```
%%          TÉCNICA DE FOURIER PARA EL PROCESO DE ETANOL DE HUNAG
%%
clear all
clc

global cont t U


T0 = 0.01;
tf = 15.7 + T0 - 6.7;
t = 0:T0:tf;

%% ORTONORMALIZACIÓN DE LA BASE POLINÓMICA

 tf = 15.7;
 T0 = 0.1;
 syms t real

% Defino el polinomio a ortonormalizar:

 u0 = 1;
 u1 = t;
 u2 = t^2;
 u3 = t^3;
 u4 = t^4;
 u5 = t^5;
 u6 = t^6;

% Ortonormalización:

 p0 = u0;
 p0 = p0/sqrt(int(t/t,0,tf));

 p1 = u1 - int(u1*p0,0,tf)*p0;        % ortogonalización
 p1 = p1/sqrt(int(p1*p1,0,tf));       % normalización

 p2 = u2 - int(u2*p0,0,tf)*p0 - int(u2*p1,0,tf)*p1;
 p2 = p2/sqrt(int(p2*p2,0,tf));

 p3 = u3 - int(u3*p0,0,tf)*p0 - int(u3*p1,0,tf)*p1 - int(u3*p2,0,tf)*p2;
 p3 = p3/sqrt(int(p3*p3,0,tf));

 p4 = u4 - int(u4*p0,0,tf)*p0 - int(u4*p1,0,tf)*p1 - int(u4*p2,0,tf)*p2 - int(u4*p3,0,tf)*p3;
 p4 = p4/sqrt(int(p4*p4,0,tf));

 p5 = u5 - int(u5*p0,0,tf)*p0 - int(u5*p1,0,tf)*p1 - int(u5*p2,0,tf)*p2 - int(u5*p3,0,tf)*p3 -
int(u5*p4,0,tf)*p4;
 p5 = p5/sqrt(int(p5*p5,0,tf));

 p6 = u6 - int(u6*p0,0,tf)*p0 - int(u6*p1,0,tf)*p1 - int(u6*p2,0,tf)*p2 - int(u6*p3,0,tf)*p3 -
int(u6*p4,0,tf)*p4 - int(u6*p5,0,tf)*p5;
 p6 = p6/sqrt(int(p6*p6,0,tf));

% Verifico los productos internos:

p0p0 = int(p0*p0,0,tf)
p0p1 = int(p0*p1,0,tf)
```

```
p0p2 = int(p0*p2,0,tf)
p0p3 = int(p0*p3,0,tf)
p0p4 = int(p0*p4,0,tf)
p0p5 = int(p0*p5,0,tf)
p0p6 = int(p0*p6,0,tf)

p1p0 = int(p1*p0,0,tf)
p1p1 = int(p1*p1,0,tf)
p1p2 = int(p1*p2,0,tf)
p1p3 = int(p1*p3,0,tf)
p1p4 = int(p1*p4,0,tf)
p1p5 = int(p1*p5,0,tf)
p1p6 = int(p1*p6,0,tf)

p2p0 = int(p2*p0,0,tf)
p2p1 = int(p2*p1,0,tf)
p2p2 = int(p2*p2,0,tf)
p2p3 = int(p2*p3,0,tf)
p2p4 = int(p2*p4,0,tf)
p2p5 = int(p2*p5,0,tf)
p2p6 = int(p2*p6,0,tf)

p3p0 = int(p3*p0,0,tf)
p3p1 = int(p3*p1,0,tf)
p3p2 = int(p3*p2,0,tf)
p3p3 = int(p3*p3,0,tf)
p3p4 = int(p3*p4,0,tf)
p3p5 = int(p3*p5,0,tf)
p3p6 = int(p3*p6,0,tf)

p4p0 = int(p4*p0,0,tf)
p4p1 = int(p4*p1,0,tf)
p4p2 = int(p4*p2,0,tf)
p4p3 = int(p4*p3,0,tf)
p4p4 = int(p4*p4,0,tf)
p4p5 = int(p4*p5,0,tf)
p4p6 = int(p4*p6,0,tf)

p5p0 = int(p5*p0,0,tf)
p5p1 = int(p5*p1,0,tf)
p5p2 = int(p5*p2,0,tf)
p5p3 = int(p5*p3,0,tf)
p5p4 = int(p5*p4,0,tf)
p5p5 = int(p5*p5,0,tf)
p5p6 = int(p5*p6,0,tf)

p6p0 = int(p6*p0,0,tf)
p6p1 = int(p6*p1,0,tf)
p6p2 = int(p6*p2,0,tf)
p6p3 = int(p6*p3,0,tf)
p6p4 = int(p6*p4,0,tf)
p6p5 = int(p6*p5,0,tf)
p6p6 = int(p6*p6,0,tf)


% El polinomio que ajusta a una función es el siguiente:
% f ~ c0*p0 + c1*p1 + c2*p2 + c3*p3 + c4*p4 + c5*p5 + c6*p6
```

```
%% Defino la base ortonormalizada

p0 = (10^(1/2)*157^(1/2))/157;
p1 = (20*30^(1/2)*157^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/24649;
p2 = -(3000*2^(1/2)*157^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/3869893;
p3 = (20000*70^(1/2)*157^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/607573201;
p4 = -(2100000*10^(1/2)*157^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/95388992557;
p5 = (25200000*110^(1/2)*157^(1/2)*(t^5 - (607573201*10^(1/2)*157^(1/2)*1570^(1/2))/6000000 +
(10^(1/2)*157^(1/2)*1570^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/40 -
(157*70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/2520 +
(24649*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/1120 -
(3869893*30^(1/2)*157^(1/2)*4710^(1/2)*(t -
(10^(1/2)*157^(1/2)*1570^(1/2))/200))/420000))/14976071831449;
p6 = -(924000000*130^(1/2)*157^(1/2)*((95388992557*10^(1/2)*157^(1/2)*1570^(1/2))/70000000 - t^6 -
(1413*10^(1/2)*157^(1/2)*1570^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/2200 +
(24649*70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/21000 -
(3869893*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/11200 +
(1822719603*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/14000000 +
(3*110^(1/2)*157^(1/2)*17270^(1/2)*(t^5 - (607573201*10^(1/2)*157^(1/2)*1570^(1/2))/6000000 +
(10^(1/2)*157^(1/2)*1570^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/40 -
(157*70^(1/2)*157^(1/2)*10990^(1/2)*(t^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/2520 +
(24649*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/1120 -
(3869893*30^(1/2)*157^(1/2)*4710^(1/2)*(t -
(10^(1/2)*157^(1/2)*1570^(1/2))/200))/420000))/1100))/2351243277537493;

%% TÉRMINOS DE FOURIER
% Defino los términos de la base de Fourier:

 w0 = ones(length(t),1);
 w1 = cos(1*2*pi/tf*t);
 w2 = cos(2*2*pi/tf*t);
 w3 = cos(3*2*pi/tf*t);
 v1 = sin(1*2*pi/tf*t);
 v2 = sin(2*2*pi/tf*t);
 v3 = sin(3*2*pi/tf*t);

% Los normalizo:

 w0 = w0/sqrt(int(w0*w0*t/t,0,tf));
 w1 = w1/sqrt(int(w1*w1,0,tf));
 w2 = w2/sqrt(int(w2*w2,0,tf));
 w3 = w3/sqrt(int(w3*w3,0,tf));
 v1 = v1/sqrt(int(v1*v1,0,tf));
 v2 = v2/sqrt(int(v2*v2,0,tf));
 v3 = v3/sqrt(int(v3*v3,0,tf));

% La base de Fourier queda definida de la siguiente manera:
% bF = a0*w0 + b1*v1 + a1*w1 + b2*v2 + a2*w2 + b3*v3 + a3*w3
% Como se busca encontrar el contenido frecuencial del polinomio, se iguala bF al polinomio anteriormente
encontrado:
 % a0*w0 + b1*v1 + a1*w1 + b2*v2 + a2*w2 + b3*v3 + a3*w3 ~ c0*p0 + c1*p1 + c2*p2 + c3*p3 + c4*p4
+ c5*p5 + c6*p6
% Los parámetros de bF se buscan, y en base a eso se calculan los del polinomio.
% Se multiplica miembro a miembro por los términos de bF para encontrar el valor de los coeficientes del
polinomio:

 w0w0 = int(w0*w0,0,tf);
 w1w1 = int(w1*w1,0,tf);
 w2w2 = int(w2*w2,0,tf);
 w3w3 = int(w3*w3,0,tf);
 v1v1 = int(v1*v1,0,tf);
 v2v2 = int(v2*v2,0,tf);
 v3v3 = int(v3*v3,0,tf);

 w0p0 = vpa(int(w0*p0,0,tf));
 v1p1 = vpa(int(v1*p1,0,tf));
 w1p2 = vpa(int(w1*p2,0,tf));
 v2p3 = vpa(int(v2*p3,0,tf));
 w2p4 = vpa(int(w2*p4,0,tf));
 v3p5 = vpa(int(v3*p5,0,tf));
 w3p6 = vpa(int(w3*p6,0,tf));

% Parámetros del sistema:

um1 = 1.8823;
um2 = 1.7098;
Yp1s1 = 0.5085;
Yp2s1 = 0.5331;
Yp1s2 =0.5098;
Yp2s2 = 0.4462;
Ks1 = 159.7525;
Ks1I = 94.2332;
Kp11 = 238.3924;
Kp1I = 2.7378;
Ks2 = 0.0726;
Ks2I = 9.0048;
Kp12 = 35.9587;
kp1I = 9.9722;
vs1p1 = 1.5051;
Ks1p1 = 1.3409;
ks1p1 = 18.6121;
vs2p2 = 0.1609;
Ks2p2 = 0.4310;
ks2p2 = 1.150;
vs1p2 = 0.00235;
Ks1p2 = 6.7116;
ks1p2 = 0.5863;
vs2p1 = 0.3321;
Ks2p1 = 0.9129;
ks2p1 = 1000;
landa = 0.5;
Sf = 300;

% Condiciones iniciales:

X0 = 1.5;
P10 = 5.3;
P20 = 0.0001;
S10 = 8.6;
S20 = 8.6;
V0 = 1.35;

% Defino los p:

p0 = (10^(1/2)*157^(1/2))/157;
p0 = p0*ones(1,length(t));
p1 = (20*30^(1/2)*157^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/24649;
p2 = -(3000*2^(1/2)*157^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/3869893;
p3 = (20000*70^(1/2)*157^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/607573201;
p4 = -(2100000*10^(1/2)*157^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t.^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/95388992557;
p5 = (25200000*110^(1/2)*157^(1/2)*(t.^5 - (607573201*10^(1/2)*157^(1/2)*1570^(1/2))/6000000 +
(10^(1/2)*157^(1/2)*1570^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t.^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/40 -
(157*70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/2520 +
(24649*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/1120 -
(3869893*30^(1/2)*157^(1/2)*4710^(1/2)*(t -
(10^(1/2)*157^(1/2)*1570^(1/2))/200))/420000))/14976071831449;
p6 = -(924000000*130^(1/2)*157^(1/2)*((95388992557*10^(1/2)*157^(1/2)*1570^(1/2))/70000000 - t.^6 -
(1413*10^(1/2)*157^(1/2)*1570^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t.^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/2200 +
(24649*70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/21000 -
(3869893*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/11200 +
(1822719603*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/14000000 +
(3*110^(1/2)*157^(1/2)*17270^(1/2)*(t.^5 - (607573201*10^(1/2)*157^(1/2)*1570^(1/2))/6000000 +
(10^(1/2)*157^(1/2)*1570^(1/2)*((3869893*10^(1/2)*157^(1/2)*1570^(1/2))/500000 - t.^4 +
(70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/350 -
(471*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/350 +
(24649*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/37500))/40 -
(157*70^(1/2)*157^(1/2)*10990^(1/2)*(t.^3 - (24649*10^(1/2)*157^(1/2)*1570^(1/2))/40000 +
(3*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/40 -
(471*30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/10000))/2520 +
(24649*2^(1/2)*157^(1/2)*314^(1/2)*((157*10^(1/2)*157^(1/2)*1570^(1/2))/3000 - t.^2 +
(30^(1/2)*157^(1/2)*4710^(1/2)*(t - (10^(1/2)*157^(1/2)*1570^(1/2))/200))/300))/1120 -
(3869893*30^(1/2)*157^(1/2)*4710^(1/2)*(t -
(10^(1/2)*157^(1/2)*1570^(1/2))/200))/420000))/1100))/2351243277537493;

% Informo los respectivos productos internos:
```

```
w0w0 = 1;
w1w1 = 1;
w2w2 = 1;
w3w3 = 1;
v1v1 = 1;
v2v2 = 1;
v3v3 = 1;

w0p0 = 1.0;
v1p1 = -0.77969680123367619495101823524062;
w1p2 = 0.96121714660194783235749232546 6;
v2p3 = -0.36923915837367337620892943804633;
w2p4 = 0.78884457979943834633924363974589;
v3p5 = 0.031010068363345013866076934913434;
w3p6 = 0.46684057854328370165852025736748;

% Para facilitar los cálculos, defino la matriz A con los productos internos anteriores:

A = [w0p0; v1p1; w1p2; v2p3; w2p4; v3p5; w3p6];

% Armo una matriz con los polinomios de la base ortonormalizada:

P = [p0; p1; p2; p3; p4; p5; p6];

%% OPTIMIZACIÓN DE LOS COEFICIENTES DE FOURIER

% Armo ciclo para suponer a0, b1, a1, b2, a2, b3 y a3, y calcular c0, c1, c2, c3, c4, c5 y c6.

% Cargo información necesaria para el ciclo:

parF = 3;        % Cantidad de parámetros de la base de Fourier

% Defino límites para el sistema:

LSU = 1;         % Valor máximo que puede tomar la acción de control
LIU = 0;         % Valor mínimo que puede tomar la acción de control
Vmax = 5;        % Volumen máximo de operación del reactor

% Cargo datos para los algoritmos de Monte Carlo (MC) y Algoritmo Genético (AG):

m = parF;           % Componentes de cada individuo (cant. de variables a optimizar)
NMC = 100;       % Población inicial generada con MC
NAG = 100;       % Población que se mantiene activa en el AG
L = 30 + 10;         % N° de generaciones

% Defino los límites para los coeficientes de Fourier

li = -3;
ls = 3;

% Defino la cantidad de individuos que se generarán en cada etapa del AG:

Se = NAG*0.2;    % Selección=20
Cr = NAG*0.2;    % Cruzamiento=20
Mu = NAG*0.4;    % Mutación=40
Al = NAG*0.2;    % Aleatorio=20
```

```
% Trunco las matrices A y P según el orden del polinomio:

A = A(1:parF);
P = P(1:parF,:);

%% MONTE CARLO

MC = zeros(NMC,parF+1);

k = 1;
while k <= NMC
   k
 a0=random('unif',-2,2);
 a1=random('unif',li,ls);
 b1=random('unif',li,ls);
 a2=random('unif',li,ls);
 b2=random('unif',li,ls);
 a3=random('unif',li,ls);
 b3=random('unif',li,ls);
 a4=random('unif',li,ls);
 b4=random('unif',li,ls);
 a5=random('unif',li,ls);
 b5=random('unif',li,ls);

 coefF = [a0; b1; a1; b2; a2; b3; a3; b4; a4; b5; a5];
 coefF = coefF(1:parF);

 % Armo un vector B que incluye los coeficientes de Fourier y los productos internos correspondientes:

 B = [a0*w0w0; b1*v1v1; a1*w1w1; b2*v2v2; a2*w2w2; b3*v3v3; a3*w3w3];

 % Trunco B dependiendo el número de parámetros a utilizar

 B = B(1:parF);

 % Calculo los coeficientes del polinomio:

 for i = 1:parF

    C(i) = [B(i)/A(i)];

 end

 % Obtengo el perfil de la acción de control:

 U = C*P;

 Volumen = sum(U)*T0 + V0;

 % Impongo condiciones para considerar o no el U calculado:

 if (min(U)>=LIU && max(U)<=LSU && Volumen<=Vmax)

         % Simulo el proceso:

         cont = 0;
```

```
          sim('Perfiles.mdl');

          MC(k,:) = [P1(length(P1)) coefF'];

          k = k+1;
 end

 end

% Ordeno las columnas de la matriz MC según P1 decreciente:

[~,s] = sort(MC(:,1),'descend');
MCord = MC(s,:);

%% ALGORITMO GENÉTICO

% Selecciono los individuos que entrarán al AG

RANKING = zeros(L,parF+1);

for y=1:L

% Armo matriz de selección truncando la matriz MCord:

SE = MCord(1:Se,:);

Padres = SE(:,2:parF+1); % Saco de la matriz anterior el valor de P1, dejando solo los coeficientes de
Fourier.

% CRUZAMIENTO:

corte = round(parF/2);  % posición en la que se corta el individuo

AA = randperm(Se);
BB = randperm(Se);

mama = Padres(:,1:corte);

papa = Padres(:,corte+1:end);

CR = [mama(AA',:) papa(BB',:)];

% MUTACIÓN:

MU = [Padres; Padres];

for n=1:Mu

 mut = round(random('unif',1,parF));  % Elijo posición a mutar.

 ab=random('unif',li,ls);

 MU(n,mut) = ab;

end
```

```
% ALEATORIO

AL = zeros(Al,11);

for z=1:Al

 a0=random('unif',li,ls);
 a1=random('unif',li,ls);
 b1=random('unif',li,ls);
 a2=random('unif',li,ls);
 b2=random('unif',li,ls);
 a3=random('unif',li,ls);
 b3=random('unif',li,ls);
 a4=random('unif',li,ls);
 b4=random('unif',li,ls);
 a5=random('unif',li,ls);
 b5=random('unif',li,ls);

 AL(z,:) = [a0 b1 a1 b2 a2 b3 a3 b4 a4 b5 a5];

end

AL = AL(:,1:parF);

% Uno las matrices generadas en una nueva matriz denominada AG:

AG = [Padres; CR; MU; AL];

% Evalúo la función objetivo en cada individuo:

b = [w0w0 v1v1 w1w1 v2v2 w2w2 v3v3 w3w3];
b = b(1:parF);

AlGen = zeros(NAG,parF+1);

for r=1:NAG

    D = AG(r,:).*b;

 % Calculo los coeficientes del polinomio:

 for q = 1:parF

     C(q) = [D(q)/A(q)];

 end

 % Obtengo el perfil de la acción de control:

 U = C*P;

 Volumen = sum(U)*T0 + V0;

 % Impongo condiciones para considerar o no el U calculado:

 if (min(U)>=LIU && max(U)<=LSU && Volumen<=Vmax)
```

```
    % Simulo el proceso:

    cont = 0;

    sim('Perfiles.mdl');

    AlGen(r,:) = [P1(length(P1)) D];

 end

 end

% Ordeno las columnas de la matriz AlGen según P1 decreciente:

[~,s] = sort(AlGen(:,1),'descend');
MCord = AlGen(s,:);

RANKING(y,:) = MCord(1,:);

end

%% METODO DEL GRADIENTE PARA AFINAR LA BUSQUEDA

% La búsqueda comienza con los coeficientes elegidos por el AG:

Elegido = MCord(1,2:parF+1);

F = Elegido.*b;

 % Calculo los coeficientes del polinomio:

 for o = 1:parF

     C(o) = [F(o)/A(o)];

 end

 % Defino el número de iteraciones:

 M = 8;

 % Defino los coeficientes como:
 clear x y z;

 C = [1.6855   -0.0857    0.9627];
 x(1) = C(1);
 y(1) = C(2);
 z(1) = C(3);


%  x(1) = 1.1278;
%  y(1) = -0.7127;
%  z(1) = 0.6970;  %%con tf = 15.7 - 4.7

% x(1) =  1.6766;
% y(1) = -0.0767;   %%con tf = 15.7 - 6.7
% z(1) = 0.9604;
%  x(1) = 0.9025;
%  y(1) = -0.9593;
%  z(1) = 0.275;

%  x(1) = 0.9229 ;
%  y(1) = -0.8899;
%  z(1) = 0.3331;

% %  x(1) = 0.9231;
% %  y(1) = -0.9065;
% %  z(1) = 0.3312;

%  x(1) = 0.9281;
%  y(1) = -0.9325;
%  z(1) = 0.3302;

% Defino los incrementos para x,y,z:

 Ix = 0.0002;
 Iy = 0.0002;
 Iz = 0.0002;

 % Defino un incremento:

 alfa = 0.0001/2;

 % Busco el extremo con el gradiente:
 clear s
 for s = 1:M

 U = x(s)*P(1,:) + y(s)*P(2,:) + z(s)*P(3,:);
 cont = 0;
 sim('Perfiles.mdl');
 J = P1(length(P1));
s
 U = (x(s)+Ix)*P(1,:) + y(s)*P(2,:) + z(s)*P(3,:);
 cont = 0;
 sim('Perfiles.mdl');
 Jx = P1(length(P1));

 U = x(s)*P(1,:) + (y(s)+Iy)*P(2,:) + z(s)*P(3,:);
 cont = 0;
 sim('Perfiles.mdl');
 Jy = P1(length(P1));

 U = x(s)*P(1,:) + y(s)*P(2,:) + (z(s)+Iz)*P(3,:);
 cont = 0;
 sim('Perfiles.mdl');
 Jz = P1(length(P1));

 Jx = (Jx-J)/Ix;
 Jy = (Jy-J)/Iy;
 Jz = (Jz-J)/Iz;

 x(s+1) = x(s) + alfa*Jx;
 y(s+1) = y(s) + alfa*Jy;
 z(s+1) = z(s) + alfa*Jz;

 end

 % Busco el valor mínimo de x,y,z:

 xMin = x(length(x));
 yMin = y(length(x));
 zMin = z(length(x));

 C = [xMin yMin zMin]

 % Calculo el caudal:

 U = C*P;

 Volumen = sum(U)*T0 + V0;

 % Simulo el proceso:

 cont = 0;

 sim('Perfiles.mdl');

%% GRAFICO LOS RESULTADOS:

U = U';

figure(1)

subplot(1,4,1); plot(U,'r')
hold
subplot(1,4,2); plot(tsimul,P1,'g', tsimul,V,'b')
subplot(1,4,3); plot(1:1:NMC, MC(:,1),'.')
subplot(1,4,4); plot(1:1:L, RANKING(:,1),'.')
```