Fundamentos de control Industrial

Práctica 2

Javier Rico Azagra

Índice

Objetivo

El objetivo principal de la práctica es aprender a trabajar con la *Control System Toolbox*[1]. Para afrontar el desarrollo de la práctica es necesario conocer el manejo básico del entorno Matlb, así como sus posibilidades de programación, la definición de variables y las instrucciones de uso general. Los conceptos estudiados se resumen a continuación:

- Aprender a definir funciones de transferencia, en el dominio de la frecuencia compleja, como modo de caracterización de los modelos de sistemas físicos.
- Conocer las instrucciones de transposición de los formatos de expresión de las funciones de transferencia (FT).
- Analizar los diagramas de polos y ceros de las FT, y ubicarlos dentro del plano complejo «S».
- Representar gráficamente la respuesta temporal de sistemas ante distintos estímulos de entrada.
- Conocer y utilizar las funciones disponibles para la evaluación en el dominio temporal de sistemas LTI (Lineales Tiempo Invariantes).

Instrucciones

A continuación se resumen las funciones o instrucciones básicas para cumplir los objetivos anteriormente descritos. Se van a presentar agrupadas por actividad.

Definición de funciones de transferencia

Podemos definir una función de transferencia empleando tres métodos diferentes. Dependiendo de los datos de partida deberemos seleccionar el más adecuado.

```
s=tf('s'); % Define la letra s como variable compleja
sys1=(s+1)/(s^2+2*s+4); % Define la FT empleando su ecuación matemática

sys=tf(num,den); % Define FT a partir de su numerador y denominador
% Ejemplo:
num=[1 1];
den=[1 2 4];
sys2=tf(num,den);

sys=zpk(ceros,polos,ganancia); % Define la FT a partir los polos ceros y ganancia
% Ejemplo:
ceros=[-2 -3 -4];
polos=[-1 ];
ganancia=2;
sys3=zpk(ceros,polos,ganancia);

% Ejemplos
s=tf('s');
P1=tf([2 4],[1 1 2 3]); % Función de transferencia creada mediante el comando TF
P2=zpk([-1 -2],[0 -4 -5 -7],[10]); % Formato ceros polos ganancia
P3=zpk([],[0 -4 -5 -7],[10]);      % El conjunto vacío indica que no existen ceros
P4=zpk([],[0],[1]);                % Función de transferencia de un integrador
P5=1/s;                            % En este caso este formato es más sencillo
```

[1] La *Control System Toolbox* es es la librería empleada por Matlab para trabajar con sistemas de control lineales.

Obsérvese que la misma función de transferencia puede almacenarse en Matlab empleando dos formatos diferentes: *tf* y *zpk*. El formato tf es asignado a las funciones de transferencia definidas con el comando *tf(num,den)* y a las definidas por su expresión matemática. El formato zpk es asignado a las funciones de transferencia definidas con el comando zpk. Puede comprobarse como P_4 y P_5 corresponden con la misma función de transferencia, pero son almacenados en el *workspace* según:

- $P4 \quad 1 \times 1 \; zpk$
- $P5 \quad 1 \times 1 \; tf$

Todas las instrucciones presentadas a lo largo del curso trabajan con los dos formatos. El usuario no debe preocuparse del formato empleado, unicamente deberá seleccionar el más adecuado en función de los datos disponibles.

Transposición entre modelos de representación

Las instrucciones mostradas a continuación permiten transformar de un modelo de definición de sistemas a otro, obteniendo información relevante del mismo.

```
sys=zpk(sys_tf);     % Guarda en sys el sistema sys_tf con formato  zpk
sys2=tf(sys_zpk);    % Guarda en sys el sistema sys_zpk con formato num-den


[num,den]=tfdata(sys);    % Guarda en num y den  el numerador y denominador de sys
[num,den]=tfdata(sys,'v'); % Resultados en formato vector (no cell-array)

[z,p,k]=zpkdata(sys);     % Guarda en z, p y k los  ceros, polos y ganancia de sys
[z,p,k]=zpkdata(sys,'v'); % Resultados en formato vector (no cell-array)

p=pole(sys);              % Guarda en p los polos de sys
z=zero(sys);              % Guarda en z los ceros de sys
[z k]=zero(sys);          % Guarda en z y k los ceros y ganancia de sys

sys_min=minreal(sys);     % Cancela polos y ceros para simplificar sys

% Ejemplo cancelación polos/ceros
P=(s+1)/s/(s+1);
P_simp=minreal(P);
```

Representación de diagramas polos-ceros

La ubicación de polos y ceros en el plano «S» se puede representar mediante la instrucción *pzmap*, sin poner argumentos de salida.

```
pzmap(sys);  % Representa la situación de polos y ceros de sys en el plano S

% Ejemplo:
G=tf([1 5 7 9],[4 5 8 7 9 3]);
pzmap(G)
```

Respuesta temporal de sistemas

A continuación se muestran las instrucciones empleadas para obtener la respuesta temporal de las funciones de transferencia.

```
impulse(sys); % Representación y cálculo de la respuesta al impulso idealizado
step(sys);     % Representación y cálculo de la respuesta al escalón unitario
lsim(sys,u,t); % Representación y cálculo de la respuesta a una entrada
               % arbitraria definida por u y t
```

Las funciones *impulse*, *step* y *lsim* admiten diferentes parámetros de entrada y salida. A continuación se muestran algunos ejemplos para la instrucción *impulse*.

```
impulse(sys); % Grafica la respuesta al impulso del sistema sys.
impulse(sys, TFINAL); % Representa la respuesta impulsional desde t=0 hasta TFINAL
impulse(sys, t); % Idem, pero utilizando el vector de tiempos t
impulse(sys1, sys2, ...); % Permite representar la respuesta de múltiples sistemas
impulse(sys1, sys2, ... ,TFINAL); % Idem, con acotación de tiempo de evaluación
impulse(sys1, sys2, ..., t); % Idem, utilizando el vector de tiempo t
[y,t]=impulse(sys); % Calcula los vectores de tiempo (t) y resultado (y)
                    % Cuando empleamos argumentos de salida no se traza el gráfico.
[y,t]=impulse(sys, TFINAL); % Idem con definición de tiempo final de evaluación
y = impulse(sys, t); % Idem con vector de tiempo de usuario
```

A continuación se muestran algunas herramientas que pueden ser útiles para el estudio de sistemas LTI y la generación de señales de excitación.

```
ltiview (sys); % Llama al visor LTI para el análisis de sistemas LTI
               % Este permite analizar las respustas temporales, sus
               % características, diagramas de Bode, Niquist, etc.

stepinfo(sys); % Obtención de valores característicos a una entrada escalón
lsiminfo(y,t); % Obtención de valores característicos para una entrada aleatoria
% Estos datos pueden obtenerse en las ventanas gráficas sin necesidad de
% emplear comandos

[u,t]=gensig(tipo,T); % Generación de señales de prueba (senoidal y cuadrada)
                      % puede ser útil para generar las señales empleadas en lsim
```

Ejercicios propuestos

Ejercicio 1: Definición de modelos LTI

El objetivo del ejercicio 1 es practicar con las instrucciones empleadas para definir funciones de transferencia, así como con las instrucciones empleadas para transformar entre los diferentes formatos de representación. Para ello se emplearan las siguientes funciones de transferencia:

$$P_1(s) = \frac{s^2 + 2s + 1}{s^4 + 4s^3 + 2s + 1}$$

$$P_2(s) = \frac{5(s+1)(s+4)}{s(s+2)(s+7)(s+9)}$$

$$P_3(s) = \frac{10s}{(s^2 + 2s + 2)(s+3)}$$

Se pide:

a. Define las siguientes funciones de transferencia:

- P1 mediante su numerador y denominador.
- P2 por su expresión matemática a partir de polos ceros y ganancia.
- P3 por su expresión matemática.

```
%% a) Define las siguientes funciones de transferencia:
s=tf('s');
P1=tf([1 2 1],[1 4 0 2 1]);
P2=zpk([-1 -4],[0 -2 -7 -9],5);
P3=10*s/(s^2+2*s+2)/(s+3);
```

b. Realiza las siguientes operaciones:

- Obtén los polos ceros y ganancia de $P_1(s)$ de dos formas distintas.
- Obtén el numerador y el denominador de $P_2(s)$.
- Traza el diagrama de polos/ceros de $P_3(s)$.

```
%% b) Realiza las siguientes operaciones:
% Obtén los polos ceros y ganancia de P1 de tres formas distintas.
[ceros,polos,ganancia]=zpkdata(P1,'v');
polos2=pole(P1);
[ceros2, ganancia2]=zero(P1);

% Obtén el numerador y el denominador de P2.
[num2, den2]=tfdata(P2,'v');

% Diagrama de polos/ceros de P3
pzmap(P3)
```

Ejercicio 2: Respuesta temporal de los sistemas de primer orden

El objetivo del segundo ejercicio es validar los conceptos estudiados en las sesiones teóricas referentes a la respuesta temporal de los sistemas de primer orden. Recuerdese que el modelo genérico de la función de transferencia de un sistema de este tipo corresponde con:

$$p(s) = \frac{k}{\tau s + 1} = \frac{k_p}{s + p},$$

donde τ corresponde con la constante de tiempo del sistema y k con la ganancia en régimen permanente.

Se pide:

a. Respuesta temporal para acciones de control básicas. Define la función de transferencia de un sistema P, que presente ganancia y constante de tiempo unitarias. Genera una figura en la que se presenten en tres sub-figuras (*subplot*) las siguientes respuestas temporales:

- Impulso unitario.
- Escalón unitario.
- Rampa unitaria. La señal debe definirse para el vector de tiempos $t = 0 : 0.1 : 10$.

Comprobar sobre las figuras que se cumple el comportamiento esperado. Parámetros característicos de las respuestas (tiempo de establecimiento, valor de régimen permanente, etc.).

```
%% a) Respuesta temporal para acciones de control básicas.
P=1/(s+1);
subplot(3,1,1),impulse(P);
subplot(3,1,2),step(P);
t=0:0.1:10;
u=t;
subplot(3,1,3),lsim(P,u,t);
```

b. Efecto de la ganancia en el comportamiento del sistema. Muestra en una figura la respuesta al escalón de cuatro sistemas de primer orden que presentan $\tau = 1$, y cuyas ganancias toman los valores

$$k = [8,\ 4,\ 2,\ 1,\ 0.5].$$

```
%% b) Efecto de la ganancia en el comportamiento del sistema
close all
k= [8 4 2 1 0.5];
hold on
for i=1:length(k)
    step(tf(k(i),[1 1]));
end
hold off
```

c. Efecto de la constante de tiempo en el comportamiento del sistema. Muestra en una figura la respuesta al escalón de cuatro sistemas de primer orden que presentan ganancia $k = 1$, y cuyas constantes de tiempo toman los valores

$$\tau = [15, 10, 5, 1, 0.5].$$

```
%% c) Efecto de la constante de tiempo en el comportamiento del sistema
close all
tau= [15 10 5 1 0.5 ];
hold on
for i=1:length(tau)
    step(tf(1,[tau(i) 1]),100);
end
hold off
```

d. Evalúa el efecto de los ceros en un sistema de primer orden (no estudiado en clase). Se desea conocer el comportamiento ante un escalón unitario del sistema

$$p(s) = \frac{k(\tau_c s + 1)}{(\tau_p s + 1)},$$

donde τ_c corresponde con la constante de tiempo del cero y τ_p con la constante de tiempo del polo.

Representa la respuesta al escalón de varios sistemas que presenten $k = 1$, $\tau_p = 1$ y ceros con constante de tiempo en

$$\tau_c = [6,\ 4,\ 2,\ 0.5,\ 0.25,\ 0.1].$$

Repite el experimento anterior fijando la ganancia de todos los sistemas a $k = 2$. Efectúa más pruebas con diferentes valores en para los parámetros hasta que observes el patrón de comportamiento.

Contesta a las siguientes preguntas:

- ¿Como afecta a la respuesta que la constante de tiempo del cero sea mayor o menor de la del polo?
- ¿Como distorsiona el cero el valor inicial de la respuesta ($y(t = 0)$)?
- ¿Como distorsiona el cero el valor final de la respuesta ($t = y(\infty)$)?
- ¿Como afecta el cero al tiempo que dura el régimen transitorio?

```
%% d) Efecto de los ceros en el comportamiento del sistema
close all
tau_c= [6 4 2  0.5 0.25 0.1];
hold on
for i=1:length(tau_c)
    step(tf([tau_c(i) 1],[1 1]));
end
hold off

% Repetimos para k=2
close all
tau_c= [6 4 2  0.5 0.25 0.1];
hold on
for i=1:length(tau_c)
    step(tf([2*tau_c(i) 2],[1 1]));
end
hold off
```

e. Definición de sistemas a partir de parámetros característicos. Define la función de transferencia para un sistema LTI de primer orden que cumpla las siguientes especificaciones:

- Constante de tiempo 10 segundos y ganancia unitaria.
- Tiempo de establecimiento 8 segundos y ganancia 5 unidades.
- Pendiente en el origen 80º y ganancia 25.
- Tiempo de subida 0.22 segundos y ganancia 10.

```
%% e) Definición de sistemas a partir de parámetros característicos
close all
% Caso 1
k1=1;
tau1=1;
P1 = k1/(tau1*s+1);
step (P1)

% Caso 2
k2=5;
tau2=8/4; % ts=4 tau
P2 = k2/(tau2*s+1);
step(P2)

% Caso 3
k3=25;
tau3=k3/tand(80); % pendiente=k/tau
P3 = k3/(tau3*s+1);
step(P3)
```

```

% Caso 4
k4=10;
tau4=0.22/2.2; % t1=2.2tau
P4 = k4/(tau4*s+1);
step(P4)
```

Ejercicio 3: Sistema de primer orden real

La planta Feedback 33-033, mostrada en la Figura 1 y disponible en el laboratorio RAII, esta compuesta principalmente por un motor de corriente continua Crouzet (imanes permanentes). La función de transferencia teórica de dicho motor esta determinada por

$$\frac{\Omega(s)}{v_i(s)} = \frac{k_i}{(Ls+R)(Js+B)+k_ik_b},$$

donde:

- Ω corresponde con la velocidad angular expresada en [rpm], rango aproximado ± 3000rpm.
- v_i corresponde con la entrada de excitación expresada en [V], rango de operación ± 10V.
- R, L resistencia e inductancia de la bobina de inducido.
- k_b constante de la fuerza contraelectromotriz [V/rad/s].
- k_i constante de par [Nm/A]
- J momento de inercia de la carga.
- B rozamiento lineal (cojinetes + freno magnetico).

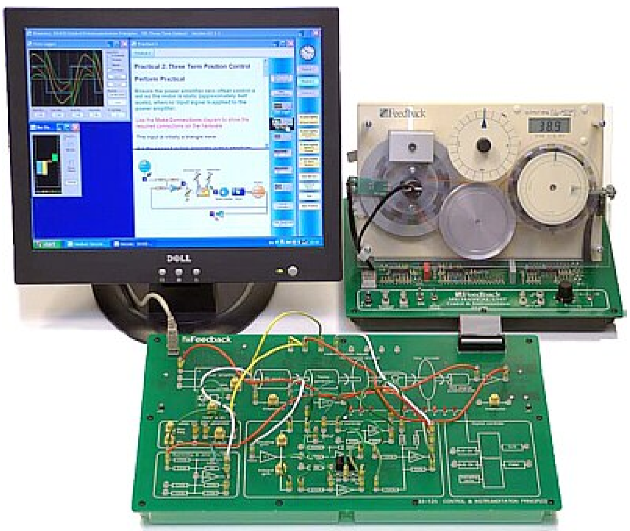

Figura 1: Maqueta docente Feedback 33-100.

En la práctica, se conoce que la constante de tiempo eléctrica L/R es mucho menor que la constante de tiempo mecánica J/B, por lo que se suele despreciar el efecto de la primera. Considerando la aproximación anterior la función de transferencia puede expresarse según

$$\frac{\Omega(s)}{v_i(s)} = \frac{k_i}{R(Js+B)+k_ik_b}.$$

Véase como R, k_i y k_b corresponden con parámetros internos del motor (no pueden variar). Sin embargo J y B dependen de la carga arrastrada por el motor, que pude configurarse mediante la manipulación del freno magnético. Este es un caso típico en el que no puede aplicarse una estrategia de control en lazo abierto, ya que la función de trasferencia del sistema cambia con la carga.
El archivo *datos_motor.mat* contiene los resultados obtenidos para tres experimentos idénticos, en los que la única diferencia corresponde con la posición del freno magnético. Las señales registradas en dicho experimento son el tiempo (t), la acción de control (tensión aplicada en el motor u) y la salida del sistema (velocidad angular en rpm y). se presentan tres conjuntos de datos:

- $[t_1,\ u_1,\ y_1]$ Resultados obtenidos con el freno magnético al 0 %.
- $[t_2,\ u_2,\ y_2]$ Resultados obtenidos con el freno magnético al 50 %.
- $[t_3,\ u_3,\ y_3]$ Resultados obtenidos con el freno magnético al 100 %.

Se pide :

a. Estudio del experimento. Representa en una figura dividida en dos sub-figuras los tres experimentos. Se desea que en la parte superior se observen las respuestas del sistema (y-velocidad en rpm) y en la parte inferior las acciones de control (u-tensión de excitación). ¿Que tipo de entrada se ha aplicado? ¿Que tipo de respuesta presenta el sistema?

```
%% a) Estudio del experimento
close all;
subplot(2,1,1), plot(t1,y1,t2,y2,t3,y3);
subplot(2,1,2), plot(t1,u1,t2,u2,t3,u3);
```

b. Extrae la primera parte de los datos del primer experimento $[t_1a,\ u_1a,\ y_1a]$, de forma que pueden analizarse por separado. Datos desde t=1 hasta t=3.5;

```
%% b) Extrae la primera parte de los datos
Ts=t1(2)-t1(1); % Calculo el tiempo de muestreo
t_fin=3.5; % Fijo el tiempo de corte
t_ini=1; % Fijo el tiempo de inicio

% Datos para freno al 0%
t1a=t1(t_ini/Ts:t_fin/Ts)-t_ini;
u1a=u1(t_ini/Ts:t_fin/Ts);
y1a=y1(t_ini/Ts:t_fin/Ts);
```

c. Emplea los datos extraídos en el apartado *b* para identificar la función de transferencia del sistema (freno magnético al 0 %). Utiliza los valores característicos estudiados en clase.

```
%% c) Identificación de la función de transferencia

%% Datos para freno al 0%
close all;
subplot(2,1,1), plot(t1a,y1a);
subplot(2,1,2), plot(t1a,u1a);
```

```
% Calculamos las plantas por inspección visual
k1=650/2; % Regimen permanente / incremento en la entrada
tau1=0.15; % Busco el tiempo en el que se alcanza 0.63yrp 0.65−0.5=0.15
P1=tf(k1,[tau1 1]);
```

d. Validación de resultados. Emplea la función de transferencia culada junto con las acciones de control reales para simular el experimento completo. Representa en una figura la respuesta teórica y real de forma superpuesta. ¿Se mantiene el ajuste a lo largo de todo el experimento?

```
%% d) Validación de resultados

%% Freno al 0%
y1ta=lsim(P1,u1a,t1a);
subplot(2,1,1),plot(t1a,y1a,t1a,y1ta)

y1t=lsim(P1,u1,t1);
subplot(2,1,2),plot(t1,y1,t1,y1t)
```

e. Crea una función *identifica_orden1* que reciba como parámetros de entrada los datos de un experimento y calcule de forma automática la función de transferencia de dicho sistema. Para ello puede emplearse la siguiente propiedad:

$$\tau = \frac{A_o}{k\Delta u}$$

donde A_o corresponde con el área encerrada entre el valor de la respuesta en régimen permanente y la respuesta del sistema.

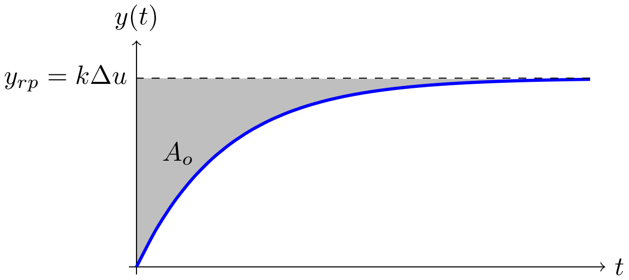

Figura 2: Esquema empleado para obtener el área A_o.

Demostración:

- La respuesta temporal de un sistema de primer orden ante un escalón unitario de amplitud Δu es

$$y(t) = \Delta uk(1 - e^{-t/\tau})$$

- El área entre encerrada entre la respuesta temporal y su valor de régimen permanente puede calcularse según:

$$A_o = \Delta u \int_0^\infty (k - y(t))dt = \Delta u \int_0^\infty (k - k - ke^{-t/\tau})dt = \int_0^\infty ke^{-t/\tau}dt = k\Delta u \left[\frac{e^{-t/\tau}}{-\tau}\right]_0^\infty = k\Delta u\tau$$

- Puede calcularse la constante de tiempo según:

$$\tau = \frac{A_o}{k\Delta u}$$

Una vez diseñada la función, identifica la funciones de transferencia para el freno al 0 %. ¿Los resultados coinciden con los obtenidos mediante la identificación por inspección visual?

```
function [P]=identifica_orden1(t,u,y)
% Identifica la función de transferencia en función de un experimento escalón
% Requisitos:
%   — La señal escalón debe aplicase despues de al menos 0.25 segundos
%   — La señal de salid debe mantenerse en rp al menos 0.25 segundos

Ts=t(2)-t(1); % Calculo el tiempo de muestreo
n=length(u);  % Calculo el numero de muestras
Au=u(n)-u(1); % Caluclo el incremento en la entrada

% Calculo la media del valor inicial y final
y_1=y(1:floor(0.25/Ts));
y_ini=sum(y_1)/length(y_1);

y_2=y(n-floor(0.25/Ts):n);
y_fin=sum(y_2)/length(y_2);

Ay=y_fin-y_ini; % Caluclo el incremento en la salida

k=Ay/Au;  % Ganancia en régimen permanente

% Tiempo de inicio del experimento
c=2;
while u(c)-u(c-1)==0
    c=c+1;
end

% Calculo la función yrp—y(t)
e=y_fin-y;


% Calculo la integral desde c hasta n
Ao=0;
for i=c+1:n
    Ao=Ao+(e(i)+e(i-1))*Ts/2;
end

tau=Ao/k/Au;
P=tf(k,[tau 1]);
```

```
%% e) identificación de la función de transferencia
P1b=identifica_orden1(t1a,u1a,y1a);
```

f. Repite la identificación y la posterior validación de resultados eliminando de la colección de datos los primero 3.5 segundos. ¿Mejora el ajuste?

```
%% f) Repetimos la identificación para el segundo escalón

t_fin=5.5;  % Fijo el tiempo de corte
t_ini=3.5; % Fijo el tiempo de inicio

```

```
% Datos para freno al 0%
t1b=t1(t_ini/Ts+1:t_fin/Ts+1)-t_ini;
u1b=u1(t_ini/Ts+1:t_fin/Ts+1)-u1(t_ini/Ts+1);
y1b=y1(t_ini/Ts+1:t_fin/Ts+1)-y1(t_ini/Ts+1);
P1c=identifica_orden1(t1b,u1b,y1b);

% Pruebo solo para los dos últimos escalones
t_fin=7.5;  % Fijo el tiempo de corte
t_ini=3.5; % Fijo el tiempo de inicio
t1c=t1(t_ini/Ts+1:t_fin/Ts+1)-t_ini;
u1c=u1(t_ini/Ts+1:t_fin/Ts+1)-u1(t_ini/Ts+1);
y1c=y1(t_ini/Ts+1:t_fin/Ts+1)-y1(t_ini/Ts+1);

y1tb=lsim(P1c,u1b,t1b);
subplot(2,1,1),plot(t1b,y1b,t1b,y1tb)
y1tb=lsim(P1c,u1c,t1c);
subplot(2,1,2),plot(t1c,y1c,t1c,y1tb)
```

g. Repite el apartado *b* para obtener mediante el empleo de la función desarrollada en el apartado *e* la función de transferencia para el motor con el freno magnético al 50 % (P_2) y al 100 % (P_3). Una vez obtenidas las funciones de transferencia valida su ajuste (repetir apartado el apartado *d*) y contestar a las siguientes preguntas:

- ¿Como afecta el aumento de la carga a la ganancia del sistema?
- ¿Como afecta el aumento de la carga a la constante de tiempo?
- ¿Los resultados reales coinciden con los esperados según la función de transferencia teórica?

```
%% g) Repetimos la identificación para el resto de experimentos

%%Extrae la primera parte de los datos
% Datos para freno al 50%
t2a=t2(t_ini/Ts:t_fin/Ts)-t_ini;
u2a=u2(t_ini/Ts:t_fin/Ts);
y2a=y2(t_ini/Ts:t_fin/Ts);

% Datos para freno al 100%
t3a=t3(t_ini/Ts:t_fin/Ts)-t_ini;
u3a=u3(t_ini/Ts:t_fin/Ts);
y3a=y3(t_ini/Ts:t_fin/Ts);

% Identificación de funciones de transferencia
P2=identifica_orden1(t2a,u2a,y2a);
P3=identifica_orden1(t3a,u3a,y3a);

% Identificación de funciones de transferencia
%% Freno al 50%
y2ta=lsim(P2,u2a,t2a);
subplot(2,1,1),plot(t2a,y2a,t2a,y2ta)

y2t=lsim(P2,u2,t2);
subplot(2,1,2),plot(t2,y2,t2,y2t)

%% Freno al 100%
```

```
y3ta=lsim(P3,u3a,t3a);
subplot(2,1,1),plot(t3a,y3a,t3a,y3ta)

y3t=lsim(P3,u3,t3);
subplot(2,1,2),plot(t3,y3,t3,y3t)
```