Anexo 2: Código utilizado en Matlab

```
%% Simulación de la Pila Modelo de cálculo pesado
Satisfacción de demanda eléctrica

clear all
close all
clc

nh=0.1/3.6; % Número de horas durante las que se
mantiene cada potencia
Pots=[0.36 0.47 0.59 0.95]; % Potencias obtenidas del
modelo en Excel en kW
Pots=Pots.*1000; % Conversión de las potencias de kW a
W

P(1:nh*3600)=ones(1,nh*3600)*Pots(1);

for i = 1:length(Pots)-1


P(i*nh*3600+1:(i+1)*nh*3600)=ones(1,nh*3600)*Pots(i+1)
;

end

Re=220^2./P;

Final=length(P);
t=[0:Final-1];

Input=[t', Re'];

R = sim('FuelCell',t, [],  Input);

Pel=R.yout{1}.Values.data;
Pter=R.yout{2}.Values.data;

subplot(2,1,1), plot(t,P,'go',t,Pel,'b'),
title('Potencia eléctrica'), grid,
legend('Referencia','P.Entregada')
subplot(2,1,2), plot(t,Pter), title('Potencia
térmica'), grid
```

```
%% Simulación control térmico clásico

clear all
close all
clc

nh=0.1/3.6; % Número de horas durante las que se
mantiene cada potencia
Term=[0.9 1.2 1.5 2.4]; % Potencias térmicas obtenidas
del modelo en Excel en kW
Term=Term.*1000; % Conversión de las potencias de kW a
W

T(1:nh*3600)=ones(1,nh*3600)*Term(1);

for i = 1:length(Term)-1

T(i*nh*3600+1:(i+1)*nh*3600)=ones(1,nh*3600)*Term(i+1)
;
end

Final=length(T);
t=[0:Final-1];

x = sin(0.1*t)*100;

Input=[t', T'];

RT = sim('FuelCell_2',t, [],  Input);

Pel=RT.yout{1}.Values.data;
Pter=RT.yout{2}.Values.data;

subplot(2,1,1), plot(t,Pel), title('Potencia
eléctrica'), grid
subplot(2,1,2), plot(t,T,'go',t,Pter), title('Potencia
térmica'), grid, legend('Referencia','P.Entregada')




%% Neuro FuelCell Ensayo

clear all
close all
clc
```

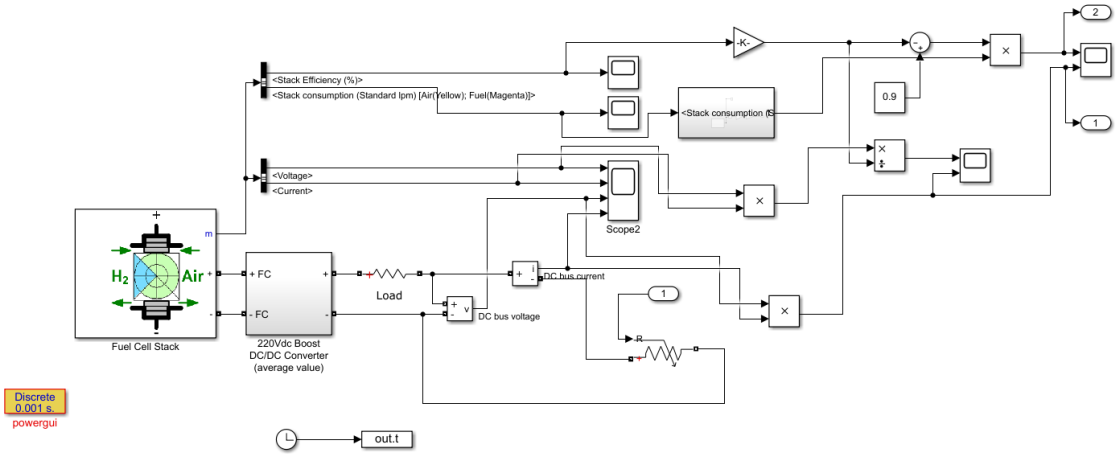
```
t=[0:0.1:2000];
Ens = sim('Ensayo_FuelCell', t, [], []);

Entrada=[Ens.yout{1}.Values.data];
Salida=[Ens.yout{2}.Values.data,
Ens.yout{3}.Values.data];



%% Simulación Neuro FuelCell

clear all
close all
clc

nh=0.1/3.6; % Número de horas durante las que se
mantiene cada potencia
Term=[0.9 1.2 1.5 2.4]; % Potencias térmicas obtenidas
del modelo en Excel en kW
Term=Term.*1000; % Conversión de las potencias de kW a
W

T(1:nh*3600)=ones(1,nh*3600)*Term(1);

for i = 1:length(Term)-1

T(i*nh*3600+1:(i+1)*nh*3600)=ones(1,nh*3600)*Term(i+1)
;
end

Final=length(T);
t=[0:Final-1];

Input=[t', T'];

RT = sim('Neuro_FuelCell',t, [],  Input);

Pel=RT.yout{1}.Values.data;
Pter=RT.yout{2}.Values.data;

subplot(2,1,1), plot(t,Pel), title('Potencia
eléctrica'), grid
subplot(2,1,2), plot(t,T,'go',t,Pter), title('Potencia
térmica'), grid

%% Simulación control térmico clásico

clear all
```

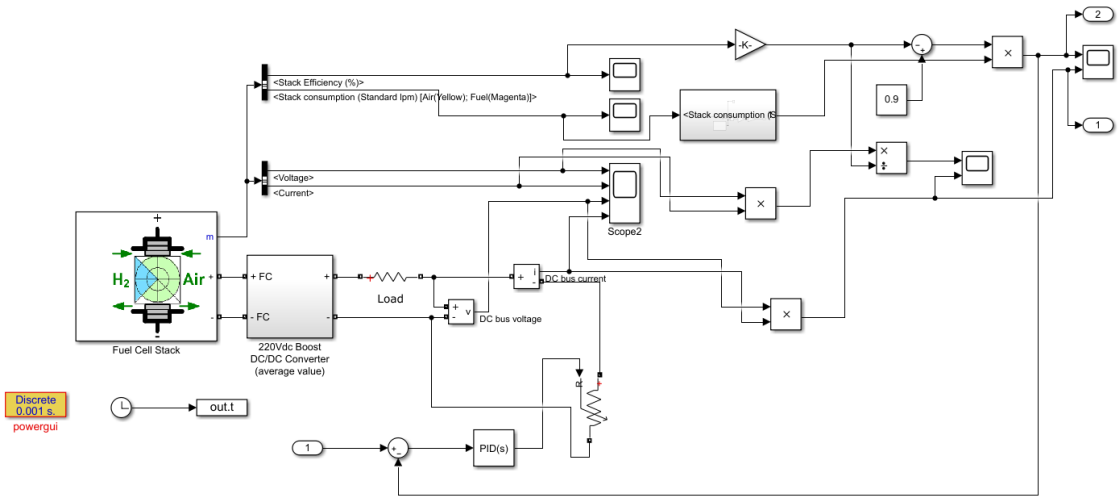
```
close all
clc

nh=0.1/3.6; % Número de horas durante las que se
mantiene cada potencia
Term=[0.9 1.2 1.5 2.4]; % Potencias térmicas obtenidas
del modelo en Excel en kW
Term=Term.*1000; % Conversión de las potencias de kW a
W

T(1:nh*3600)=ones(1,nh*3600)*Term(1);

for i = 1:length(Term)-1

T(i*nh*3600+1:(i+1)*nh*3600)=ones(1,nh*3600)*Term(i+1)
;
end

Final=length(T);
t=[0:Final-1];

x = sin(0.1*t)*250;

Input=[t', (T+x)'];

RT = sim('FuelCell_3',t, [],  Input);

Pel=RT.yout{1}.Values.data;
Pter=RT.yout{2}.Values.data;

subplot(2,1,1), plot(t,Pel), title('Potencia
eléctrica'), grid
subplot(2,1,2), plot(t,T+x,'go',t,Pter),
title('Potencia térmica'), grid,
legend('Referencia','P.Entregada')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%
%% Dibujo de las curvas de polarización
```

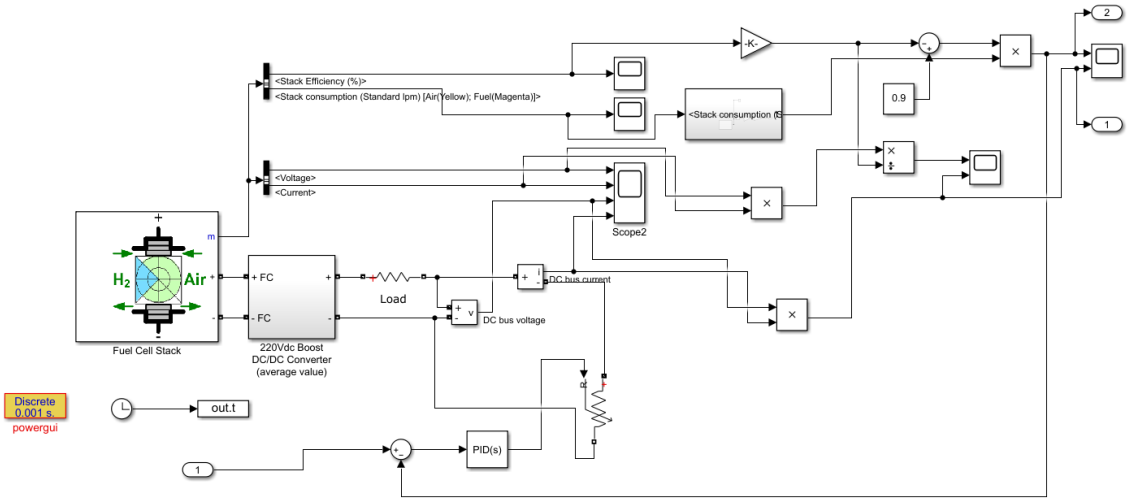
```
clear all
close all
clc

t=[0:0.1:600];
Ens = sim('Ensayo_FuelCell_Curvas', t, [], []);

I=[Ens.yout{1}.Values.data];
V=[Ens.yout{2}.Values.data];
Pel=[Ens.yout{3}.Values.data];
Pter=[Ens.yout{4}.Values.data];
Rend=[Ens.yout{5}.Values.data];
PH2=[Ens.yout{6}.Values.data];

%%

clc

eje=linspace(0,320,3200)        %(0,320,3200);
g=3;

CVI=polyfit(I,V,g);
ErrorV=immse(V,polyval(CVI,I))
CVIe=polyval(CVI,eje);

CVP=polyfit(I,Pel,g);
ErrorP=immse(Pel,polyval(CVP,I))
CVPe=polyval(CVP,eje);

CVT=polyfit(I,Pter,g);
ErrorT=immse(Pter,polyval(CVT,I))
CVTe=polyval(CVT,eje);

CVR=polyfit(I,Rend,g);
ErrorR=immse(Rend,polyval(CVR,I))
CVRe=polyval(CVR,eje);

CVH=polyfit(I,PH2,g);
ErrorH=immse(PH2,polyval(CVH,I))
CVHe=polyval(CVH,eje);

figure
hold on

subplot(2,1,1), plot(eje,CVIe,'b',eje,CVRe,'-g'),
grid, title('Curva de polarización'), xlabel('I(A)'),
ylabel('V'), legend('Curva V-I','Curva Rendimiento-I')
```

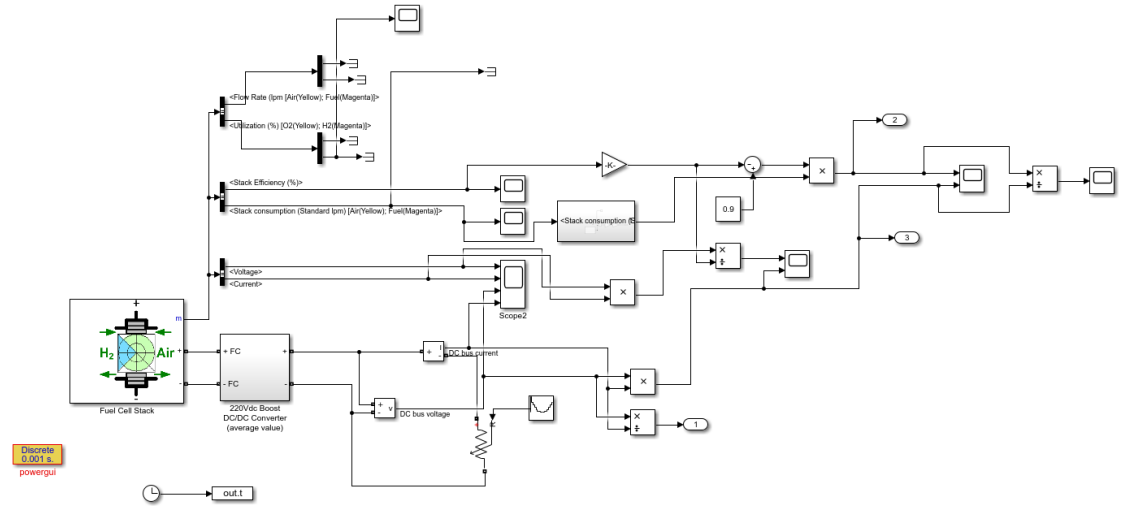
```
subplot(2,1,2), plot(eje,CVPe,eje,CVTe,eje,CVHe),
grid, title('Potencias'), xlabel('I(A)'), ylabel('W'),
legend('Curva Pelectrica-I','Curva Ptermica-I','Curva
PH2-I')

hold off

RCE=CVTe./CVPe;
% figure
% plot(eje,RCE), grid, axis([0 250 0 1])

%% Conclusión

clc

Pelec=CVP;
Pterm=CVT;
% Palim=CVH;

% Problema de optimizacion 1

q = polyder(Pelec-Pterm);

raices=roots(q)

RCEMin=polyval(CVT,raices(2))/polyval(CVP,raices(2))

%plot(eje,y)



%% Cálculo simbólico

clear all
close all
clc

%d(i*(1/C-(2*(Er-(R*T/a*F)*ln((i+iloss)/io)-
(R*T/n*F)*ln(iL/(iL-i))-i*Ri))))/di

syms Vfc(i) i Er R T a F iloss i0 n iL Ri C f(i)

Vfc(i)=Er-R*T/a/F*log((i+iloss)/i0)-
R*T/n/F*log(iL/(iL-i))-i*Ri;
f(i)=diff(i*(1/C-2*Vfc),i)

solve(f==0,i)

%%
```

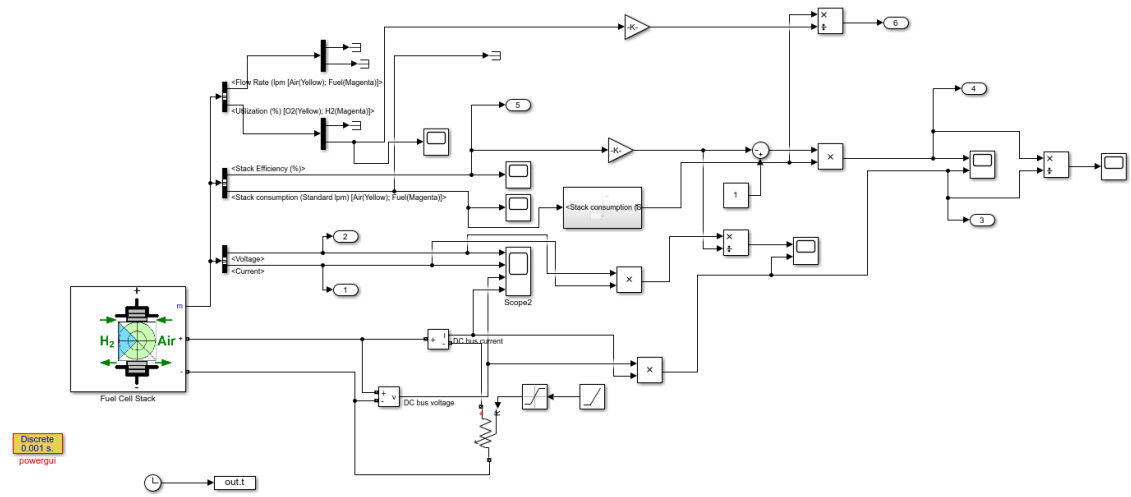
```
clear all
close all
clc

syms Vfc(i) i C V0 k f(i)

Vfc(i)=V0-k*i;
f(i)=diff(i*(1/C-2*Vfc),i)

solve(f==0,i)
```

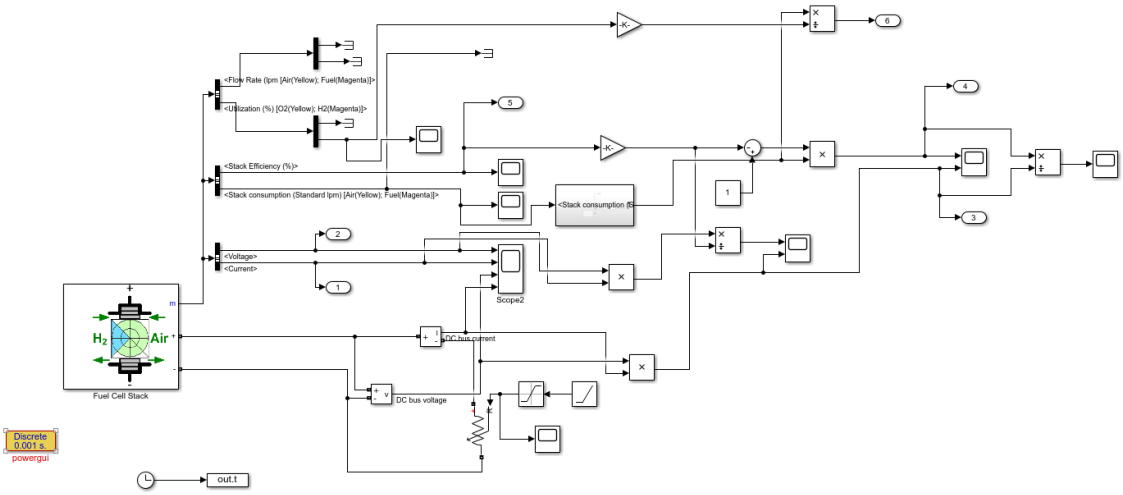


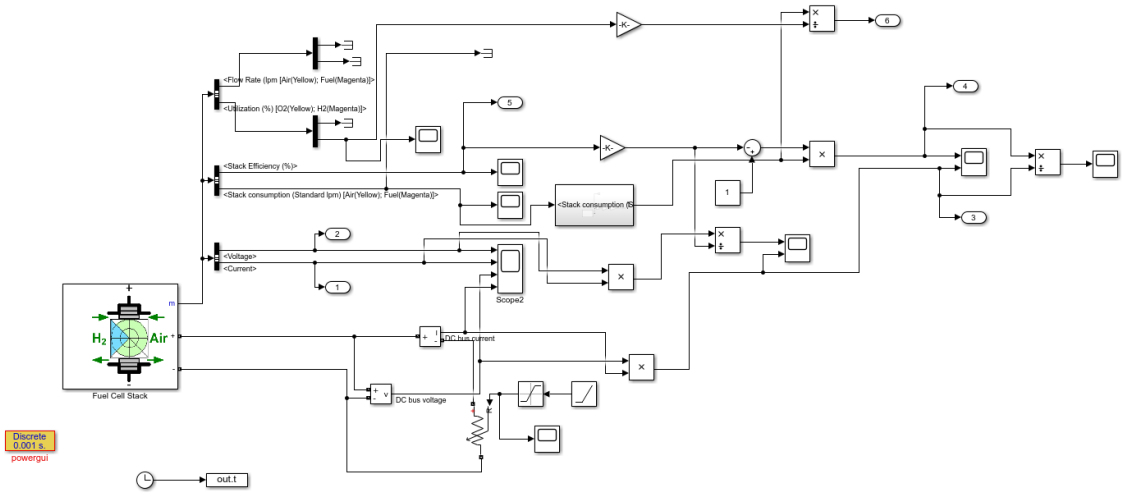


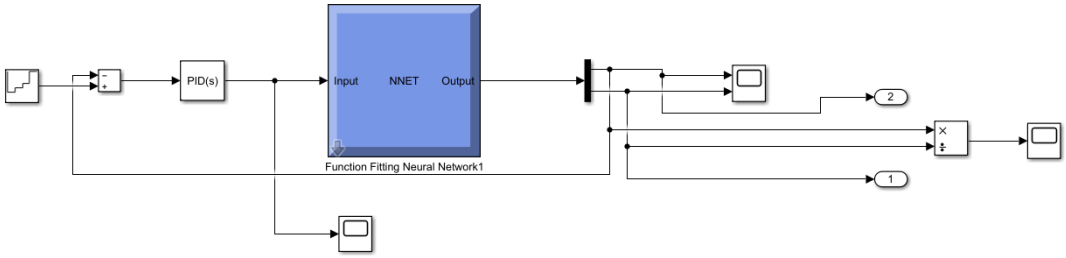

<Stack Efficiency (%)>
<Stack consumption (Standard lpm) [Air(Yellow); Fuel(Magenta)]>
<Voltage>
<Current>
Scope2
-K-
0.9
Load
DC bus current
DC bus voltage
Fuel Cell Stack
220Vdc Boost
DC/DC Converter
(average value)
Discrete
0.001 s.
powergui
out.t
PID(s)
1
2


<Flow Rate (lpm [Air(Yellow); Fuel(Magenta)]>
<Utilization (%) [O2(Yellow); H2(Magenta)]>
<Stack Efficiency (%)>
<Stack consumption (Standard lpm) [Air(Yellow); Fuel(Magenta)]>
<Voltage>
<Current>
Scope2
-K-
0.9
DC bus current
DC bus voltage
Fuel Cell Stack
220Vdc Boost
DC/DC Converter
(average value)
Discrete
0.001 s.
powergui
out.t
1
2
3


<Flow Rate (lpm [Air(Yellow); Fuel(Magenta)]>
<Utilization (%) [O2(Yellow); H2(Magenta)]>
<Stack Efficiency (%)>
<Stack consumption (Standard lpm) [Air(Yellow); Fuel(Magenta)]>
<Voltage>
<Current>
Scope2
-K-
1
DC bus current
DC bus voltage
Fuel Cell Stack
Discrete
0.001 s.
powergui
out.t
1
2
3
4
5
6


-K-
6
<Flow Rate (lpm [Air(Yellow); Fuel(Magenta)]>
<Utilization (%) [O2(Yellow); H2(Magenta)]>
5
4
<Stack Efficiency (%)>
<Stack consumption (Standard lpm) [Air(Yellow); Fuel(Magenta)]>
<Stack consumption (S
1
3
2
<Voltage>
<Current>
Scope2
m
H2
Air
DC bus current
Fuel Cell Stack
DC bus voltage
Discrete
0.001 s.
powergui
out.t


-K-
6
<Flow Rate (lpm [Air(Yellow); Fuel(Magenta)]>
<Utilization (%) [O2(Yellow); H2(Magenta)]>
5
4
<Stack Efficiency (%)>
<Stack consumption (Standard lpm) [Air(Yellow); Fuel(Magenta)]>
<Stack consumption (S
1
3
2
<Voltage>
<Current>
Scope2
m
H2
Air
DC bus current
Fuel Cell Stack
DC bus voltage
Discrete
0.001 s.
powergui
out.t


PID(s)
Input
NNET
Output
Function Fitting Neural Network1
2
1