



    k_1 = F_xy(x(i));
    k_2 = F_xy(x(i)+0.5*h*k_1);
    k_3 = F_xy((x(i)+0.5*h*k_2));
    k_4 = F_xy((x(i)+k_3*h));


    x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

    y(i+1)=y(i)+h*x(i+1);

    e(i+1)=1-y(i+1);



    ed=e(i+1)-e(i);



    u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;



end


F_xy = @(u,y,x) u-y-2*x;


for i=(de+1):(tf/2)


k_1 = F_xy(u(i-de),y(i),x(i));

k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));
```

```
k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;


y(i+1)=y(i)+h*x(i+1);

e(i+1)=1-y(i+1);


ed=e(i+1)-e(i);

u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;


end


u((tf/2)+1)=-20;



F_xy = @(u,y,x) u-y-2*x;


for i = (tf/2)+1:length(t)

k_1 = F_xy(u(i-de),y(i),x(i));

k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));


x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);

e(i+1)=1-y(i+1);


ed=e(i+1)-e(i);



u(i+1)=kp*e(i+1)+ki*sum(e)+kd*ed;


end


z=y(1:tf+1);

figure(1)
```

```
plot(t,z)

xlabel('Time t ')
ylabel('Response y')
title(' exp(-0.2s) /(s+1)^2  ')

grid on
end

%%%time specification

y;
yy = y(1:(tf/2));
[yymax tp]=max(yy);
peak_time=(tp-1)*0.1;

overshoot=(yymax-1)*100
rr=1;

while y(rr)<1.0001

rr=rr+1;
end
rise_time=(rr-1)*0.1



s=(tf/2);
while y(s)>0.98 & y(s)<1.02;
s=s-1;
end


settling_time=(s-1)*0.1



iae=0.1*sum(abs(e))
```

```
h=0.1;

for i= 1:length(t)

g(i)=0.01*i*e(i);

end

itae=sum(abs(g));

itae
```

****

****

PSO BASED ADAPTIVE PID CONTROLLER PROGRAM:

```
%%%%pso based adaptive PID controller design%%%%%%
%%%%consider the palnt model = exp(-0.2s) /(s+1)^2
%%%%%%%%%aznpid seven -pso-algo

clc ;
clear all;

popsize=10;


npar=7;


c1=2;
c2=2;


no_of_variable=7;




par=rand(popsize,npar);

vel=rand(popsize,npar);



%%%initialization

ku=10.5;
tu=2.0333;


kp=0.6*ku;
ti=tu/2;
td=tu/8;


ki=kp*(0.1/ti);

kd=kp*(td/0.1);



%%%%%%%%%%%%%%%%%5555
```

```
xh1=kp+0.20*kp;
xl1=kp-0.20*kp;


xh2=ki+0.20*ki;
xl2=ki-0.20*ki;

xh3=kd+0.20*kd;
xl3=kd-0.20*kd;


xh4=5;
xl4=0;

xh5=5;
xl5=0;


xh6=30;
xl6=0;

xh7=1;
xl7=0.1;


h=0.1;
t = 0:h:16;
tf=16/h;


w1=0;
w2=1;


delay=0.2;
de=2;



%%%% bring position vector in range



for i=1:10

x1(i,1)=xl1+((xh1-xl1)/(0.9999-0.0001))*par(i,1);

x2(i,1)=xl2+((xh2-xl2)/(0.9999-0.0001))*par(i,2);

x3(i,1)=xl3+((xh3-xl3)/(0.9999-0.0001))*par(i,3);
```

```
x4(i,1)=xl4+((xh4-xl4)/(0.9999-0.0001))*par(i,4);

x5(i,1)=xl5+((xh5-xl5)/(0.9999-0.0001))*par(i,5);

x6(i,1)=xl6+((xh6-xl6)/(0.9999-0.0001))*par(i,6);

x7(i,1)=xl7+((xh7-xl7)/(0.9999-0.0001))*par(i,7);


end


p=[x1 x2 x3 x4 x5 x6 x7 ] ;



%%%define the rande of velocity


vh1=(xh1-xl1);
vl1= -(xh1-xl1);


vh2=(xh2-xl2);
vl2= -(xh2-xl2);


vh3=(xh3-xl3);
vl3= -(xh3-xl3);



vh4=(xh4-xl4);
vl4= -(xh4-xl4);


vh5=(xh5-xl5);
vl5= -(xh5-xl5);


vh6=(xh6-xl6);
vl6=-(xh6-xl6);


vh7=(xh7-xl7);
vl7=-(xh7-xl7);
```

```
%%% Bring velocity vector in range

for i=1:10

v1(i,1)=vl1+((vh1-vl1)/(0.9999-0.0001))*vel(i,1);

v2(i,1)=vl2+((vh2-vl2)/(0.9999-0.0001))*vel(i,2);

v3(i,1)=vl3+((vh3-vl3)/(0.9999-0.0001))*vel(i,3);


v4(i,1)=vl4+((vh4-vl4)/(0.9999-0.0001))*vel(i,4);

v5(i,1)=vl5+((vh5-vl5)/(0.9999-0.0001))*vel(i,5);

v6(i,1)=vl6+((vh6-vl6)/(0.9999-0.0001))*vel(i,6);

v7(i,1)=vl7+((vh7-vl7)/(0.9999-0.0001))*vel(i,7);

end


v=[v1 v2 v3 v4 v5 v6 v7 ] ;


p;

%%%%call of objective function


[iae ]=objectivefunctionsevenvariable(p);

[iae ind]=sort(iae);

p=p(ind,:);
v=v(ind,:);

localpar=p(1,:)
localminima(1,1)=iae(1,1);

globalpar=p(1,:)
globalminima(1,1)=iae(1,1);


display('start of while loop')

iter=0;

m=1;
n=10;
```

```
maxit=50;


while iter < maxit


iter=iter+1;

r1=rand(popsize,npar);

r2=rand(popsize,npar);

wt=(maxit-iter)/maxit;




for i=1:10

v(i,:)= wt*v(i,:) + c1*r1(i,:).*(localpar-p(i,:))  + c2*r2(i,:).*(globalpar-
p(i,:)    );


end


v;

v1=v(:,1);

v2=v(:,2);

v3=v(:,3);

v4=v(:,4);
v5=v(:,5);
v6=v(:,6);
v7=v(:,7);

%%in main equation of PSO there are three term of addition so sometimes it
%%may cross the boundry limit so we will do with the following method


for i=1:10

if v1(i,1)  > vh1

v1(i,1)= vh1;

elseif v1(i,1)  < vl1
v1(i,1)= vl1;


end
```

```
end

for i=1:10

if v2(i,1)  > vh2
v2(i,1)= vh2;

elseif v2(i,1)  < vl2
v2(i,1)= vl2;


end

end


for i=1:10

if v3(i,1)  > vh3
v3(i,1)= vh3;

elseif v3(i,1)  < vl3

v3(i,1)= vl3;


end




end


for i=1:10

if v4(i,1)  > vh4
v4(i,1)= vh4;

elseif v4(i,1)  < vl4

v4(i,1)= vl4;


end

end



for i=1:10

if v5(i,1)  > vh5
```

```
v5(i,1)= vh5;

elseif v5(i,1)  < vl5

v5(i,1)= vl5;


end

end



for i=1:10

if v6(i,1)  > vh6
v6(i,1)= vh6;

elseif v6(i,1)  < vl6

v6(i,1)= vl6;


end

end


for i=1:10

if v7(i,1)  > vh7
v7(i,1)= vh7;

elseif v7(i,1)  < vl7

v7(i,1)= vl7;


end

end





v1;
v2;
v3;
v4;
v5;
v6;
v7;

v=[v1 v2 v3 v4 v5 v6 v7];
```

```
%%%%%new position of partice will be

    p=p+v;

%%% bring p in the range

x1=p(:,1);

x2=p(:,2);

x3=p(:,3);

x4=p(:,4);
x5=p(:,5);
x6=p(:,6);
x7=p(:,7);


%%%bring kp ki kd in range....it means x1 x2 x3

for i=1:10

if x1(i,1)  > xh1

x1(i,1)= xh1;

elseif x1(i,1)  < xl1
x1(i,1)= xl1;


end


end



for i=1:10

if x2(i,1)  > xh2

x2(i,1)= xh2;


elseif x2(i,1)  < xl2

x2(i,1)= xl2;


end


end
```

```
for i=1:10
if x3(i,1)  > xh3
x3(i,1)= xh3;

elseif x3(i,1)  < xl3
x3(i,1)= xl3;

end


end



for i=1:10
if x4(i,1)  > xh4
x4(i,1)= xh4;

elseif x4(i,1)  < xl4
x4(i,1)= xl4;

end

end
for i=1:10
if x5(i,1)  > xh5
x5(i,1)= xh5;

elseif x5(i,1)  < xl5
x5(i,1)= xl5;

end
```

```
end

for i=1:10

if x6(i,1)  > xh6

x6(i,1)= xh6;


elseif x6(i,1)  < xl6

x6(i,1)= xl6;


end


end

for i=1:10

if x7(i,1)  > xh7

x7(i,1)= xh7;


elseif x7(i,1)  < xl7

x7(i,1)= xl7;


end


end



x1;
x2;
x3;
x4;
x5;
x6;
x7;



v1;
v2;
v3;
v4;
v5;
v6;
v7;
```

```
v=[v1 v2 v3 v4 v5 v6 v7];

p=[x1 x2 x3 x4 x5 x6 x7];

p;

%%%objective function

[ iae ]=objectivefunctionsevenvariable(p);



[iae ind]=sort(iae);


p=p(ind,:);

v=v(ind,:);


localpar=p(1,:);


localminima(iter+1,1)=iae(1,1);

ss=length(localminima);

tt=length(globalminima);


if  (localminima(ss,1) < globalminima(tt,1))

  globalpar= p(1,:);

  globalminima(tt+1,1)= iae(1,1);


else


   globalminima(tt+1,1)= globalminima(tt,1);

end


vv(m:n,:)=v(1:10,:);
pp(m:n,:)=p(1:10,:);

m=n+1;
n=n+10;


end
```

```
display('please note the kp ki kd -this is the result of pso algorithm')

globalpar

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear v;

for j=1

    kp=globalpar(j,1);
    ki=globalpar(j,2);
    kd=globalpar(j,3);

    k1=globalpar(j,4);
    k2=globalpar(j,5);
    k3=globalpar(j,6);

    k4=globalpar(j,7);



 y = zeros(1,length(t)+100);
 x = zeros(1,length(t)+100);
 z = zeros(1,length(t)+100);
 u = zeros(1,length(t)+100);


 e = zeros(1,length(t)+100);
 er = zeros(1,length(t)+100);
 ed = zeros(1,length(t)+100);

 kpp = zeros(1,length(t)+100);
 kii = zeros(1,length(t)+100);
 kdd = zeros(1,length(t)+100);
 v = zeros(1,length(t)+100);
 g = zeros(1,length(t)+100);



r= 1;

setpoint=r;
y(1) = 0;
x(1)=0 ;
e(1)=r - y(1);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



u(1)=kp*e(1)+ki*sum(e);

kpp(1)=0;
kii(1)=0;
kdd(1)=0;
v(1)=0;


F_xy = @(x) -2*x;

for i = 1:de


k_1 = F_xy(x(i));
k_2 = F_xy(x(i)+0.5*h*k_1);
k_3 = F_xy((x(i)+0.5*h*k_2));
k_4 = F_xy((x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);

e(i+1)=r-y(i+1);

er=sum(e);

ed=e(i+1)-e(i);


v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(k4+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end




F_xy = @(u,y,x) u-y-2*x;


for i = de+1:(tf/2)
```

```
k_1 = F_xy(u(i-de),y(i),x(i));

k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);


e(i+1)=r-y(i+1);

ed=e(i+1)-e(i);

er=sum(e);

v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(k4+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;



end


u((tf/2)+1)= -20;

F_xy = @(u,y,x) u-y-2*x;


for i = (tf/2)+1:length(t)


k_1 = F_xy(u(i-de),y(i),x(i));

k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);
```

```
e(i+1)=r-y(i+1);

ed=e(i+1)-e(i);

er=sum(e);

v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(k4+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end



z=y(1:(tf+1));


figure(1)

plot(t,z)


xlabel('Time t ')

ylabel('Response y')

title(' exp(-0.2s) /(s+1)^2  aznpid pso (i.e.PSO-APID) kp ki kd k1 k2 k3 k4 ')

grid on
end

%%%%time specification

y;

yy = y(1:(tf/2));

[yymax tp]=max(yy);

peak_time=(tp-1)*0.1;

overshoot=(yymax-1)*100



rr=1;
```

```
while y(rr)<1.0001


    rr=rr+1;

end

rise_time=(rr-1)*0.1



s=(tf/2);

while y(s)>0.98 & y(s)<1.02;
    s=s-1;
end


settling_time=(s-1)*0.1



iae1=0.1*sum(abs(e))


h=0.1;

for i= 1:length(t)

g(i)=0.01*i*e(i);

end

itae1=sum(abs(g));

itae1



%%%%%%%OBJECTIVE FUNCTION PROGRAM%%%%%%


function [iae]=objectivefunctionsevenvariable(p)



h=0.1;
t = 0:h:16;
tf=160 ;


w1=1;
w2=0;
```

```
delay=0.2;
de=2;



for j=1:10


        kp=p(j,1);
        ki=p(j,2);
        kd=p(j,3);

        k1=p(j,4);
        k2=p(j,5);
        k3=p(j,6);
        k4=p(j,7);


y = zeros(1,length(t)+100);
x = zeros(1,length(t)+100);

u = zeros(1,length(t)+100);


e = zeros(1,length(t)+100);

kpp = zeros(1,length(t)+100);
kii = zeros(1,length(t)+100);
kdd = zeros(1,length(t)+100);


v = zeros(1,length(t)+100);
g = zeros(1,length(t)+100);


r= 1;

setpoint=r;
y(1) = 0;
x(1)=0 ;
e(1)=r - y(1);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



u(1)=kp*e(1)+ki*sum(e);

kpp(1)=0;
kii(1)=0;
kdd(1)=0;
v(1)=0;


F_xy = @(x) -2*x;
```

```
for i = 1:de


k_1 = F_xy(x(i));
k_2 = F_xy(x(i)+0.5*h*k_1);
k_3 = F_xy((x(i)+0.5*h*k_2));
k_4 = F_xy((x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);

e(i+1)=r-y(i+1);

er=sum(e);

ed=e(i+1)-e(i);


v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(k4+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));


u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end




F_xy = @(u,y,x) u-y-2*x;


for i = de+1:(tf/2)



k_1 = F_xy(u(i-de),y(i),x(i));

k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);


e(i+1)=r-y(i+1);
```

```
ed=e(i+1)-e(i);

er=sum(e);

v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(k4+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));

u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;



end

u((tf/2)+1)= -20;

F_xy = @(u,y,x) u-y-2*x;

for i = (tf/2)+1:length(t)

k_1 = F_xy(u(i-de),y(i),x(i));

k_2 = F_xy(u(i-de)+0.5*h,y(i)+0.5*h,x(i)+0.5*h*k_1);

k_3 = F_xy((u(i-de)+0.5*h),(y(i)+0.5*h),(x(i)+0.5*h*k_2));

k_4 = F_xy((u(i-de)+h ) ,(y(i)+h) ,(x(i)+k_3*h));

x(i+1) = x(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h;

y(i+1)=y(i)+h*x(i+1);

e(i+1)=r-y(i+1);

ed=e(i+1)-e(i);

er=sum(e);

v(i+1)=(e(i+1)/r)*(ed/r);

kpp(i+1)=kp*(1+k1*abs(v(i+1)));

kii(i+1)=ki*(k4+k2*v(i+1));

kdd(i+1)=kd*(1+k3*abs(v(i+1)));
```

```
u(i+1)=kpp(i+1)*e(i+1)+ kii(i+1)*sum(e)+ kdd(i+1)*ed;

end



z=y(1:(tf+1));

figure(j+1)

plot(t,z)

xlabel('Time t ')

ylabel('Response y')

title(' exp(-0.2s) /(s+1)^2  loop ')

grid on

%%%%%%%%   IAE   AND ITAE    %%%%%

iaeiae(j,1)=0.1*sum(abs(e));

for i= 1:length(t)

g(i)=0.01*i*e(i);

end

itaeitae(j,1)=sum(abs(g));

end

iae = w1*iaeiae+w2*itaeitae;
```