

6 ANEXOS

6.1 HOJA DE CARACTERISTICAS DE LA TARJETA DT2811

Table 1: DT2811 Supported Options

DT2811	A/D	D/A	DIN	DOUT	Serial I/O	Counter/Timer
Total Subsystems on Board	1	1	1	1	0	0
Maximum Throughput OLSSCE_MAX_THROUGHPUT	100 Hz (*1)	50 kHz	0	0		
Minimum Throughput OLSSCE_MIN_THROUGHPUT	1 Hz	1 Hz	0	0		
Maximum Retrigger Frequency OLSSCE_MAXRETRIGGER	0	0	0	0		
Minimum Retrigger Frequency OLSSCE_MINRETRIGGER	0	0	0	0		
Maximum External Clock Divider OLSSCE_MAXCLOCKDIVIDER	4.29e9	1	1	1		
Minimum External Clock Divider OLSSCE_MINCLOCKDIVIDER	1	1	1	1		
Base Clock Frequency OLSSCE_BASECLOCK	600 kHz	600 kHz	0	0		
SE Channels OLSSC_MAXSECHANS	16 (*2)	0	1	1		
DI Channels OLSSC_MAXDICHANS	8 (*2)	2	0	0		
CGL Depth OLSSC_CGLDEPTH	1	1	1	1		
Number of Filters OLSSC_NUMFILTERS	1	1	1	1		
Number of Gains OLSSC_NUMGAINS	4	1	1	1		
Number of Voltage Ranges OLSSC_NUMRANGES	1	1	1	1		
Number of DMA Channels OLSSC_NUMDMACHANS	0	0	0	0		
Number of Channels OLSSC_NUMCHANNELS	16SE/8DI (*2)	2	1	1		
Number of Extra Clocks OLSSC_NUMEXTRACLOCKS	0	0	0	0		

Table 1: DT2811 Supported Options

DT2811	A/D	D/A	DIN	DOUT	Serial I/O	Counter/Timer
Total Subsystems on Board	1	1	1	1	0	0
Number of Extra Triggers OLSSC_NUMEXTRATRIGGERS	0	0	0	0		
Number of Resolutions OLSSC_NUMRESOLUTIONS	1	1	1	1		
Interrupt Support OLSSC_SUP_INTERRUPT	Yes					
SE Support OLSSC_SUP_SINGLEENDED	Yes (*2)		Yes	Yes		
DI Support OLSSC_SUP_DIFFERENTIAL	Yes (*2)					
Binary Coding Support OLSSC_SUP_BINARY	Yes					
Two's Complement Support OLSSC_SUP_2SCOMP						
Software Trigger Support OLSSC_SUP_SOFTTRIG	Yes	Yes	Yes	Yes		
External Trigger Support OLSSC_SUP_EXTERNTRIG	Yes					
Threshold Trigger Support OLSSC_SUP_THRESHTRIGPOS						
Threshold Trigger Support OLSSC_SUP_THRESHTRIGNEG						
Analog Event Trigger Support OLSSC_SUP_ANALOGEVENTTRIG						
Digital Event Trigger Support OLSSC_SUP_DIGITALEVENTTRIG						
Timer Event Trigger Support OLSSC_SUP_TIMEREVENTTRIG						
Triggered Scan Support OLSSC_SUP_TRIGSCAN						
Internal Clock Support OLSSC_SUP_INTCLOCK	Yes	Yes	Yes	Yes		
External Clock Support OLSSC_SUP_EXTCLOCK	Yes					

Table 1: DT2811 Supported Options

DT2811	A/D	D/A	DIN	DOUT	Serial I/O	Counter/Timer
Total Subsystems on Board	1	1	1	1	0	0
Software Calibration Support OLSSC_SUP_SWCAL						
DT2896 Channel Expansion Support OLSSC_SUP_EXP2896						
DT727 Channel Expansion OLSSC_SUP_EXP727						
Filter/Channel Support OLSSC_SUP_FILTERPERCHAN						
DT-Connect Support OLSSC_SUP_DTCONNECT						
FIFO in Data Path Support OLSSC_SUP_FIFO						
Programmable Gain Support OLSSC_SUP_PROGRAMGAIN	Yes					
Data Processing Capability OLSSC_SUP_PROCESSOR						
Software Programmable Resolution OLSSC_SUP_SWRESOLUTION						
Continuous Operation Support OLSSC_SUP_CONTINUOUS	Yes					
Pause Operation Support OLSSC_SUP_PAUSE						
Multiple Buffer Wrap Mode Support OLSSC_SUP_WRPMULTIPLE	Yes					
Single Buffer Wrap Mode Support OLSSC_SUP_WRPSINGLE	Yes					
Single Value Operation Support OLSSC_SUP_SINGLEVALUE	Yes	Yes	Yes	Yes		
Asynchronous Operation Support OLSSC_SUP_POSTMESSAGE	Yes					
Cascading Support OLSSC_SUP_CASCADING						
Event Count Mode Support OLSC_SUP_CTMODE_COUNT						

Table 1: DT2811 Supported Options

DT2811	A/D	D/A	DIN	DOUT	Serial I/O	Counter/Timer
Total Subsystems on Board	1	1	1	1	0	0
Rate Generation Mode Support OLSSC_SUP_CTMODE_RATE						
One-Shot Mode Support OLSSC_SUP_CTMODE_ONESHOT						
Repeatable One-Shot Mode Support OLSSC_SUP_CTMODE_ONESHOT_RPT						
Maximum Synchronous Digital I/O Value OLSSC_MAX_DIGITALIOLIST_VALUE						
Continuous DT-Connect Support OLSSC_SUP_DTCONNECT_CONTINUOUS						
Burst DT-Connect Support OLSSC_SUP_DTCONNECT_BURST						
Channel List Inhibit Support OLSSC_SUP_CHANNEL_LIST_INHIBIT						
Synchronous Digital I/O Support OLSSC_SUP_SYNCHRONOUS_DIGITALIO						
Simultaneous Start List Support OLSSC_SUP_SIMULTANEOUS_START						
Inprocess Buffer Flush Support OLSSC_SUP_INPROCESSFLUSH	Yes					
Range per Channel Support OLSSC_SUP_RANGEPERCHANNEL						
Simultaneous Sample and Hold Support OLSSC_SUP_SIMULTANEOUS_SH						
Random Channel Gain List Support OLSSC_SUP_RANDOM_CGL	Yes	Yes	Yes	Yes		
Sequential Channel Gain List Support OLSSC_SUP_SEQUENTIAL_CGL	Yes	Yes	Yes	Yes		
Zero Start Sequential Channel Gain List Support OLSSC_SUP_ZEROSEQUENTIAL_CGL	Yes	Yes	Yes	Yes		
Supports Gap Free Data with No DMA OLSSC_SUP_GAPFREE_NODMA	Yes					
Supports Gap Free Data with Single DMA OLSSC_SUP_GAPFREE_SINGLEDMA						

6.3 VENTANAS DE LA APLICACIÓN

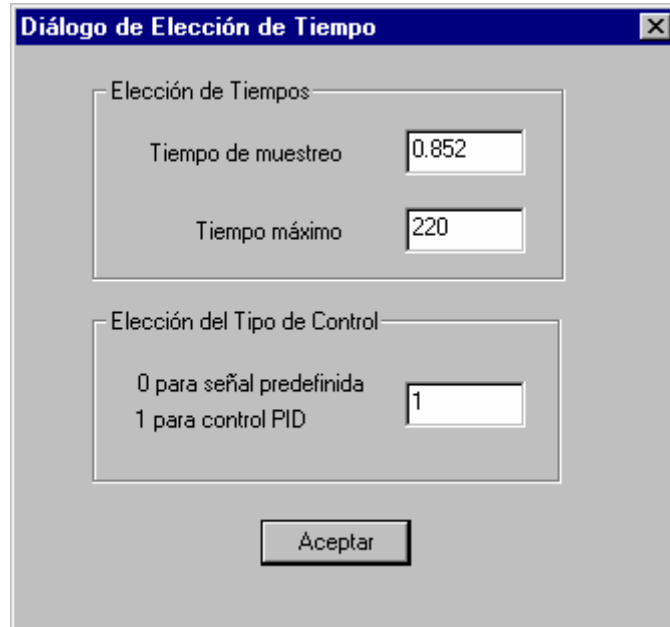


Figura A : Cuadro de diálogo inicial.

Utilizado para la elección del tiempo de muestreo, el tiempo final del experimento y el tipo de control que se utilizará.

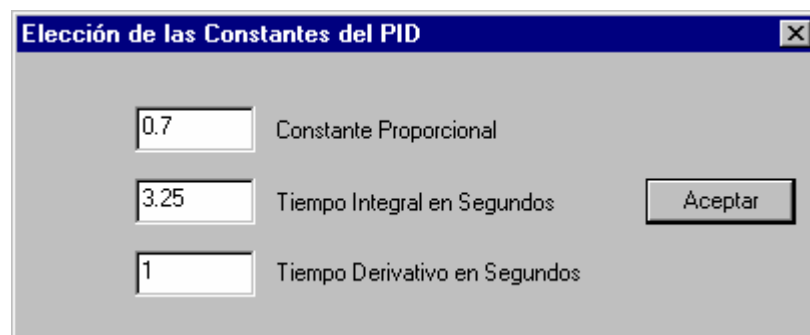


Figura B : Cuadro de diálogo PID.

Utilizado para la elección de los parámetros de un PID en el caso de tipo de control PID. Los parámetros son la constante proporcional, el tiempo integral y el tiempo derivativo.

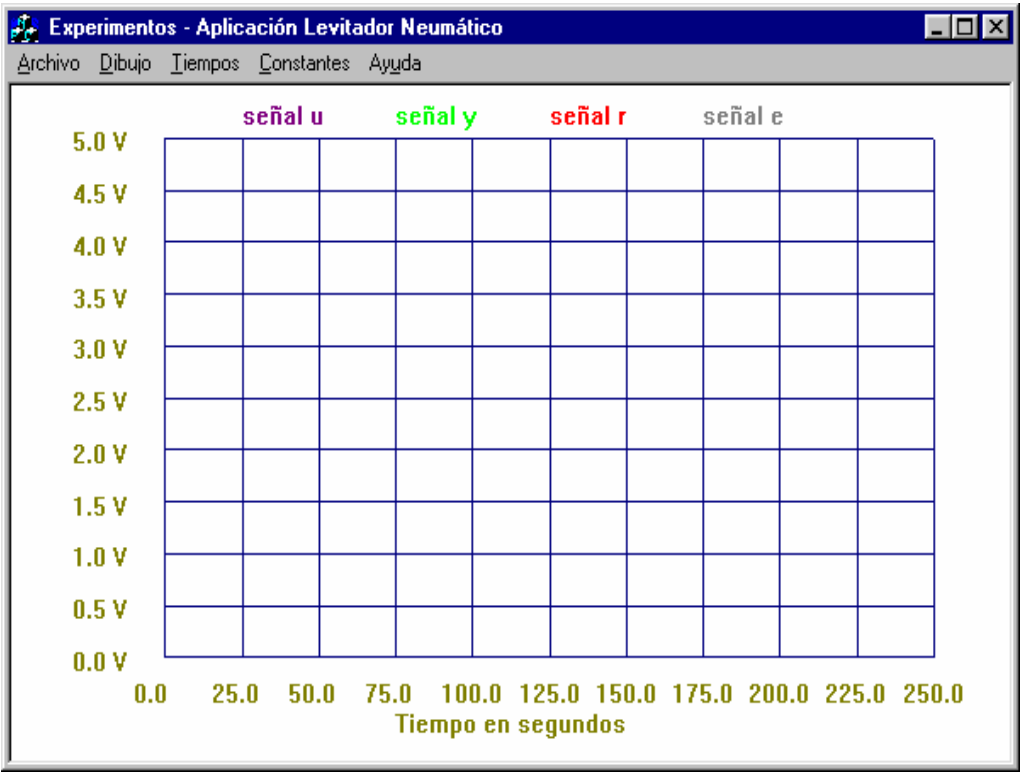


Figura C : Ventana principal de la aplicación de levitador neumático

6.4 ABREVIATURAS USADAS EN EL PROYECTO

- **A/D** : Conversión Analógico/Digital
- **ANSI** : Instituto Nacional Americano de Estandarización. American National Standards Institute
- **API** : Application Program Interface.
- **COM** : Modelo de objetos componentes. Component Object Model
- **CPU** : Unidad Central de Proceso. Central Process Unit
- **D/A** : Conversión Digital/Analógica
- **DFT** : Transformada Discreta de Fourier
- **DLL** : Biblioteca de Enlace Dinámico. Dynamic Link Library
- **DMA** : Acceso Directo a Memoria. Direct Memory Access
- **GDI** : Dispositivo de Interfaz Gráfico. Graphics Device Interface
- **GPC** : Control Predictivo Generalizado
- **GUI** : Interfaz de Usuario Gráfico. Graphics User's Interface
- **MDI** : Interfaz de Documento Múltiple. Multiple Document Interface
- **MFC** : Clases Base de Microsoft. Microsoft Foundation Classes
- **MS-DOS** : Sistema Operativo de Disco de Microsoft. Microsoft Disk Operating System.
- **OEM** : Juego de Caracteres OEM (Original Equipment Manufacturer)
- **PID** : Control Proporcional Derivativo e Integral
- **PWM** : Modulación por Ancho de Pulso. Pulse Width Modulation
- **SDI** : Interfaz Simple de Documento. Single Document Interface
- **SDK** : Herramientas de Desarrollo de Programas. Software Development Kit

6.5 LISTADO DE FUNCIONES MATLAB

En este anexo se incluye el listado de las funciones programadas en MATLAB que se han mencionado en el proyecto.

Función Identif.m

```
% Fichero usado para facilitar el proceso de identificación.
% presenta menús y submenús para elegir los diferentes
% parámetros usados en la identificación.

% Valores por defecto.
clear all;
Ts=0.213;
l=0;
n_a=4;
n_b=3;
factor=4;
filtro=[0.75 0.25];
com=-1;
final=0;
ch_datos=1;

while final~=1,

% Elección del menú principal. Incluye todos los parámetros.
while com<6,
com=menu('Menú principal', 'Factor de submuestreo', 'retardo',
'grado de ai', 'grado de bi', 'filtro para u', 'continuar',
'terminar');
    if com==1,
        disp(' ');
        disp('Valor actual: ');
        disp(factor);
        factor=input('Nuevo factor de sobremuestreo? ');
    end
    if com==2,
        disp(' ');
        disp('Valor actual: ');
        disp(l);
        l=input('Nuevo retardo?');
    end
    if com==3,
        disp(' ');
        disp('Valor actual: ');
        disp(n_a);
        n_a=input('Nuevo grado para ai?');
    end
    if com==4,
        disp(' ');
        disp('Valor actual: ');
```

```

        disp(n_b);
        n_b=input('Nuevo grado para bi?');
    end
    if com==5,
        disp(' ');
        disp('Filtro actual; valores de alfa0 y alfa1:');
        disp(filtro);
        filtro1=input('Nuevo filtro f(k+1)=alfa1*f(k)+alfa2*u(k).
Elija alfa1; alfa2=1-alfa1. ');
        filtro=[filtro1 1-filtro1];
    end
end

if com==7,
    final=1;
end

if com==6,

ch_datos=menu('Menú de cambio de datos', 'Cambiar datos',
'Mantener datos anteriores');

% Si se quiere cambiar datos.
if ch_datos==1,

% Elección del fichero con los datos.
com=menu('Nombre del fichero de datos','iden??','otro');
if com==1,
    ext=input('Teclee el número "??" del experimento "iden??".',
's');
    nom_fic=['iden' ext];
end
if com==2,
    nom_fic=input('Teclee el nombre del fichero de datos, sin
extensión.', 's');
end
eval(nom_fic);

% Gráficas de los datos escogidos.
figure;
subplot(2,1,1); ylabel('Señal de control'); hold on;
title('Datos del experimento');
plot(u);
subplot(2,1,2); ylabel('Señal de salida'); hold on;
plot(y);
pause;

% Posibilidad de eliminar muestras no deseadas del principio.
com=input('Desea eliminar muestras iniciales (1/0)?');
if com==1,
    disp('Pinche con el ratón sobre la gráfica en el lugar donde
quiera empezar');
    [x_pun,y_pun]=ginput(1);
    x_pun=ceil(x_pun);
    y1=y(x_pun:length(y),1);
    u1=u(x_pun:length(u),1);
end
if com==0,

```

```
        y1=y;
        u1=u;
    end

    close;

    % Sobremuestreo y Filtrado de la señal u.
    u2=u1(1:factor:length(u1),1);
    uf(1,1)=u2(1,1);
    for k=2:length(u2),
        uf(k,1)=filtro*[uf(k-1,1);u2(k,1)];
    end

    % Adaptación de la señal y.
    y2=pros_sig(y1,u1,Ts);
    pause;
    close;
    yf=y2(1:factor:length(y2),1);

    % Cierro el if ch_datos==1.
    end

    % Presentación de los datos que van a ser usados.
    figure;
    t=[0:Ts*factor:Ts*factor*(length(uf)-1)]';
    subplot(2,1,1); ylabel('Señal de control'); hold on;
    title('Datos de entrada al identificador');
    plot(t,uf);
    axis tight;
    subplot(2,1,2); ylabel('Señal de salida'); hold on;
    plot(t,yf);
    axis tight;
    pause;
    close;

    [b,a,dist]=mincuad(yf,uf,n_a,n_b,l);
    ysim=genera_y(yf,uf,b,a,l);
    figure;
    plot(yf,'r'); hold on;
    plotyy([1:length(ysim)],ysim,[1:length(uf)],uf);
    pause;
    com=input('Guardar datos (1/0)?');
    if com==1,
        save resultados ysim uf yf b a factor l;
    else
        close;
    end
    % Cierro el if com==6.
    end
    % Cierro el while final~=1.
    end
```

Función Mincuad.m

```

function [b,a,v]=mincuad(y,u,n,m,l)
% [b,a,distancia]=mincuad(y,u,n,m,l)
% La función mincuad realiza una identificación del sistema
% mediante mínimos cuadrados. Devuelve los coeficientes ai
% correspondientes a los yk-i y los coeficientes bi que
% corresponden a los uk-i-1. El número de coeficientes ai
% es n+1 y el de bi es de m+1. El retardo para los uk es de l.
% El modelo ajustado es
%  $a_0 y_k + a_1 y_{k-1} + \dots + a_n y_{k-n} = -k + (b_0 u_{k-l} + b_1 u_{k-l-1} + \dots + b_m u_{k-l-m}) / P(y)$ 
% donde P(y) es un polinomio ajustado para eliminar la no
% linealidad
% y k es el término debido a la fuerza gravitatoria.
% k se fija a 1.5, y se añaden dos restricciones más a los
% mínimos
% cuadrados, que la suma de los ai sea -0.7 y la suma de los bi
% sea 1.
% Estas restricciones son necesarias para que P(y) sea
% correcto.
% Devuelve como distancia la norma 2 del vector que da la
% distancia
% de la solución obtenida al conjunto de las restricciones.

r=length(y);
p=[18 -33 18.5 0.17 1];
if n>=m+1,
    for i=1:r-n,
        pol=polyval(p,y(n+i-1));
        A(i,:)= [y(n+i:-1:i)' -u(n+i-l:-1:n+i-m-l)'/pol];
        bb(i)=-1.5;
    end
    w1=diag(ones(length(A),1));
end
if n<m+1,
    for i=1:r-m-1,
        pol=polyval(p,y(i+m+1-1));
        A(i,:)= [y(m+1+i:-1:m+1-n+i)' -u(m+i:-1:i)'/pol];
        bb(i)=-1.5;
    end
    w1=diag(ones(length(A),1));
end

w2=diag(ones(2,1));
w=[w1 zeros(length(A),2);zeros(2,length(A)) w2];
i=i+1;
A(i,:)= [ones(1,n+1) zeros(1,m+1)];
bb(i)=-0.7;
i=i+1;
A(i,:)= [zeros(1,n+1) ones(1,m+1)];
bb(i)=1;

if det(A'*A)==0,

```

```

    disp('Matriz A es no singular');
else
    x=inv(A'*W*A)*A'*W*bb';
    a=x(1:n+1);
    b=x(n+2:m+n+2);
    v_dist=bb'-A*x;
    v=v_dist'*v_dist;
end

```

Función Genera_y.m

```

function y_g=genera_y(y_real,u_real,b,a,l)
% y_generad=genera_y(y_real,u_real,b,a,l)
% genera_y es la función encargada de simular el
% sistema para unos vectores 'a' y 'b' suministrados, y una
% señal 'u' dada. La función genera_y simula la salida 'y' del
% sistema siguiente:
%  $ao*y_k+a1*y_{k-1}+...+an*y_{k-n} = -k + (bo*uk-l+b1*uk-l-1+...+bm*uk-l-m)/P(y)$ 
% donde los ai son los coeficientes del vector 'a' y los bi son los
% coeficientes del vector 'b'. El retardo viene dado por el
% parámetro
% 'l'. El factor K vale 1.5 y P(y) vale para cada 'k' el
% resultado de
% evaluar el polinomio P(y) usando yk-1 como argumento.

r=length(y_real);
n=length(a)-1;
m=length(b)-1;
b
a
l
p=[18 -33 18.5 0.17 1];
if n>=m+1
    y_g=y_real(1:n);
    for i=1:r-n,
        pol=polyval(p,y_g(n+i-1));
        fac1=y_g(n+i-1:-1:i);
        fac2=u_real(n+i-1:-1:n+i-m-1)/pol;
        y_g(n+i)=(-1.5+b'*fac2-a(2:length(a))*fac1)/a(1);
        if y_g(n+i)>0.95,
            y_g(n+i)=0.95;
        end
        if y_g(n+i)<0,
            y_g(n+i)=0;
        end
    end
end
if n<m+1,
    y_g=y_real(1:m+1);
    for i=1:r-m-1,
        %pol=polyval(p,y_real(i+m+1));
        pol=polyval(p,y_g(i+m+1-1));

```

```

        fac1=y_g(m+l+i-1:-1:m+l+i-n);
        fac2=u_real(m+i:-1:i)/pol;
        y_g(m+l+i)=(-1.5+b'*fac2-a(2:length(a))*fac1)/a(1);
        if y_g(m+l+i)>0.95,
            y_g(m+l+i)=0.95;
        end
        if y_g(m+l+i)<0,
            y_g(m+l+i)=0;
        end
    end
end
end

```

Función Mipred.m

```

% mipredi(num,den,d,Ts,N,lambda,tfinal)
% Función que calcula los coeficientes de la ley de control
% para un control predictivo sin restricciones.
% El modelo del sistema viene dado como función de
% transferencia, con 'num' y 'den' como numerador y denominador
% de la función de transferencia, 'd' es el retardo del sistema
% 'N' es el horizonte de control, 'lambda' es el factor de
% ponderación. El orden de 'num' y 'den' está prefijado.
% También simula el comportamiento del control para una
% referencia
% predefinida. 'Ts' es el tiempo de muestreo y 'tfinal' es el
% tiempo total de la simulación
function [u,yk]=mipred(num,den,d,Ts,N,lambda,tfinal)

tf=round(tfinal/Ts);

numd=num';
dend=[1 den'];

E(1,1)=1;
aacute=-conv([1,-1],dend);
F(1,1:length(aacute)-1)=aacute(2:length(aacute));
for j=2:N+d
    E(j,j)=F(j-1,1);
    E(j,1:j-1)=E(j-1,1:j-1);
    for i=1:length(dend)-1
        F(j,i)=F(j-1,i+1)+F(j-1,1)*aacute(i+1);
    end
    F(j,length(dend))=F(j-1,1)*aacute(length(dend)+1);
end

for i=1:N+d
    GA(i,:)=conv(E(i,:),B);
end

for i=1:N
    for j=1:i
        G(i,j)=GA(i,i-j+1);
    end
end

```

```

        for j=i+1:length(GA(d+1,:))
            Gprima(i,j-i) = GA(i+d,j);
        end
    end

% Efectuar el control

for i=1:tf+2*N+1+2*d
    if (i<20+N+d)
        r(i) = 0;
    else
        r(i) = 0.3;
    end
end

w(1) = 0;
for i=2:tf+2*N+1+2*d
    w(i) = w(i-1)*0.3+r(i)*0.7;
end

FF=F(d+1:d+N,:);

% IMPORTANTE: t=0 equivale a i=N+d+1 para w,u,du,yk
for i=1:N+d+1
    u(i) = 0;
    du(i) = 0;
    yk(i) = 0;
end

for i=N+d+1+1:tf+N+d
    for k=1:N
        W(k)=w(i+d+k);
    end
    uk1 = u(i-1-d);
    uk2 = u(i-2-d);

    % CALCULO DE YK

    if pepo==0,
        keyboard;
        pepo=1;
    end

    yk(i)=(-dend(2))*yk(i-1)+(-dend(3))*yk(i-2)+(-dend(4))*yk(i-3)+(-dend(5))*yk(i-4)+numd(1)*uk1+numd(2)*uk2;

%     APLICAR GPC

    for j=1:length(dend)
        y(j) = yk(i-j+1-d);
    end
    for j=1:d+1
        iu(j) = du(i-j);
    end

    du(i) = M(1,:)*( FF*y' + Gprima(:,1:d+1)*iu' - W');

```

```
        u(i) = u(i-1)+du(i);

end

figure;
subplot(2,1,1);
plot (w, 'r');
hold on;
pause
plot (yk, 'g');
title('Comparación entre la referencia y la salida de la
simulación');
xlabel('Número de la muestra');
ylabel('Valor en metros');
pause
subplot(2,1,2);
plot(u, 'b');
title('Señal de control aplicada');
xlabel('Número de la muestra');
ylabel('Valor en voltios');

M(1,:)*FF
M(1,:)*Gprima(:,1:d+1)
-M(1,:)*W'
```