

Identificación de Sistemas.
Aplicación al modelado de un motor de continua.

Departamento: **ELECTRÓNICA**

M^a Elena López Guillén

ÍNDICE

1. Introducción.....	1
1.1. Concepto de sistema.....	1
1.2. Modelo de un sistema.....	1
1.3. Tipos de modelos.	2
1.4. Métodos de obtención de modelos.....	3
2. Identificación de sistemas.	4
2.1. El proceso de identificación.....	4
2.2. Métodos de identificación.	6
3. Técnicas de identificación no paramétrica.....	6
3.1. Conceptos previos.	7
3.2. Identificación no paramétrica en el dominio del tiempo.....	7
3.3. Identificación no paramétrica en el dominio de la frecuencia.....	10
4. Técnicas de identificación paramétrica.....	10
4.1. Tipos de modelos paramétricos.	11
4.2. Métodos para el ajuste de parámetros.	13
4.2.1. Errores de predicción o residuos de un modelo.	14
4.2.2. Regresión lineal.....	14
4.2.3. Método de mínimos cuadrados (LSE).....	14
4.2.4. Método de variables instrumentales.....	16
5. Consideraciones prácticas sobre identificación.	17
5.1. De la obtención de los datos.	17
5.2. Del pretratamiento de los datos.	19
5.3. De la conversión de modelos no paramétricos a paramétricos.....	22
5.4. De la validación del modelo.	25
6. Identificación recursiva.	32
7. Aplicación práctica: Identificación y modelado de una planta con motores CC.....	33
7.1. Planteamiento del problema.	33
7.2. Registro de los datos de entrada-salida.	36
7.3. Tratamiento previo de los datos.....	37
7.4. Identificación no paramétrica mediante la respuesta transitoria.....	38
7.5. Identificación paramétrica.	40
7.6. Validación del modelo	41
APÉNDICE: Identificación de sistemas con MATLAB	42
BIBLIOGRAFÍA	60

1. Introducción

El diseño de un controlador continuo o discreto, ya sea mediante técnicas clásicas o en variables de estado, requiere de un modelo de la planta a controlar que caracterice su comportamiento dinámico. Este modelo permite al diseñador realizar y validar mediante simulación el ajuste de los parámetros del controlador que permiten obtener una respuesta que satisfaga las especificaciones de diseño. En este tema se estudian diferentes alternativas para obtener el modelo de un sistema como paso previo al diseño de un controlador.

1.1. Concepto de sistema

Un *sistema* es toda realidad en la que interactúan variables de diferentes tipos para producir señales observables. Las señales observables que son de interés para el observador se denominan *salidas* del sistema, mientras que las señales que pueden ser manipuladas libremente por dicho observador son las *entradas* del mismo. El resto de señales que influyen en la evolución de las salidas pero no pueden ser manipuladas por el observador se denominan *perturbaciones*.

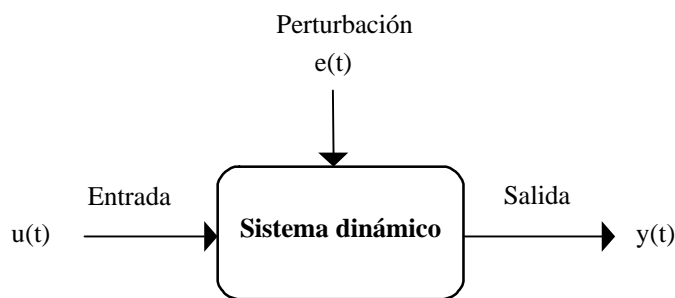


Figura 1. Sistema dinámico con entrada $u(t)$, perturbación $e(t)$ y salida $y(t)$.

1.2. Modelo de un sistema

Cuando se hace necesario conocer el comportamiento de un sistema en unas determinadas condiciones y ante unas determinadas entradas, se puede recurrir a la experimentación sobre dicho sistema y a la observación de sus salidas. Sin embargo, en muchos casos la experimentación puede resultar compleja o incluso imposible de llevar a cabo, lo que hace necesario trabajar con algún tipo de representación que se aproxime a la realidad, y a la que se conoce como *modelo*.

Básicamente, un modelo es una herramienta que permite predecir el comportamiento de un sistema sin necesidad de experimentar sobre él.

1.3. Tipos de modelos

Los modelos de sistemas físicos pueden ser de muy diversos tipos. Una clasificación, en función del grado de formalismo matemático que poseen, es la siguiente:

1. *Modelos mentales, intuitivos o verbales.* Estos modelos carecen de formalismo matemático. Para conducir un coche, por ejemplo, se requiere un modelo mental o intuitivo sobre el efecto que produce el movimiento del volante, pero no es necesario caracterizar dicho efecto mediante ecuaciones matemáticas exactas.
2. *Modelos no paramétricos.* Muchos sistemas quedan perfectamente caracterizados mediante un gráfico o tabla que describa sus propiedades dinámicas mediante un número no finito de parámetros. Por ejemplo, un sistema lineal queda definido mediante su respuesta al impulso o al escalón, o bien mediante su respuesta en frecuencia.
3. *Modelos paramétricos o matemáticos.* Para aplicaciones más avanzadas, puede ser necesario utilizar modelos que describan las relaciones entre las variables del sistema mediante expresiones matemáticas como pueden ser ecuaciones diferenciales (para sistemas continuos) o en diferencias (para sistemas discretos). En función del tipo de sistema y de la representación matemática utilizada, los sistemas pueden clasificarse en:
 - **Determinísticos o estocásticos.** Se dice que un modelo es determinístico cuando expresa la relación entre entradas y salidas mediante una ecuación exacta. Por contra, un modelo es estocástico si posee un cierto grado de incertidumbre. Estos últimos se definen mediante conceptos probabilísticos o estadísticos.
 - **Dinámicos o estáticos.** Un sistema es estático cuando la salida depende únicamente de la entrada en ese mismo instante (un resistor, por ejemplo, es un sistema estático). En estos sistemas existe una relación directa entre entrada y salida, independiente del tiempo. Un sistema dinámico es aquél en el que las salidas evolucionan con el tiempo tras la aplicación de una determinada entrada (por ejemplo, una red RC). En estos últimos, para conocer el valor actual de la salida es necesario conocer el tiempo transcurrido desde la aplicación de la entrada.
 - **Continuos o discretos.** Los sistemas continuos trabajan con señales continuas, y se caracterizan mediante ecuaciones diferenciales. Los sistemas discretos trabajan con señales muestreadas, y quedan descritos mediante ecuaciones en diferencias.

Todo modelo matemático o paramétrico, por tanto, consta de una o varias ecuaciones que relaciona/n la/s entrada/s y salida/s (en los modelos dinámicos la variable t -tiempo- juega también un papel primordial).

Ejemplo 1.

Un ejemplo de modelo matemático de un sistema dinámico con dos entradas u_1 y u_2 y una salida y puede ser el siguiente:

$$y(t) = 5 \cdot \frac{du_1(t)}{dt} + 3 \cdot u_1(t) - 2 \cdot u_2(t)$$

donde hay que distinguir:

1. la *estructura* del modelo: $y(t) = k_1 \cdot \frac{du_1(t)}{dt} + k_2 \cdot u_1(t) + k_3 \cdot u_2(t)$
2. los *parámetros* del modelo: $k_1 = 5; \quad k_2 = 3; \quad k_3 = -2;$

De ahí que a los modelos matemáticos se les conozca más comúnmente como *modelos paramétricos*, ya que pueden definirse mediante una *estructura* y un número finito de *parámetros*.

1.4. Métodos de obtención de modelos

Existen dos métodos principales para obtener el modelo de un sistema:

1. *Modelado teórico*. Se trata de un método analítico, en el que se recurre a leyes básicas de la física para describir el comportamiento dinámico de un fenómeno o proceso.
2. *Identificación del sistema*. Se trata de un método experimental que permite obtener el modelo de un sistema a partir de datos reales recogidos de la planta bajo estudio.

El modelado teórico tiene un campo de aplicación restringido a procesos muy sencillos de modelar, o a aplicaciones en que no se requiera gran exactitud en el modelo obtenido. En muchos casos, además, la estructura del modelo obtenido a partir del conocimiento físico de la planta posee un conjunto de parámetros desconocidos y que sólo se pueden determinar experimentando sobre el sistema real. De ahí la necesidad de recurrir a los métodos de *identificación de sistemas*.

Los modelos obtenidos mediante técnicas de identificación tienen, sin embargo, las siguientes desventajas:

1. Su rango de validez suele ser limitado (sólo son aplicables a un determinado punto de trabajo, un determinado tipo de entrada o un proceso concreto).
2. En muchos casos es difícil dar significado físico al modelo obtenido, puesto que los parámetros identificados no tienen relación directa con ninguna magnitud física. Estos parámetros se utilizan sólo para dar una descripción aceptable del comportamiento conjunto del sistema.

En la práctica, lo ideal es recurrir a una mezcla de ambos métodos de modelado para obtener el modelo final. El uso de datos reales para identificar los parámetros del modelo provee a éste de una gran exactitud, pero el proceso de identificación se ve tanto más facilitado cuanto mayor sea el conocimiento sobre las leyes físicas que rigen el proceso.

2. Identificación de sistemas

Se entiende por identificación de sistemas a la obtención de forma experimental de un modelo que reproduzca con suficiente exactitud, para los fines deseados, las características dinámicas del proceso objeto de estudio.

2.1. El proceso de identificación

En términos generales, el proceso de identificación comprende los siguientes pasos:

1. *Obtención de datos de entrada - salida.* Para ello se debe excitar el sistema mediante la aplicación de una señal de entrada y registrar la evolución de sus entradas y salidas durante un intervalo de tiempo.
2. *Tratamiento previo de los datos registrados.* Los datos registrados están generalmente acompañados de ruidos indeseados u otro tipo de imperfecciones que puede ser necesario corregir antes de iniciar la identificación del modelo. Se trata, por tanto, de ‘preparar’ los datos para facilitar y mejorar el proceso de identificación.
3. *Elección de la estructura del modelo.* Si el modelo que se desea obtener es un modelo paramétrico, el primer paso es determinar la estructura deseada para dicho modelo. Este punto se facilita en gran medida si se tiene un cierto conocimiento sobre las leyes físicas que rigen el proceso.

4. *Obtención de los parámetros del modelo.* A continuación se procede a la estimación de los parámetros de la estructura que mejor ajustan la respuesta del modelo a los datos de entrada-salida obtenidos experimentalmente.
5. *Validación del modelo.* El último paso consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión. Si se llega a la conclusión de que el modelo no es válido, se deben revisar los siguientes aspectos como posibles causas:
 - a) El conjunto de datos de entrada-salida no proporciona suficiente información sobre la dinámica del sistema.
 - b) La estructura escogida no es capaz de proporcionar una buena descripción del modelo.
 - c) El criterio de ajuste de parámetros seleccionado no es el más adecuado.

Dependiendo de la causa estimada, deberá repetirse el proceso de identificación desde el punto correspondiente. Por tanto, el proceso de identificación es un proceso iterativo, cuyos pasos pueden observarse en el organigrama de la figura 2.

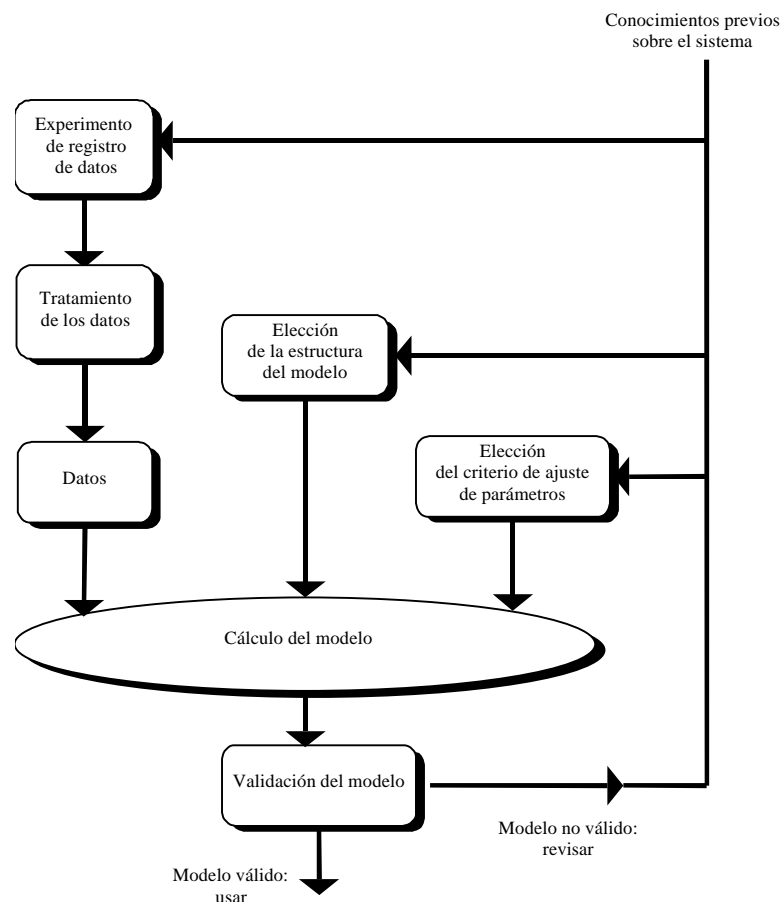


Figura 2. El proceso de identificación

2.2. Métodos de identificación

Existen diversos métodos de identificación, que pueden clasificarse según distintos criterios:

- Dependiendo del tipo de modelo obtenido:
 1. *Métodos no paramétricos*, que permiten obtener modelos no paramétricos del sistema bajo estudio. Algunos de estos métodos son: análisis de la respuesta transitoria, análisis de la respuesta en frecuencia, análisis de la correlación, análisis espectral, análisis de Fourier, etc.
 2. *Métodos paramétricos*, que permiten obtener modelos paramétricos. Estos métodos requieren la elección de una posible estructura del modelo, de un criterio de ajuste de parámetros, y por último de la estimación de los parámetros que mejor ajustan el modelo a los datos experimentales.
- Dependiendo de la aplicación:
 1. *Métodos de identificación off-line (a posteriori)*, utilizados en aquellas aplicaciones en que no se requiera un ajuste continuado del modelo. En estos casos, se realiza la identificación previa de la planta, considerándose que la validez de los parámetros obtenidos no se verá alterada con el paso del tiempo.
 2. *Métodos de identificación on-line (identificación recursiva)*, en los que los parámetros se van actualizando continuamente a partir de los nuevos datos de entrada-salida obtenidos durante la evolución del proceso. Estos métodos son muy utilizados en sistemas de control adaptativo.
- Dependiendo del criterio de ajuste de los parámetros. Existen diversos métodos matemáticos para ajustar los parámetros de una estructura a un conjunto de datos de entrada-salida. Algunos de los más utilizados en el campo de la identificación son el método de mínimos cuadrados y el método de las variables instrumentales.

3. Técnicas de identificación no paramétrica

Los métodos de identificación no paramétricos permiten obtener modelos o representaciones no paramétricas de la planta bajo estudio.

3.1. Conceptos previos

Supóngase el siguiente sistema:

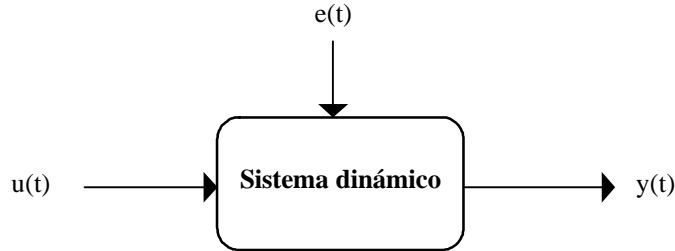


Figura 3. Señales de interés para el modelado no paramétrico

Suponiendo que el sistema es lineal, la relación entre la salida del sistema $y(t)$, su entrada $u(t)$ y el ruido $e(t)$ puede expresarse del siguiente modo:

$$y(t) = G(q^{-1}) \cdot u(t) + e(t) \quad (ec.1)$$

donde q^{-1} es el operador retardo y el producto $G(q^{-1})u(t)$ representa la siguiente secuencia:

$$G(q^{-1}) \cdot u(t) = \sum_{k=1}^{\infty} g(k)u(t-k) \quad (ec.2)$$

y por tanto:

$$G(q^{-1}) = \sum_{k=1}^{\infty} g(k)q^{-k} \quad (ec.3)$$

La secuencia $g(k)$ se conoce como *respuesta al impulso* del sistema, y coincide con la salida del mismo cuando a la entrada se aplica un impulso unitario. Por otro lado, la función $G(z)$ es la *función de transferencia* del sistema. Evaluando esta última a lo largo del círculo unidad ($z=e^{j\omega}$) se obtiene la llamada *respuesta en frecuencia* del sistema, $G(e^{j\omega})$.

La respuesta al impulso es un modelo no paramétrico que se define en el dominio del tiempo, mientras que la respuesta en frecuencia es una descripción no paramétrica en el dominio de la frecuencia.

3.2. Identificación no paramétrica en el dominio del tiempo

Mediante esta técnica de identificación se pretende obtener la respuesta al impulso del sistema, o bien la respuesta al escalón del mismo (pudiendo obtenerse esta última mediante una integración de la primera). Para ello, debe registrarse la evolución temporal de la salida del sistema tras la aplicación de una señal impulso o escalón. Obviamente, la imposibilidad de

conseguir este tipo de señales en la práctica lleva a utilizar un método indirecto para obtener la respuesta impulsiva, conocido como *análisis de la correlación*.

Si se escoge como entrada al sistema $u(t)$ un ruido blanco, cuya función de covarianza es:

$$R_u(\tau) = E[u(t+\tau)u(t)] = \begin{cases} \lambda & \text{si } \tau = 0 \\ 0 & \text{si } \tau \neq 0 \end{cases} \quad (ec.4)$$

entonces la correlación cruzada entre la entrada y la salida puede ponerse del siguiente modo:

$$R_{yu}(\tau) = E[y(t+\tau)u(t)] = \lambda g(\tau) \quad (ec.5)$$

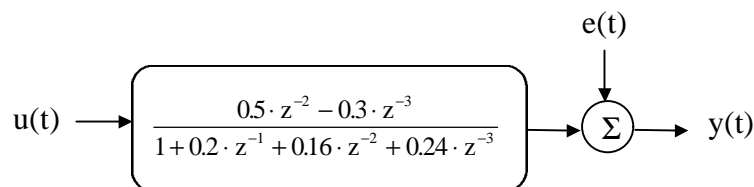
y por tanto la respuesta al impulso puede obtenerse a partir de las N muestras registradas de la entrada y la salida del sistema del siguiente modo:

$$g(\tau) = \frac{1}{\lambda N} \sum_{t=1}^N y(t+\tau)u(t) \quad (ec.6)$$

Si la entrada al sistema no es un ruido blanco puro, puede considerarse como el resultado de la aplicación de un filtro $L(q^{-1})$ a un ruido blanco. Aplicando dicho filtro a los datos de salida del sistema puede realizarse la misma operación indicada anteriormente para obtener los coeficientes de $g(k)$.

Este método es muy apropiado para obtener una idea rápida de la relación entre distintas señales del sistema, retardos, constantes de tiempo y ganancias estáticas del mismo.

Ejemplo 2. Supóngase el siguiente sistema:



En primer lugar se realizará una simulación del mismo en Matlab incluyendo la presencia de un ruido aleatorio para obtener un conjunto de datos de entrada-salida que sirvan de punto de partida para la identificación (revísense las funciones del Toolbox de Identificación en el apéndice).

```
>> numd=[0.5 -0.3];           %numerador de la función de transferencia en z
>> dend=[1 0.2 0.16 0.24];    %denominador de la función de transferencia en z
```

```
>> u=idinput(1000,'PRBS',[0 0.25],[0 50]); %Generación de una entrada binaria
pseudoaleatoria con 1000 muestras
>> e=randn(1000,1); %Generación del ruido aleatorio, también con 1000 muestras
>> y_sin=dlsim(numd,dend,u); %Obtención mediante simulación de la salida sin ruido
>> y=y_sin+e; %Adición del ruido a la salida.
>> datos=[y u]; %Construcción de la matriz con los datos de entrada-salida
```

A continuación se realiza un análisis de la correlación a partir de los datos de entrada-salida anteriores y se comprueba la similitud de la respuesta al impulso resultante con la obtenida directamente mediante la simulación del sistema real ante la aplicación de una entrada impulso.

```
>> cra(datos); %Análisis y representación gráfica de la respuesta impulsiva obtenida
por análisis de correlación
>>hold on; dimpulse(numd,dend); %Representación gráfica de la respuesta impulsiva
obtenida por simulación del sistema real
```

En la figura 4, los círculos corresponden al análisis de correlación, mientras que los escalones representan la respuesta al impulso del sistema real, observándose una gran similitud entre ambos resultados.

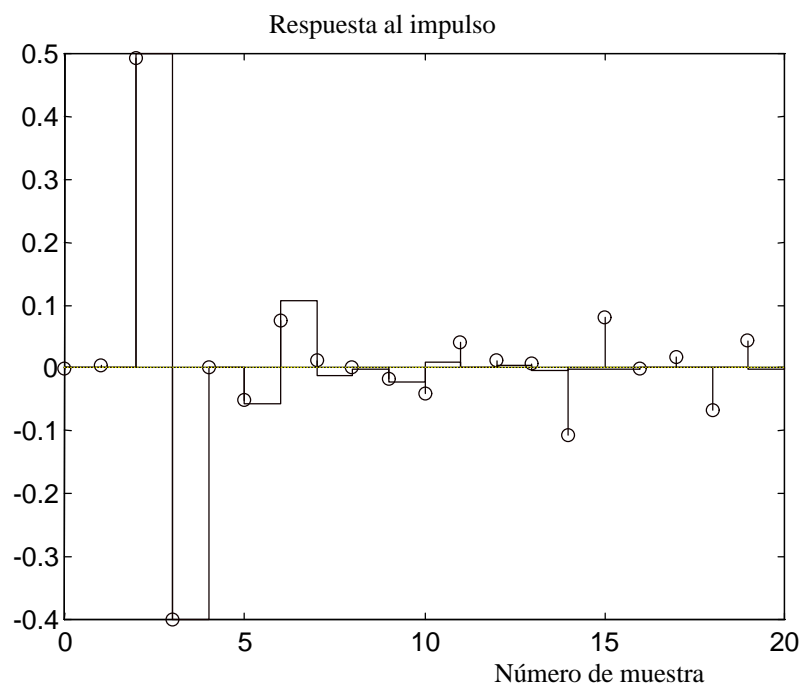


Figura 4. Respuestas al impulso real (escalones) y estimada (círculos) mediante análisis de correlación.

3.3. Identificación no paramétrica en el dominio de la frecuencia

En este caso, el modelo resultante es una representación de la respuesta en frecuencia del sistema, obtenida mediante la aplicación de señales de entrada sinusoidales de distintas frecuencias. Cuando no sea posible aplicar este tipo de entradas, puede recurrirse a la aplicación de un ruido blanco, que permite obtener la respuesta en frecuencia mediante el conocido *análisis espectral*. Este análisis se basa en la realización de la transformada de Fourier de las funciones de covarianza de la entrada y la salida y la correlación entre la entrada y la salida. Por tanto, definiendo las siguientes funciones de correlación:

$$\begin{aligned} R_u(\tau) &= E[u(t + \tau) \cdot u(t)] = \frac{1}{N} \cdot \sum_{t=1}^N u(t + \tau) \cdot u(t) \\ R_{yu}(\tau) &= E[y(t + \tau) \cdot u(t)] = \frac{1}{N} \cdot \sum_{t=1}^N y(t + \tau) \cdot u(t) \end{aligned} \quad (ec.7)$$

y sus transformadas de Fourier:

$$\begin{aligned} \Phi_u(\omega) &= \sum_{\tau=-\infty}^{\infty} R_u(\tau) \cdot e^{-j\omega\tau} \\ \Phi_{yu}(\omega) &= \sum_{\tau=-\infty}^{\infty} R_{yu}(\tau) \cdot e^{-j\omega\tau} \end{aligned} \quad (ec.8)$$

se demuestra que puede obtenerse la respuesta en frecuencia del sistema mediante la siguiente expresión:

$$G(e^{j\omega}) = \frac{\Phi_{yu}(\omega)}{\Phi_u(\omega)} \quad (ec.9)$$

Las principales ventajas de este método son el no requerir un procesamiento complejo de los datos, ni ningún tipo de conocimiento previo sobre la planta, a excepción de que ésta sea lineal. Además, permite concentrar los datos obtenidos en torno al margen de frecuencias de interés. El principal inconveniente es que el modelo resultante no puede usarse directamente para simulación.

4. Técnicas de identificación paramétrica

Los modelos paramétricos, a diferencia de los anteriores, quedan descritos mediante una estructura y un número finito de parámetros que relacionan las señales de interés del sistema (entradas, salida y perturbaciones). En muchas ocasiones es necesario realizar la identificación de un sistema del cual no se tiene ningún tipo de conocimiento previo. En estos casos, se suele recurrir a modelos estándar, cuya validez para un amplio rango de sistemas dinámicos ha sido comprobada experimentalmente. Generalmente estos modelos permiten

describir el comportamiento de cualquier sistema lineal. La dificultad radica en la elección del tipo de modelo (orden del mismo, número de parámetros, etc.) que se ajuste satisfactoriamente a los datos de entrada - salida obtenidos experimentalmente.

4.1. Tipos de modelos paramétricos

Generalmente los modelos paramétricos se describen en el dominio discreto, puesto que los datos que sirven de base para la identificación se obtienen por muestreo. En el caso de que se requiera un modelo continuo, siempre es posible realizar una transformación del dominio discreto al continuo.

La expresión más general de un modelo discreto es del tipo:

$$s(t) = \eta(t) + w(t) \quad (ec.10)$$

donde $w(t)$ es el término que modela la salida debida a las perturbaciones, $\eta(t)$ la salida debida a la entrada, y $s(t)$ la salida medible del sistema. Cada uno de estos términos puede desarrollarse de la siguiente forma:

$$\eta(t) = G(q^{-1}, \theta) \cdot u(t) \quad (ec.11)$$

$$w(t) = H(q^{-1}, \theta) \cdot e(t) \quad (ec.12)$$

$$s(t) = A(q^{-1}, \theta) \cdot y(t) \quad (ec.13)$$

donde q^{-1} es el operador retardo, θ representa un vector de parámetros, $u(t)$ y $e(t)$ son la entrada al sistema y el ruido de entrada al mismo respectivamente e $y(t)$ es la salida de interés del sistema (que puede no coincidir con la salida medible). Tanto $G(q^{-1}, \theta)$ como $H(q^{-1}, \theta)$ son cocientes de polinomios del tipo:

$$G(q^{-1}, \theta) = \frac{B(q^{-1})}{F(q^{-1})} = \frac{b_1 \cdot q^{-nk} + b_2 \cdot q^{-nk-1} + \dots + b_{nb} \cdot q^{-nk-nb+1}}{1 + f_1 \cdot q^{-1} + \dots + f_{nf} \cdot q^{-nf}} \quad (ec.14)$$

$$H(q^{-1}, \theta) = \frac{C(q^{-1})}{D(q^{-1})} = \frac{1 + c_1 \cdot q^{-1} + \dots + c_{nc} \cdot q^{-nc}}{1 + d_1 \cdot q^{-1} + \dots + d_{nd} \cdot q^{-nd}}$$

y $A(q^{-1}, \theta)$ un polinomio del tipo: $A(q^{-1}, \theta) = 1 + a_1 \cdot q^{-1} + \dots + a_{na} \cdot q^{-na} \quad (ec.15)$

El vector de parámetros θ contiene los coeficientes a_i , b_i , c_i , d_i y f_i de las funciones de transferencia anteriores. La estructura genérica de estos modelos es por tanto:

$$A(q^{-1}) \cdot y(t) = G(q^{-1}, \theta) \cdot u(t) + H(q^{-1}, \theta) \cdot e(t) = \frac{B(q^{-1})}{F(q^{-1})} \cdot u(t) + \frac{C(q^{-1})}{D(q^{-1})} \cdot e(t) \quad (ec.16)$$

Para elegir la *estructura* de este tipo de modelos hay que determinar el orden de cada uno de los polinomios anteriores, es decir na, nb, nc, nd, nf y el retardo entre la entrada y la salida nk. Una vez elegidos estos valores, sólo queda determinar el vector de coeficientes θ (a_i , b_i , c_i , d_i y f_i) que hacen que el modelo se ajuste a los datos de entrada - salida del sistema real.

En muchos casos, alguno de los polinomios anteriores no se incluye en la descripción del modelo, dando lugar a los siguientes casos particulares, entre otros:

Tipo de modelo	Condición	Estructura resultante
Modelo ARX	$F(q^{-1})=D(q^{-1})=C(q^{-1})=1$	$A(q^{-1})y(t) = B(q^{-1}) \cdot u(t) + e(t)$
Modelo Output Error (OE)	$C(q^{-1})=D(q^{-1})=A(q^{-1})=1$	$y(t) = \frac{B(q^{-1})}{F(q^{-1})} \cdot u(t) + e(t)$
Modelo ARMAX	$F(q^{-1})=D(q^{-1})=1$	$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + C(q^{-1}) \cdot e(t)$
Modelo Box Jenkins (BJ)	$A(q^{-1})=1$	$y(t) = \frac{B(q^{-1})}{F(q^{-1})} \cdot u(t) + \frac{C(q^{-1})}{D(q^{-1})} \cdot e(t)$

Tabla 1. Diferentes estructuras de modelos paramétricos.

En la figura 5 se muestra el diagrama de bloques equivalente para cada uno de los modelos anteriores.

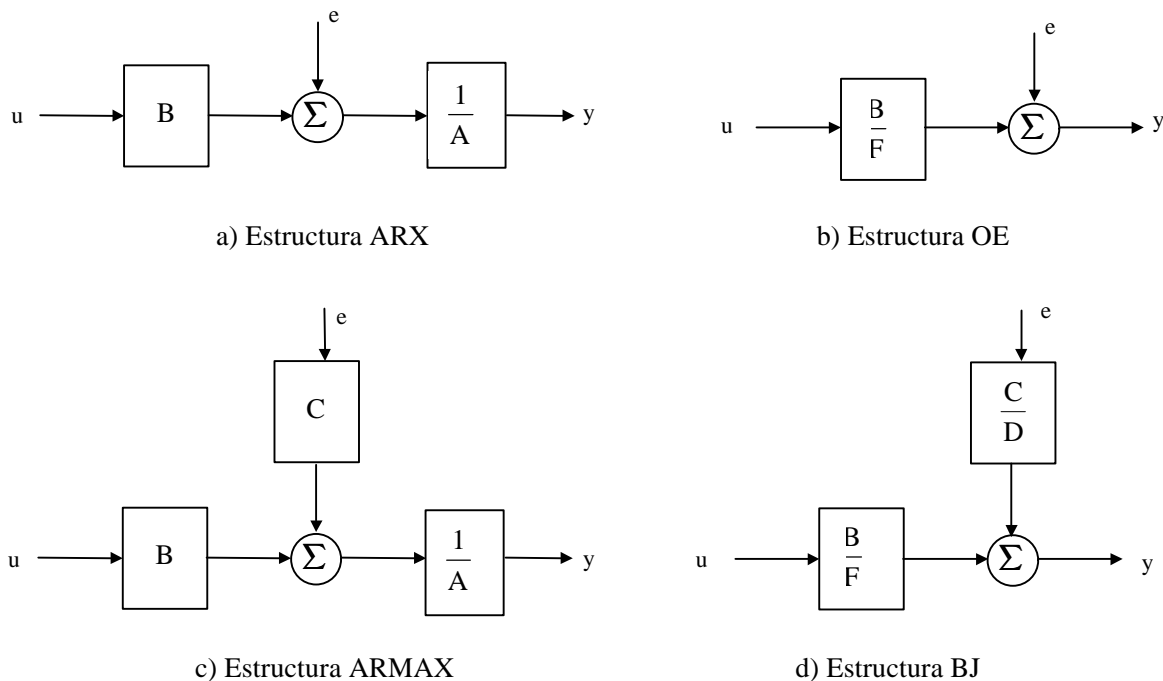


Figura 5. Diagramas de bloques de las estructuras de la tabla 1.

La anulación de alguno de los polinomios, resultando estructuras simplificadas, facilita el proceso de ajuste de parámetros. Cada una de las estructuras (ARX, ARMAX, OE o BJ) tiene sus propias características y debe ser elegida fundamentalmente en función del punto en el que se prevé que se añade el ruido en el sistema. En cualquier caso, puede ser necesario ensayar con varias estructuras y con varios órdenes dentro de una misma estructura hasta encontrar un modelo satisfactorio.

Ejemplo 3

Para el sistema del ejemplo 2, y dado que el ruido se suma directamente a la salida, el tipo de estructura más apropiada para identificación debe ser del tipo “Output Error” (OE). Dentro de este tipo, el número de parámetros óptimo de los polinomios B y F son los que coinciden con la estructura real del sistema, es decir:

$$y(t) = \frac{B(q^{-1})}{F(q^{-1})} \cdot u(t) + e(t) = \frac{b_1 \cdot q^{-2} + b_2 \cdot q^{-3}}{1 + f_1 \cdot z^{-1} + f_2 \cdot z^{-2} + f_3 \cdot z^{-3}} \cdot u(t) + e(t)$$

y por tanto $nb=2$, $nf=3$ y $nk=2$.

Sin embargo, es importante tener en cuenta que en la mayoría de los casos el diseñador no dispone de esta información sobre el sistema real, por lo que será necesario ensayar con varios tipos de estructuras y número de parámetros hasta dar con el modelo que mejor se ajusta a los datos de entrada-salida registrados.

4.2. Métodos para el ajuste de parámetros

Una vez elegida la estructura del modelo (tanto el tipo - ARX, ARMAX, BJ, OE...- como los órdenes de cada polinomio), es necesario determinar el valor de los parámetros del mismo que ajustan la respuesta del modelo a los datos de entrada - salida experimentales. Es importante destacar, sin embargo, que esta etapa del proceso de identificación se ve facilitada por la existencia de herramientas software que proporcionan diferentes algoritmos para el ajuste de parámetros. Una de estas herramientas es el *Toolbox de Identificación* de Matlab (ver apéndice).

Existen varios métodos o criterios para realizar este ajuste de parámetros, entre los que cabe destacar el método de mínimos cuadrados y el de variables instrumentales.

4.2.1. Errores de predicción o residuos de un modelo

Todo modelo matemático es capaz de predecir el valor de la salida del sistema en función de las entradas y salidas en instantes anteriores. Se llama error de predicción $\varepsilon(t, \theta)$ a la diferencia entre la salida estimada por el modelo y la salida real del sistema en un determinado instante de tiempo:

$$\varepsilon(t, \theta) = y(t) - y_e(t, \theta) \quad (ec.17)$$

donde $y_e(t, \theta)$ es la salida estimada por el modelo en el instante t .

4.2.2. Regresión lineal

Se dice que una estructura posee regresión lineal cuando la salida estimada puede expresarse como:

$$y_e(t, \theta) = \varphi^T(t) \cdot \theta \quad (ec.18)$$

donde $\varphi^T(t)$ es un vector columna formado por las salidas y entradas anteriores (conocido como *vector de regresión*), y θ es el vector de parámetros del modelo.

El modelo ARX es un claro ejemplo de estructura con regresión lineal, definiendo:

$$\theta = [a_1 \quad a_2 \quad \cdots \quad a_{na} \quad b_1 \quad \cdots \quad b_{nb}]^T \quad (ec.19)$$

$$\varphi^T(t) = [-y(t-1) \quad \cdots \quad -y(t-na) \quad u(t-nk) \quad \cdots \quad u(t-nk-nb+1)]$$

4.2.3. Método de mínimos cuadrados (LSE)

Aplicando los criterios fijados en los dos apartados anteriores, la expresión del error de predicción es:

$$\varepsilon(t, \theta) = y(t) - \varphi^T(t) \cdot \theta \quad (ec.20)$$

Se define la siguiente función del error:

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} \cdot [y(t) - \varphi^T(t) \cdot \theta]^2 \quad (ec.21)$$

conocida como *criterio de mínimos cuadrados* para una regresión lineal.

Existe un valor de θ que minimiza la función anterior y que constituye la *estimación del modelo por mínimos cuadrados*:

$$\theta_{\text{LSE}} = \text{sol} \left\{ \frac{1}{N} \cdot \sum_{t=1}^N \phi^T(t) \cdot [y(t) - \phi^T(t) \cdot \theta] = 0 \right\} \quad (\text{ec.22})$$

Para este vector de parámetros, la función de error V_N toma su valor mínimo, siendo éste la *función de pérdidas* del modelo estimado.

Una variante del método anterior, conocido como *Criterio de Akaike* consiste en minimizar otra función de pérdidas distinta a la anterior, que puede obtenerse a partir de ésta del siguiente modo:

$$V_{\text{AIC}}(\theta) = V_N(\theta) \cdot (1 + 2 \cdot d/N) \quad (\text{ec.23})$$

siendo d el número de parámetros del modelo y N el número de muestras de los datos de entrada-salida utilizados para su identificación.

Ejemplo 4

Utilice el método de mínimos cuadrados para estimar los parámetros del modelo OE escogido en el ejemplo 3, utilizando para ello los datos de entrada-salida generados en el ejemplo 2. Estime también un modelo ARX del mismo orden que el anterior, y compare los parámetros de los modelos obtenidos con los del sistema real.

```
>> th_oe = oe(datos,[2 3 2]);
```

```
>> present(th_oe)
```

This matrix was created by the command OE on 4/30 1999 at 10:53

Loss fcn: 0.967 Akaike`s FPE: 0.9767 Sampling interval 1

The polynomial coefficients and their standard deviations are

B =

```
0      0  0.4988 -0.2973
0      0  0.0023  0.0034
```

F =

```
1.0000  0.1995  0.1529  0.2421
0  0.0070  0.0063  0.0058
```

Por tanto, el modelo OE obtenido es el siguiente:

$$y(t) = \frac{0.4988 \cdot q^{-2} - 0.2973 \cdot q^{-3}}{1 + 0.1995 \cdot q^{-1} + 0.1529 \cdot q^{-2} + 0.2421 \cdot q^{-3}} \cdot u(t)$$

$$\frac{Y(z)}{U(z)} = \frac{0.4988 \cdot z^{-2} - 0.2973 \cdot z^{-3}}{1 + 0.1995 \cdot z^{-1} + 0.1529 \cdot z^{-2} + 0.2421 \cdot z^{-3}}$$

siendo sus parámetros prácticamente idénticos a los del sistema real. Entre la información del modelo se encuentran tanto la función de pérdidas por el método de mínimos cuadrados simple (0.967) como con el criterio de Akaike (0.9767).

A continuación se realiza la estimación del modelo ARX:

```
>> th_arx = arx(datos,[3 2 2]);
>> present(th_arx)
This matrix was created by the command ARX on 4/30 1999 at 11:8
Loss fcn: 1.119 Akaike's FPE: 1.13 Sampling interval 1
The polynomial coefficients and their standard deviations are
B = 0 0 0.4963 -0.3054
    0 0 0.0025 0.0031
A =
    1.0000 0.1766 0.1398 0.2204
         0 0.0073 0.0069 0.0065
```

Por tanto, el modelo ARX obtenido es el siguiente:

$$y(t) = \frac{0.4963 \cdot q^{-2} - 0.3054 \cdot q^{-3}}{1 + 0.1766 \cdot q^{-1} + 0.1398 \cdot q^{-2} + 0.2204 \cdot q^{-3}} \cdot u(t)$$

$$\frac{Y(z)}{U(z)} = \frac{0.4963 \cdot z^{-2} - 0.3054 \cdot z^{-3}}{1 + 0.1766 \cdot z^{-1} + 0.1398 \cdot z^{-2} + 0.2204 \cdot z^{-3}}$$

En este caso los parámetros se desvían más que en el anterior puesto que se supone el ruido añadido en otro punto del sistema (ver figura 5). Puede observarse que las funciones de pérdidas son superiores que en el caso anterior.

4.2.4. Método de variables instrumentales

Mediante este método, el vector de parámetros debe cumplir la relación:

$$\theta_{IV} = \text{sol} \left\{ \frac{1}{N} \cdot \sum_{t=1}^N \xi(t) [y(t) - \phi^T(t) \cdot \theta] = 0 \right\} \quad (ec.24)$$

donde los elementos del vector $\xi(t)$ son las llamadas *variables instrumentales*, que resultan de aplicar algún tipo de filtro lineal al vector de regresión lineal $\phi^T(t)$. Este método es en realidad una generalización del método de mínimos cuadrados, que proporciona mejores resultados en aquellos casos en que existe algún tipo de correlación entre el ruido y la salida del sistema.

5. Consideraciones prácticas sobre identificación

En este apartado se revisan algunas consideraciones prácticas a tener en cuenta durante el proceso de identificación.

5.1. De la obtención de los datos

El primer paso dentro del proceso de identificación es realizar algún tipo de experimento sobre el sistema bajo estudio para obtener los datos de entrada-salida que servirán de base para la obtención del modelo final.

Para que el proceso de identificación sea satisfactorio, es necesario que los datos utilizados para tal fin contengan información significativa sobre el sistema. Esto implica un cuidadoso diseño del experimento de adquisición de datos, debiéndose tomar una serie de decisiones respecto a las señales que deben ser medidas, el periodo de muestreo a utilizar, el tipo de entrada más adecuada, el número de datos a almacenar, etc.

5.1.1. Elección de las señales a medir

La primera decisión es qué señales se deben registrar (mediante algún tipo de sistema de adquisición y el correspondiente sistema de almacenamiento de datos), y qué señales deben ser manipuladas para excitar al sistema durante el experimento. Se debe tener en cuenta que pueden existir señales que, aunque afecten a la evolución de la salida, no pueden considerarse como entradas debido a la imposibilidad de actuar sobre ellas. En el caso de que estas señales puedan ser medidas, pueden considerarse también como entradas al sistema (midiéndose sus valores durante el experimento). En caso contrario, deben ser consideradas como perturbaciones.

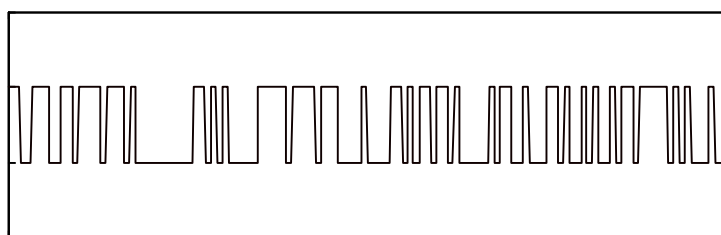
5.1.2 Elección del tipo de entrada/s

La/s entrada/s al sistema deben ser cuidadosamente elegidas de forma que los datos recogidos proporcionen toda la información posible sobre el sistema. A este respecto, conviene tener en cuenta los siguientes aspectos:

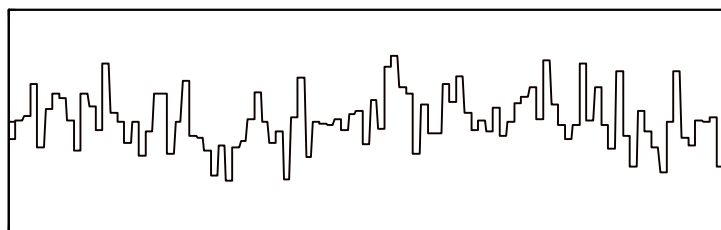
- La señal de entrada debe contener el mayor número de frecuencias posibles. Por ejemplo, una señal senoidal pura no es adecuada en un experimento de identificación, puesto que

sólo se obtendrá la respuesta del sistema para la frecuencia de dicha señal. Por el contrario, las señales escalonadas (con cambios bruscos) son muy utilizadas, puesto que contienen un espectro suficientemente amplio de frecuencias.

- Para sistemas lineales, basta con utilizar dos niveles de entrada, preferiblemente barriendo todo el rango de variación permitido. En este tipo de sistemas se suelen utilizar señales binarias de duración aleatoria (conocidas como señales binarias aleatorias o pseudoaleatorias), como la mostrada en la figura 6.a). Sin embargo, para sistemas no lineales es necesario trabajar con más de dos niveles de entrada, como se muestra en la figura 6.b).



a)



b)

Figura 6. a) Entrada binaria aleatoria para sistemas lineales.
b) Entrada escalonada aleatoria para sistemas no lineales

- Si se sabe que el sistema va a trabajar preferentemente en torno a un determinado punto de trabajo, es conveniente realizar el registro de datos en ese mismo entorno. Este aspecto adquiere especial importancia si el sistema no es lineal.

5.1.3. Elección del periodo de muestreo

La elección del periodo de muestreo está directamente relacionada con las constantes de tiempo del sistema, y tiene una influencia decisiva en el experimento de identificación. Así, un periodo de muestreo muy pequeño puede llevar a la obtención de datos redundantes, que no aportan información sobre el sistema (pero sí ocupan espacio en la memoria del dispositivo de almacenamiento de datos), mientras que un periodo de muestreo demasiado grande provoca grandes dificultades a la hora de identificar la dinámica del sistema.

Una regla comúnmente usada consiste en escoger una frecuencia de muestreo alrededor de diez veces el ancho de banda del sistema. Esto corresponde aproximadamente a muestrear en torno a cinco u ocho valores del tiempo de subida de la respuesta al escalón del sistema.

5.1.4. Elección del número de muestras a tomar

En principio, cuanta más información se tenga sobre el sistema, más exacto será el proceso de identificación. En la práctica, el número de muestras a recoger durante el experimento de identificación viene limitado por la capacidad del dispositivo de memoria utilizado. Por tanto, es importante llegar a un buen compromiso en la elección del periodo de muestreo y el número de muestras a tomar.

5.2. Del pretratamiento de los datos

Los datos registrados pueden tener deficiencias que implican efectos devastadores en el resto del proceso de identificación, como son las siguientes:

- Presencia de perturbaciones de alta frecuencia, por encima de las frecuencias de interés en la respuesta del sistema.
- Datos claramente erróneos, producidos por fallos en el hardware o software utilizados en el experimento de recogida de muestras.
- Desviaciones, desplazamientos o perturbaciones de baja frecuencia.

A continuación, se verá la forma de tratar cada una de estas deficiencias para conseguir unos datos adecuados para el proceso de identificación.

5.2.1. Eliminación de perturbaciones de alta frecuencia

Estas perturbaciones se producen por fuentes de ruido ajenas al sistema y pueden ser evitadas mediante una correcta elección del período de muestreo. Si, tras el experimento, se observa que el período de muestreo escogido era innecesariamente pequeño (captándose por tanto estas perturbaciones indeseadas), se puede recurrir al diezmado de los datos, para evitar repetir el experimento con un período de muestreo mayor.

5.2.2. Eliminación de datos erróneos

Estos datos suelen presentarse de forma aislada, pero pueden tener un efecto muy negativo en el proceso de identificación. Por tanto, es fundamental eliminarlos antes de iniciar el proceso. Esto se realiza generalmente manualmente, eliminando dicho dato y aproximando su nuevo

valor mediante interpolación. Para aplicaciones más avanzadas, existen algoritmos de detección de fallos que permiten corregir estos datos de forma casi automática.

5.2.3. Tratamiento de niveles de continua

Las perturbaciones de baja frecuencia, desplazamientos, desviaciones o variaciones periódicas en los datos de entrada-salida pueden causar problemas si no son tratados convenientemente. Los modelos paramétricos usados para identificación responden a determinadas estructuras, como puede ser la ARX mostrada a continuación:

$$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + e(t) \quad (ec.25)$$

Se trata de una ecuación en diferencias que establece una relación lineal entre la secuencia de salida $y(t)$, la secuencia de entrada $u(t)$ y una fuente de ruido $e(t)$, siendo q^{-1} el operador retardo. Este modelo, en principio, debería caracterizar tanto la dinámica del sistema (variaciones en torno a un punto de trabajo), como su respuesta en régimen permanente, es decir, cuando $u(t)$ e $y(t)$ se estabilizan en un valor que llamaremos u_0 e y_0 respectivamente. Para este último caso, la ecuación anterior equivale a:

$$A(1) \cdot y_0 = B(1) \cdot u_0 \quad (ec.26)$$

En la práctica, los valores obtenidos para la identificación ($y_m(t)$ y $u_m(t)$), se miden en unidades físicas, cuyos niveles pueden ser arbitrarios. Por tanto, una vez encontrados los polinomios $A(q^{-1})$ y $B(q^{-1})$ de la ecuación (ec.25) que satisfacen la dinámica del sistema, es muy probable que no cumplan la ecuación (ec.26) que relaciona sus niveles de continua. Y en el caso de que se desee encontrar dos polinomios $A(q^{-1})$ y $B(q^{-1})$ que satisfagan simultáneamente la características estáticas y dinámicas del sistema, estos polinomios deben ser de un grado muy superior al realmente necesario para describir el sistema. Luego se hace necesario aportar una solución a la presencia de niveles de continua en los datos obtenidos para identificación.

Existen distintas vías para solucionar el problema anterior, algunas de las cuales se comentan a continuación:

- Si la planta bajo estudio va a trabajar en torno a un punto de trabajo conocido, basta con modelar el comportamiento del sistema en torno a dicho punto de operación. Por tanto, el modelo sólo debe satisfacer las condiciones dinámicas del sistema, no debiendo cumplir la relación (ec.26). Una vez determinado el punto de trabajo deseado, (y_0 , u_0), se realiza el siguiente tratamiento sobre los datos de entrada - salida:

$$\begin{aligned} y(t) &= y_m(t) - y_0 \\ u(t) &= u_m(t) - u_0 \end{aligned} \quad (ec.27)$$

Los nuevos datos de entrada - salida $y(t)$ y $u(t)$ representan las desviaciones de los datos originales en torno al punto de equilibrio, y satisfacen simultáneamente las ecuaciones (ec.25) y (ec.26). En el caso de la ecuación (ec.26), ambos miembros de la igualdad se hacen cero, al ser (0,0) el nuevo punto de equilibrio de los datos.

- En el caso de que la planta vaya a trabajar en torno a puntos de trabajo desconocidos, se puede realizar una aproximación al método anterior considerando los valores medios de las entradas y salidas como un posible punto de equilibrio del sistema. Es decir:

$$y_0 = \frac{1}{N} \cdot \sum_{t=1}^N y_m(t) \quad ; \quad u_0 = \frac{1}{N} \cdot \sum_{t=1}^N u_m(t) \quad (ec.28)$$

Si una entrada que varía en torno a u_0 produce una salida que varía en torno a y_0 , entonces el punto (u_0, y_0) es un buen candidato a ser un punto de equilibrio del sistema, o estar cercano a él.

- Como ya se ha comentado, el modelo obtenido por cualquiera de los dos métodos anteriores caracteriza correctamente la dinámica del sistema, tanto para los pares de datos $(y(t), u(t))$ como para los pares $(y_m(t), u_m(t))$, que sólo difieren en sus niveles continuos. Sin embargo, dicho modelo sólo satisface el comportamiento estático para los pares $(y(t), u(t))$, obtenidos a partir de la operación (ec.27) (puesto que ambos miembros de la ecuación 3.26 se hacen 0). Lógicamente, podría darse el caso de que la ecuación (ec.26) también se cumpliera de forma fortuita con los datos $(y_m(t), u_m(t))$, de forma que el modelo caracterizaría completamente al sistema. Si éste no es el caso, y se desea un modelo que describa completamente al sistema (no sólo su dinámica), al modelo $A(q^{-1}) B(q^{-1})$ obtenido por los métodos anteriores, se le debe añadir una constante que realice la corrección oportuna de los niveles de continua:

$$A(q^{-1}) \cdot y_m(t) = B(q^{-1}) \cdot u_m(t) + \alpha + e(t) \quad (ec.29)$$

donde $\alpha = A(1) \cdot y_0 - B(1) \cdot u_0$.

- El modelo anterior no es un modelo lineal, y por tanto, no puede ser tratado mediante la teoría de sistemas lineales. Una solución a este problema consiste en considerar la constante α como una perturbación constante, que puede modelarse como:

$$\frac{\alpha}{1 - q^{-1}} \delta(t) \quad (ec.30)$$

siendo $\delta(t)$ el impulso unitario. De esta forma, el modelo completo es:

$$y_m(t) = \frac{A(q^{-1})}{B(q^{-1})} \cdot u_m(t) + \frac{1}{(1-q^{-1}) \cdot A(q^{-1})} \cdot w(t) \quad (ec.31)$$

donde $w(t) = \alpha \cdot \delta(t) + e(t)$.

El desplazamiento α , por tanto, puede ser descrito cambiando el modelo de ruido de $1/A(q^{-1})$ a $1/((1-q^{-1}) \cdot A(q^{-1}))$. Esto es equivalente a realizar un prefiltrado de los datos a través de un filtro $L(q^{-1}) = 1 - q^{-1}$, es decir, a realizar una diferenciación de los datos:

$$\begin{aligned} y_m^F(t) &= L(q^{-1}) \cdot y_m(t) = y_m(t) - y_m(t-1) \\ u_m^F(t) &= L(q^{-1}) \cdot u_m(t) = u_m(t) - u_m(t-1) \end{aligned} \quad (ec.32)$$

Es importante notar que el modelo (ec.31) es un caso especial del (ec.25) si los órdenes de $A(q^{-1})$ y $B(q^{-1})$ en este último se incrementan en uno. Esto supone que un modelo de orden superior del tipo (ec.25) puede converger con los datos $(y_m(t), u_m(t))$, como ya se comentó al presentar la problemática de los niveles de continua.

5.3. De la conversión de modelos no paramétricos a paramétricos

Los modelos no paramétricos se basan siempre en la obtención de la respuesta transitoria o en frecuencia del sistema que, si bien pueden utilizarse directamente para el diseño de controladores, no sucede lo mismo para simulación. Si se desea simular dichos modelos, es necesario obtener una aproximación paramétrica de los mismos. A modo de ejemplo, se expondrán a continuación varias aproximaciones paramétricas de modelos no paramétricos obtenidos mediante el análisis de la respuesta transitoria.

La señal más simple que puede utilizarse para el análisis del transitorio es, sin duda, la función escalón. La respuesta de un sistema simple a la señal escalón puede aproximarse generalmente mediante uno de los tres siguientes modelos paramétricos en el dominio continuo:

1. Modelo de primer orden con retardo:

$$G(s) = \frac{A}{\tau \cdot s + 1} \cdot e^{-L \cdot s} \quad (ec.33)$$

2. Modelo de segundo orden sobreamortiguado con retardo:

$$G(s) = \frac{A}{(\tau_1 \cdot s + 1) \cdot (\tau_2 \cdot s + 1)} \cdot e^{-L \cdot s} \quad (ec.34)$$

3. Modelo de segundo orden subamortiguado con retardo:

$$G(s) = \frac{A}{s^2/\omega_n^2 + 2 \cdot \xi \cdot s/\omega_n + 1} \cdot e^{-L \cdot s} \quad (ec.35)$$

donde ξ es el coeficiente de amortiguamiento y ω_n la frecuencia natural del sistema.

La elección de uno u otro de los modelos anteriores depende de la forma de la respuesta transitoria y del grado de precisión que se desea en el ajuste. Los modelos del tipo (1) pueden utilizarse en procesos simples u otros más complejos si no se desea un alto grado de exactitud. Los modelos del tipo (2) pueden usarse prácticamente en muchos casos salvo en procesos cuya respuesta es francamente oscilatoria. Los modelos del tipo (3) están destinados al caso de procesos oscilatorios y también en algunos casos de procesos sobreamortiguados en los que se logra un mejor ajuste que con un modelo del tipo (2).

Como ejemplo típico, se expone a continuación el método gráfico que se ha utilizado tradicionalmente para la estimación de los parámetros de un modelo de primer orden con retardo.

Si la respuesta al escalón del proceso tiene la forma de la figura 7, que responde a un sistema de primer orden, la expresión que describe la respuesta es:

$$y(t) = A \cdot U \cdot (1 - e^{-\frac{-(t-L)}{\tau}}) \quad (ec.36)$$

donde U es la amplitud del escalón aplicado a la entrada.

Para $t = L + \tau$, $y(L + \tau) = A \cdot U \cdot (1 - e^{-1}) = 0.632 \cdot A \cdot U$. De esta forma, localizando el punto donde la respuesta alcanza el 63.2% de su valor final, se obtiene el valor de τ . La ganancia A se determina como:

$$A = \frac{Y_{est}}{U}$$

donde Y_{est} es el valor estacionario de la salida.

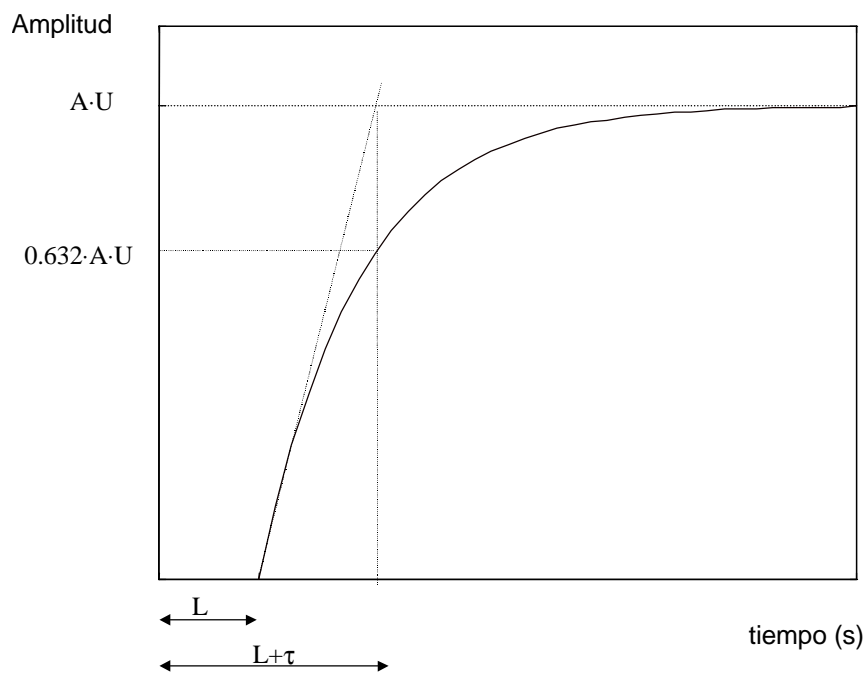


Figura 7. Respuesta al escalón de un sistema de primer orden con retardo.

También es frecuente el caso de que la respuesta tenga la forma mostrada en la figura 8. En este caso se trata de aproximar la respuesta de un sistema de orden superior mediante un modelo de primer orden. Esta aproximación se usa frecuentemente para diversos propósitos (ajustes preliminares de los parámetros de un PID, compensación feed-forward, predictor de Smith, etc.).

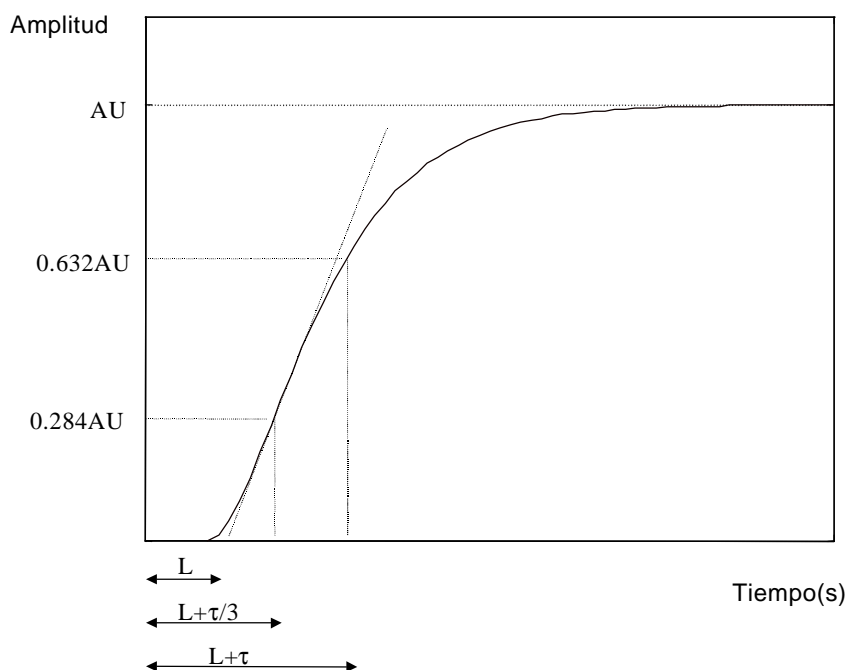


Figura 8. Aproximación de un sistema de orden mayor a uno de primer orden.

Los parámetros del modelo pueden calcularse en este caso usando la construcción gráfica mostrada en la figura 8 y siguiendo el procedimiento:

1. Se toman dos puntos de la curva que corresponden al 63.2% y al 28.4% del valor estacionario final y se determinan $t_{0,284}$ y $t_{0,632}$.
2. Se plantean las ecuaciones:

$$\begin{aligned}t_{0,284} &= L + \frac{\tau}{3} \\ t_{0,632} &= L + \tau\end{aligned}$$

3. Se calculan los parámetros L y τ .

5.4. De la validación del modelo

En todo proceso de identificación es conveniente probar varias estructuras y diferentes órdenes dentro de cada estructura hasta dar con el modelo que mejor se ajuste a los datos obtenidos experimentalmente de la planta real. En definitiva, se trata de determinar cuándo un determinado modelo es lo suficientemente exacto para la aplicación requerida, proceso que se conoce habitualmente como *validación del modelo*.

En general, la mayoría de los métodos de validación tratan de determinar si la respuesta del modelo se ajusta con suficiente exactitud a los datos de entrada-salida obtenidos mediante experimentación. A continuación se exponen algunos criterios típicos a la hora de descartar o elegir unos modelos respecto a otros.

a) Validación en base a la aplicación del modelo

Puesto que en la práctica es imposible determinar si un modelo responde exactamente al comportamiento de un sistema real, suele ser suficiente comprobar que el modelo es capaz de resolver el problema para el cual ha sido hallado (simulación, predicción, diseño de un controlador, etc.). Así, por ejemplo, si el controlador que ha sido ajustado por medio del modelo da buen resultado sobre el sistema real, se puede asegurar que el modelo era ‘válido’ para esta aplicación.

b) Comprobación de parámetros físicos

Para una determinada estructura que haya sido parametrizada en función de magnitudes físicas, un método importante de validación consiste en comparar el valor estimado de dichos parámetros y el que sería de esperar mediante el conocimiento previo que se tiene de la planta.

c) Coherencia con el comportamiento de entrada-salida

Para determinar si el comportamiento de entrada-salida está suficientemente caracterizado, puede ser necesario recurrir a diferentes métodos de identificación y comparar los resultados obtenidos. Por ejemplo, comparando los diagramas de Bode de los modelos obtenidos mediante identificación paramétrica de diferentes estructuras, por el método de variables instrumentales y por análisis espectral, se puede determinar si la dinámica del sistema ha quedado suficientemente caracterizada.

Ejemplo 5

Compare la respuesta en frecuencia obtenida mediante identificación no paramétrica directa del sistema de los ejemplos anteriores con la obtenida indirectamente a través del modelo paramétrico del ejemplo 4 (modelo OE).

```
>> g=spa(datos); %Identificación no paramétrica en el dominio de la frecuencia  
>> g1=th2ff(th_oe); %Obtención de la respuesta en frecuencia a partir del modelo OE  
>> ffplot([g g1]); % Representación gráfica de ambas
```

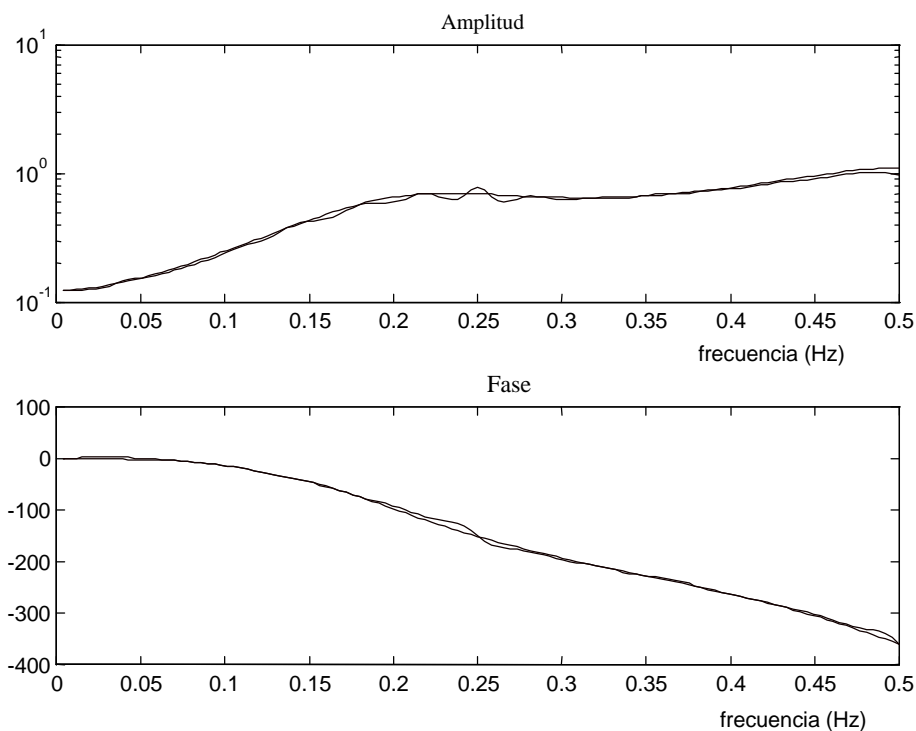


Figura 9. Comparación de la respuesta en frecuencia obtenida mediante dos métodos de identificación distintos.

d) Reducción del modelo

Un procedimiento para determinar si un modelo proporciona una descripción simple y apropiada de un sistema consiste en aplicarle algún método de reducción de modelos. Si una reducción en el orden del modelo no produce alteraciones apreciables en el comportamiento de entrada-salida del mismo, entonces el modelo original era innecesariamente complejo.

Ejemplo 6

Realice la estimación por el método de mínimos cuadrados de un modelo OE con un parámetro más en los polinomios B y F que los que tiene el sistema real, y realice una representación de los ceros y polos del modelo obtenido.

```
>> th_oe1=oe(datos,[3 4 2]); %Estimación de los parámetros del modelo
>> zp=th2zp(th_grande);      %Obtención de ceros y polos del modelo resultante
>> zpplot(zp);               %Representación del diagrama de ceros y polos
```

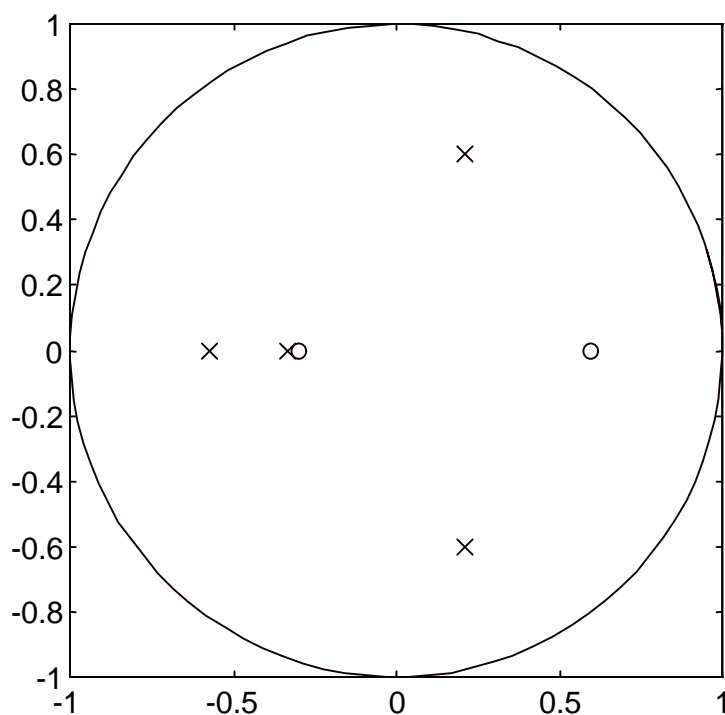


Figura 10. Diagrama de ceros y polos obtenido

La presencia de un cero y un polo que prácticamente se cancelan informa sobre la posibilidad de reducir en uno el orden de los polinomios B y F.

e) Intervalos de fiabilidad de parámetros

Otro método para determinar si el modelo bajo estudio contiene demasiados parámetros consiste en comparar los parámetros estimados con su desviación estándar. Si el intervalo de confianza de un parámetro contiene el valor cero, se debe considerar la posibilidad de eliminar dicho parámetro.

Ejemplo 7

Analice las desviaciones estándar del modelo obtenido en ejemplo anterior, y compruebe que a partir de ellas también se deduce la posibilidad de reducir el orden del modelo.

```
>> present(th_op1)
```

This matrix was created by the command OE on 4/30 1999 at 12:41

Loss fcn: 1.01 Akaike's FPE: 1.024 Sampling interval 1

The polynomial coefficients and their standard deviations are

B =

<i>0</i>	<i>0</i>	<i>0.4931</i>	<i>-0.1479</i>	<i>-0.0869</i>
<i>0</i>	<i>0</i>	<i>0.0024</i>	<i>0.2361</i>	<i>0.1425</i>

F =

<i>1.0000</i>	<i>0.4930</i>	<i>0.2204</i>	<i>0.2906</i>	<i>0.0786</i>
<i>0</i>	<i>0.4778</i>	<i>0.0908</i>	<i>0.0754</i>	<i>0.1135</i>

Puede comprobarse que el último parámetro de cada polinomio tiene una desviación superior al propio parámetro, lo que informa sobre la posibilidad de eliminar dicho parámetro.

f) Simulación

Un procedimiento muy habitual que puede ser considerado como otra técnica de validación de modelos consiste en simular el modelo con un conjunto de entradas distintas a las utilizadas para identificación, y comparar la respuesta del modelo con la obtenida del sistema real.

Ejemplo 8

Valide mediante simulación el modelo OE de orden [2 3 2]. Para ello, aplique la misma entrada al sistema real y al modelo, y compare las respuestas obtenidas. Se propone como ejercicio ensayar diferentes estructuras y órdenes, y contrastar los resultados.

```
>> u_valid=idinput(500,'PRBS',[0 0.2],[0 100]);
>> y_valid=dlsim(numd,dend,u_valid);
>> datos_valid=[y_valid u_valid];
>> compare(datos_valid,th_oe);
```

g) Análisis de residuos

Se conocen como *residuos* de un sistema a los errores de predicción obtenidos según la expresión:

$$\varepsilon(t) = \varepsilon(t, \theta) = y(t) - y_e(t, \theta) \quad (ec.37)$$

siendo θ el vector de parámetros del modelo, $y(t)$ la respuesta real del sistema e $y_e(t)$ la respuesta estimada por el modelo para la misma entrada.

Idealmente, estos residuos deben ser independientes de la entrada. Si no sucede así, significa que hay componentes en $\varepsilon(t)$ que proceden de la entrada $u(t)$, lo cual a su vez significa que el modelo no es capaz de describir completamente la dinámica del sistema.

Para realizar el estudio anterior, suele comprobarse la correlación entre el error de predicción y la entrada al sistema, según la expresión:

$$R_{\varepsilon u} = \frac{1}{N} \sum_{t=1}^N \varepsilon(t + \tau) \cdot u(t) \quad (ec.38)$$

El modelo será tanto más exacto cuanto más se acerquen a cero los términos de la correlación anterior. Puede demostrarse que si $\varepsilon(t)$ y $u(t)$ son realmente independientes, la expresión anterior (para valores grandes de N) es una distribución normal, con media cero y varianza

$$P_r = \frac{1}{N} \sum R_{\varepsilon}(k) \cdot R_u(k) \quad (ec.39)$$

donde R_{ε} y R_u son las covarianzas de $\varepsilon(t)$ y $u(t)$ respectivamente.

Generalmente, $R_{eu}(\tau)$ se representa en un diagrama junto con las líneas $\pm 3 \cdot \sqrt{P_r}$. Si $R_{eu}(\tau)$ sobrepasa dichas líneas para algún valor de τ , significa que $\varepsilon(t+\tau)$ y $u(t)$ probablemente no son independientes para ese valor de τ .

A la hora de examinar la función de correlación, es importante tener en cuenta los siguientes aspectos:

- Si existe correlación para valores negativos de τ , esto indica que existe realimentación de la salida hacia la entrada, no que el modelo sea deficiente.
- Si la estimación de un modelo y su expresión de correlación entre residuos y entrada R_{eu} se han determinado utilizando los mismos datos de entrada, entonces $R_{eu}(\tau)=0$ para $\tau=nk, \dots, nk+nb-1$.
- Si $R_{eu}(\tau)$ es considerablemente distinto de cero para un valor τ_0 , esto indica que el término $u(t-\tau_0)$ debería ser incluido en el modelo. Éste es un buen método para ajustar el orden más apropiado de la estructura del modelo.

Obviamente, el análisis de los residuos será un método de validación más eficaz si el conjunto de datos utilizados para realizar la correlación es distinto que el usado para la identificación del modelo.

Ejemplo 9

Realice una representación de los residuos y la correlación cruzada entre la entrada y los residuos para determinar la validez del modelo OE de orden [2 3 2] obtenido anteriormente:

```
>> resid(datos_valid,th_oe);
```

Puesto que la correlación no se sale del margen de validez, se deduce que la dinámica del sistema queda suficientemente caracterizada con el modelo escogido.

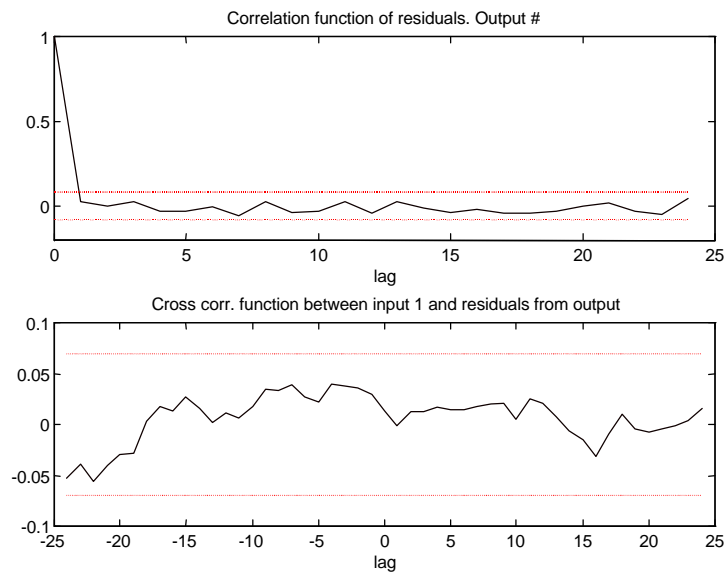


Figura 11. Análisis de residuos para el modelo OE de orden [2 3 2].

Sin embargo, si se estima un modelo de orden inferior (por ejemplo, con un parámetro menos en el polinomio F), el análisis de residuos sería el mostrado en la figura 12.

```
>> th_red=oe(datos,[2 2 2]);
>> resid(datos_valid,th_red);
```

El pico que aparece en $\tau=3$ informa de la necesidad de introducir un parámetro más al modelo, concretamente con retardo q^{-3} , que es el que se ha eliminado respecto al sistema real.

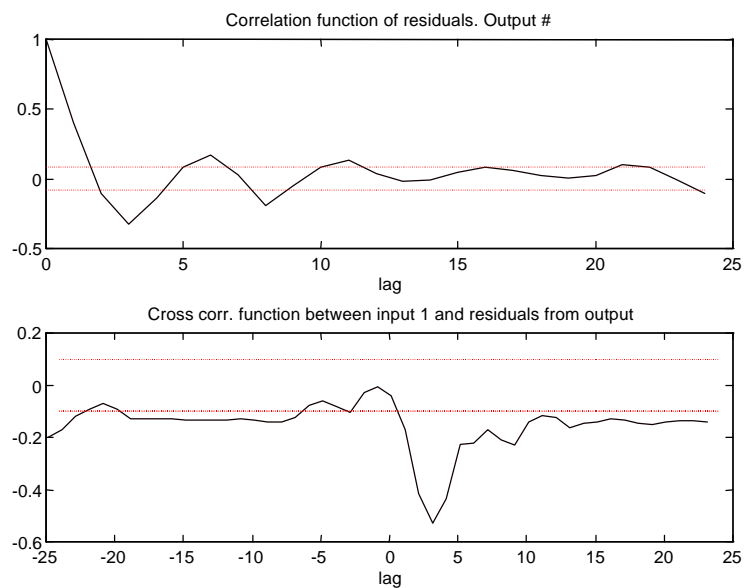


Figura 12. Análisis de residuos para el modelo OE de orden [2 2 2].

6. Identificación recursiva

En determinadas ocasiones puede ser necesario estimar los parámetros de un modelo a la vez que se reciben los datos de entrada-salida del mismo, como sucede en los sistemas de control adaptativo. Esto permite actualizar los parámetros del modelo en el caso de que se produzcan variaciones en la planta. Este tipo de algoritmos se conocen como *métodos de identificación recursiva*.

Un algoritmo de identificación recursiva típico es el siguiente:

$$\theta(t) = \theta(t-1) + K(t) \cdot (y(t) - y_e(t)) \quad (ec.40)$$

donde $\theta(t)$ es el vector de parámetros estimado en el instante t , $y(t)$ la salida real del sistema en dicho instante de tiempo, $y_e(t)$ la salida estimada con los parámetros actuales, y $\theta(t-1)$ el vector de parámetros del modelo en el instante de tiempo anterior. De esta forma, $K(t)$ determina el modo en que el error de predicción $y(t)-y_e(t)$ afecta en la actualización on-line de los parámetros del modelo. Generalmente $K(t)$ se elige del siguiente modo:

$$K(t) = Q(t) \cdot \psi(t) \quad (ec.41)$$

donde $\psi(t)$ representa el gradiente de la salida estimada respecto al vector de parámetros $y_e(t|\theta)$. Por otro lado, la matriz $Q(t)$ puede ser escogida según diferentes criterios, dando lugar a diferentes métodos de adaptación.

Uno de los criterios más utilizados consiste en suponer que los parámetros reales de la planta θ_0 cambian aleatoriamente del siguiente modo:

$$\theta_0(t) = \theta_0(t-1) + w(t) \quad (ec.42)$$

siendo $w(t)$ un ruido blanco gaussiano con matriz de covarianza:

$$R_1 = Ew(t) \cdot w^T(t) \quad (ec.43)$$

Suponiendo que el modelo puede escribirse como una regresión lineal, uno de los métodos más sencillos de escoger la matriz $Q(t)$ está basado en el filtro de Kalman, dando lugar al siguiente algoritmo:

$$\begin{aligned} \theta(t) &= \theta(t-1) + K(t) \cdot (y(t) - y_e(t)) \\ y_e(t) &= \psi^T(t) \cdot \theta(t-1) \\ K(t) &= Q(t) \cdot \psi(t) \end{aligned} \quad (ec.44)$$

$$Q(t) = \frac{P(t-1)}{R_2 + \psi^T(t) \cdot P(t-1) \cdot \psi(t)}$$
$$P(t) = P(t-1) + R_1 - \frac{P(t-1) \cdot \psi(t) \cdot \psi^T(t) \cdot P(t-1)}{R_2 + \psi^T(t) \cdot P(t-1) \cdot \psi(t)}$$

donde R_2 es la varianza del error de predicción del modelo.

7. Aplicación práctica: Identificación y modelado de una planta con motor de continua

Como ejemplo práctico de identificación de un sistema real se propone en esta práctica la obtención de forma experimental del modelo de una planta basada en un motor de continua. Dicho modelo puede obtenerse mediante varios métodos o, en la mayoría de los casos, mediante una combinación de éstos:

1. Conocimiento de las leyes físicas que rigen el comportamiento del motor.
2. Información proporcionada por los fabricantes sobre sus parámetros principales.
3. Identificación experimental basada en el registro de datos de entrada-salida.

Si bien el estudio del comportamiento físico de un motor de continua permite obtener un modelo teórico del mismo cuyos parámetros principales son proporcionados por los fabricantes, es conveniente obtener y validar el modelo de forma experimental, mediante técnicas de identificación. Esto permite eliminar las tolerancias de los datos proporcionados por los fabricantes y modelar efectos de los cuales no informan, como puede ser el retardo entre la entrada y la salida.

7.1. Planteamiento del problema

Las alternativas hardware para controlar un motor de continua desde un dispositivo digital son múltiples. La mayor parte de ellas actúan sobre el motor a través de un sistema electrónico que convierte un código digital (V_{cod}) en una tensión proporcional al mismo (V_m) capaz de excitar al motor y leen la velocidad de éste (W_m) mediante un tacómetro digital que genera un código proporcional a la velocidad (W_{cod}), según la estructura mostrada en la figura 13.

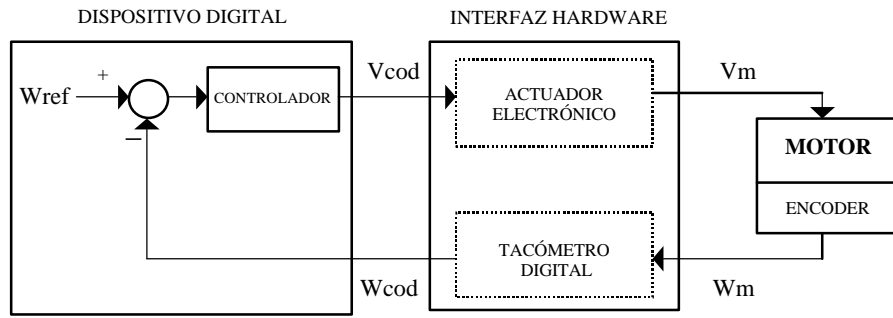


Figura 13. Diagrama de bloques de un sistema de control digital de un motor CC.

Desde este punto de vista, la planta sobre la que actúa el controlador digital está formada por el conjunto Actuador+Motor+Tacómetro, cuyo modelado es necesario abordar como etapa previa al diseño del controlador.

La función de transferencia teórica que modela cada uno de los bloques que componen la planta es la siguiente:

1. *Actuador electrónico.* La relación entre la tensión que llega al motor V_m , y el código digital V_{cod} debe ser lineal, de forma que el actuador electrónico se puede modelar como una ganancia de valor:

$$K_{act} = \frac{V_m(s)}{V_{cod}(s)} = \frac{V_{m_{\max}}}{V_{cod_{\max}}} = \frac{V_N}{2^n - 1}$$

siendo V_N la tensión que llega al motor cuando V_{cod} toma su valor máximo ($2^n - 1$ siendo n el número de bits de V_{cod}).

2. *Motor de continua.* El comportamiento bastante lineal de estos motores facilita la caracterización de los mismos. En vacío, sin la acción de carga externa, la velocidad angular de salida W_m en función de la tensión continua V_m aplicada puede simplificarse mediante la relación:

$$M(s) = \frac{W_m(s)}{V_m(s)} = \frac{K_E^{-1}}{\frac{R_a \cdot J_m}{K_M \cdot K_E} \cdot s + 1} \cdot e^{-L_m \cdot s} = \frac{A_m}{\tau_m \cdot s + 1} \cdot e^{-L_m \cdot s}$$

siendo K_E la constante de fuerza contraelectromotriz, K_M la constante de par, R_a la resistencia de armadura, J_m la inercia del rotor y L_m el retardo. Estos parámetros pueden englobarse en una ganancia estática A_m (rpm/V), en una constante de tiempo electromecánica τ_m (s) y en un retardo L_m (s).

3. *Tacómetro digital*. Realiza la cuenta de los pulsos de encoder en intervalos fijos y conocidos de tiempo T_s , lo cual permite obtener un código digital proporcional a la velocidad de giro del motor según la siguiente expresión:

$$W_{cod} = \frac{W_m \cdot N \cdot T_s}{60}$$

siendo W_m la velocidad del motor en revoluciones por minuto (rpm), N el número de pulsos por vuelta generados por el encoder y T_s el periodo de lectura del tacómetro en segundos. Fijado el valor de este último, el tacómetro se puede modelar también como una ganancia de valor:

$$K_{tac} = \frac{W_{cod}(s)}{W_m(s)} = \frac{N \cdot T_s}{60}$$

Por tanto, la función de transferencia $P(s)$ de la planta completa (conjunto actuador + motor + tacómetro) responde a una expresión del siguiente tipo:

$$P(s) = K_{act} \cdot M(s) \cdot K_{tac} = \frac{K_{act} \cdot A_m \cdot K_{tac}}{\tau_m \cdot s + 1} \cdot e^{-L_m \cdot s} = \frac{A}{\tau \cdot s + 1} \cdot e^{-L \cdot s}$$

en la que aparecen como parámetros la ganancia estática total A , la constante de tiempo total τ y el retardo total entre la entrada y la salida L .

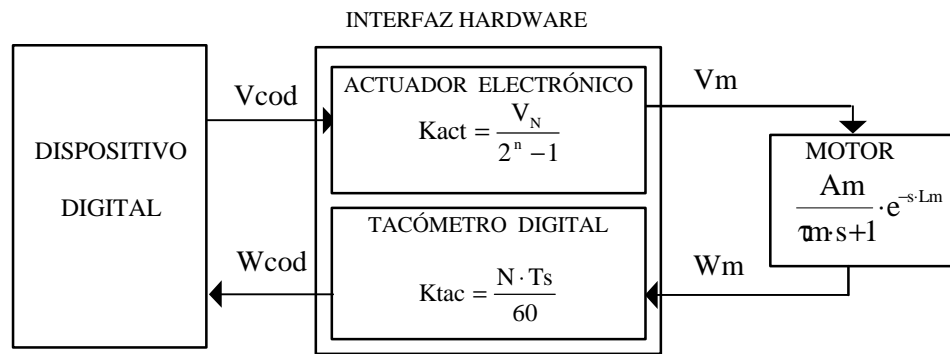


Figura 14. Modelo teórico de la planta a identificar.

Conocida la estructura teórica del modelo de la planta, es necesario obtener el valor de sus parámetros. Para ello puede recurrirse a los datos proporcionados por el fabricante, con la limitación de que dichos datos se refieren al modelo del motor, pudiendo estar sujetos a importantes tolerancias. La alternativa consiste en realizar una identificación experimental basada en el registro de datos de entrada-salida y la posterior aplicación de las técnicas de identificación de sistemas vistas en este capítulo.

7.2. Registro de datos de entrada-salida

Para que el proceso de identificación sea satisfactorio, los datos recogidos deben contener información significativa sobre la planta.

1. Escriba un programa en C que utilice el puerto paralelo de un PC para enviar distintas consignas de tensión al motor (V_{cod}) y leer la respuesta del mismo (W_{cod}).

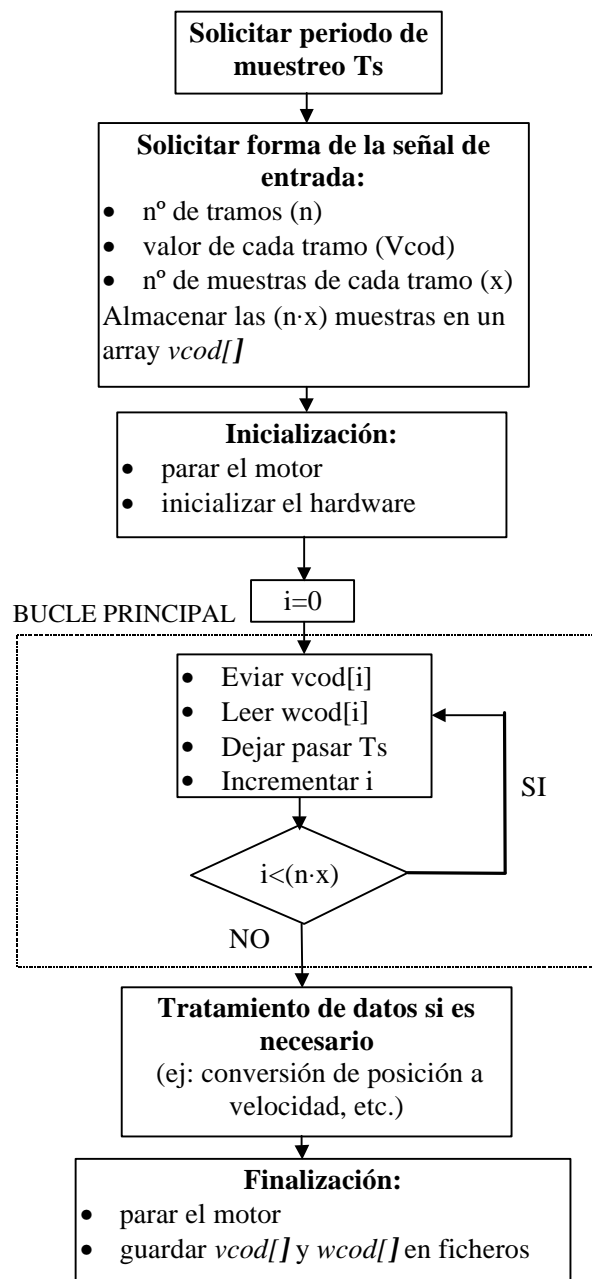


Figura 15. Organigrama del programa de registro de datos de entrada-salida.

El programa debe permitir:

- Seleccionar distintos periodos de muestreo.
- Seleccionar distintas entradas (sucesión de escalones de distinta amplitud y duración).
- Almacenar los datos de entrada y salida en ficheros binarios para poder recuperarlos después desde Matlab, para el posterior análisis de los mismos con el Toolbox de Identificación (ver apéndice).

Un posible organigrama del programa a realizar se muestra en la figura 15.

2. ¿Qué factores son determinantes en la elección del periodo de muestreo? Justifíquese el valor escogido para realizar el registro de datos y comente qué sucede al aumentar o disminuir su valor.

7.3. Tratamiento previo de los datos

Antes de proceder a realizar la identificación es necesario analizar los datos registrados y decidir si son o no adecuados para el proceso de identificación, o si necesitan algún tipo de tratamiento previo, como puede ser el filtrado de ruidos, eliminación de componentes de variación lenta, etc.

En la figura 16 se muestra un ejemplo de datos registrados desde un PC utilizando el puerto paralelo para enviar consignas de tensión a través del bus de datos (8 bits) y leer la velocidad codificada por un contador a través del mismo bus.

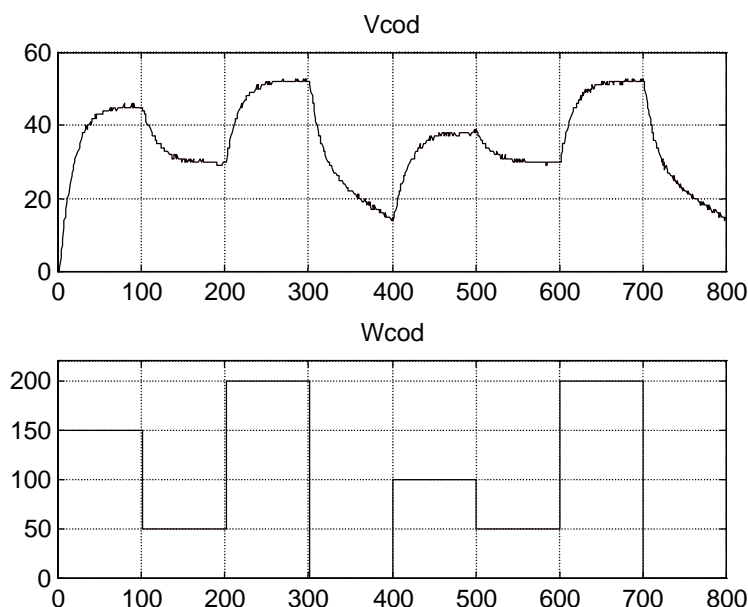


Figura 16. Ejemplo de registro de datos de entrada-salida de una planta basada en motor CC.

1. Recupere en Matlab los datos recogidos en la fase anterior mediante el uso de las funciones *fopen()*, *fread()* y *fclose()*.
2. Realice la representación gráfica de los datos utilizando para ello la función del Toolbox de Identificación *idplot()*.
3. Decida si es necesario realizar un tratamiento de los datos. En caso de que sea necesario, el Toolbox de Identificación proporciona, entre otras, las siguientes funciones para la manipulación de datos: *dtrend()* y *idfilt()*.

7.4. Identificación no paramétrica mediante la respuesta transitoria

El análisis de la respuesta transitoria registrada permite obtener de forma gráfica la ganancia estática, constante de tiempo y retardo de la planta a identificar.

Ejemplo 10.

Del análisis del primer escalón de la figura 16 pueden extraerse las siguientes conclusiones:

1. El retardo puede obtenerse realizando un listado de los primeros datos contenidos en los vectores *Vcod* y *Wcod*:

```
>> [Vcod(1:5) Wcod(1:5)]
```

```
150    0
150    0
150    1
150    4
```

Puesto que las dos primeras muestras son nulas, puede asegurarse que $T_s \leq L < 2 \cdot T_s$.

2. La ganancia estática se obtiene de la relación entre el valor final de *Wcod* y *Vcod*, tal y como se muestra en la siguiente figura:

$$A = \frac{W_{cod_{ss}}}{V_{cod_{ss}}} = \frac{45}{150}$$

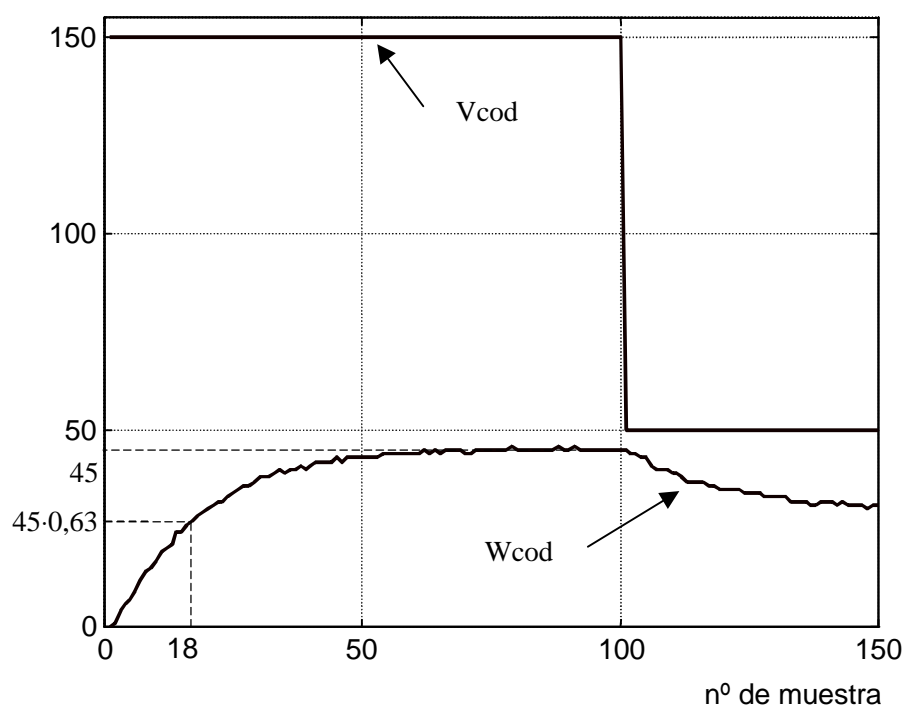


Figura 17. Análisis de la respuesta transitoria

3. Por último, la constante de tiempo se obtiene multiplicando el número de muestras que transcurren desde que la señal comienza a evolucionar hasta que alcanza el 63% de su valor final por el periodo de muestreo con el que se han adquirido los datos. En este caso:

$$\tau = (18 - 2) \cdot T_s$$

1. A partir de los datos de entrada-salida obtenidos, realice una estimación de la constante de tiempo de la planta completa (conjunto del actuador, motor y tacómetro digital), su ganancia estática y su retardo. Indique el procedimiento seguido para obtener dichos valores.
2. Obtenga a partir de los resultados del apartado anterior el valor de la ganancia estática, constante de tiempo y retardo del motor. Para ello, mida previamente sobre el interfaz hardware el valor exacto de las constantes K_{act} y K_{tac} .
3. Compare los resultados obtenidos con la información proporcionada por el fabricante del motor.
4. Indique el efecto del periodo de muestreo en la exactitud de la identificación de cada uno de los parámetros: ganancia estática, retardo y constante de tiempo.

7.5. Identificación paramétrica

Los métodos de identificación paramétrica se basan en la obtención del modelo a partir de una estructura y un número finito de parámetros. Entre las distintas estructuras (AR, ARX, ARMAX, OE, BJ...), se ensayará únicamente la ARX, por ser una de las más sencillas. Dicha estructura responde a la siguiente ecuación en diferencias para caracterizar el sistema:

$$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + e(t)$$

Un ejemplo de modelo ARX puede ser el siguiente:

$$y(t) + a_1 \cdot y(t-1) + a_2 \cdot y(t-2) = b_0 \cdot u(t-3) + b_2 \cdot u(t-4)$$

o en el dominio transformado:

$$\frac{y(z)}{u(z)} = z^{-3} \cdot \frac{b_0 + b_1 \cdot z^{-1}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}}$$

La obtención de un modelo de este tipo se realiza en dos pasos:

1. *Elección de la estructura del modelo.* En función de los conocimientos previos sobre el sistema, debe escogerse el número de parámetros de los polinomios A y B, así como el retardo del sistema. En el ejemplo anterior, el número de parámetros de $A(q^{-1})$ es 2, el de $B(q^{-1})$ es 2 y el retardo del sistema es 3.
2. *Obtención de los parámetros del modelo* (en el ejemplo a_1 , a_2 , b_0 , y b_1) que ajustan la respuesta del mismo a los datos de entrada salida obtenidos experimentalmente.

El primer paso es tarea del diseñador, mientras que el segundo puede realizarse mediante el uso de herramientas software como el Toolbox de Identificación.

1. Conociendo la función de transferencia teórica del sistema a identificar, determine cuál puede ser la estructura del modelo ARX que mejor se adapte a los datos de entrada-salida obtenidos.
2. Utilice la función ARX para obtener los parámetros del modelo con la estructura escogida en el apartado anterior.

3. El modelo obtenido anteriormente pertenece al dominio discreto, pero puede convertirse al dominio continuo mediante la función D2CM. Obtenga la función de transferencia de la planta completa en el dominio continuo.
4. Una vez traducido al dominio continuo, compare los resultados con los obtenidos mediante el estudio de la respuesta transitoria, tanto para la planta completa como para el motor.

7.6. Validación del modelo

Una vez obtenido el modelo, debe decidirse si éste se ajusta correctamente al comportamiento de la planta. Para ello pueden seguirse diversos métodos.

1. Utilice un nuevo conjunto de datos de entrada tanto para excitar a la planta real como para simular el modelo mediante la función IDSIM. Pruebe varias estructuras y decida cuál de ellas se ajusta mejor a la planta.
2. Realice un análisis de los residuos de los modelos anteriores para determinar la validez del número de parámetros escogidos. Realice también el estudio mediante la representación de polos y ceros de los diferentes modelos y de las desviaciones estándar de sus parámetros.
3. Indique cuáles son los valores definitivos de los parámetros A , τ y L obtenidos del proceso completo de identificación.

APÉNDICE: Identificación de sistemas con MATLAB

A.1. Introducción al Toolbox de Identificación

Matlab dispone de una amplia colección de funciones aplicables al campo de la identificación de sistemas, agrupadas en el *System Identification Toolbox*, que constituyen una herramienta de gran utilidad para la identificación y modelado de sistemas dinámicos. Estas funciones incluyen diferentes algoritmos para el ajuste de parámetros en todo tipo de modelos lineales. Así mismo, permiten examinar las propiedades de los modelos obtenidos, e incluso realizar un preprocesamiento de los datos utilizados para la identificación, en caso de que sea necesario.

Las versiones del Toolbox de Identificación a partir de la 4.0 permiten trabajar en dos modos distintos:

1. *Modo comando*. En este modo, se trabaja directamente desde la ventana de comandos de Matlab, realizando llamadas a las diferentes funciones que componen el Toolbox de Identificación y que se describen en este apéndice.
2. *Mediante Interfaz de Usuario*. A diferencia de otros toolboxes, el Toolbox de Identificación proporciona un interfaz gráfico que facilita el trabajo con las funciones anteriores, realizando llamadas a las mismas de forma transparente al usuario.

En este apéndice se realiza una descripción de las principales funciones del Toolbox de Identificación para trabajar en modo comando, así como una breve introducción al Interfaz de Usuario.

A.2. Los datos de entrada - salida

Como es sabido, todo proceso de identificación parte de un conjunto de datos de entrada - salida obtenidos de forma experimental a partir del sistema físico que se pretende modelar. Generalmente, sólo se utiliza una parte de esos datos para realizar la identificación del modelo, y los restantes se reservan para realizar la validación del mismo.

Los datos anteriores suelen encontrarse almacenados en ficheros ascii, que pueden ser cargados en el Workspace de Matlab mediante la función *load* para el posterior trabajo con los mismos.

Todas las funciones del Toolbox de Identificación que requieren como parámetro los datos de entrada-salida, deben recibir dichos datos en una matriz de dos columnas y N filas, siendo N el número de datos (muestras) de entrada-salida registrados. En la primera columna deben

aparecer las salidas en sucesivos periodos de muestreo y en la segunda las correspondientes entradas.

Mediante la función *idplot* se puede realizar una representación de los datos de entrada - salida, muy útil para observar posibles deficiencias en los mismos que pueden dificultar el proceso de identificación, haciendo necesario un tratamiento previo de los mismos.

Ejemplo A.1.

El Toolbox de Identificación proporciona un fichero (dryer2) con datos de entrada-salida correspondientes a un secador de mano. A continuación se muestran los comandos y funciones necesarios para introducir dichos datos en el Workspace, seleccionar la mitad de ellos para identificar y la otra mitad para validar, y realizar una representación gráfica de los datos seleccionados para identificar.

```
>> load dryer2
>> who
Your variables are:
u2    y2
>> tam=length(u2);    %tamaño de los vectores de datos u2 e y2 (deben ser iguales)
>> datos_ident=[y2(1:tam/2) u2(1:tam/2)];%selección de primera mitad para identificar
>> datos_val=[y2(tam/2+1: tam) u2(tam/2+1: tam)]; % y segunda mitad para validar
>> idplot(datos_ident); % representación gráfica de los datos reservados para identificar
```

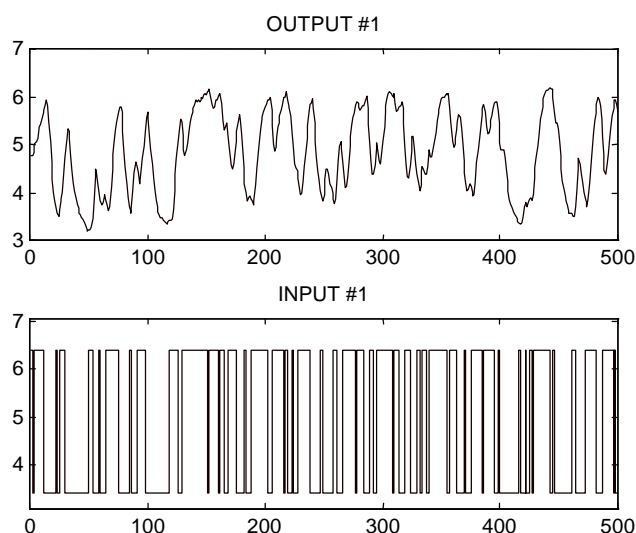


Figura A.1. Representación de datos de entrada-salida mediante *idplot*.

A.3. Tratamiento previo de los datos

El Toolbox de Identificación proporciona un conjunto de funciones que permiten realizar diferentes tratamientos de los datos de entrada - salida, para hacerlos lo más adecuados posible al proceso de identificación. Entre ellas existen funciones de filtrado, eliminación de valores medios, etc.

Generación y manipulación de datos	
<i>dtrend</i>	Eliminación de niveles de continua de un conjunto de datos.
<i>idfilt</i>	Filtrado de datos mediante filtros Butterworth.
<i>idinput</i>	Generación de señales de entrada para identificación.
<i>Idresamp</i>	Cambio del periodo de muestreo (diezmado o interpolación de los datos).

Tabla A.1. Funciones para la generación y manipulación de datos de entrada-salida

Ejemplo A.2

Compruebe el funcionamiento de la función *dtrend* aplicándola a los datos del ejemplo anterior, y realizando una nueva representación de los mismos. Téngase en cuenta que, después de utilizar *dtrend*, se mantiene la información de la dinámica del sistema, pero no de su comportamiento estático.

```
>> datos_ident=dtrend(datos_ident);
>> idplot(datos_ident);
```

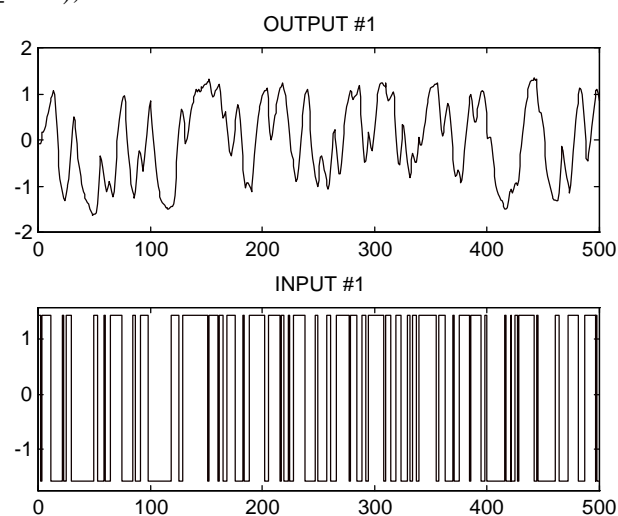


Figura A.2. Representación de datos tras la ejecución de *dtrend*.

C.4. Identificación no paramétrica

A partir de los datos de entrada-salida es posible obtener representaciones no paramétricas de la planta, tales como su respuesta en frecuencia (análisis espectral), análisis de correlación, etc. Las funciones del Toolbox de Identificación relacionadas con la identificación no paramétrica de sistemas se muestran en la tabla A.2.

Identificación no paramétrica	
<i>covf</i>	Estimación de la función de covarianza de una matriz de datos.
<i>cra</i>	Análisis de correlación para obtención de la respuesta al impulso.
<i>etfe</i>	Estimación de función de transferencia empírica y periodograma.
<i>spa</i>	Análisis espectral.
Presentación de resultados	
<i>bodeplot</i>	Diagrama de Bode de una función de transferencia o espectro.
<i>ffplot</i>	Representación de funciones frecuenciales.
<i>nyqplot</i>	Diagrama de Nyquist de una función de transferencia.
<i>zpplot</i>	Representación de ceros y polos.

Tabla A.2. Funciones para la identificación no paramétrica y representaciones gráficas.

Ejemplo A.3

Obtenga la respuesta en frecuencia del secador de mano de los ejemplos anteriores, y represente sus diagramas de Bode y de Nyquist.

```
>> g=spa(datos_ident);
>> bodeplot(g);
>> nyqplot(g);
```

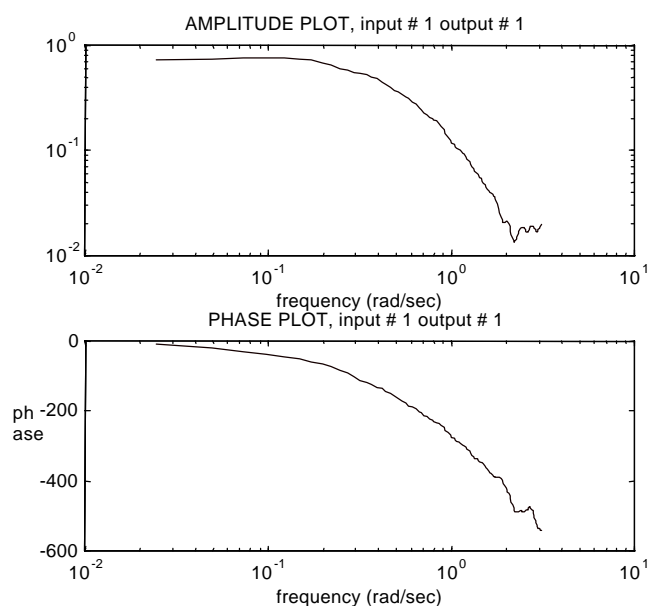


Figura A.3. Diagrama de Bode obtenido

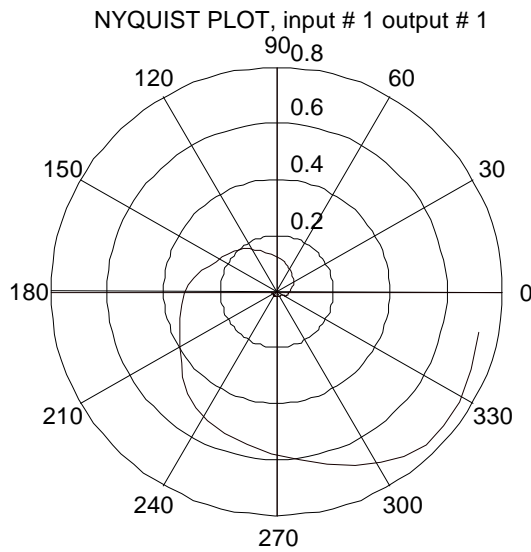


Figura A.4. Diagrama de Nyquist obtenido.

A.5. Identificación paramétrica

Se incluyen a continuación las principales funciones relacionadas con la identificación paramétrica proporcionadas por el Toolbox de Identificación.

A.5.1. Tipos de modelos paramétricos

El Toolbox de Identificación contempla una gran variedad de modelos paramétricos para sistemas lineales. Todos ellos se ajustan a la siguiente estructura general:

$$A(q^{-1}) \cdot y(t) = \frac{B(q^{-1})}{F(q^{-1})} \cdot u(t) + \frac{C(q^{-1})}{D(q^{-1})} \cdot e(t)$$

donde $u(t)$, $y(t)$ y $e(t)$ son la entrada, salida y ruido del sistema respectivamente, y A , B , C , D y F son polinomios función del operador desplazamiento (q^{-1}).

Escoger una estructura significa escoger los órdenes de todos los polinomios que intervienen. En muchas ocasiones esta elección lleva a simplificaciones típicas de la estructura general anterior (conocida como PEM), entre las que hay que destacar:

ARX: $A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + e(t)$

ARMAX: $A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) + C(q^{-1}) \cdot e(t)$

OE (Output Error): $y(t) = [B(q^{-1})/F(q^{-1})] \cdot u(t) + e(t)$

BJ (Box Jenkins): $y(t) = [B(q^{-1})/F(q^{-1})] \cdot u(t) + [C(q^{-1})/D(q^{-1})] \cdot e(t)$

Ejemplo A.4

Supóngase un sistema dinámico cuyo comportamiento puede aproximarse por la siguiente ecuación en diferencias:

$$y(t) - 1.5y(t-1) + 0.7y(t-2) = 2.5u(t-2) + 0.9u(t-3)$$

La ecuación anterior corresponde a un modelo ARX en el que:

$$\begin{aligned} A(q^{-1}) &= 1 - 1.5q^{-1} + 0.7q^{-2} \\ B(q^{-1}) &= 0 + 0q^{-1} + 2.5q^{-2} + 0.9q^{-3} \end{aligned}$$

A.5.2. Funciones para identificación de modelos paramétricos

Todas las funciones del Toolbox de Identificación para identificación paramétrica responden al siguiente formato de llamada:

```
>> th = funcion ([y u], ths)
```

donde y y u son vectores columna que contienen las muestras de salida y entrada respectivamente, ths es un vector con información sobre la estructura escogida y th es el modelo estimado en formato codificado (*formato theta*). Se utilizará una función u otra dependiendo del tipo de modelo escogido (ARX, ARMAX, OE, BJ) o el método de ajuste de los parámetros (mínimos cuadrados o variables instrumentales).

El formato del vector ths es el siguiente:

$$ths = [na \quad nb \quad nc \quad nd \quad nf \quad nk]$$

siendo na , nb , nc , nd y nf el número de coeficientes de los polinomios A , B , C , D y F de la estructura escogida, y nk el número de retardos entre la entrada y la salida. En caso de que el modelo escogido no tenga alguno/s de los polinomios anteriores, se suprimen del vector anterior el/los componente/s correspondientes a dicho/s polinomio/s.

Todas las funciones devuelven los parámetros del modelo en formato codificado. Para presentar dichos parámetros en pantalla puede utilizarse la función:

```
>> present(th)
```

que muestra por cada polinomio del modelo un vector que contiene los coeficientes del mismo comenzando por el término independiente a la izquierda, y continuando con las potencias crecientes del operador retardo q^{-1} .

La siguiente tabla muestra las distintas funciones de identificación paramétrica disponibles en el Toolbox de Identificación:

Identificación paramétrica	
<i>ar</i>	Estimación de un modelo AR usando mínimos cuadrados recursivos.
<i>armax</i>	Estimación de un modelo ARMAX usando mínimos cuadrados recursivos.
<i>arx</i>	Estimación de un modelo ARX usando mínimos cuadrados recursivos.
<i>bj</i>	Estimación de un modelo Box-Jenkins usando mínimos cuadrados recursivos.
<i>oe</i>	Estimación de un modelo Output Error usando mínimos cuadrados recursivos.
<i>pem</i>	Estimación de un modelo lineal genérico usando mínimos cuadrados recursivos.
<i>ivar</i>	Estimación de un modelo AR usando variables instrumentales.
<i>ivx</i>	Estimación de un modelo ARX usando variables instrumentales.
<i>iv4</i>	Estimación de un modelo ARX usando variables instrumentales de 4 etapas.
Presentación de resultados	
<i>present</i>	Presentación de modelos paramétricos en formato theta.

Tabla A.3. Funciones para la identificación paramétrica y presentación de modelos paramétricos.

Ejemplo A.5

Para el ejemplo del secador de mano, obtenga un modelo que se ajuste a la siguiente estructura:

$$y(t) + a_1 y(t - T) + a_2 y(t - 2T) = b_1 u(t - 3T) + b_2 u(t - 4T)$$

La ecuación anterior corresponde a un modelo ARX con dos coeficientes para el polinomio A, dos para el polinomio B y tres retardos entre la entrada y la salida. La función *arx* permite obtener los coeficientes del modelo por el método de mínimos cuadrados recursivos, para que éste se ajuste a los datos de entrada salida contenidos en la matriz *datos_ident*:

```
>> th=arx(datos_ident,[2 2 3]);
```

```
>> present(th)
```

This matrix was created by the command ARX on 9/21 1998 at 19:27

Loss fcn: 0.001687 Akaike's FPE: 0.001714 Sampling interval 1

The polynomial coefficients and their standard deviations are

B =

*0 0 0 0.0652 0.0452 /*Coeficientes del polinomio B*/*

*0 0 0 0.0016 0.0025 /*Desviaciones de los coeficientes del polinomio B*/*

A =

*1.0000 -1.2765 0.3967 /*Coeficientes del polinomio A*/*

*0 0.0159 0.0146 /*Desviaciones de los coeficientes del polinomio A*/*

Por tanto, los polinomios $A(q^{-1})$ y $B(q^{-1})$ resultantes de la identificación son los siguientes:

$$A(q^{-1}) = 1 - 1.2765q^{-1} + 0.3967q^{-2}$$

$$B(q^{-1}) = 0.0652q^{-3} + 0.0452q^{-4}$$

y el modelo paramétrico (ecuación en diferencias) del sistema discreto con periodo de muestreo igual al utilizado para el registro de datos es el siguiente:

$$y(t) - 1.2765y(t - T) + 0.3967y(t - 2T) = 0.0652u(t - 3T) + 0.0452u(t - 4T)$$

A.5.3. Elección de la estructura óptima

La elección de la estructura del modelo es una de las decisiones más importantes y difíciles que debe tomar el diseñador. Si no se tiene ningún conocimiento del sistema que facilite dicha elección, puede recurrirse a varias funciones del Toolbox de Identificación que permiten calcular las funciones de pérdidas de muchas estructuras, y escoger de entre ellas aquella cuyo resultado sea óptimo. Estas funciones se muestran en la siguiente tabla:

Selección de la estructura del modelo	
<i>arxstruc</i>	Cálculo de las funciones de pérdidas de un conjunto de estructuras ARX.
<i>ivstruc</i>	Cálculo de las funciones de pérdidas de un conjunto de estructuras OE.
<i>selstruc</i>	Selección de la estructura con menor función de pérdidas.
<i>struc</i>	Generación de un conjunto de estructuras.

Tabla A.4. Funciones para la selección de la estructura óptima de un modelo.

Ejemplo A.6

Para el ejemplo del secador de mano, obtenga la estructura que mejor ajusta el comportamiento del sistema a un modelo ARX

La función *struc* devuelve una matriz con todas las estructuras resultantes de combinar los órdenes *na*, *nb* y *nk* pasados como parámetros de entrada. La función *arxstruc*, a la cual se le pasan la matriz de datos de identificación, la de datos de validación y la de estructuras generada anteriormente, devuelve las funciones de pérdidas de todas las estructuras. Por último, la función *selstruc* selecciona de todas las estructuras aquella que presenta una menor función de pérdidas.

```
>> nn=struc([1:10],[1:10],[1:10]);
>> v=arxstruc(datos_ident,datos_val,nn);
>> nn=selstruc(v)
nn = 10 5 2
```

Por tanto, la estructura ARX óptima para los datos del secador de mano es aquella que tiene diez coeficientes para $A(q^{-1})$, 5 para $B(q^{-1})$ y dos retardos entre la entrada y la salida.

A.6. Conversión entre modelos

El Toolbox de Identificación proporciona varias funciones que permiten traducir los modelos identificados de un formato a otro, incluyendo el paso de modelos paramétricos a no paramétricos y viceversa. Dichas funciones se muestran en la siguiente tabla:

Conversión entre modelos	
<i>idmodred</i>	Reducción del orden del modelo.
<i>thc2thd</i>	Transformación del dominio continuo al discreto.
<i>thd2thc</i>	Transformación del dominio discreto al continuo.
<i>th2arx</i>	Transformación del formato theta a parámetros ARX.
<i>th2ff</i>	Transformación del formato theta a respuesta en frecuencia o espectro.
<i>th2par</i>	Transformación del formato theta a parámetros y varianzas.
<i>th2poly</i>	Transformación del formato theta a polinomios
<i>th2ss</i>	Transformación del formato theta a modelo en variables de estado.
<i>th2tf</i>	Transformación del formato theta a función de transferencia.
<i>th2zp</i>	Transformación del formato theta a ceros, polos y ganancia estática.
Manipulación de modelos	
<i>sett</i>	Inclusión del periodo de muestreo en un modelo.

Tabla A.5. Funciones para conversión y manipulación de modelos.

Ejemplo A.7

Convierta el modelo del secador de mano obtenido en el ejemplo A.5 a función de transferencia en el dominio continuo.

Puesto que la conversión del dominio discreto al continuo depende del periodo de muestreo utilizado, el primer paso es incluir en el formato *theta* del modelo obtenido en el ejemplo A.5 la información del periodo de muestreo (supóngase de 1 ms). A continuación, mediante las funciones *thd2thc* y *th2tf* se transforma el modelo del dominio discreto al continuo primero, y después del formato *theta* al de función de transferencia (compuesto por numerador y denominador). Por último, la función de transferencia puede presentarse en pantalla mediante la función *printsys*.

```
>> th=sett(th,0.001);
>> present(th)
This matrix was created by the command ARX on 9/22 1998 at 10:37
Loss fcn: 0.001687 Akaike`s FPE: 0.001714 Sampling interval 0.001
The polynomial coefficients and their standard deviations are

B =
    0    0    0  0.0652  0.0452
    0    0    0  0.0016  0.0025
A =
    1.0000 -1.2765  0.3967
    0  0.0159  0.0146

>> thc=thd2thc(th);
>> [numc,denc]=th2tf(thc);
>> printsys(numc,denc,'s')
num/den =
    2.481 s + 1.718e+005
-----
    s^2 + 924.5 s + 1.87e+005
```

A.7. Validación de modelos

Los métodos de validación proporcionados por el Toolbox de Identificación se basan en la simulación del modelo obtenido, y la comparación entre su salida y la respuesta real del sistema. Las funciones disponibles para ello se muestran en la siguiente tabla:

Validación de modelos	
<i>Compare</i>	Comparación de la salida simulada con la salida real del sistema.
<i>Idsim</i>	Simulación de un modelo.
<i>Pe</i>	Cálculo de errores de predicción de un modelo.
<i>Predict</i>	Predicción de salidas futuras de un modelo.
<i>Resid</i>	Cálculo de los residuos de un modelo.

Tabla A.6. Funciones para validación de modelos.

Ejemplo A.8

Compare la respuesta real del sistema con la salida de simulación del modelo usando para ello los datos de entrada-salida reservados para validación del modelo. Calcule también los residuos del modelo.

```
>> compare(datos_valid,th);
```

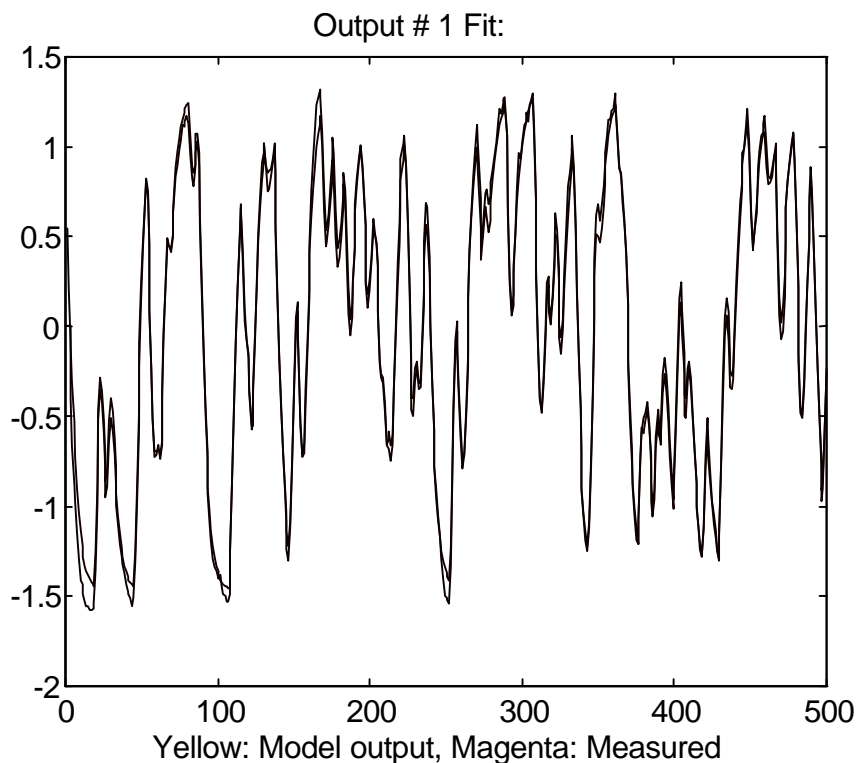


Figura A.5. Comparación de las salidas real y simulada.

```
>> resid(datos_valid,th);
```

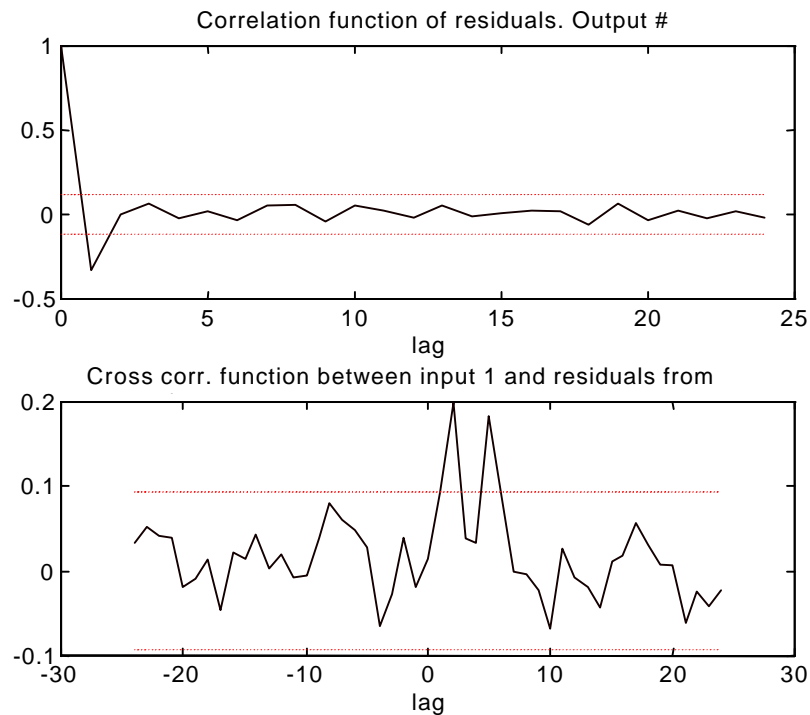



Figura A.6. Representación de los residuos del modelo.

A.8. Identificación recursiva

En la tabla A.7 se muestran todas las funciones para identificación recursiva proporcionadas por el Toolbox de Identificación:

Identificación Recursiva	
<i>Rarmax</i>	Estimación recursiva de modelos ARMAX.
<i>Rarx</i>	Estimación recursiva de modelos ARX.
<i>Rbj</i>	Estimación recursiva de modelos Box Jenkins (BJ)
<i>Roe</i>	Estimación recursiva de modelos Output Error (OE - filtros IIR)
<i>Rpem</i>	Estimación recursiva de modelos genéricos mediante errores de predicción.
<i>rplr</i>	Estimación recursiva de modelos genéricos mediante regresión pseudo-lineal.

Tabla A.7. Funciones para identificación recursiva.

Todas las funciones anteriores se ajustan al siguiente formato:

`[thm, yh] = funcion (datos, nn, adm, adg);`

donde *datos* contiene los datos de entrada - salida, *nn* especifica la estructura del modelo y los parámetros *adm* y *adg* hacen referencia al método de adaptación y sus parámetros típicos respectivamente (dependen de la función utilizada).

A.9. Introducción al interfaz gráfico de usuario (GUI)

Las últimas versiones del Toolbox de Identificación (a partir de la 4.00) proporcionan un Interfaz Gráfico de Usuario que permite manejar todas las funciones anteriores de forma prácticamente transparente al usuario, sin necesidad de llamarlas desde la línea de comandos. Para arrancar el interfaz de usuario (GUI), basta con teclear en la línea de comandos de Matlab:

```
>> ident
```

apareciendo la ventana mostrada en la figura A.7.

A.9.1. Estructura de la ventana principal

La ventana principal se divide en las siguientes partes:

a) *Los tableros de datos y de modelos.*

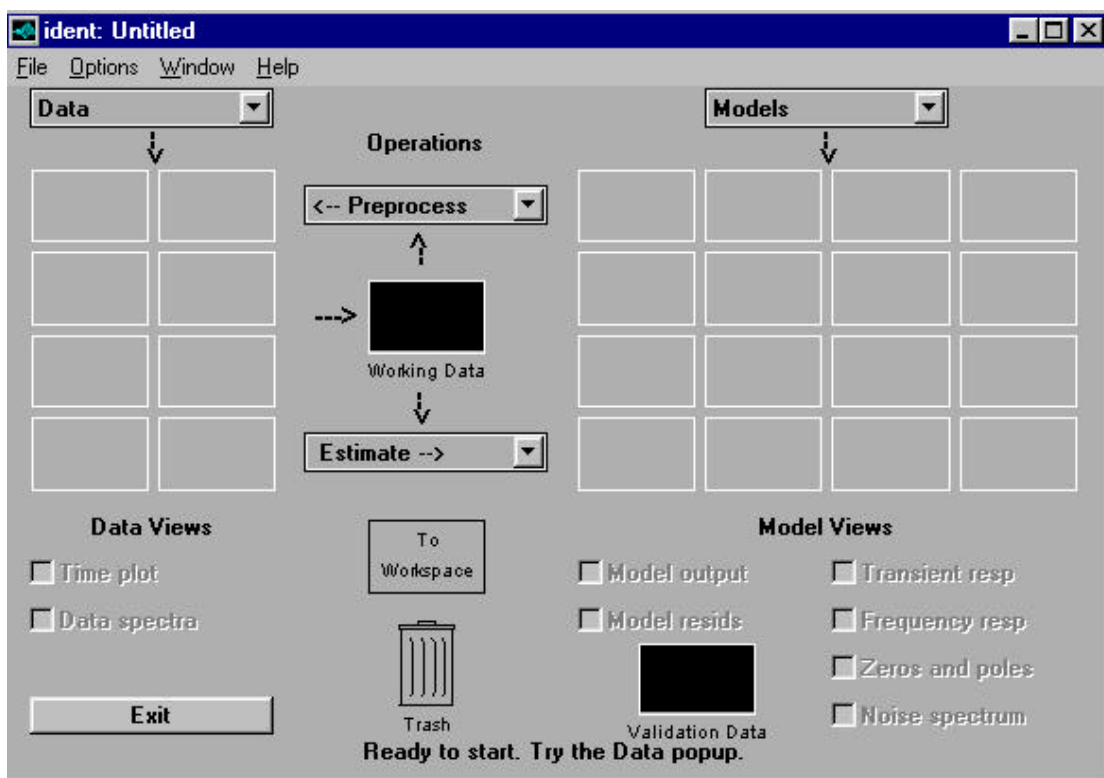


Figura A.7. Aspecto del Interfaz Gráfico de Usuario (GUI).

Como se observa en la figura, la ventana dispone de dos zonas con varios recuadros cada una:

- El tablero de datos está situado en la zona izquierda de la pantalla, y permite incluir en cada uno de los recuadros un conjunto distinto de datos de entrada-salida, representados por un icono.
- El tablero de modelos está en la zona derecha de la pantalla, y puede contener en cada uno de sus recuadros diferentes modelos obtenidos a partir de la identificación realizada con datos del tablero de datos. Cada modelo quedará representado también por un icono distinto.

Los datos del tablero de datos pueden provenir de las siguientes fuentes:

1. De otras sesiones anteriores con el GUI.
2. Del Workspace de Matlab.
3. Del tratamiento de otro conjunto de datos contenido en el tablero de datos.

Por su parte, los modelos pueden provenir de las siguientes fuentes:

1. De sesiones anteriores con el GUI.
2. Del Workspace de Matlab.
3. De la identificación a partir de los datos contenidos en el tablero de datos.

b) *Los datos de trabajo*

Todos los procesos realizados mediante el GUI actúan sobre los llamados “Datos de Trabajo”, contenidos en el recuadro central de la ventana (“*working data*”). Para modificar los datos de trabajo basta con arrastrar con el ratón el icono con los nuevos datos de trabajo desde el tablero de datos hasta el recuadro “*working data*”.

c) *Los datos de validación*

Del mismo modo, todos los procesos del GUI que necesiten datos para validación los tomarán del recuadro (“*validation data*”) situado debajo del tablero de modelos.

d) *Las representaciones (“Views”)*

Tanto los datos de entrada-salida como los modelos pueden representarse en pantalla de diversas formas.

Para representar en pantalla un conjunto de datos del tablero de datos, en primer lugar hay que hacer “click” con el ratón sobre su icono, quedando éste resaltado mediante una línea más gruesa. Pueden seleccionarse varios conjuntos de datos simultáneamente. Para desactivar un conjunto de datos, se vuelve a hacer “click” con el ratón sobre su icono. A continuación se selecciona en el menú de Data Views el tipo de representación que se desea: representación temporal de las señales (“Time plot”) o del espectro de las mismas (“Data spectra”).

Con los modelos se procede de igual manera, seleccionando con el ratón aquéllos que se quieren representar, y escogiendo el tipo de representación entre salida del modelo (“Model output”), residuos del modelo (“Model resids”), respuesta transitoria (“Transient resp”), respuesta frecuencial (“Frequency resp”), ceros y polos (“Zeros and poles”) y espectro del ruido (“Noise spectrum”).

e) Variables del Workspace

Los conjuntos de datos o los modelos creados mediante el interfaz gráfico generalmente no están visibles desde el Workspace. Sin embargo, esta información puede ser exportada en cualquier momento al Workspace sin más que arrastrar con el ratón el icono de los datos o el modelo correspondiente. El nombre de la matriz con la información del modelo o de los datos coincidirá con el del icono dentro del interfaz gráfico.

C.9.2. Flujo de trabajo

El trabajo con el GUI suele realizarse en los siguientes pasos:

- Se introducen de los datos de entrada-salida en el tablero de datos.
- Se examinan dichos datos mediante su representación temporal o espectral del menú “data views”.
- Se realiza el preprocesamiento de los datos, como eliminación de niveles de continua, filtrado, o división del conjunto de datos en dos partes: una para identificar y que se colocará en el recuadro de ‘working data’ y otra para validar en el recuadro de ‘validation data’. Todas las opciones de preprocesamiento se encuentran en el menú desplegable “Preprocess”.
- A continuación se realiza la estimación de uno o varios modelos, mediante el menú desplegable de “Estimate”. Los modelos obtenidos se incluirán automáticamente en el tablero de modelos.
- Por último se procede a analizar las propiedades de los modelos obtenidos mediante diversas representaciones de los mismos, escogidas en el menú de “Model Views”.

A.10. Ejercicios propuestos

A.10.1. Utilizando como punto de partida el fichero “dryer2” proporcionado por Matlab, con datos de entrada-salida correspondientes a un secador de mano obtenidos con un periodo de muestreo de 0.08 segundos, se pide:

1. Cargue los datos en el Workspace, y separe los trescientos primeros datos para identificar y el resto para validar. Realice la representación gráfica de ambos conjuntos de datos.
2. Elimine los niveles de continua del conjunto de datos para identificación y realice una nueva representación gráfica de los mismos.
3. Realice una representación de la respuesta al impulso del sistema mediante un análisis de la correlación (utilice para ello la función *cra*). Estime de forma aproximada a partir del resultado cuáles son las constantes de tiempo del sistema y el retardo del mismo.
4. Obtenga la respuesta al escalón del sistema mediante una integración de la respuesta al impulso (se propone utilizar para ello la función *cumsum*).
5. Obtenga un modelo ARX con dos polos, un cero y tres retardos. Escriba la ecuación en diferencias correspondiente al modelo obtenido.
6. Para validar el modelo anterior, realice una simulación del mismo con los datos reservados para la validación, y compare su respuesta con la respuesta real del sistema.
7. Obtenga los polos y ceros del modelo anterior.
8. Obtenga la función de transferencia discreta del modelo anterior.
9. Compare la función de transferencia anterior (obtenida de forma indirecta a partir de una identificación paramétrica de un modelo ARX) con la obtenida de la identificación no paramétrica directa mediante un análisis espectral.

A.10.2. El objetivo de este ejercicio es comparar diferentes métodos de identificación. Para ello, se obtendrá un conjunto de datos de entrada-salida mediante la simulación de un modelo ARX cuyos polinomios A y B son los siguientes:

$$B = [0 \ 1 \ 0.5];$$

$$A = [1 \ -1.5 \ 0.7];$$

1. Obtenga el formato theta del sistema anterior, utilizando para ello la función *poly2th*, utilizando el tercer parámetro de la misma para caracterizar la presencia de perturbaciones en el sistema.

2. Genere mediante la función *rand* un vector de entradas aleatorias *u*, otro de ruido aleatorio *e*, y simule la respuesta del sistema en dichas condiciones utilizando para ello la función *idsim*. Realice una representación de los datos de entrada-salida obtenidos, que se utilizarán en adelante para obtener varios modelos y comparar sus características.
3. Obtenga para los datos de entrada-salida anteriores un modelo ARX (por el método de mínimos cuadrados) con dos polos, un cero y un retardo entre la entrada y la salida.
4. Obtenga otro modelo por el método de variables instrumentales, también con dos polos, un cero y un retardo entre la entrada y la salida.
5. Obtenga las funciones de transferencia de los dos modelos obtenidos en los apartados 3 y 4. Represente sobre una misma gráfica los diagramas de Bode correspondientes a dichas funciones de transferencia.
6. Calcule y represente los residuos del modelo obtenido por el método de variables instrumentales.
7. Obtenga ahora un modelo ARMAX y otro BJ ambos de segundo orden (dos coeficientes para todos los polinomios que intervienen) con un retardo entre la entrada y la salida. Obtenga las funciones de transferencia para ambos modelos y compare los diagramas de Bode con el obtenido por el método de variables instrumentales.
8. Represente gráficamente los residuos de los modelos ARMAX y BJ.
9. Simule los modelos ARMAX y BJ utilizando para ello el vector de entrada *u* obtenido en el apartado 1. Dibuje sobre la misma gráfica la salida real y las salidas de la simulación de ambos modelos.
10. Represente los polos y ceros de los modelos ARMAX y BJ.

A.10.3. Para la realización de este ejercicio se utilizarán también los datos del fichero “dryer2” proporcionado por Matlab. Se parte de dichos datos divididos en dos mitades: la primera para identificar y la segunda para validar. A ambos conjuntos de datos se les ha eliminado la componente continua.

Se pretende trabajar con diferentes modelos, todos ellos con la siguiente estructura:

$$y(t) + a_1 \cdot y(t-1) + \dots + a_{n_a} \cdot y(t-n_a) = b_1 \cdot u(t-n_k) + \dots + b_{n_b} \cdot u(t-n_b-n_k+1)$$

1. Se desea obtener el retardo n_k . Para ello, se escogerán estructuras con $n_a=n_b=2$, y se ensayarán todos los modelos con retardos comprendidos entre 1 y 10. Obtenga las funciones de pérdidas para todos los modelos anteriores, y escoja la estructura óptima (utilice las funciones *arxstruc* y *selstruc*).

2. A continuación se buscará el orden del sistema óptimo, ensayando los 25 modelos posibles de combinar órdenes de A y B comprendidos entre 1 y 5. Utilice como retardo el obtenido en el apartado anterior.
3. Como norma general, los modelos obtenidos son tanto más exactos cuanto mayor sea su orden. Sin embargo, debe buscarse el modelo más reducido que satisface la dinámica del sistema. Realice una representación de los ceros y polos del modelo seleccionado en el apartado anterior y decida, en función de la proximidad de los mismos (tendencia a cancelarse) si se puede reducir el orden del sistema. En caso afirmativo, realice la simulación del modelo óptimo y del reducido y analice las diferencias.

NOTA: resuélvanse todos los ejercicios desde la línea de comandos de Matlab primero y a continuación.

BIBLIOGRAFÍA

- L. Ljung. “*System Identification. Theory for the user*”. Prentice Hall. 1987.
- T. Söderström y P. Stoica. “*System Identification*”. Prentice Hall. 1989.
- M.K. Masten. “*Modern Control Systems. Study Guide*”. IEEE/EAB Self-Study Course. 1995.
- L Ljung. “*The System Identification Toolbox: The Manual*”. The MathWorks Inc. 1986.
- L. Ljung y T. Glad. “*Modeling of Dynamic Systems*”. Prentice Hall, 1994.
- J. Schoukens y R. Pintelon. “*Identification of Linear Systems: a Practical Guideline to Accurate Modeling*”. Pergamon Press, 1991.