ANEXOS

1. Cálculo de la matriz RGA

```
clear all;
close all;
clc;
s = tf('s');
G11 = (0.0301*(40.90*s+1))/((148.61*s+1)*(147.85*s+1))
G12 = 0;
G21 = (-9.79*(10.67*s+1))/((29.31*s+1)*(488.27*s+1))
G22 = (-10.67*(3.07*s+1))/((8.42*s+1)*(268.6*s+1))
G = [G11 G12
     G21 G22];
G1 = dcgain(G)
Gt = inv(G1')
RGA = G1.*Gt
```

2. Newton- Raphson

```
function newtonecuacion
%w0 es el valor inicial,
syms w
func= input('ingrese la función: ');
w=input('Ingrese el valor inicial: ');
precision=input('Ingrese la precisión: ');
dfunc=diff(func,1);
it=0; w0=w;
d=subs(func,w0)/subs(dfunc,w0);
while abs(d)>precision
    w1=w0-d;
    it=it+1;
    w0=w1;
    d=subs(func,w0)/subs(dfunc,w0);
end;
respuesta=w0
iteracion=it
```

3. Diseño de controladores - Método MIGO

```
close all; clc; clear all;

%%Diseño para el controlador de Gp11 //Gp22
s = tf('s');
w = sym('w');

disp('La función de transferencia tiene la forma:
%K(T1*s+1)/((T2*s+1)*(T3*s+1))')
K  = input('K = ');
T1 = input('T1 = ');
T2 = input('T2 = ');
T3 = input('T3 = ');

K = 0.0301; T1 = 40.90; T2 = 148.61; T3 = 147.85; %Gp11
K = -10.67; T1 = 3.07; T2 = 8.42; T3 = 268.6; %Gp22

s = tf('s');
w = sym('w');
r2 = 0.6;
c2 =- 0.55;
```

```
Kd = -5;

%Función de transferencia
P = (K*(T1*s+1))/((T2*s+1)*(T3*s+1))
[num,den] = tfdata(P,'v');

r = w*i;

%Numerador
N = num(1,1)*r^2 + num(1,2)*r + num(1,3);
Nn = real(N) + imag(N)*i;

%Denominador
D = den(1,1)*r^2 + den(1,2)*r + den(1,3);
Dn = real(D) + imag(D)*i;
Dc = conj(Dn);

%Pasando a la forma: a + b*i
P1 = Nn*Dc;
P11 = real(P1);
P12 = imag(P1);
P2 = Dn * Dc;
a = P11/P2;
b = P12/P2;

a2b2 = (a*a)+(b*b);
c = b/a;

alfa = sqrt(a2b2);
alfad = diff(alfa,'w');
phi = atan(c); %para senos y cosenos
phi2 = atan(c)*(180/pi); %solo
phid = diff(phi,'w');

E = (r2+c2*sin(phi));
F = (alfad/alfa)-(1/w);
G = E*F;
H =2*Kd*alfa;
J = cos(phi)*-c2*phid;
hpid =   G + J + H;
%w = solve(hpid)--Newton Raphson
w = input ('w a probar:  ');
alfa1 = eval(alfa);
alfa2 = eval(alfad);
phi3 = eval(phi);%para senos y cosenos
phi4= eval(phi2); %solo
phi5 = eval(phid); %derivada
hpid1 = eval(hpid);

%Controlador
Kp2 = -c2*cos(phi3)/alfa1;
Ki2 = (-w*(r2+c2*sin(phi3)))/alfa1 + (w*w*Kd);
Kd2 = Kd;
T12 = -Kp2/Ki2;
Td2 = Kd2/Kp2;

C = Kp2 + Ki2/s + Kd2*s;
G = P*C;
```

```
%Diagrama de Bode
S   = 1/(1+G);
[mag,fase,w] = bode(S);
[Mp,k] = max(mag);
w1 = w(k);
i = sqrt(-1);
r3 = w1*i;

% Sensibilidad
[num,den] = tfdata(S,'v');
num(1,1); num(1,2);
den(1,1); den(1,2);
A = num(1,1)*r3^3 + num(1,2)*r3^2 + num(1,3)*r3^1 + num(1,4);
B=  den(1,1)*r3^3 + den(1,2)*r3^2 + den(1,3)*r3^1 + den(1,4);
T = A/B;
T_1= abs(T)

nyquist(G)
```