



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

INTEGRACION DE UN SISTEMA PARA LA
IDENTIFICACION DE PROCESOS LINEALES

T E S I S

QUE PARA OBTENER EL TITULO DE:

Ingeniero Mecánico Electricista
AREA ELECTRICA ELECTRONICA

P R E S E N T A :

ALFONSO PRIETO GUERRERO

DIRECTOR DE TESIS :

M. EN I. PABLO FRANCISCO LARA REYES

CODIRECTORA : DRA. CRISTINA VERDE RODARTE



MEXICO, D. F.

1993

TESIS CON
FALLA DE ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1 <i>Conceptos Generales</i>	
1.1 Procedimiento de Identificación	3
1.2 Restricciones de los Procesos a Identificar	5
1.3 Verificación del Conocimiento A Priori del Proceso	7
1.4 Señales de Excitación	7
1.5 Captura de los Datos experimentales	10
1.6 Modelos No Paramétricos	11
1.7 Modelos Paramétricos	12
1.8 Validación del Modelo Estimado	17
CAPÍTULO 2 <i>Descripción del Sistema Para Identificación</i>	
2.1 Esquema de Bloques	19
2.2 Descripción del Osciloscopio Digital	19
2.3 Descripción de la Tarjeta de Adquisición de Datos	20
2.4 Descripción del Generador de Secuencias Binarias Seudoaleatorias	20
2.5 Descripción del Paquete MATLAB	22
2.6 Rutina de Estimación	23
2.7 Procedimiento de Uso del Sistema	25
2.8 Identificación de un Proceso Térmico	27
REFERENCIAS	39
APÉNDICE A <i>Ejemplos de Aplicación</i>	40

APÉNDICE B <i>Prontuario para la Utilización del Osciloscopio y la Tarjeta de Adquisición</i>	50
APÉNDICE C <i>Datos Técnicos del Generador SBS</i>	56
APÉNDICE D <i>Listados de Programas</i>	64

INTRODUCCIÓN

En muchos casos, los sistemas físicos son demasiado complejos o no se cuenta con suficiente información sobre ellos; lo que hace impráctico, y a veces imposible, modelarlos a través de leyes físicas. Debido a lo anterior, surge una forma diferente de obtener la descripción del sistema; la cual está basada en la inferencia de las propiedades del sistema, a partir del análisis de la relación entre datos experimentales de las señales de entrada y de salida. A este procedimiento se le conoce como *Identificación de Sistemas*.

Una de las áreas de mayor aplicación del procedimiento de identificación es la ingeniería de control; por ello, se decidió integrar un sistema que permitiese simplificar y automatizar dicho proceso, para fines ya sean didácticos o de investigación en dicho campo. Los usuarios inmediatos de este equipo son la Coordinación de Automatización del Instituto de Ingeniería y el Departamento de Control de la División de Estudios de Posgrado de la Facultad de Ingeniería. El equipo está constituido por un generador de secuencias binarias pseudoaleatorias (que se diseñó y construyó como parte del trabajo), un osciloscopio digital, una tarjeta de adquisición de datos y un programa desarrollado en el paquete de asistencia matemática MATLAB. El objetivo primordial del presente trabajo, es diseñar e integrar un sistema que permita obtener un modelo paramétrico restringido a procesos lineales, invariantes con el tiempo y de una entrada-una salida.

A continuación se explica brevemente el contenido del trabajo escrito.

En el capítulo primero se explica en forma breve el procedimiento general de identificación.

En el segundo capítulo se describen los elementos que integran el sistema propuesto y el uso del mismo. Al final de este capítulo, se describe en detalle un ejemplo de identificación aplicando el sistema mencionado.

Finalmente, se presentan las conclusiones sobre el desempeño del sistema realizado, el cual fue en general satisfactorio, ya que se verificó que el sistema integra de manera armónica los elementos más importantes del procedimiento de identificación, y que la información, tanto técnica como práctica que se provee al usuario, demostró ser de gran utilidad. Por otro lado se comentan las limitaciones del sistema, las precauciones que se aconsejan seguir para su correcta aplicación y para terminar esta sección se señalan las directrices que seguirán futuros trabajos destinados a mejorar y ampliar la capacidad del mismo.

La información que complementa el trabajo se provee en los siguientes apéndices: el apéndice A contiene dos ejemplos de aplicación del sistema. Un prontuario para uso del osciloscopio digital y de la tarjeta de adquisición forma el contenido del apéndice B. Los datos técnicos referentes al generador de secuencias binarias pseudoaleatorias se incluyen en el apéndice C. Por último, en el apéndice D, se proporcionan los listados de la rutina de estimación realizada en MATLAB.

CAPÍTULO 1

CONCEPTOS GENERALES

1.1 PROCEDIMIENTO DE IDENTIFICACIÓN

En general, el procedimiento de identificación está organizado de forma iterativa como se muestra en la figura 1.1. (Ref 2, pág 576).

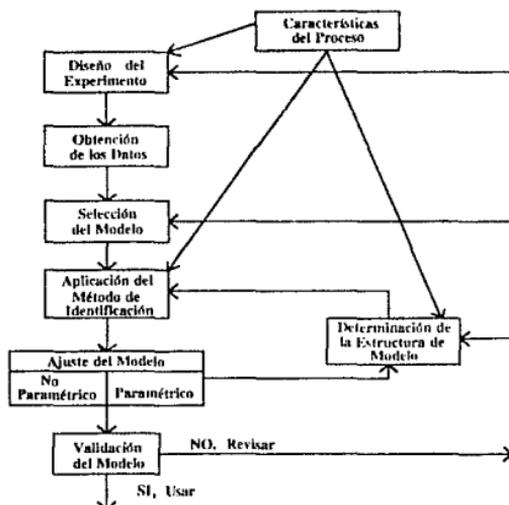


Fig 1.1 Esquema general de Identificación

A continuación, se explican brevemente los pasos principales del procedimiento.

Se observa que debe existir un conocimiento *a priori* del proceso; de tal manera que permita suponer consideraciones de características tales como, tiempo de

asentamiento, constante de tiempo dominante, retardo, linealidad, concentración de parámetros, invariancia en el tiempo, etc.

El diseño del experimento, involucra conceptos tales como tipo de entrada aplicada al proceso, periodo de muestreo, selección del tipo de identificación: en línea o fuera de línea, etc. Una buena identificación está supeditada al diseño adecuado del experimento, a través del cual se adquiere información del comportamiento del proceso (datos de entrada y salida).

Posterior a la obtención de los datos, se selecciona el tipo de modelo considerado (paramétrico o no paramétrico). En el caso paramétrico se deberá proponer una estructura para el modelo.

Una vez determinado el tipo del modelo (y su estructura, si es el caso), se elige la técnica para ajustar el modelo. Los principales métodos de identificación, y el tipo de modelo que puede obtenerse a través de ellos, se anotan en la tabla 1.1 (Ref 2. pág 576).

Métodos de Identificación	Modelo	Modelo
	No Paramétrico	Paramétrico
Técnicas de respuesta al escalón	X	X
Técnicas de respuesta en frecuencia	X	X
Análisis de Fourier y espectral	X	
Técnicas de correlación	X	
Estimación de Parámetros		X

Tabla 1.1 Principales métodos de identificación

El último paso del esquema se refiere a la validación del modelo obtenido, ésta consiste en realizar una evaluación de la capacidad del modelo para reproducir el comportamiento del proceso. Esto lleva normalmente al cálculo del error entre el proceso y el modelo, como se ilustra en la figura 1.2.

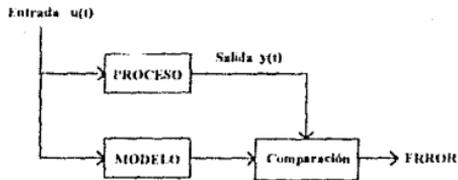


Fig 1.2 Representación esquemática del error

Como se mencionó, el procedimiento de identificación es por lo general iterativo: ya que, si el modelo no es adecuado, se debe ajustar otro, ya sea cambiando el método usado, determinando otra estructura, modificando las suposiciones sobre el proceso o incluso modificando el diseño del experimento.

Con el sistema que se propone integrar, se desea conseguir un modelo paramétrico utilizando la técnica de estimación de parámetros a través del algoritmo de mínimos cuadrados. Como ayuda en la determinación del modelo paramétrico, se hará uso del análisis de Fourier y espectral, así como de las técnicas de correlación.

1.2 RESTRICCIONES DE LOS PROCESOS A IDENTIFICAR

La aplicación del sistema propuesto, está restringida a procesos cuyos modelos son de parámetros concentrados, lineales, invariantes en el tiempo y de una entrada-una salida del tipo función de transferencia. Debido al uso de computadoras digitales, se consideran únicamente modelos del tipo discreto. En forma general, un proceso de este tipo puede ser representado de la siguiente manera (fig 1.3)

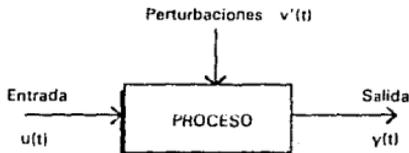


Fig 1.3 Representación esquemática de un proceso

En donde el término $v'(t)$, representa las perturbaciones que afectan al proceso, las cuales se consideran como ruido blanco y que, por lo general, no pueden ser medidas.

Cuando se tienen perturbaciones resulta complicado obtener una descripción matemática de éste. Debido a que el sistema es lineal, se puede descomponer la salida en dos términos, uno que depende de la entrada $u(t)$ y el segundo de $v'(t)$, de tal forma que el sistema puede ser descrito por (Ref 4, pp 13-20)

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t) \quad (1.1)$$

donde $g(k)$ es la respuesta al impulso del sistema y el segundo término corresponde a

$$v(t) = \sum_{k=0}^{\infty} h(k)v'(t-k) \quad (1.2)$$

donde $h(k)$ es la respuesta al impulso del ruido. $v(t)$ puede ser descrito a través del primer y segundo momentos estadísticos (i.e. media y covariancia).

Para manejar notaciones más cortas de las ecuaciones anteriores, se introduce el operador retardo (q^{-1}), que se define como

$$q^{-1}u(t) = u(t-1) \quad (1.3)$$

Aplicando este operador a las ecuaciones (1.1) y (1.2) y combinando los resultados se obtiene

$$y(t) = G(q)u(t) + H(q)v'(t) \quad (1.4)$$

donde

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k} \quad (1.5)$$

$$H(q) = \sum_{k=0}^{\infty} h(k)q^{-k} \quad (1.6)$$

Esta es la representación para los procesos lineales que se usará en el presente trabajo.

1.3 VERIFICACIÓN DEL CONOCIMIENTO A PRIORI DEL PROCESO

Para validar las hipótesis más importantes acerca de los procesos a identificar (linealidad e invariancia en el tiempo), se sugiere tomar en cuenta las siguientes recomendaciones:

VERIFICACIÓN DE LA LINEALIDAD: Una de las formas de verificar la linealidad del proceso es asegurar que su comportamiento ante diferentes amplitudes de la entrada, satisface el principio de homogeneidad. En caso de no verificarse será necesario reentregar el modelo a un rango de operación determinado, donde el comportamiento del proceso sea prácticamente homogéneo.

VERIFICACIÓN DE LA INVARIANCIA EN EL TIEMPO: Para poder asegurar que el proceso en estudio es invariante en el tiempo, se debe analizar su comportamiento bajo mismas condiciones de operación, en tiempos diferentes.

1.4 SEÑALES DE EXCITACIÓN

En el diseño del experimento de identificación, un punto importante es la elección de la señal que será aplicada al proceso. Esta señal debe cumplir con ciertas características en función del método de identificación.

Las señales de excitación más utilizadas en el procedimiento de identificación son (Ref 6, pp 96-112):

- Señal Escalón

Esta señal se define como

$$u(t) = U_0 \quad t \geq 0 \quad (1.7)$$

donde U_0 es una constante.

Esta señal es empleada cuando el proceso presenta una razón señal a ruido grande, y puede proporcionar información acerca de la dinámica del proceso, tal como el tiempo de levantamiento, la constante de tiempo dominante, sobrepaso, ganancia estática, retardo, etc.

- Suma de Senoides

La suma de funciones senoidales se obtiene de la siguiente forma

$$u(t) = \sum_{j=1}^m a_j \text{sen}(\omega_j t + \phi_j) \quad (1.8)$$

donde

a_j son las amplitudes de las senoides

ϕ_j son las fases de las senoides

ω_j son diferentes velocidades angulares, que cumplen con

$$0 \leq \omega_1 < \omega_2 < \dots < \omega_m \leq 2\pi$$

Con esta señal es posible determinar la respuesta en frecuencia del proceso y de esta forma determinar frecuencias de corte, ancho de banda, e incluso la función de transferencia.

- Secuencias Binarias Seudoaleatorias (SBS)

En general, la señal tiene las siguientes características (Ref 1, pág 531):

- La señal posee únicamente dos niveles (V_{max} , V_{min}).
- Puede variar de un nivel a otro sólo después de un intervalo mínimo de tiempo, conocido como periodo básico, representado con el símbolo (Δ).
- Esta señal es periódica, con periodo (M) dado por:

$$M = N\Delta \quad (1.9)$$

donde N es un entero y representa el número de cambios de nivel que presenta la secuencia antes de repetirse.

Un ejemplo de una señal SBS se muestra a continuación, en la figura 1.4.

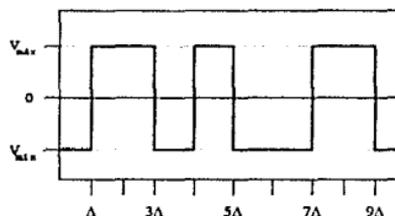


Fig 1.4 Secuencia Binaria Seudoaleatoria

Esta señal, es una de las más utilizadas en los métodos de identificación, por sus características generales parecidas al ruido blanco: presenta excitación persistente (excita todos los modos del sistema) y la función de correlación indica cierta independencia entre cada una de los datos que la constituyen. En la figura 1.5 se muestra la autocorrelación de la SBS.

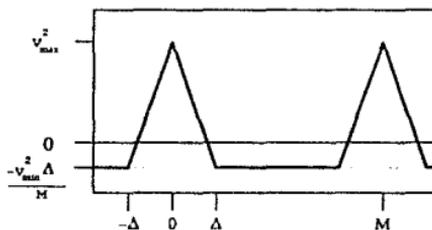


Fig 1.5 Autocorrelación de la SBS

Las SBS más utilizadas están basadas en secuencias de longitud máxima, conocidas como secuencias-m; para las cuales:

$$N = 2^n - 1 \quad (1.10)$$

donde n es un entero que determina el número de cambios de la secuencia antes de repetirse.

Para el ajuste de los parámetros, se decidió utilizar señales SBS como las entradas que son aplicadas al proceso.

1.5 CAPTURA DE LOS DATOS EXPERIMENTALES

Como la captura de las señales de entrada y salida es discreta, es necesario determinar adecuadamente el periodo de muestreo (T) de las mismas. Dicho periodo debe elegirse de acuerdo a las características dinámicas del proceso. Existen algunas reglas de dedo para dicha elección. A continuación se presentan tres de ellas:

- Si el sistema tiene una constante de tiempo (τ) dominante, es recomendable que T sea menor a ésta; una buena elección es la siguiente (Ref 4, pág 385)

$$T \cong 0.5(\tau) \quad (1.11)$$

- Si el sistema presenta sobrepaso el periodo de muestreo puede ser elegido

a) como el 10% del tiempo de asentamiento (T_a) de la respuesta al escalón del sistema (Ref 6, pág 473)

$$T \cong 0.1(T_a) \quad (1.12)$$

b) a través de la siguiente expresión (Ref 2, pág 577)

$$T \cong \frac{0.95(T_a)}{NM} \quad (1.13)$$

donde NM es un número entero entre 6 y 15 y T_a es el tiempo de asentamiento.

En cualquiera de los casos anteriores, si se tienen problemas para ajustar T, es preferible tomar tiempos más pequeños que más grandes en relación con el tiempo de referencia (τ ó T_a). Sin embargo, hay que prever que cuando se tienen periodos de

muestreo muy pequeños, se pueden presentar problemas prácticos; por ejemplo, que todos los polos del sistema se agrupen alrededor del punto $z=1$ en el plano complejo, o que se presenten problemas numéricos en la estimación de parámetros (Ref 6, pág 473).

Una vez seleccionada la frecuencia de muestreo, se elige el periodo básico de la secuencia binaria, este debe cumplir, para su correcta recuperación (de acuerdo con el teorema de Shannon) con la siguiente desigualdad

$$\Delta \geq 2(T) \quad (1.14)$$

1.6 MODELOS NO PARAMÉTRICOS

La identificación no parametrizada se realiza con el fin de tener una idea de la dinámica de los procesos por identificar. Los modelos paramétricos consisten de gráficas o valores muestrales. Los principales métodos para obtener modelos no paramétricos están basados en la obtención de la respuesta al impulso, la respuesta al escalón y la respuesta en frecuencia del proceso.

Con las técnicas de correlación, es posible estimar la respuesta al impulso del proceso a partir de la ecuación (Ref 11, pág 55)

$$R_{uy}(\tau) = g(\tau) * R_{uu}(\tau) \quad (1.15)$$

donde

$g(\tau)$ es la respuesta al impulso del sistema.

$R_{uu}(\tau)$ es la función de autocorrelación de la entrada $u(t)$.

$R_{uy}(\tau)$ es la función de correlación entre la entrada $u(t)$ y la salida $y(t)$ del sistema.

El asterisco (*) denota la convolución de las señales.

A partir de la ecuación 1.15 y siguiendo el procedimiento presentado en (Ref 11, pp 55-56) se obtiene la respuesta $g(\tau)$ para sistemas lineales.

1.7 MODELOS PARAMÉTRICOS

La obtención de modelos paramétricos, está basada en estructuras particulares que toman únicamente en cuenta la relación entrada-salida, a través de funciones de transferencia. Las estructuras más utilizadas son las siguientes (Ref 4, pp 69-77):

- ESTRUCTURA ARX

Las siglas ARX significan Autorregresión con Entrada Externa (del inglés *AutoRegressive with eXternal input*). Esta estructura es la más simple y se describe a través de la ecuación en diferencias lineal de orden na (con $na \geq nb$)

$$y(t) + a_1y(t-T) + \dots + a_ny(t-na(T)) = b_1u(t-nk(T)) + b_2u(t-(nk+1)T) + \dots + b_nu(t-(nk+nb-1)T) + v(t) \quad (1.16)$$

donde los parámetros a estimar son $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n$. T corresponde al tiempo de muestreo de las señales y nk representa el número de periodos de muestreo que constituye el retardo que presenta la salida con respecto a la entrada y que, de acuerdo con la ecuación 1.16, corresponde al orden relativo del modelo (diferencia entre polos y ceros).

Utilizando el operador retardo (cc 1.3), se obtiene

$$A(q)y(t) = B(q)u(t-(nk)T) + v(t) \quad (1.17)$$

donde

$$A(q) = 1 + a_1q^{-1} + \dots + a_nq^{-na} \quad (1.18)$$

$$B(q) = b_1 + b_2q^{-1} + \dots + b_nq^{-(n-1)} \quad (1.19)$$

na y nb representan los ordenes de los polinomios $A(q)$ y $B(q)$ respectivamente.

De acuerdo con la representación de la ecuación 1.4, la estructura ARX puede ser descrita a través de las funciones racionales

$$G(q) = q^{(nb)} \frac{B(q)}{A(q)} \quad (1.20)$$

$$H(q) = \frac{1}{A(q)} \quad (1.21)$$

En la función de transferencia $G(q)$ puede observarse que el número de polos está determinado por el orden del polinomio $A(q)$, dado por na , y el número de ceros por el orden del polinomio $B(q)$, simbolizado por nb .

De acuerdo a lo anterior, el esquema de la estructura ARX se muestra en la siguiente figura

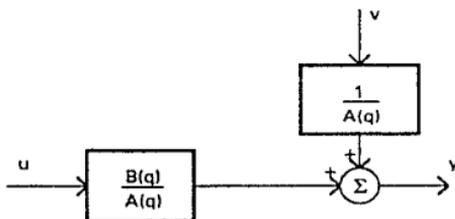


Fig 1.6 Diagrama de bloques de la estructura ARX

- ESTRUCTURA ARMAX

La estructura ARMAX, Autorregresión con Promedio Móvil y Entrada Externa (siglas del inglés *AutoRegressive Moving Average with eXternal input*) presenta básicamente la misma forma que la estructura ARX, agregándole únicamente un promedio móvil que le da flexibilidad para describir el error $v(t)$. La estructura tiene la forma

$$A(q)y(t) = B(q)u(t-nk(T)) + C(q)v(t) \quad (1.22)$$

donde los polinomios $A(q)$ y $B(q)$ están definidos por (1.18) y (1.19) y $C(q)$ se define como

$$C(q) = 1 + C_1q^{-T} + \dots + C_{nc}q^{-(nc)T} \quad (1.23)$$

De aquí

$$G(q) = q^{-(nk)T} \frac{B(q)}{\Lambda(q)} \quad (1.24)$$

$$H(q) = \frac{C(q)}{\Lambda(q)} \quad (1.25)$$

A continuación se muestra el diagrama esquemático de la estructura ARMAX

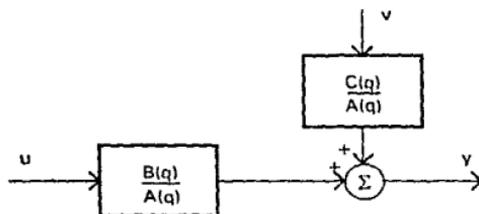


Fig 1.7 Diagrama de bloques de la estructura ARMAX

- ESTRUCTURA OE

La estructura OE, Error de Salida (siglas del inglés *Output Error*), a diferencia de las dos estructuras anteriores, supone el error $v(t)$ de manera aditiva en la salida del sistema. Por tanto, el modelo se reduce a

$$y(t) = \frac{B(q)}{F(q)} u(t-nk)T + v(t) \quad (1.26)$$

donde $B(q)$ se define como en (1.19) y $F(q)$ por

$$F(q) = 1 + f_1 q^{-1} + \dots + f_m q^{-m}T \quad (1.27)$$

El diagrama esquemático de esta estructura se muestra a continuación

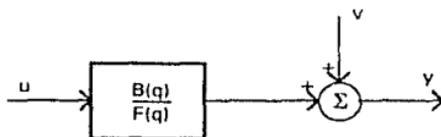


Fig 1.8 Diagrama de bloques de la estructura OE

- ESTRUCTURA BJ

La estructura de modelo Box-Jenkins (BJ), basada en la estructura OE, describe el error como un modelo ARMA, de tal manera que

$$y(t) = \frac{B(q)}{F(q)} u(t-nk)T + \frac{C(q)}{D(q)} v(t) \quad (1.28)$$

donde

$$D(q) = 1 + d_1 q^{-T} + \dots + d_{nd} q^{-(nd)T} \quad (1.29)$$

$$C(q) = 1 + c_1 q^{-T} + \dots + c_{nc} q^{-(nc)T} \quad (1.30)$$

de lo anterior

$$G(q) = q^{-(nk)T} \frac{B(q)}{F(q)} \quad (1.31)$$

$$H(q) = \frac{C(q)}{D(q)} \quad (1.32)$$

Se observa en las ecuaciones (1.31) y (1.32) que las funciones de transferencia $G(q)$ y $H(q)$ son parametrizadas de forma independiente. La forma esquemática de este modelo es

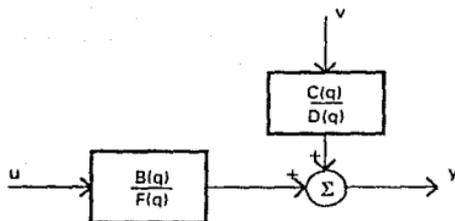


Fig 1.9 Diagrama de bloques de la estructura BJ

- ESTRUCTURA IV4

La estructura de modelo de Variable Instrumental (IV4), siglas del inglés *Instrumental Variable - 4 stage*; es en principio una estructura ARX. Cuando se utiliza la estructura ARX y se estiman los parámetros $(a_1, a_2, \dots, a_{na})$ y $(b_1, b_2, \dots, b_{nb})$ utilizando el algoritmo de mínimos cuadrados y, además, se consideran los errores correlacionados (método de variable instrumental), se denota a la estructura como IV4.

1.7.1 MÉTODO DE LOS MÍNIMOS CUADRADOS PARA LA ESTIMACIÓN DE PARÁMETROS.

En el método de los mínimos cuadrados propuesto por Karl Friedrich Gauss, se propone que los valores apropiados para los parámetros desconocidos, son los más probables. Gauss definió el valor más probable del parámetro desconocido, como aquel para el cual la suma de los cuadrados de las diferencias entre las observaciones y los cálculos es mínima. A continuación se describe la aplicación del método de Gauss para la estimación de los parámetros de las estructuras consideradas.

A partir de la estructura ARX (1.16) se puede obtener (Ref 8, pp 535-538)

$$y(t) = \phi^t(t)\theta + v(t) \quad (1.33)$$

donde

$$\phi^t(t) = [-y(t-1), -y(t-2), \dots, -y(t-na), u(t-nk), u(t-nk-1), \dots, u(t-nk-nb)]^T$$

$$\theta = [a_1, a_2, \dots, a_{na}, b_1, b_2, \dots, b_{nb}]$$

Ahora bien, definiendo el error $\epsilon(t)$ como

$$\epsilon(t) = y_p(t) - \phi^T(t)\theta \quad (1.34)$$

donde $\epsilon(t) = [\epsilon(1), \epsilon(2), \dots, \epsilon(N)]$, y N es el número de mediciones realizadas. Este se denota normalmente como error de ecuación, error de medición o residuo. $y_p(t)$ es la salida medida del proceso.

Así el problema se reduce a encontrar el vector θ que minimice la suma de los cuadrados de los residuos. Esta suma es conocida como función de pérdida o función de costo y se representa por $V(\theta)$ y está dada por

$$V(\theta) = \frac{1}{N} \sum_{t=1}^N [y_p(t) - \phi^T(t)\theta]^2 \quad (1.35)$$

Para obtener su mínimo, se deriva la función de pérdida con respecto al vector θ y se iguala a cero, al resolver la ecuación resultante se obtiene el vector de parámetros estimado

$$\theta = [R(N)]^{-1} \frac{1}{N} \sum_{t=1}^N \phi(t)y(t) \quad (1.36)$$

donde la matriz

$$R(N) = \frac{1}{N} \sum_{t=1}^N \phi(t)\phi^T(t) \quad (1.37)$$

es conocida como la matriz de covariancia del sistema.

1.8 VALIDACIÓN DEL MODELO ESTIMADO

Las principales técnicas de validación son:

a) Validación de cruce

Esta técnica permite validar el orden y retardo del modelo, con base en la comparación de las funciones de pérdida (ec 1.35), para diferentes órdenes y retardos.

b) Comparación de la salida real contra la simulada

Este es uno de los procedimientos de validación más directos y requiere de las salidas del proceso y del modelo. Los datos reales de la salida del proceso, se obtienen del experimento de identificación y los datos del modelo se generan en una computadora a partir del modelo estimado excitándolo con los datos de la entrada real. Esta validación se implanta comparando visualmente los valores reales de la salida con los datos simulados.

c) Análisis Residual

Este análisis se realiza sobre los residuos (1.34). Consiste básicamente en la obtención de las funciones de autocorrelación de los residuos y de la de correlación entre éstos y la excitación del proceso.

Para la estructuras de modelo consideradas, los residuos deben ser lo menos correlacionados posible, es decir; deben ser independientes entre sí. De igual forma debe existir independencia entre los residuos y las entradas pasadas.

CAPÍTULO 2

DESCRIPCIÓN DEL SISTEMA PARA IDENTIFICACIÓN

2.1 ESQUEMA DE BLOQUES

El esquema de bloques del sistema de identificación implantado, se muestra en la figura 2.1 y a continuación se describen los bloques más importantes de éste.

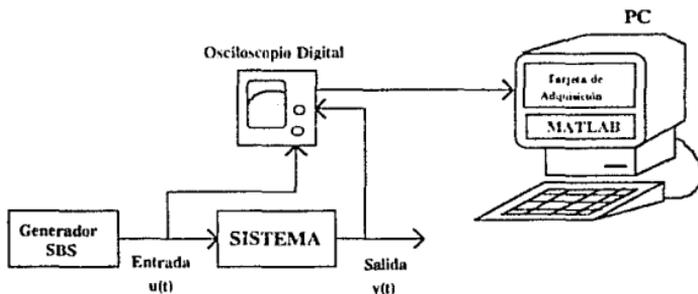


Fig 2.1 Esquema de bloques del equipo implantado

2.2 DESCRIPCIÓN DEL OSCILOSCOPIO DIGITAL

El instrumento de medición, mediante el cual se obtienen las señales de entrada y salida del proceso en estudio, es un osciloscopio digital TEKTRONIX 11402A. Algunas de sus características particulares son (Ref 10):

- Captura, muestreo y almacenamiento de señales en forma digital.
- La captura de las formas de onda se realiza con una resolución vertical de 10 bits.
- El mínimo y máximo de puntos de la captura de las señales, son 512 y 10240, respectivamente.

- Pantalla sensible al tacto, en la que se pueden seleccionar opciones de menús y submenús de la funciones del osciloscopio.
- El osciloscopio puede conectarse a una computadora personal a través de un puerto serie (RS232) o utilizando una tarjeta especial que requiere el GPIB (*General Purpose Interface Bus*), la cual se describe posteriormente.
- Es posible conectar directamente una impresora al osciloscopio e imprimir la señal mostrada en la pantalla de éste.

2.3 DESCRIPCIÓN DE LA TARJETA DE ADQUISICIÓN DE DATOS

La tarjeta de adquisición de datos AT-GPIB 488.2, diseñada de acuerdo al estándar IEEE 488.2 sirve como interfaz entre el osciloscopio digital y una computadora personal IBM AT o compatible.

Para realizar la adquisición de datos con dicha tarjeta, se cuenta con el paquete, denominado SPDMENU (Ref 7), que consta de rutinas especiales para la adquisición y procesamiento digital de señales, como son: correlaciones, transformada rápida de Fourier, convoluciones, etc.

2.4 DESCRIPCIÓN DEL GENERADOR DE SECUENCIAS BINARIAS SEUDOALEATORIAS

Para generar las señales de excitación fue necesario contruir un generador de secuencias binarias pseudoaleatorias, al que se le nombró Generador SBS. Dicho generador está basado en una secuencia-m donde el entero n de la ec (1.10) toma el valor de 11. Este valor se seleccionó tomando en cuenta que el número de cambios de la secuencia antes de repetirse fuera grande para propósitos de identificación, además que su implantación para n=11 es convencional. Por tanto, el periodo de la secuencia binaria pseudoaleatoria para el caso elegido (1.9) es

$$M = 2047(\Delta) \quad (2.1)$$

El esquema general del Generador SBS se presenta en la siguiente figura

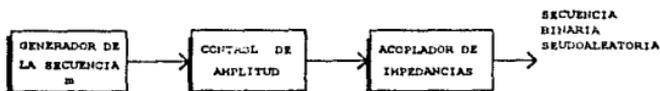


Fig 2.2 Esquema de bloques del Generador SBS

El circuito del generador de la secuencia- m se muestra en la figura 2.3 y se observa que está compuesto por un registro de corrimiento de 11 *bits*, una compuerta OR-Exclusiva y la señal de reloj que activa a los registros. El funcionamiento del generador de la secuencia- m es como sigue: al registro de corrimiento se les carga, de manera inicial, un dato diferente de cero. Una vez hecho esto, se realimenta el primer *bit* con la suma módulo-dos de los *bits* 9 y 11, realizada a través de una compuerta OR-exclusiva. El valor del *bit* 11, constituye la secuencia- m .

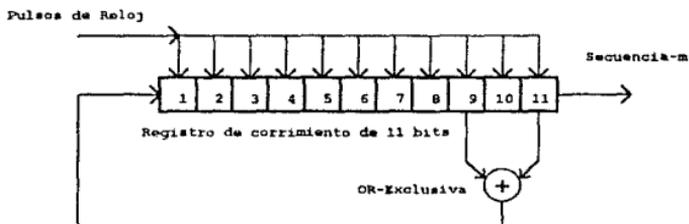


Fig 2.3 Diagrama del generador de la secuencia- m

Las características del Generador SBS se presenta a continuación (los datos anotados entre paréntesis, corresponden a la simbología utilizada en los diagramas electrónicos anexados en el apéndice C).

El generador de la secuencia- m está construido con base en dos registros de corrimiento (U_2 y U_3) del tipo universal de 8 *bits* cada uno, conectados en serie. Se utilizan únicamente 11 *bits* del conjunto. En el generador es posible realizar variaciones en el periodo básico, simplemente variando la frecuencia de la señal de reloj aplicada al registro de corrimiento.

Con el fin de manipular las amplitudes de la secuencia binaria, se dotó al Generador SBS de un módulo de control de amplitud (fig 2.2), en el cual se conecta la salida del *bit* 11 del registro, a la entrada de un comparador de voltaje basado en un amplificador operacional (U6A). Cuando la señal tiene un nivel alto, el voltaje de salida es el de la fuente de alimentación positiva del amplificador operacional (en este caso 12V), cuando sucede lo contrario, el voltaje de salida es el de la fuente de alimentación negativa (-12V). Con la señal anterior y el uso de recortadores de señal se puede variar la amplitud del nivel alto de la señal de 0 a 12V y el nivel bajo de -12 a 0V. El circuito de los recortadores de señal se construyó con base en amplificadores operacionales (U7A, U7B, U7C y U7D) y permite variar los niveles de la señal a través de potenciómetros (POT2 y POT3). Como etapa final (acoplador de impedancia, fig 2.2) se usa un amplificador operacional (U8) en configuración seguidor.

El rango total del reloj va desde 0.1Hz hasta 20kHz. Este rango se determinó con el fin de poder identificar tanto procesos relativamente lentos (como los térmicos) como rápidos (como el que corresponde a la dinámica eléctrica de un motor). El reloj se construyó con base en un circuito estable (U1) y la frecuencia del mismo se establece con arreglos de capacitores (C2, C3, C4, C5 y C6), resistencias (R2, R3, R4, R5 y R6) y un potenciómetro (POT1). Un conmutador de tipo selector (SW2) permite seleccionar los arreglos de capacitores y resistencias adecuados para determinados rangos de frecuencia del reloj; con lo cual se consigue tener mayor precisión en la generación de la señal. Mediante el potenciómetro (POT1) se realiza el ajuste fino del reloj.

2.5 DESCRIPCIÓN DEL PAQUETE MATLAB

El paquete MATLAB (*MATrix LABoratory*), es una poderosa herramienta de asistencia matemática, que está constituida de un paquete principal de análisis matricial y un conjunto de paquetes especiales denominados "Cajas de Herramientas" (*Toolboxes*). Estas cajas de herramientas contienen funciones o comandos de aplicación específica en diferentes áreas de la ingeniería (Ref 5).

La caja de herramientas *System Identification Toolbox*, proporciona los comandos comúnmente utilizados en el área de identificación, como son: análisis no paramétrico (análisis de Fourier, espectral y de correlación), estimación paramétrica de diversas estructuras, etc. Con esta caja es posible estimar parámetros tanto con entradas

externas, ya sea para sistemas SISO (*Single Input-Single Output*, una entrada-una salida) o sistemas MIMO (*Multi Input-Multi Output*, varias entradas-varias salidas); como de procesos sin señal de excitación medible (Ref 9).

2.6 RUTINA DE ESTIMACIÓN

El programa que realiza el proceso matemático del procedimiento de identificación, se llevó a cabo en la versión 3.2 de MATLAB, con las siguientes cajas de herramientas: *System Identification Toolbox*, *Control System Toolbox* y *Signal Processing Toolbox*.

A continuación, se describe el formato para manejar las estructuras de los modelos a estimar dentro de MATLAB (Ref 9) y que se conoce como formato THETA. A continuación, se lista la información principal que maneja:

- ◆ Valores medios y desviaciones estándares de los parámetros estimados.
- ◆ Periodo de muestreo utilizado.
- ◆ Valor de la función de pérdida.
- ◆ Tiempo y fecha de la creación del modelo.
- ◆ Provee información general acerca de la estructura de modelo utilizada.

Toda esta información es posible visualizarla utilizando la instrucción *present(th)*.

Como ejemplo, se muestra a continuación como se almacena la información de la estructura para el caso ARX con un periodo de muestreo unitario.

```
B = 0.0000  0.0000  0.0000  1.5415  
      0.0000  0.0000  0.0000  0.0321
```

```
A = 1.0000 -1.5200  0.6983  
      0.0000  0.0204  0.0022
```

Los primeros renglones de *B* y de *A* corresponden a los parámetros nominales. En este caso el modelo estimado nominal tiene la siguiente forma

$$y(t) - 1.52y(t-1) + 0.6983y(t-2) = 1.5415u(t-3)$$

El segundo renglón de las matrices A y B corresponde a las desviaciones estándares de los parámetros estimados. Lo cual es válido para cualquier estructura de modelo utilizada.

En el caso del primer renglón de la matriz B , el número de valores nulos que aparecen en las primeras columnas es igual a nk (ec 1.16). Los valores siguientes al último valor nulo, son los parámetros estimados del polinomio $B(q)$.

El valor unitario que aparece en la primera columna del primer renglón de la matriz A , corresponde al primer término de la estructura general del polinomio $A(q)$; los números siguientes en el primer renglón de A son los parámetros estimados del polinomio $A(q)$ (recuérdese que, $A(q)$ es mónico). Los polinomios $C(q)$, $D(q)$, $E(q)$ y $F(q)$, presentes en otras estructuras, se representan de modo semejante al polinomio $A(q)$.

EJECUCIÓN DE LA RUTINA DE ESTIMACIÓN

Para ejecutar la rutina es necesario estar en el ambiente de MATLAB; para ello, dentro del directorio adecuado se tecldea el comando *matlab*. Una vez en dicho ambiente se ejecuta la rutina de identificación tecleando el comando *idensis*.

- En primer término, la rutina pregunta por los nombres de los archivos de los datos de entrada, de salida y el tiempo de muestreo del osciloscopio. Esta rutina permite dividir el conjunto de datos de entrada y salida en dos grupos, uno para realizar la estimación y otro que se reserva para la validación del modelo obtenido.
- La rutina, ofrece la opción de estimar modelos no paramétricos; los cuales consisten en las gráficas del espectro de la entrada y el de la salida, las respuesta al impulso y al escalón estimadas, las funciones de covariancia y de correlación entre la entrada y la salida, además de la respuesta en frecuencia del sistema en cuestión. Todo esto se realiza con el fin de obtener una idea de la dinámica del proceso.
- Para realizar la estimación paramétrica, es posible elegir entre diferentes estructuras de modelos (ARX, ARMAX, OE, BJ e IV4). Para cada una de éstas se muestra en pantalla una breve información acerca de los polinomios que las integran. Para cada una de las estructuras elegidas, el usuario debe indicar el

número de parámetros a estimar. Una vez seleccionada la estructura, se realiza la estimación de los parámetros. Cuando se utilizan las estructuras ARX e IV4 se muestra información acerca de la estimación, que permite elegir más fácilmente el orden del retardo y del modelo. En particular se lleva a cabo el cálculo de la función de pérdida $V(0)$ (ec 1.35) y además se incluye la posibilidad de mostrar gráficamente los valores de $V(0)$, para un determinado valor de retardo y diferentes combinaciones de n_a , n_b y n_k . Por ello, es conveniente hacer uso de estas estructuras como un paso inicial en la estimación.

- Una vez obtenidos los valores de los parámetros, se procede a validar el modelo; para ello, el modelo estimado es simulado con los datos de entrada reservados para este caso y su salida se muestra gráficamente junto con la salida real del sistema. La rutina provee la opción de poder realizar un acercamiento de la gráfica obtenida. También se muestran la respuesta en frecuencia del modelo paramétrico y la obtenida con el procedimiento no paramétrico (si se realizó éste). A continuación se muestran las funciones de autocorrelación de los residuos y la de correlación entre éstos y la entrada, simultáneamente se indican los niveles de confianza (del 99%) para las señales.
- Posteriormente a la validación del modelo, es posible obtener, a partir del modelo en el formato theta, las funciones de transferencia del sistema identificado, ya sea en el dominio de tiempo discreto o continuo. Una vez que se tiene la función de transferencia en el tiempo continuo del sistema identificado, se obtiene la respuesta al escalón de éste. También se grafican los polos y ceros del modelo estimado.
- Si la identificación se considera satisfactoria la rutina termina almacenando los datos del modelo en el formato theta. En caso contrario, se pueden calcular otros parámetros, utilizar otra estructura de modelo o realizar la estimación de otro proceso.

2.7 PROCEDIMIENTO DE USO DEL SISTEMA

A continuación se indica la metodología práctica a seguir para el uso del sistema de identificación integrado.

I.- OBTENCIÓN DE LOS DATOS EXPERIMENTALES

Consiste principalmente de los siguientes pasos:

En primer lugar, se determina el periodo de muestreo, T , de las señales de entrada y salida, de acuerdo con las reglas establecidas en las ecuaciones (1.11), (1.12) y (1.13).

Una vez elegido el periodo de muestreo, se calcula el periodo básico (Δ) de la secuencia binaria pseudoaleatoria, aplicando la ecuación (1.14).

Después de determinar el periodo básico, se debe definir el tiempo total de barrido en la pantalla ó tiempo de ventana del osciloscopio (T_V), y el número de puntos (NP) que muestreará el osciloscopio, tomando en cuenta que sólo una parte de los datos se usará en la validación. Además debe buscarse una combinación de T_V y NP que establezca un buen compromiso entre lograr que el tiempo de muestreo (T_{ms}) sea lo más cercano a T y el de que el número de cambios (NC) de la SBS sea menor a 2047. En la literatura estudiada, se manejan órdenes de NC cercanos a 1000. Algunas recomendaciones para encontrar la combinación adecuada se presentan en el apéndice B.

Una vez seleccionados el tiempo de ventana, el número de puntos de muestreo y el periodo básico de la secuencia binaria, se excita al proceso en estudio con dicha señal y se capturan y almacenan las señales de entrada y salida en el osciloscopio, como se indica en el apéndice B.

II.- TRANSPORTE DE LOS DATOS A ARCHIVOS PARA PC

Utilizando la tarjeta de adquisición AT-GPIB 488.2, el programa SPDMENU y algún editor de texto, los datos almacenados en el osciloscopio se convierten en archivos ASCII que puedan ser interpretados correctamente por MATLAB. En el apéndice B se incluye el prontuario del procedimiento para dicha conversión.

III.- EJECUCIÓN DE LA RUTINA DE ESTIMACIÓN

Con los datos obtenidos, se está en condiciones de correr el programa *idensis.m*.

2.8 IDENTIFICACIÓN DE UN PROCESO TÉRMICO

En este apartado, se presenta en detalle la aplicación del sistema sobre el proceso térmico de laboratorio (*Feedback's Process Trainer PT326*). Éste funciona de la manera siguiente: el aire es capturado a través de un ventilador centrífugo, se hace pasar por un calentador de resistencia y después es transportado a lo largo de un conducto para ser expulsado al exterior. El calentamiento de la resistencia se controla por medio de una señal de voltaje continuo. La señal de salida es medida a través de un transductor de temperatura a voltaje. El conducto de transporte del aire posee tres aberturas donde es posible colocar el sensor de temperatura. Para el experimento, el sensor se colocó en la parte más alejada del ventilador.

Con el fin de tener una idea de la dinámica del sistema, se generó la respuesta al escalón del proceso, mostrada en la figura 2.4.

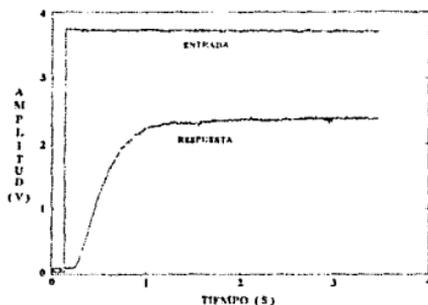


Fig 2.4 Respuesta al escalón del proceso

De la figura 2.4 se puede obtener la constante de tiempo dominante del sistema

$$\tau = 0.366 \text{ s} \quad (2.2)$$

Con base en la ecuación (1.11) y el valor de τ se calcula el periodo de muestreo

$$T = 0.183 \text{ s} \quad (2.3)$$

Así, de acuerdo a la relación (1.14), el periodo básico de la secuencia binaria pseudoaleatoria debe cumplir

$$\Delta \geq 0.366 \text{ s} \quad (2.4)$$

Se decide, entonces utilizar un periodo básico igual a tres veces τ , esto es, 1.098 s.

Con base en los valores seleccionados de τ , T y Δ , se elaboró la siguiente tabla tomando en cuenta las ecuaciones (B.1) y (B.2)

OPCIÓN	NP	TV [s]	NC
1	512	93.696	85.33
2	1024	187.392	170.67
3	2048	374.784	341.33
4	4096	749.568	682.67
5	5120	936.96	853.33
6	8192	1499.136	1365.33
7	10240	1873.92	1706.67

Tabla 2.1 Combinaciones de NP, TV y NC

Buscando la opción que contenga el tiempo de ventana más cercano a uno de los accesibles del osciloscopio y que NC sea cercano a 1000, se toma la opción 5, a la que corresponde el tiempo de ventana estándar del osciloscopio más cercano de 1000 s. El tiempo de muestreo del osciloscopio (T_{osc}) está dado por la ecuación 2.5 y resulta

$$T_{osc} = \frac{TV}{NP} = \frac{1000}{5120} = 0.1953 \text{ s} \quad (2.5)$$

y se observa, que es bastante parecido al tiempo de muestreo calculado, ecuación 2.3.

Una vez ajustado el osciloscopio, se excita al proceso con la señal del Generador SBS. Para este caso, las amplitudes máxima y mínima de la secuencia binaria se establecieron en 4V y 0V, respectivamente.

Una vez grabadas las señales en el osciloscopio, se transfirieron a la PC a través de la tarjeta de adquisición y se editaron para la interpretación adecuada de éstas por MATLAB.

Dentro de la rutina de estimación, y después de proporcionar los archivos de los datos de entrada y salida y el valor del periodo de muestreo, se selecciona un grupo de datos para usarse en la estimación y otro para la validación. En la figura 2.5, se muestra una porción de los datos destinados a la estimación.

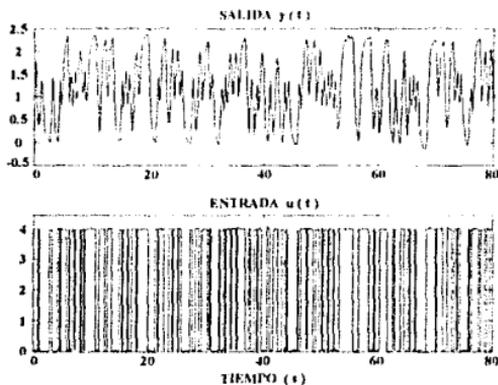


Fig 2.5 Gráficas de los datos muestreados

Se decide realizar un análisis no paramétrico de los datos, la rutina presenta un conjunto de modelos no paramétricos dado por las funciones de covariancia, respuesta al impulso, respuesta al escalón y diagrama de Bode estimados.

En primer lugar, se muestran las funciones de covariancia del proceso (después de realizarle un filtrado blanco a los datos) y la respuesta al impulso estimada del sistema (fig 2.6)

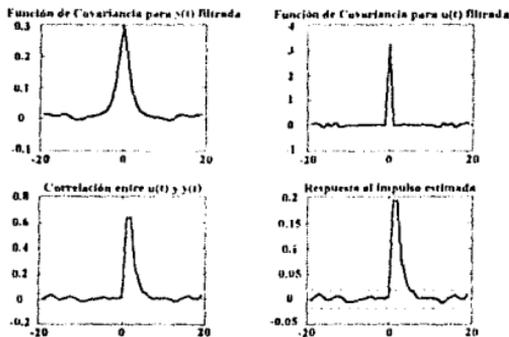


Fig 2.6 Funciones de Covariancia y respuesta al impulso estimadas

A partir de la respuesta al impulso, se obtiene la respuesta al escalón en forma estimada con lo que se genera la segunda y tercera pantalla. En la primera, se grafica ésta en función de los periodos de muestreo (fig 2.7) y en la segunda, directamente en función del tiempo (fig 2.8). Con la primera de estas gráficas es más fácil determinar, cuantos periodos de muestreo está retrasada la señal de salida. Para el ejemplo, se observa que la señal presenta un retraso cercano a un periodo de muestreo; la segunda gráfica es útil para estimar la constante de tiempo y, en algunos casos, para compararla con datos experimentales. En el proceso térmico la constante de tiempo estimada resultó ser de aproximadamente 0.37; valor bastante cercano al obtenido mediante la salida real del proceso (fig 2.4).

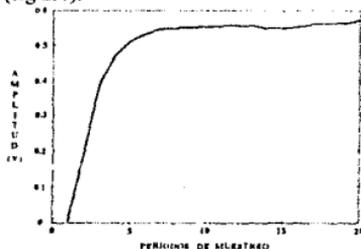


Fig 2.7 Respuesta al escalón unitario estimada, en función de los periodos de muestreo

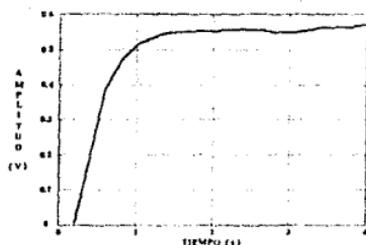


Fig 2.8 Respuesta al escalón unitario estimada, en función del tiempo

Finalmente se muestra el diagrama de Bode estimado del proceso

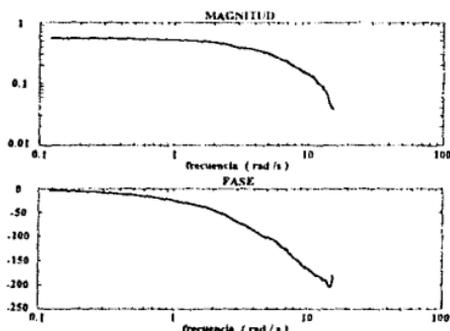


Fig 2.9 Diagrama de Bode estimado del modelo no paramétrico

Para realizar la estimación paramétrica se eligió la estructura ARX con polinomios $A(q)$ y $B(q)$ de orden 2 ($n_a=n_b=2$). Como resultado se obtiene

$$B = \begin{bmatrix} 0 & 0.1963 & 0.0377 \\ 0 & 0.0019 & 0.0044 \end{bmatrix}$$

$$A = \begin{bmatrix} 1.0000 & -0.7383 & 0.1727 \\ 0 & 0.0183 & 0.0130 \end{bmatrix}$$

Se observa de B un orden del retardo igual a uno y que en las matrices las desviaciones estándares de los parámetros son muy pequeñas, esto significa que los residuos (ec 1.34) deben ser pequeños. Como información adicional, el programa muestra la gráfica de los polos y ceros del modelo estimado (fig 2.10).

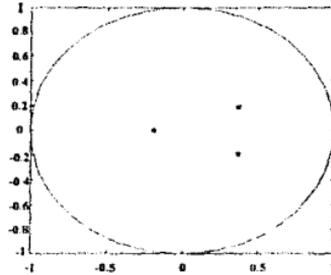


Fig 2.10 Polos y ceros del modelo

A continuación se muestra el diagrama de Bode estimado del modelo paramétrico (Fig 2.11).

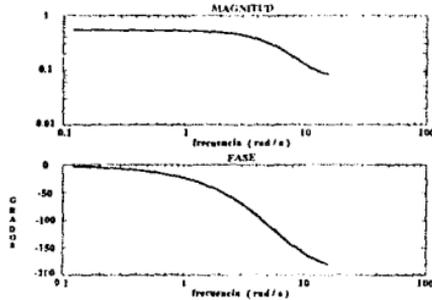


Fig 2.11 Diagramas de Bode del modelo paramétrico

Con objeto de comparar los modelos estimados, paramétrico y no paramétrico, se presentan (Fig 2.12), en un mismo diagrama de Bode, ambas respuestas frecuenciales;

de donde se ve que las respuestas en bajas frecuencias son muy similares en ambos casos. Por el contrario, a partir de 10 rad/s la diferencia entre ambas respuestas es apreciable.

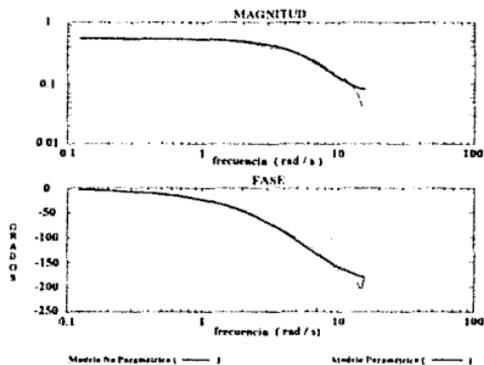


Fig 2.12 Comparación de los diagramas de Bode

Similarmente, se presentan en la figura 2.13 las respuesta al escalón unitario para ambos modelos. En este caso la diferencia entre las respuestas es apreciable a bajas frecuencias; la constante de tiempo es prácticamente igual en ambos modelos.

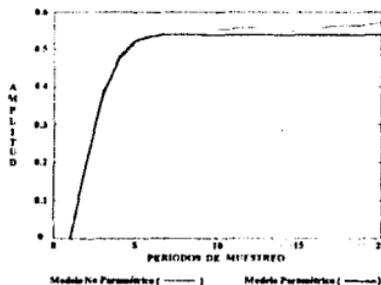


Fig 2.13 Comparación de las respuestas al escalón estimadas

Por último, se procede a validar el modelo. Para ello, el sistema se auxilia de varias presentaciones (Fig 2.14 a la 2.17). La figura 2.14, muestra una porción de la salida medida y de la simulada, con el modelo bajo la misma excitación, y donde se ha eliminado la componente de corriente directa. En la figura 2.15 se aprecia el error entre las salidas.

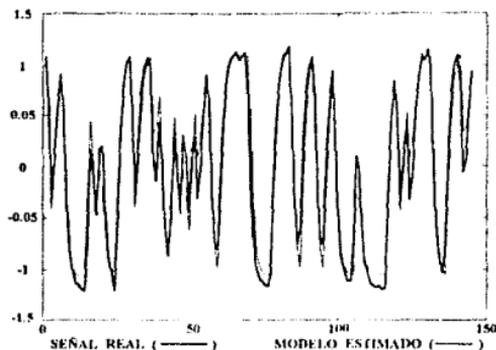


Fig 2.14 Acercamiento de los datos medidos vs datos simulados

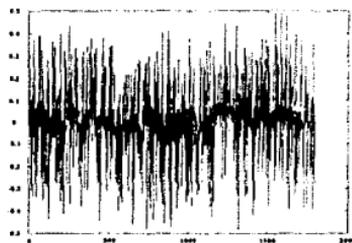


Fig 2.15 Error entre las salidas

Las siguientes pantallas (figuras 2.16 y 2.17), corresponden a la función de correlación entre los residuos y la excitación, y a la función de autocorrelación entre los residuos, respectivamente. Se observa en ambas gráficas que los niveles de correlación están limitados en su mayor parte por los niveles de confianza recomendados (los cuales se

muestran con líneas punteadas). Una de las consideraciones supuestas más importantes, que se debe verificar, es la de excitación persistente. La gráfica de autocorrelación muestra claramente que la suposición es válida.

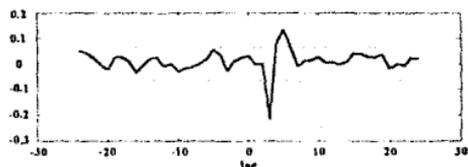


Fig 2.16 Función de Correlación entre los residuos y la entrada

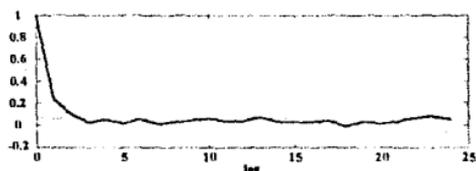


Fig 2.17 Función de Autocorrelación de los residuos

Finalmente, el sistema presenta la función de transferencia discreta estimada, obtenida a partir del modelo theta.

$$H(z) = \frac{0.1963z + 0.03767}{z^2 - 0.7383z + 0.1727} \quad (2.6)$$

De esta función se observa que el orden relativo impuesto al modelo por medio de nk , se mantiene (i.e. $nk=1$).

Utilizando la aproximación matemática de Tustin para un integrador, se obtiene

$$H(s) = \frac{-0.08302s^2 - 0.3942s + 12.24}{s^2 + 8.658s + 22.73} \quad (2.7)$$

Debe tomarse en cuenta que en este caso la aproximación genera una función propia (mismo número de polos y ceros).

Por último, la función de transferencia en el dominio del tiempo continuo es simulada, obteniéndose la respuesta al escalón que se ilustra en la figura 2.18.

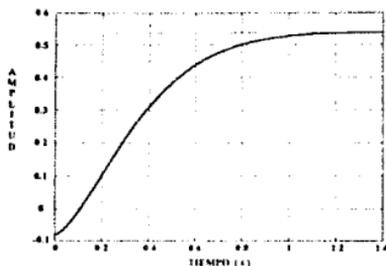


Fig 2.18 Respuesta al escalón de $H(s)$

Comparando esta última grafica con la figura 2.4, se observa que el modelo en el tiempo continuo cumple con las características de la forma de la respuesta y ganancia estática del sistema, aunque el cero inestable en el modelo continuo provoca valores negativos al inicio de la respuesta. Con esta figura termina el ejemplo presentado.

CONCLUSIONES

El presente trabajo consiste en implantar un sistema para realizar, de manera integrada, los pasos necesarios para una identificación de procesos dinámicos, fuera de línea y con el auxilio de una computadora personal. El sistema está integrado por la computadora personal, un generador de secuencias binarias pseudoaleatorias (Generador SBS), un osciloscopio digital, una tarjeta de adquisición de datos y el paquete de asistencia matemática MATLAB, y permite identificar bajo la suposición de linealidad e invariancia en el tiempo procesos de una entrada y una salida.

El presente reporte describe brevemente el tipo de modelos que pueden considerarse en el procedimiento de identificación; además, se anexa un prontuario para el uso adecuado de la tarjeta de adquisición y del osciloscopio digital, así como la descripción detallada, dotada de ejemplos, de la rutina de estimación diseñada. Dicha rutina incorpora pantallas, tanto gráficas como de texto, con información sobre los pasos de la misma.

A partir de la experiencia adquirida en la implantación y utilización del sistema, pueden hacerse las siguientes recomendaciones, para la correcta operación del mismo.

Un conocimiento, al menos somero de las características dinámicas del proceso es crucial para una buena identificación. Por ejemplo, al no considerar adecuadamente la dinámica del proceso, es muy probable que la señal de excitación seleccionada no excite todas las dinámicas importantes de éste, ó que el tiempo de muestreo no permita recuperar correctamente las señales de entrada y salida.

Para usar adecuadamente la rutina de estimación es importante una correcta interpretación de los polinomios que componen las estructuras de los modelos propuestos. Debe ponerse especial atención, en la interpretación de n_k , que determina el orden relativo de dichos modelos.

La aplicación inmediata del sistema, es la de facilitar la implantación de las técnicas de identificación, en los procesos del Laboratorio de Control tanto del Instituto de Ingeniería como de la División de Estudios de Posgrado de la Facultad de Ingeniería.

Finalmente, se proponen las siguientes directrices para futuros trabajos orientados a mejorar el sistema:

- Ampliar la capacidad del sistema para abarcar el análisis de procesos multivariables (varias entradas-varias salidas) utilizando las funciones que para tal caso se encuentran en el *System Identification Toolbox*, de MATLAB.
- Automatizar la captura de datos a través de programas en lenguaje C y de una interfaz entre dicho lenguaje y MATLAB, de modo que, desde la rutina de estimación sea posible acceder los datos del osciloscopio digital. Esto no pudo llevarse a cabo para este trabajo por no contar con la paquetería adecuada (*Microsoft C*, versión 6.0).
- Existe la posibilidad de realizar directamente la captura de los datos e incluso excitar los procesos utilizando tarjetas de conversión analógica-digital y digital-analógica, manejadas con programas en C residentes en MATLAB, lo cual se ahorraría el uso del osciloscopio digital.

REFERENCIAS

- [1] Godfrey, K. R.. Correlations Methods. Automatica, vol 16. 1980.
- [2] Isermann, R.. Practical Aspects of Process Identification. Automatica, vol 16. 1980.
- [3] Lancaster, D.. *Active Filter Cookbook*. Howard W. Sams & Co., Inc. 1982.
- [4] Ljung, L.. *System Identification, Theory for the user*. Prentice-Hall, Englewood Cliffs, N.J.. 1987.
- [5] MATLAB User's Guide, Math Works Inc. 1992.
- [6] Söderström, T. and Stoica P.. *System Identification*. Prentice-Hall, Englewood Cliff, N.J.. 1989.
- [7] SPD Signal Processing & Display Programs, Tektronix. 1992.
- [8] Strej V.. Least Squares Parameter Estimation. Automatica, vol 16. 1980.
- [9] System Identification Toolbox User's Guide, Math Works Inc. 1992.
- [10] The 11402A&11403 Digitizing Oscilloscopes, TUTORIAL, Tektronix. 1989.
- [11] Wellstead, P.E.. Non-Parametric Methods of System Identification. Automatica, vol 17, No. 1. 1981.

APÉNDICE A

EJEMPLOS DE APLICACIÓN

A continuación, se describen brevemente dos ejemplos de aplicación del equipo integrado.

IDENTIFICACIÓN DE UN FILTRO ACTIVO

El filtro propuesto es uno pasobajas de segundo orden, cuyo diagrama eléctrico (Ref. 3, pág. 139) se muestra en la figura A.1.

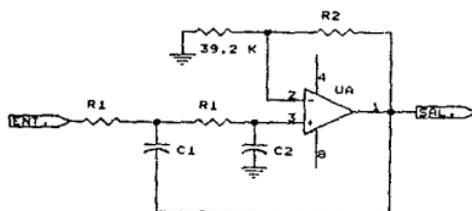


Fig A.1 Filtro pasobajas de segundo orden

De acuerdo a la figura A.1, la función de transferencia del filtro está dada por la siguiente expresión

$$H(s) = \frac{K}{s^2 + \frac{R1C1 + R2C2 + (1-K)R1C1}{R2C2R1C1}s + \frac{1}{R1C1R2C2}} \quad (A.1)$$

donde K es la ganancia estática del filtro.

Al medir, con un multímetro, los valores reales de los elementos constituyentes del filtro, y con un osciloscopio las amplitudes de entrada y salida, se obtiene

$$R1 = 11.7k\Omega \quad R2 = 48.7k\Omega$$

$$C1 = 18.77nF \quad C2 = 18.55nF$$

$$K = \frac{2.21V}{1.01V} = 2.188$$

De donde

$$H(s) = \frac{50.3645 \times 10^6}{s^2 + 2787.1185s + 21.1615 \times 10^6} \quad (A.2)$$

Ahora bien, la respuesta al escalón, experimental, del filtro es la siguiente

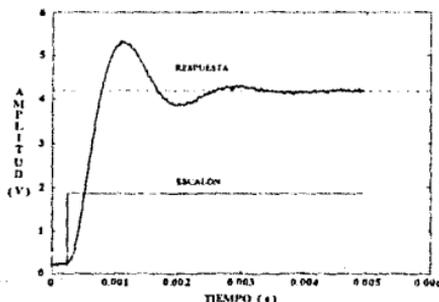


Fig A.2 Respuesta al escalón del proceso

Se excitó al filtro con una señal SBS. Una porción de los datos capturados se muestra en la figura A.3.

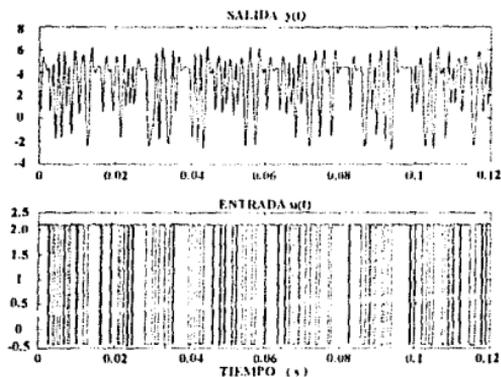


Fig A.3 Gráfica de los datos muestreados

La respuesta al escalón estimada a partir del análisis no paramétrico es la siguiente

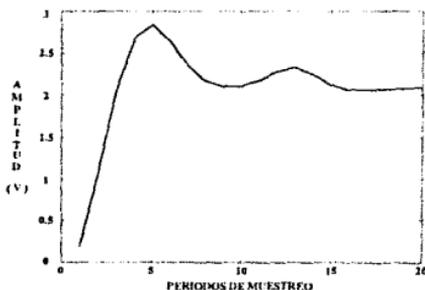


Fig A.4 Respuesta al escalón estimada

Para el análisis paramétrico, se eligió una estructura ARMAX (con $n_a = 2$, $n_b = 2$, $n_c = 2$ y $n_k = 0$), el resultado de la estimación es

$$B = \begin{bmatrix} 0.1853 & 0.6746 \\ 0.0055 & 0.0060 \end{bmatrix}$$

$$A = \begin{bmatrix} 1.0000 & -1.1864 & 0.5896 \\ 0 & 0.0049 & 0.0047 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.0000 & 0.2956 & 0.0376 \\ 0 & 0.0229 & 0.0229 \end{bmatrix}$$

La comparación de la simulación del modelo propuesto y la salida real se muestra en la gráfica A.5.

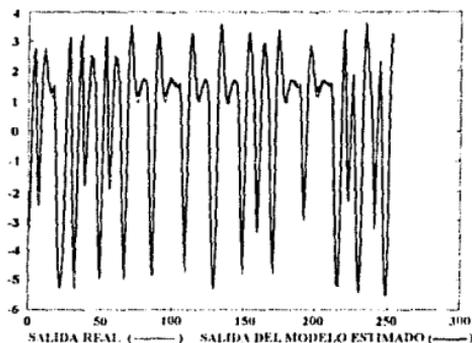


Fig A.5 Comparación de datos simulados vs datos reales

Se observa un buen seguimiento de los valores simulados a los valores reales, lo cual indica que el modelo obtenido es adecuado.

A continuación se presenta el análisis residual hecho sobre el modelo obtenido, del cual resultan las figuras A.6 y A.7. En estas gráficas se muestra que los errores de ecuación son, en general, independientes entre sí.

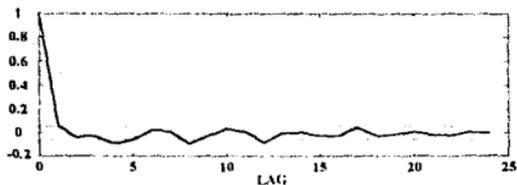


Fig A.6 Correlación de los residuos

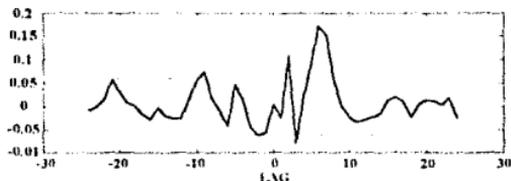


Fig A.7 Correlación entre $u(t)$ y los residuos

La función de transferencia, en el tiempo discreto, del modelo estimado es la siguiente

$$H(z) = \frac{0.1853z^2 + 0.6746z}{z^2 - 1.1864z + 0.5896} \quad (\text{A.3})$$

A partir de la ecuación anterior, se obtiene (a través de la aproximación de Tustin) la función de transferencia en el dominio del tiempo continuo (ec A.4).

$$H(s) = \frac{-0.1762s^2 + 1335s + 3.097 \cdot 10^7}{s^2 + 2957s + 1.452 \cdot 10^7} \quad (\text{A.4})$$

Por último, se obtiene la respuesta al escalón de esta última función de transferencia. El resultado se muestra en la figura siguiente (fig A.8).

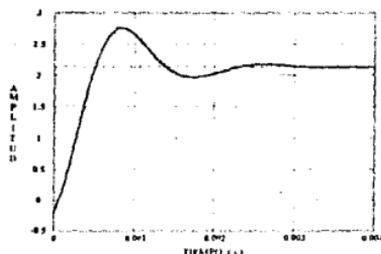


Fig A.8 Respuesta al escalón de H(s)

Las diferencias más notables entre el modelo teórico (ec A.2) y los estimados (ecs A.3 y A.4) se encuentran en la aparición de ceros en los modelos estimados. En el caso del modelo continuo se generan ceros de fase no mínima. Se intentó, modificando el valor de nk , obtener un modelo sin ceros o al menos sin ceros de fase no mínima; sin embargo, los modelos encontrados diferían sensiblemente del comportamiento del filtro real. La mejor aproximación del modelo es la que se indica en A.3 y A.4.

IDENTIFICACIÓN DE UN TRANSFORMADOR

Como segundo, y último ejemplo, se decidió identificar un transformador, cuyo modelo teórico aproximado es similar a un filtro pasobanda de segundo orden, esto es, consta de dos polos (los cuales determinan las frecuencias de corte alta y baja del sistema) y un cero en el origen. Como primer paso se analizó la respuesta en frecuencia de éste. Los resultados obtenidos de este análisis son los siguientes

$$f_{cb} = 7\text{Hz}$$

$$f_{ca} = 1.2\text{MHz}$$

donde f_{ca} y f_{cb} son las frecuencia de corte baja y alta del transformador, respectivamente.

Con base en estos datos se determinó que con el sistema integrado no era posible identificar la dinámica rápida ya que la frecuencia asociada a ésta es demasiado grande.

La rutina de estimación muestra los siguientes resultados.

Parte de los datos muestreados de las señales de entrada y salida se presentan en la figura A.10.

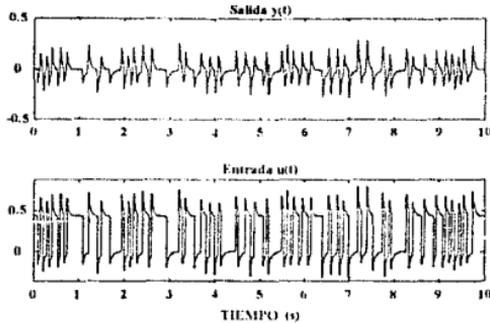


Fig A.10 Datos de la entrada y salida del proceso

Con base en estos datos, en el análisis no paramétrico, se obtuvo la respuesta al escalón estimada (figura A.11).

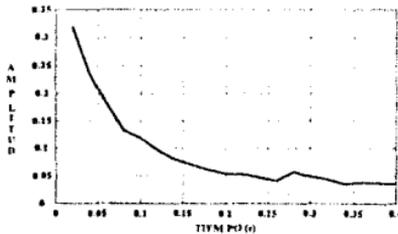


Fig A.11 Respuesta al escalón estimada

El diagrama de Bode estimado de este análisis es el siguiente, puede observarse que la frecuencia de corte baja, se aproxima a la medida (7Hz).

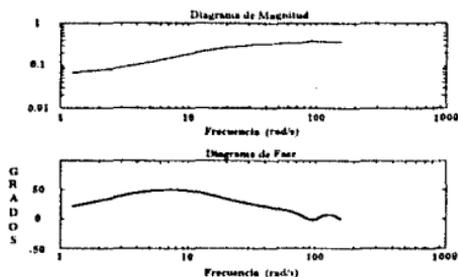


Fig A12 Diagrama de Bode estimado

En el análisis paramétrico, se eligió una estructura Box-Jenkins. Como primera aproximación, se probó con un modelo de primer orden, con la cual se obtuvo una mala aproximación del proceso. Se consideró posteriormente un modelo de segundo orden con $nb=2$, $nc=2$, $nd=2$, $nf=2$ y $nk=0$. De esta forma el modelo resultante (parámetros estimados y desviaciones estándares) fue

$$B = \begin{matrix} 0.0308 & -0.3035 \\ 0.0011 & 0.0017 \end{matrix}$$

$$F = \begin{matrix} 1.0000 & -0.7059 & -0.0207 \\ 0 & 0.0051 & 0.0035 \end{matrix}$$

$$C = \begin{matrix} 1.0000 & -0.1444 & -0.0311 \\ 0 & 0.3036 & 0.0570 \end{matrix}$$

$$D = \begin{matrix} 1.0000 & -0.6992 & -0.0239 \\ 0 & 0.3029 & 0.2162 \end{matrix}$$

La comparación entre la salida del modelo y los datos medidos se muestra en la figura A.13.

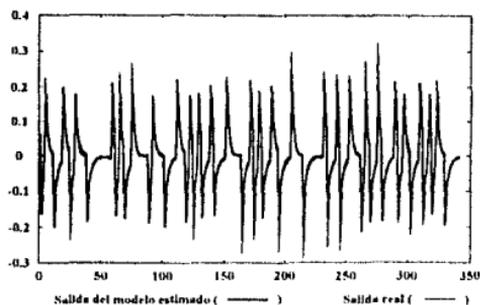


Fig A13 Salidas simulada y real

Se observa en la figura A13 una buena concordancia entre ambas señales, lo cual indica que el nivel de los residuos es pequeño, en comparación con los niveles de ambas señales.

Con base en el modelo estimado, se obtiene el diagrama de Bode (fig A14). Se observa de nuevo que la frecuencia de corte baja se aproxima a la frecuencia de corte medida.

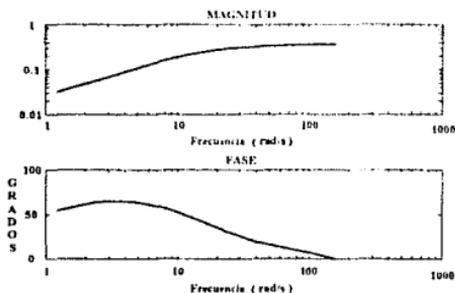


Fig A14 Diagrama de Bode del modelo paramétrico

A continuación se presentan las funciones de autocorrelación de los residuos y la de correlación entre éstos y la entrada aplicada (figuras A15 y A16).

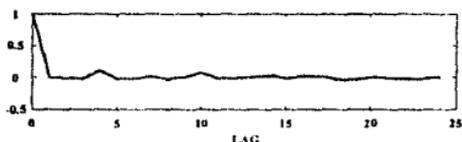


Fig A15 Función de Autocorrelación de los residuos

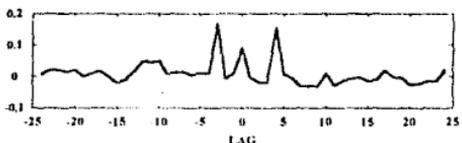


Fig A16 Función de Correlación entre los residuos y la excitación

La mayoría de los niveles de correlación de las señales están dentro de los límites establecidos, con lo cual podemos concluir satisfactoriamente la validación del modelo.

Por último, la función de transferencia en el tiempo discreto resulta

$$H(z) = \frac{0.3038z^2 - 0.3035z}{z^2 - 0.7059z - 0.02071} \quad (\text{A.5})$$

Cabe hacer notar que el sistema identifica correctamente el cero en el origen, se debe tener cuidado con el restante cero, pues se encuentra muy cerca del círculo unitario. Este cero no está incorporado en el modelo teórico aproximado. Un paso que ya no se realizó por las limitaciones de tiempo impuestas a este trabajo, es el de comparar los resultados con modelos más completos que incluyan los componentes parásitos.

APÉNDICE B

PRONTUARIO PARA LA UTILIZACIÓN DEL OSCILOSCOPIO Y LA TARJETA DE ADQUISICIÓN

SELECCIÓN DEL TIEMPO DE VENTANA (TV) Y DEL NÚMERO DE PUNTOS MUESTREADOS DEL OSCILOSCOPIO (NP)

El tiempo de ventana del osciloscopio se determina a partir de la siguiente ecuación

$$TV = T(NP) \quad (B.1)$$

donde, T es el periodo de muestreo calculado para la captura de las señales.

El número de cambios (NC) de la secuencia binaria que aparecerán en la ventana de tiempo, se obtiene de la expresión

$$NC = \frac{TV}{\Delta} \quad (B.2)$$

Con las ecuaciones anteriores, y en función de los valores posibles de NP que provee el osciloscopio, se elabora la siguiente tabla

NP	TV	NC
512	TV1	NC1
1024	TV2	NC2
2048	TV3	NC3
4096	TV4	NC4
5120	TV5	NC5
8192	TV6	NC6
10240	TV7	NC7

Tabla B.1 Selección de TV del osciloscopio

Una vez calculados los elementos de la tabla B.1, se eligen las combinaciones de TV y NC con el número de cambios menor a 2047, y cercanos a 1000 cambios. Finalmente, de las combinaciones elegidas se selecciona la que tenga el tiempo de ventana más próximo a uno de los estándares del osciloscopio: 10ns, 20ns, 50ns, 100ns, 200ns, 500ns, 1000ns, 2000ns, 5000ns, 10 μ s, 20 μ s, 50 μ s, 100 μ s, 200 μ s, 500 μ s, 1000 μ s, 2000 μ s, 5000 μ s, 10ms, 20ms, 50ms, 100ms, 200ms, 500ms, 1000ms, 2000ms, 5000ms, 10s, 20s, 50s, 100s, 200s, 500s, 1000s.

TRANSPORTE DE LOS DATOS A ARCHIVOS PARA PC CON FORMATO DE MATLAB

A continuación se explica la forma de grabar los datos en el osciloscopio digital y de convertirlos a archivos para computadora personal (PC).

Para detener el trazado de las señales en la pantalla del osciloscopio y fijar la imagen, se debe oprimir el botón *STOP*. Para grabar las señales se presiona el botón *STORE/RECALL*, que sitúa al usuario en un menú, que sirve para introducir la información para almacenar las señales, en dicho menú se elige un número para cada señal.

Una vez que se tienen las señales grabadas en el osciloscopio, se procede de la forma siguiente:

- 1.- Conectar el osciloscopio digital a la PC a través de la tarjeta de adquisición de datos AT-GPIB IEEE 488.2.
- 2.- Ubicarse en la PC en el directorio donde se encuentra el programa *SPDMENU.BAT*.
- 3.- Ejecutar la instrucción *SPDMENU*.
- 4.- Una vez que aparece en pantalla el menú de opciones, se ignoran éstas y se tecldea directamente el comando *GET11K*, al hacer esto aparece en pantalla el siguiente formato

Acquired Waveform: _____
Waveform Source: STO
Waveform number: _____
GPIB address: POLL
Timeout: 30sec
Normalize Data (Offset and Scale Factor)?: YES

Fig B.1 Formato para la adquisición de señales

Dicho formato, presenta campos iluminados (señalados con negritas o en subrayado en la figura B.1). Para tener acceso a cada uno de los campos se utilizan ya sean las flechas del teclado que indican arriba y abajo ó la tecla *ENTER*. Para solicitar información de ayuda respecto a los campos, se oprime la tecla *F1*. Cuando se comete algún error y se quiere borrar tootalmente el contenido de algún campo se oprimen las teclas *CTRL* y *BACKSPACE* simultáneamente.

El campo asociado a *Acquired Waveform* corresponde al nombre que se dará a la señal por adquirir, y debe empezar siempre con una letra. El campo asociado a *Waveform number* se refiere al número en que fue almacenada en el osciloscopio digital la forma de onda a ser adquirida (Vgr. *STO5*).

Una vez que se tiene el formato correctamente llenado, se presiona la tecla *F2* que da comienzo a la adquisición; si ésta se realizó en forma exitosa, la forma de la señal aparecerá en el monitor, de lo contrario aparecerán los posibles errores que encontró el programa durante la adquisición de la señal; cuando esto último suceda, se deben verificar los siguientes puntos:

- 1.- Revisar que la forma contenga todos los datos descritos.
- 2.- Revisar que el conector que une la tarjeta de adquisición y el osciloscopio digital esté correctamente instalado.
- 3.- Revisar que el número anotado en la forma de adquisición coincida con el número de la forma de onda almacenada en el osciloscopio.

Todas las formas de onda son almacenadas en la memoria RAM de la PC. Para conocer el tamaño en *bytes* de cada una de ellas se presiona la tecla *F8* y se tecléa el comando *DISP*. En la pantalla aparecerán dos opciones, si se selecciona la número dos se mostrarán los nombres de las formas de ondas existentes y el número de *bytes* de cada una de ellas.

Para poder manipular adecuadamente los archivos, es necesario escribirlos en disco duro. Para ello, se tecléa nuevamente *F8*, se escribe el comando *WRITE* y se presiona la tecla *ENTER*; al hacerlo aparecerá en pantalla la forma en la que se proporcionarán los datos necesarios para la escritura en disco duro de las señales almacenadas en la memoria RAM. La forma tiene la siguiente estructura

```
Input Waveform or '*': NOMBRE
Output Waveform File Name: NOMBRE.WAV
Mode: ASCII
```

Fig B.2 Formato para guardar en disco las señales adquiridas

En el campo *NOMBRE* se anota el nombre de la forma de onda almacenada en RAM que se quiere escribir en el disco duro; si se tecléa un nombre incorrecto, el programa marcará un error. Es posible tecléar únicamente un asterisco, con lo que se indica que se crearán archivos en disco de cada una de las formas de onda existentes en RAM.

Al anotar el nombre y presionar la tecla *ENTER*, se introduce automáticamente, al campo iluminado asociado con *Output Waveform File Name*, el nombre mencionado anteriormente, con extensión *.WAV*; excepto si se eligió escribir todas las formas de ondas existentes; entonces aparecerá el letrero *ALL WAVEFORMS*.

El campo asociado a *Mode* se refiere al formato en que serán creados los archivos; existen dos formatos: ASCII y Binario. En este caso para transportar los datos a MATLAB, el formato adecuado es el ASCII. Una vez que la forma está correctamente escrita, se presiona la tecla *F2* para crear los archivos; una vez creados se presiona la tecla *F3* para regresar al menú principal y se escribe el comando *QUIT* para terminar la adquisición de las señales.

Los archivos creados aparecen en el directorio donde se encuentra el programa SPDMENU.BAT. Estos archivos deben ser modificados para su uso posterior con el paquete MATLAB; para ello, con algún editor, se deben modificar los archivos eliminando las líneas que aparecen al principio y los paréntesis que aparecen al final. Las comas que aparecen después de cada dato deben ser reemplazadas por espacios en blanco utilizando la opción de reemplazo del editor. Enseguida se presenta un ejemplo de un archivo sin modificar (lista B.1) y uno modificado (lista B.2).

```
ADIF = u3pt (STO (VER 1.0)
```

```
  IDE (
```

```
    NAM "STO5"
```

```
    DAT 1993-4-15
```

```
    TIM 12:01:11
```

```
    HIS "SPD V3.0
```

```
  )
```

```
  ENC (
```

```
    FOR SFP32
```

```
    HIR 3.4028235e+038
```

```
  )
```

```
  DIM = X1 (
```

```
    TYP IMP
```

```
    SCA = 0.004
```

```
    OFF 0
```

```
    SIZ 2048
```

```
    UNI "SECONDS"
```

```
  )
```

```
  DIM = Y2 (
```

```
    TYP EXP
```

```
    SCA 1
```

```
    OFF 0
```

```
    UNI "VOLTS"
```

```
  )
```

```
  TRA = SPD1 (
```

```
    IND (LABEL X1)
```

```
    DEP (LABEL Y2)
```

```
  )
```

DAT (CUR (VAL

1.97,
1.96,
1.97,
1.955,
1.955,
1.96)))

Lista B.1 Archivo sin modificar

1.97
1.96
1.97
1.955
1.955
1.96

Lista B.2 Archivo modificado

Antes de modificar los archivos, se debe anotar el dato que aparece en la línea que principia con la palabra *SCA*, incluida dentro del bloque que comienza con el letrero $DIM = X1$ (ver lista B.1), que corresponde al tiempo de muestreo del osciloscopio (T_{mc}), mismo que es necesario conocer para su uso dentro de la rutina de estimación en MATLAB (en el archivo de ejemplo de la lista B.1 el tiempo de muestreo es de 0.004s).

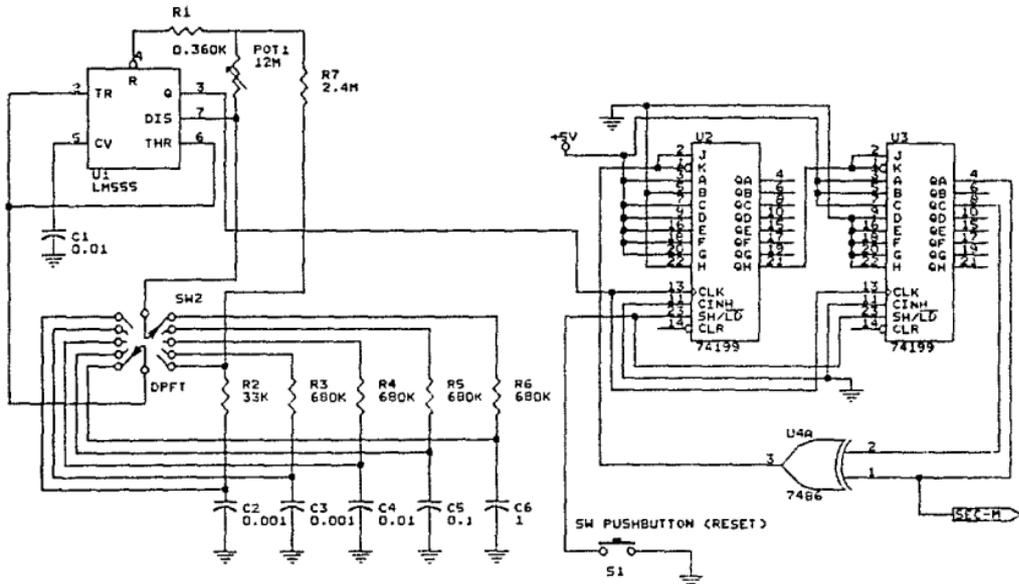
Cuando se tienen los archivos de las señales de entrada y de salida editados, es ya posible utilizarlos en la rutina de estimación implantada en MATLAB, para ello es necesario grabar los archivos en alguno de los subdirectorios desde donde el programa MATLAB puede ejecutar comandos.

APÉNDICE C

DATOS TÉCNICOS DEL GENERADOR SBS

En este apartado, se presenta la siguiente información del generador de secuencias binarias pseudoaleatorias (Generador SBS): diagramas electrónicos, de disposición, además de las listas de partes, ordenados de la siguiente forma:

	Pág
GENERADOR DE LA SECUENCIA-M	
- Diagrama Electrónico	57
- Lista de Partes	58
CIRCUITO DE CONTROL DE AMPLITUD Y ACOPLAMIENTO	
- Diagrama Electrónico	59
- Lista de Partes	60
FUENTES DE PODER	
- Diagrama Electrónico	61
- Lista de Partes	62
DIAGRAMA DE DISPOSICIÓN	63

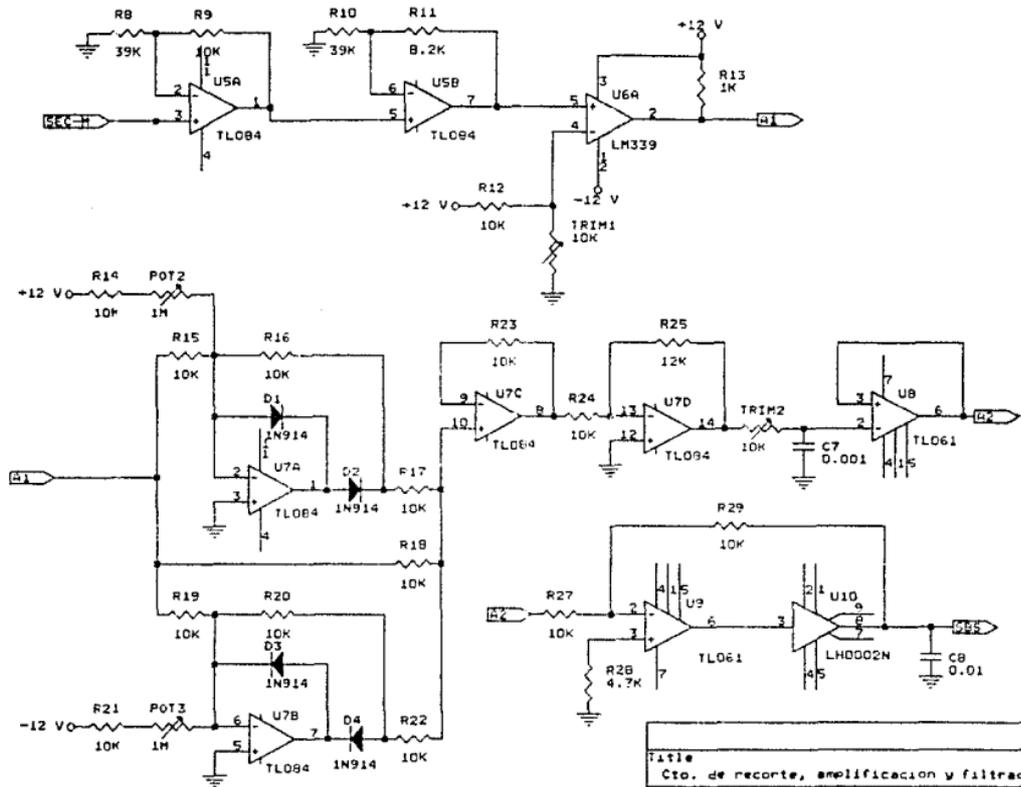


Title		Generador de la secuencia-M	
Size Document Number		REV	
A	1		
Date:	April 18, 1993	Sheet	1 of 3

LISTA DE PARTES DEL GENERADOR DE LA SECUENCIA-M

CONCEPTO	CANTIDAD	REFERENCIA	CARACTERÍSTICAS
1	2	C1,C4	0.01 μ F, 250V
2	2	C2,C3	0.001 μ F, 250V
3	1	C5	0.1 μ F, 250V
4	1	C6	1 μ F, 250V
5	1	POT1	12M Ω , \pm 5%, 1/2W
6	1	R1	0.360k Ω , \pm 5%, 1/2W
7	1	R2	33k Ω , \pm 5%, 1/2W
8	4	R3,R4,R5,R6	680k Ω , \pm 5%, 1/2W
9	1	R7	2.4M Ω , \pm 5%, 1/2W
10	1	S1	SW PUSHBUTTON
11	1	SW2	DPFT
12	1	U1	LM555
13	2	U2,U3	74199
14	1	U4	7486

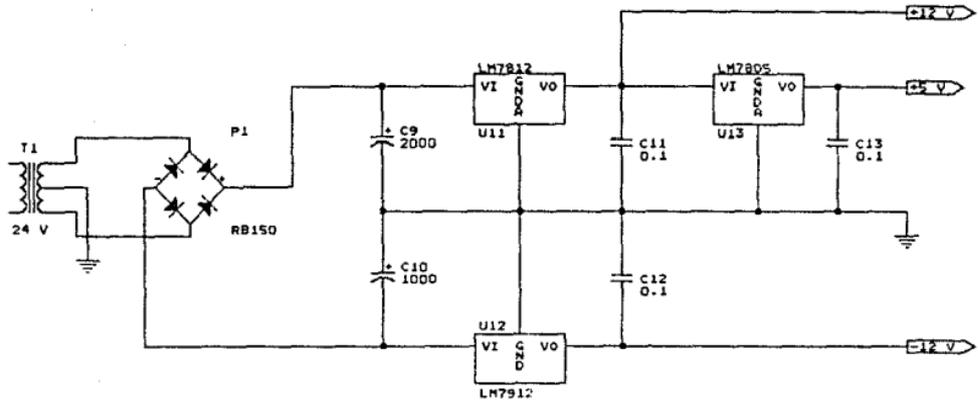
59



Title	
Cto. de recorte, amplificación y filtrado	
Size Document Number	
A	2
Date:	June 24, 1993 Sheet 2 of 3

**LISTA DE PARTES DEL CIRCUITO DEL CONTROL DE AMPLITUD Y
ACOPLAMIENTO**

CONCEPTO	CANTIDAD	REFERENCIA	CARACTERÍSTICAS
1	1	C7	0.001 μ F, 250V
2	1	C8	0.01 μ F, 250V
3	4	D1,D2,D3,D4	1N914
4	2	POT2,POT3	1M Ω , \pm 5%, 1/2W
5	2	R8,R10	39k Ω , \pm 5%, 1/2W
6	17	TRIM1,TRIM2,R9,R12, R14,R15,R16,R17,R18 R19,R20,R21,R22,R23 R24,R27,R29	10k Ω , \pm 5%, 1/2W
7	1	R11	8.2k Ω , \pm 5%, 1/2W
8	1	R13	1k Ω , \pm 5%, 1/2W
9	1	R25	12k Ω , \pm 5%, 1/2W
10	1	R28	4.7k Ω , \pm 5%, 1/2W
11	2	U5,U7	TL084
12	1	U6	LM339
13	2	U8,U9	TL061
14	1	U10	LH0002N



Title

Fuentes de poder

Size Document Number

A

3

REV

Date: April 18, 1993 Sheet 3 of 3

LISTA DE PARTES DE LAS FUENTES DE PODER

CONCEPTO	CANTIDAD	REFERENCIA	CARACTERÍSTICAS
1	1	C9	2000 μ F, 250V
2	1	C10	1000 μ F, 250V
3	3	C11,C12,C13	0.1 μ F, 250V
4	1	P1	RB150, 250V
5	1	T1	110V/24V, 500mA
6	1	U11	LM7812
7	1	U12	LM7912
8	1	U13	LM7805

APÉNDICE D

LISTADOS DE PROGRAMAS

ÍNDICE DE PROGRAMAS

	Pág
IDENSIS.M	65
IDENLIST.	70
MPARLIST.M	70
ARX11.M	71
ARMAX11.M	74
OE11.M	75
BJ11.M	76
IV411.M	77
VALMOD.M	81

IDENSIS.M

```
% OPCION DEL ANALISIS NO PARAMETRICO
d = 0;
while (d < 1)|(d > 2)
    clc
    disp('');
    disp('');
echo off
clc, clg
help idenlist
pause
clc
m=3;
disp('');
disp(' Espere un momento...');
while m == 3
    clc
    disp('          OBTENCION DE LOS DATOS EXPERIMENTALES');
    disp('');
    disp('Dé los nombres de los archivos de datos de entrada y salida entre');
    disp('apóstrofes y sin ninguna extensión:');
    disp('');
    entrada=input(' Nombre del archivo de los datos de la entrada: ');
    salida = input(' Nombre del archivo de los datos de la salida: ');

eval(['load ', entrada, '.wav']);
eval(['load ', salida, '.wav']);

% El intervalo de muestreo es pedido
disp('');
T=input(' Dé el intervalo de muestreo T del osciloscopio [segundos] = ');
clc
disp('');
disp('');
disp('A continuación se grafican todos los puntos que constituyen los');
disp('archivos de la entrada y de la salida');
disp('');
disp('');
disp('');
disp('          Presione cualquier tecla para continuar');
pause
% Se grafican todos los puntos que constituyen la muestra
Ntotal = length(eval(entrada));
z21=[eval([salida,'(1:Ntotal)']) eval([entrada,'(1:Ntotal)'])];
```

```
idplot(z21, [], T), xlabel('TIEMPO [s]'),  
pause
```

% Se piden los datos que serán utilizados para la obtención del modelo y
% los que serán utilizados para la validación

```
clc  
disp('');  
disp('');  
disp('El número total de puntos que constituyen a cada uno de los archivos');  
fprintf('es de %g.\n', Ntotal);  
disp('');  
disp('Del total de puntos, es recomendable utilizar una parte para la');  
disp('estimación y otra para la validación.')
```

```
c=0;  
while c==0,  
    disp('');  
    disp('');  
    disp('Indique los valores inicial y final de los puntos a ser utilizados para');  
    disp('la estimación:');  
    disp('');  
  
    n1=input(' Valor inicial = ');  
    n2=input(' Valor final = ');  
    if (n1 <= 0)|(n2 > Ntotal)|(n1 > n2),  
        disp('');  
        disp(' ERROR fuera del intervalo permitido');  
        c=0; disp(' Presione cualquier tecla para continuar'); pause  
    else  
        c=1;  
    end  
end  
clc  
end
```

```
c=0;  
while c==0,  
    disp('');  
    disp('');  
    disp('Indique los valores inicial y final de los puntos a ser utilizados');  
    disp('para la validación:');  
    disp('');  
    disp('');  
    n11=input(' Valor inicial = ');  
    n22=input(' Valor final = ');  
    if (n11 <= 0)|(n22 > Ntotal)|(n11 > n2),  
        disp('');  
    end  
end
```

```

disp(' ERROR, fuera del intervalo permitido');
c=0; disp(' Presione cualquier tecla para continuar'); pause
else
c=1;
end
clc
end
z2=[eval([salida,'(n1:n2)']) eval([entrada,'(n1:n2)'])];

% Se grafican los datos elegidos
disp('');
disp('');
disp('A continuación se muestran los datos elegidos para la estimación');
disp('');
disp('');
disp('                Presione cualquier tecla para continuar');
pause

idplot(z2,[], T),xlabel('TIEMPO [s]'),
pause
clc

% Se eliminan los valores de dc y se hace la media de los valores igual a cero
disp('');
disp(' Espere un momento...');
z2=dtrend(z2);
disp(' El análisis no paramétrico da una idea de la dinámica del sistema');
disp(' a identificar. ');
disp('');
disp(' ¿ Desea obtener un análisis no paramétrico de los datos ?');
disp('');
disp('                1) SI');
disp('                2) NO');
disp('');
d = input(' opción: ');
end

if d == 1,
clc
disp('                ANALISIS NO PARAMETRICO DE LOS DATOS');
disp('');
disp(' El programa desarrolla un análisis de correlación y se estima con');
disp(' éste la respuesta al impulso. A partir de lo anterior, se obtienen');
disp(' las funciones de covariancia. Posteriormente, a partir de la respuesta');
disp(' al impulso estimada se obtiene la respuesta al escalón estimada. ');

```

```

disp('');
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

clc
disp('Espere un momento ...');
ir=cra(z2); pause

clc
% Se obtiene una aproximación de la respuesta al escalón del sistema
disp('');
disp('');
disp(' A continuación se muestra la respuesta al escalón estimada. En primer');
disp(' lugar, aparece con el eje de las abscisas en términos del número de');
disp(' periodos de muestreo, lo cual da una idea del retardo que presenta');
disp(' el sistema. ');
disp('');
disp(' Después aparece la misma repuesta al escalón, pero en el eje de las');
disp(' abscisas aparece el tiempo, esto da una idea de la constante de tiempo');
disp(' dominante del sistema. ');
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

stepr = cumsum(ir);
plot(stepr), title('RESPUESTA AL ESCALON ESTIMADA'),
xlabel('PERIODOS DE MUESTREO'), ylabel('AMPLITUD')
pause
t=1:length(stepr);
plot(t*T,stepr), xlabel(' TIEMPO [s]'), ylabel(' AMPLITUD')
title('RESPUESTA AL ESCALON ESTIMADA');
grid
pause
clc
disp('');
disp(' Espere un momento..');

[gs,fiv] = spa(z2);
gs = sett(gs,T);
clc
disp('');
disp('');

```

```

disp(' A continuación se muestra la respuesta en frecuencia del sistema,');
disp(' obtenida a través de un análisis espectral de los datos. ');
disp('');
disp('');
disp('                               Presione cualquier tecla para continuar');
pause
bodeplot(gs)
pause
clc

else
end

m=2;
% A continuación se selecciona la estructura del modelo a utilizar
while m==2
    clc,clg,
    help mparlist
    k=input('Seleccione el número de la estructura elegida: ');
    m=1;
    while m==1,
        if k==1, arx11,    valmod,else, end
        if k==2, armax11, valmod,else, end
        if k==3, oe11,    valmod,else, end
        if k==4, bj11,    valmod,else, end
        if k==5, iv411,   valmod,else, end

    while(1)
        clc
        disp('');
        disp('                               Se tienen las siguientes opciones:');
        disp('');
        disp(' 1) Cambiar el número de parámetros a ser estimado');
        disp(' 2) Probar con otra estructura de modelo ');
        disp(' 3) Identificar otro proceso');
        disp(' 4) Abandonar el programa');
        disp('');
        disp('');
        m = input('    opción: ');
        if (m >= 1) |(m <= 4), break , end
    end
end
end
end
end

```

IDENTLIST.M

```
% El problema de identificación consiste en inferir modelos matemáticos
% de sistemas a partir de datos experimentales.
%
% Esta rutina permite aplicar los principales métodos de identificación
% tanto paramétricos como no paramétricos para la obtención de modelos
% de sistemas dinámicos. Los sistemas que pueden ser identificados son del
% tipo lineal.
%
% Los sistemas a identificar son sistemas SISO (Single Input-Single Output,
% una entrada-una salida).
%
%
% Presione cualquier tecla para continuar
```

MPARLIST.M

```
% SELECCION DE LA ESTRUCTURA PARA EL MODELO PARAMETRICO
%
% A continuación se listan las estructuras de los modelos utilizados
% en este programa para la identificación de sistemas dinámicos SISO.
%
% Las primeras cuatro estructuras utilizan un MÉTODO DE PREDICCIÓN DE
% ERROR, la última (IV4) es una estructura ARX que utiliza un MÉTODO
% DE VARIABLE INSTRUMENTAL.
%
% Si se desea tener cierta idea acerca del orden del retardo, es preferible
% en primer término; utilizar la estructura ARX o la IV4.
%
% ** ESTRUCTURAS DE LOS MODELOS **
%
% 1) ARX
% 2) ARMAX
% 3) OE
% 4) BJ
% 5) IV4
%
```

ARX11.M

```
echo off
clc
disp('          ESTRUCTURA DE MODELO ARX');
disp('');
disp('    ARX son las siglas de "AutoRegression with eXternal input"');
disp('    Esto es, una autorregresión con entrada externa. ');
disp('    La estructura del modelo es de la forma:');
disp('');
disp('           $A(q)*y(t) = B(q)*u(t-nk)T + e(t)'$ );
disp('');
disp('    ó en forma explícita:');
disp('');
disp('     $y(t) + a_1*y(t-T) + \dots + a_{na}*y(t-na)T = b_1*u(t-nk)T + \dots'$ );
disp('           $+ b_{nb}*u(t-(nb+nk-1)T) + v(t)'$ );
disp('');
disp('    Los parámetros a ser estimados son a1,a2,...,ana,b1,b2,...,bnb');
disp('');
disp('    Donde:');
disp('    T representa el tiempo de muestreo de las señales');
disp('    nk representa el número de periodos de muestreo de retraso que presenta ');
disp('    la señal de salida con respecto a la entrada');
disp('    na representa el orden del polinomio A')
disp('    nb representa el orden del polinomio B')
disp('');
disp('');
disp('          Presione cualquier tecla para continuar');
pause

ze=[eval([salida,'(1:(Ntotal/2))']) eval([entrada,'(1:(Ntotal/2))'])];
zr=[eval([salida,'((Ntotal/2 + 1):Ntotal)']) eval([entrada,'((Ntotal/2 + 1):Ntotal)'])];
ze = dtrend(ze); zr = dtrend(zr);

clc
disp('');
disp('El número de polos del modelo está dado por:');
disp('');
disp('          na ');
disp('');
disp('El número de ceros está dado por la siguiente relación:');
disp('');
disp('          número de ceros = (nb-1)');
disp('');
disp('');
```

```

na = input(' Dé el orden del polinomio A (na) = ');
nb = input(' Dé el orden del polinomio B (nb) = ');
clc
disp('');
disp(' Espere un momento...');

% Se estima el orden (periodos de muestreo) del retardo con na y nb indicados
V = arxstruc(ze,zr,estruc(na,nb,1:10));
[nn1,Vm] = selstruc(V,0);
clc
disp('');
fprintf(' El orden del retardo con los valores de na= %g y nb= %g es\n',na,nb);
disp(' el siguiente:');
disp('');
fprintf(' nk = %g\n',nn1(3))
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

clc
disp('');
disp(' Realizando una estimación de na y nb, espere un momento...');
V = arxstruc(ze,zr,estruc(1:10,1:10,nn1(3)));
nn = selstruc(V,0);
clc
disp('');
disp(' La mejor estimación de na y nb considerando el orden del retardo');
fprintf(' anterior (nk = %g) es la siguiente:\n', nn(3))
disp('');
fprintf(' na = %g\n',nn(1))
fprintf(' nb = %g\n',nn(2))
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause
l=0;
while ( l < 1)|(l > 2)
    clc
    disp('');
    disp(' Las gráficas de las funciones de pérdida indican el número de parámetros');
    disp(' necesarios para tener funciones de pérdida menores. ');
    disp('');
    disp(' ¿ Desca ver las funciones de pérdida ?');
    disp('');

```

```

disp('                1) SI');
disp('                2) NO');
disp('');
disp('');
l=input('    opción: ');
end
if l == 1
    clc
    disp('');
    disp(' A continuación se presenta la gráfica de las funciones de pérdida');
    disp(' para diferentes valores de na y nb. ');
    disp('');
    disp(' A partir de esta información se indica el número de parámetros a ser');
    disp(' estimados, que será igual a la suma de na y nb. ');
    disp('');
    disp(' Si sólo se presiona "ENTER" se utilizarán los valores por omisión');
    disp(' que corresponden a na, nb y nk estimados. ');
    disp('');
    disp('');
    disp('');
    disp('                Presione cualquier tecla para continuar');
    pause
    clc

    nns = selstruc(V)
    disp('');
    disp('');
    disp('                Presione cualquier tecla para continuar');
    pause
    clc
else
end

% Se construye el modelo
if l == 1
    th=arx(z2,nns);
else
    p=0;
    while (p < 1) | (p > 3)
        clc
        disp('');
        disp('');
        disp(' Para la construcción del modelo ARX se tienen las opciones ');
        disp(' siguientes:');
        disp('');

```

```

fprintf('      1) {na nb nk} propuestos : [%g %g %g]\n', nn1(1), nn1(2), nn1(3))
fprintf('      2) {na nb nk} estimados : [%g %g %g]\n', nn(1),nn(2),nn(3))
fprintf('      3) Otros valores de {na nb nk}\n');
disp('');
disp('');
p=input(' opción: ');
end
if p==1
    th=arx(z2,nn1);
elseif p==2
    th=arx(z2,nn);
else
    clc
    disp('');
    na2=input(' Dé el orden del polinomio A (na) = ');
    nb2=input(' Dé el orden del polinomio B (nb) = ');
    nk2=input(' Dé el orden (periodos de muestreo) del retardo (nk) = ');
    th=arx(z2,[na2 nb2 nk2]);
end
end

```

ARMAX11.M

```

echo off
clc
disp('          ESTRUCTURA DE MODELO ARMAX');
disp('');
disp(' ARMAX son las siglas de "AutoRegressive Moving Average with eXternal');
disp(' input");
disp(' La estructura del modelo es de la forma:');
disp('');
disp('y(t) + a1*y(t-T) + ... + ana*y(t-(na)T)');
disp('=b1*u(t-(nk)T)+...+ bnb*u(t-(nb+nk-1)T) + v(t) + c1v(t-T) + ...+cncv(t-');
disp(' (nc)T)');
disp('');
disp(' Los parámetros a ser estimados son a1,a2,...,ana,b1,b2,...,bnb,');
disp(' c1,c2,...,cnc.'):
disp('');
disp(' Donde:');
disp(' T representa el tiempo de muestreo de las señales');
disp(' nk representa el número de periodos de muestreo del retraso que presenta. ');
disp(' la salida con respecto a la entrada');
disp(' na representa el orden del polinomio A')

```

```

disp(' nb representa el orden del polinomio B')
disp(' nc representa el orden del polinomio C');
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

```

```

clc
disp('');
disp('');
disp(' El número de polos del sistema corresponde a:');
disp('');
disp(' na ');
disp('');
disp(' El número de ceros está dado por la siguiente relación:');
disp('');
disp(' número de ceros = nb - 1');
disp('');
na = input('Dé el orden del polinomio A (na) = ');
nb = input('Dé el orden del polinomio B (nb) = ');
nc = input('Dé el orden del polinomio C (nc) = ');
nk = input('Dé el orden (periodos de muestreo) del retardo (nk) = ');

```

```

clc

```

```

% Se construye el modelo

```

```

th=armax(z2,[na nb nc nk]);

```

OE11.M

```

echo off
clc
disp(' ESTRUCTURA DE MODELO OE');
disp('');
disp(' OE son las siglas de "Output Error");
disp(' La estructura del modelo es de la forma:');
disp('');
disp(' y(t) = [B(q)/F(q)]*u(t-(nk)T) + v(t)');
disp('');
disp(' Los parámetros a ser estimados son b1,b2,...,bnb,f1,f2,...,fnf');
disp('');
disp(' Donde:');

```

```

disp(' T representa el tiempo de muestreo de las señales');
disp(' nk representa el número de periodos de muestreo de retraso que presenta');
disp(' la salida con respecto a la entrada');
disp(' nb representa el orden del polinomio B')
disp(' nf representa el orden del polinomio F')
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

```

```

clc
disp('');
disp('');
disp(' El número de polos del modelo está dado por:');
disp('');
disp(' nf ');
disp('');
disp(' El número de ceros está dado por la siguiente relación:');
disp('');
disp(' número de ceros = nb - 1');
disp('');
nb = input('Dé el orden del polinomio B (nb) = ');
nf = input('Dé el orden del polinomio F (nf) = ');
nk = input('Dé el orden (periodos de muestreo) del retardo (nk) = ');

```

% Se construye el modelo

```
th=oc(z2,[nb nf nk]);
```

BJ11.M

```

echo off
clc
disp(' ESTRUCTURA DE MODELO BJ');
disp('');
disp(' BJ son las siglas de "Box-Jenkins".');
disp(' La estructura del modelo es de la forma:');
disp('');
disp('  $y(t) = [B(q)/T(q)]*u(t - (nk)T) + [C(q)/D(q)]*v(t)$  ');
disp('');
disp(' Los parámetros a ser estimados son b1,b2,...,bnb,c1,c2,...,cnc');
disp(' d1,d2,...,dnd,f1,f2,...,fnf. ');
disp('');
disp(' Donde:');

```

```

disp(' T representa el tiempo de muestreo de las señales');
disp(' nk representa el número de periodos de muestreo de retraso que presenta');
disp(' la salida con respecto a la entrada');
disp(' nb,nc,nd y nf representan los órdenes de los polinomios B,C, D y F')
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

```

```

clc
disp('');
disp(' El número de polos del sistema está dado por:');
disp('');
disp('          nf ');
disp('');
disp(' El número de ceros está dado por la siguiente relación:');
disp('');
disp('          número de ceros = nb - 1');
disp('');
nb = input('Dé el orden del polinomio B (nb) = ');
nc = input('Dé el orden del polinomio C (nc) = ');
nd = input('Dé el orden del polinomio D (nd) = ');
nf = input('Dé el orden del polinomio F (nf) = ');
nk = input('Dé el orden (número de periodos) del retardo (nk) = ');

```

% Se construye el modelo

```
th=bj(z2,[nb nc nd nf nk]);
```

IV411.M

```

echo off
clc
disp('          ESTRUCTURA DE MODELO IV4');
disp('');
disp(' IV son las siglas de "Instrumental Variables");
disp(' La estructura del modelo es de la forma:');
disp('');
disp('           $A(q)*y(t) = B(q)*u(t-(nk)T) + v(t)'$ );
disp('');
disp(' ó en forma explícita:');
disp('');
disp('  $y(t) + a_1*y(t-T) + \dots + a_n*y(t-(n)T) = b_1*u(t-(nk)T) + \dots'$ );

```

```

disp('
+ bnb*u(t-(nb+nk-1)T) + v(t)');
disp('');
disp(' Los parámetros a ser estimados son a1,a2,...,ana,b1,b2,...,bnb');
disp('');
disp(' Donde:');
disp(' T representa el tiempo de muestreo de las señales');
disp(' nk representa el número de periodos de muestreo de retraso que presenta');
disp(' la salida con respecto a la entrada');
disp(' na representa el orden del polinomio A')
disp(' nb representa el orden del polinomio B')
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

ze = [eval([salida,'(1:(Ntotal/2))']) eval([entrada,'(1:(Ntotal/2))'])];
zr = [eval([salida,'((Ntotal/2 + 1):Ntotal)']) eval([entrada,'((Ntotal/2 + 1):Ntotal)'])];
zc = dtrend(ze); zr = dtrend(zr);

clc
disp('');
disp('El número de polos estimado esta dado por:');
disp('');
disp(' na');
disp('');
disp('El número de ceros está dado por la siguiente relación:');
disp('');
disp(' número de ceros = (nb-1)');
disp('');
disp('');
na = input(' Dé el orden del polinomio A (na) = ');
nb = input(' Dé el orden del polinomio B (nb) = ');
clc
disp(' Espere un momento...');

% Se estima el número de retardos con na y nb indicados
V = ivstruc(ze,zr,estruc(na,nb,1:10));
[nn1,Vm] = selstruc(V,0);
clc
disp('');
fprintf(' El mejor orden del retardo con los valores de na=%g y nb=%g es\n',na,nb);
disp(' el siguiente:');
disp('');
fprintf(' nk = %g\n',nn1(3))
disp('');

```

```

disp('');
disp(' Presione cualquier tecla para continuar');
pause

clc
disp('');
disp(' Realizando una estimación de na y nb, espere un momento...');
V = ivstruc(ze,zr, struc(1:10,1:10,nn1(3)));
nn = selstruc(V,0);
clc
disp('');
disp(' La mejor estimación de na y nb considerando el orden del retardo');
fprintf(' anterior (nk = %g) es la siguiente:\n', nn(3))
disp('');
fprintf(' na = %g\n',nn(1))
fprintf(' nb = %g\n',nn(2))
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause
l=0;
while ( l < 1 )|( l > 2)
    clc
    disp('');
    disp(' Las gráficas de las funciones de pérdida indican el número de parámetros');
    disp(' necesarios para tener funciones de pérdida menores. ');
    disp('');
    disp(' ¿ Desea ver las funciones de pérdida ?');
    disp('');
    disp(' 1) SI');
    disp(' 2) NO');
    disp('');
    disp('');
    l=input(' opción: ');
end
if l == 1
    clc
    disp('');
    disp(' A continuación se presenta la gráfica de las funciones de pérdida');
    disp(' correspondientes a determinados valores de na y nb. ');
    disp('');
    disp(' A partir de esta información se indica que orden de na y nb se desea');
    disp(' utilizar para la construcción del modelo paramétrico. Se debe teclear');
    disp(' el número de parámetros a ser estimado, que será igual a la suma de na');
    disp(' y nb. Si sólo se presiona "ENTER" se utilizarán los valores por omisión');

```

```

disp(' que corresponden a na, nb y nk estimados. ');
disp('');
disp('');
disp('                               Presione cualquier tecla para continuar');
pause
clc

nns = seistruc(V)
disp('');
disp('');
disp('                               Presione cualquier tecla para continuar');
pause
clc
else
end

% Se construye el modelo
if l == 1
    th = iv4(z2, nns);
else
    p = 0;
    while (p < 1) | (p > 3)
        clc
        disp('');
        disp(' Para la construcción del modelo IV4 se tienen las opciones ');
        disp(' siguientes: ');
        disp('');
        fprintf(' 1) [na nb nk] propuestos : [%g %g %g]\n', nn1(1), nn1(2), nn1(3))
        fprintf(' 2) [na nb nk] estimados : [%g %g %g]\n', nn(1), nn(2), nn(3))
        fprintf(' 3) Otros valores de [na nb nk]\n');
        disp('');
        disp('');
        p = input(' opción: ');
    end
    if p == 1
        th = iv4(z2, nn1);
    elseif p == 2
        th = iv4(z2, nn);
    else
        clc
        disp('');
        na2 = input(' Dé el orden del polinomio A (na) = ');
        nb2 = input(' Dé el orden del polinomio B (nb) = ');
        nk2 = input(' Dé el orden (periodos de muestreo) del retardo (nk) = ');
        th = iv4(z2, {na2 nb2 nk2});
    end
end

```

```
end
end
```

VALMOD.M

```
echo off
th=sett(th,T); % Se ajusta el factor de escala
clc
disp('');
disp('          PRESENTACIÓN DEL MODELO PARAMÉTRICO');
disp('');
disp(' A continuación se muestran las matrices que contienen la información');
disp(' del modelo paramétrico. El primer renglón de las matrices corresponde');
disp(' a los parámetros estimados, el segundo renglón contiene los valores de');
disp(' de la desviación estándar de cada uno de los parámetros estimados. ');
disp('');
disp(' El orden del retardo estimado está indicado con el número de ceros que');
disp(' aparecen al principio en el polinomio B. ');
disp('');
disp('');
disp('          Presione cualquier tecla para continuar');
pause
clc

present(th) % Se realiza la presentación del modelo obtenido
disp('');
disp('          Presione cualquier tecla para continuar');
pause
clc

zpth=th2zp(th); % Se obtienen los polos y ceros del sistema
zpplot(zpth), % Se grafican los polos y ceros del sistema
xlabel('POLOS Y CEROS DEL SISTEMA')
pause
clc
% A continuación se procede a validar el modelo obtenido
% a través de un análisis que se realizará con los puntos no utilizados
% Se eliminan los valores constantes y se hace la media igual a cero

u=dtrend(eval([entrada,'(n11:n22)'])); y=dtrend(eval([salida,'(n11:n22)']));

yh=idsim(u,th); % Simulación del modelo obtenido

disp('');
```

```

disp('                VALIDACIÓN DEL MODELO ');
disp(' A continuación se realiza una presentación de la salida simulada con el');
disp(' modelo obtenido y la salida medida para el juego de valores reservados');
disp(' para tal propósito. ');
disp('');
disp('');
disp('                Presione cualquier tecla para continuar');
pause

```

```

plot([y,yh]), title('VALIDACION DEL MODELO ESTIMADO');
xlabel('salida medida (rojo) salida del modelo estimado(verde)');
pause
clc

```

```

s=0;
while (s < 1) | (s > 2)
    clc
    disp('');
    disp(' ¿ Desea realizar un acercamiento de las gráficas mostradas ?');
    disp('');
    disp('                1) SI');
    disp('                2) NO');
    disp('');
    s=input(' opción: ');
end
if s == 1
    clc
    disp('');
    disp(' Elija con la flecha los valores mínimo y máximo del eje de las');
    disp(' abscisas. ');
    disp('');
    disp('');
    disp('                Presione cualquier tecla para continuar');
    pause
    clc
    [ex1 ey1]=ginput(1);
    hold on
    plot(ex1,ey1,'b+')
    [ex2 ey2]=ginput(1);
    plot(ex2,ey2,'b+')
    pause(2)
    hold off
    if ex2 < ex1
        ex11=ex1;
        ex22=ex2;

```

```

ex1=ex22;
ex2=ex11;
else
end
plot([y(fix(ex1):fix(ex2)), yh(fix(ex1):fix(ex2))])
title(' ACERCAMIENTO DE LAS GRAFICAS MOSTRADAS');
xlabel('salida medida(rojo) salida del modelo estimado(verde)'),
pause
else
end

clc
gth=th2ff(th); % Se obtiene la respuesta en frecuencia del modelo theta
if d == 2
    disp('');
    disp(' A continuación se muestra la respuesta en frecuencia del modelo ');
    disp(' paramétrico estimado. ');
    disp('');
    disp('');
    disp(' Presione cualquier tecla para continuar');
    pause
    bodeplot(gth), % Se grafica el diagrama de Bode
    pause
else
    disp('');
    disp(' A continuación se muestra la respuesta en frecuencia del modelo ');
    disp(' paramétrico estimado. ');
    disp('');
    disp('');
    disp(' Presione cualquier tecla para continuar');
    pause
    bodeplot(gth),
    pause
    clc
    disp('');
    disp(' A continuación se comparan los diagramas de Bode');
    disp(' obtenidos del análisis no paramétrico (señal roja) con');
    disp(' el obtenido con el modelo paramétrico (señal verde). ');
    disp('');
    disp('');
    disp(' Presione cualquier tecla para continuar');
    pause
    bodeplot([gs gth])
    pause
    clc

```

```

disp('');
disp(' A continuación se muestran las respuestas al escalón obtenidas con el');
disp(' el método no paramétrico (señal roja) y con el paramétrico (señal verde).');
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause

% se muestran las respuestas al escalón
stepr1=ones(length(stepr),1);
mstepr=idsim(stepr1,th);
plot([stepr,mstepr]), xlabel('PERIODOS DE MUESTREO')
ylabel('AMPLITUD');
pause
end

clc
% A continuación se calculan y grafican los residuos del modelo
disp('');
disp(' A continuación se presenta un análisis de los residuos relacionados con');
disp(' los datos experimentales (entrada y salida) y el modelo paramétrico');
disp(' elegido. ');
disp('');
disp(' En el análisis que se realiza, se considera que los residuos son ideal-');
disp(' mente blancos e independientes de la entrada u. A partir de lo anterior,');
disp(' se obtiene la función de autocorrelación de los residuos (e) y la corre-');
disp(' lación entre los residuos e y la entrada u. ');
disp('');
disp(' En estas gráficas se muestran intervalos de confianza de las variables');
disp(' analizadas. Si las funciones de correlación (en su gran mayoría) quedan');
disp(' fuera de estos intervalos de confianza, no se debe aceptar el modelo y');
disp(' se deben elegir otros parámetros u otra estructura de modelo. ');
disp('');
disp('');
disp(' Presione cualquier tecla para continuar');
pause
clc
disp('');
disp(' Espere un momento... ');
e = resid(z2, th);
pause
plot(e), title('RESIDUOS'),
pause

```

```

clc
disp('');
disp(' Espere un momento...');
[nz dz] = th2tf(th);
clc
disp('');
disp(' La función de transferencia del sistema en el dominio de z es:');
disp('');
printsys(nz,dz,'z')
disp('');
disp('');
disp('          Presione cualquier tecla para continuar');
pause
clc
disp('');
disp(' Espere un momento...');
[nc dc] = d2cm(nz,dz,T,'tustin');
clc
disp('');
disp(' La función de transferencia del sistema en el dominio de s es:');
disp('');
printsys(nc,dc,'s')
disp('');
disp('');
disp('          Presione cualquier tecla para continuar');
pause
clc
disp('');
disp(' Con la función de transferencia anterior, se obtiene la respuesta al');
disp(' escalón del modelo obtenido. ');
disp('');
disp('');
disp('          Presione cualquier tecla para continuar');
pause

step(nc,dc)
grid
title('RESPUESTA AL ESCALON DEL MODELO EN EL DOMINIO DE S'),
pause
clc

```