

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

“Índice de la Calidad de la Identificación”

Autor: Ailet Abreu López

Tutor(es): Msc. Iván Iglesias Navarro

Msc. María del Carmen Hernández Caruz

Santa Clara

2013

"Año 55 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

“Índice de la Calidad de la Identificación”

Trabajo de Diploma presentado en opción al Título Académico de
Ingeniero en Automática

Autor: Ailet Abreu López

email: ailet@uclv.edu.cu

Tutor(es): Msc. Iván Iglesias Navarro

email: iglesias@uclv.edu.cu

Msc. María del Carmen Hernández Caruz

email: carmen@uclv.edu.cu

Santa Clara

2013

"Año 55 de la Revolución"



Hago constar que el presente Trabajo de Diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Automática, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Ailet Abreu López
Autor

Fecha

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Ailet Abreu López
Autor

Fecha

Boris Luis Martínez Jiménez, Dr.C.
Jefe del Departamento

Fecha

Responsable ICT o J de Carrera (Dr.C., Msc. o Ing.)
Responsable de Información Científico-Técnica

Fecha

PENSAMIENTO

“Es gracia divina empezar bien.

Gracia mayor persistir en el camino correcto

Pero la mayor gracia de las gracias es no desistir nunca”.

DEDICATORIA

A mami y papi, por la VIDA.

A mis hermanos, los mejores que alguien pudiera tener.

A Jorge, por estar todo este tiempo a mi lado apoyándome.

A Liz Mariam, Eduardito y Luis Ernesto, por ser los mejores regalos.

AGRADECIMIENTO

A:

Mami y Papi, por el ejemplo que me han dado y por el apoyo en la realización de este trabajo.

Mis hermanos y el resto de mi familia, por su inmenso cariño.

Jorge, por ser él, y por representar tanto para mí.

Los Royas, por ser más que amigos, mis hermanos.

Iván, por la presencia siempre que lo he necesitado.

Alejandro, por los desvelos.

Carmen y Samy, por la amistad de siempre.

Mis amistades del grupo; Dianelis, Ana, Leisis, Magdiel, Rolando, Pineda, por las historias juntos.

Los profesores del departamento de Automática, por estos años de guía.

Mary, Diamir, Luis Martínez, Miguel y Alain, por ser la otra parte de mi familia.

RESUMEN

En la práctica ingenieril la utilización de modelos es una necesidad insoslayable, ya que permite el análisis y diseño de sistemas sin necesidad de utilizar la instalación real. Existen diferentes formas de obtener un modelo a partir de datos reales de un proceso o sistema, de estos los más utilizados son los denominados modelos paramétricos, que describen las relaciones entre las variables del sistema mediante expresiones matemáticas.

En todo proceso de identificación es conveniente probar varias estructuras y diferentes órdenes dentro de cada estructura, hasta obtener la que mejor se ajuste a los datos obtenidos experimentalmente, proceso que se conoce como validación del modelo.

Matlab, mediante su herramienta *Ident*, permite solamente la validación a través del cálculo de diferentes criterios gráficos. Por lo que se lleva a cabo la presente investigación para desarrollar una interfaz gráfica capaz de calcular y mostrar (de manera muy sencilla) criterios numéricos que apoyen al proceso de selección de estructuras identificadas. Tales como: *Cuadrados medios del residuo (o del error) (CME)*, *Criterio de información de Akaike (AIC)*, *Criterio de información bayesiano (BIC)*. *Coefficiente de determinación (R^2)*. Por otro lado se describen las pruebas realizadas para la validación de la efectividad de la interfaz, evidenciando muy buenos resultados en su funcionamiento. También se probó con datos reales, obtenidos de la planta de tratamiento de residuales de la refinería Camilo Cienfuegos, con buenos resultados en lo que a la obtención de los valores de los criterios de bondad se refiere.

INDICE:

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTO	iii
RESUMEN	iv
INTRODUCCIÓN	7
CAPÍTULO 1. Identificación de Sistemas, importancia del control de la calidad de la identificación	13
1.1 Modelo de un sistema.....	13
1.1.1 Métodos de obtención de modelos.....	14
1.2 Identificación de sistemas.....	15
1.3 Pasos de la identificación.....	15
1.3.1 Métodos de identificación.....	16
1.3.1.1 Identificación paramétrica.....	17
1.3.1.2 Tipos de modelos paramétricos.....	17
1.3.1.3 Métodos para el ajuste de parámetros:.....	21
1.3.1.4 Validación de modelos:.....	22
1.4 Interfaces Gráfica	30
1.4.1 MatLab como herramienta en el desarrollo de aplicaciones automáticas y el diseño de interfaces gráficas	31
1.4.2 Herramienta GUIDE del MatLab.....	31
1.4.3 Ventajas de la herramienta GUIDE.....	32
1.5 Consideraciones finales del capítulo:.....	32
CAPÍTULO 2: <i>BondTool</i> : Interfaz Gráfica para el Cálculo de la Bondad de Ajuste.	34
2.1. Generalidades de la aplicación <i>BondTool</i>	36
2.2. Componentes de la aplicación <i>BondTool</i>	37
2.3. Paneles de la aplicación, componentes, funcionalidad y validaciones.....	38
2.3.1. Panel “Iniciar”.....	38
2.3.1.1. Validaciones del panel “Iniciar”.	39
2.3.2. Panel “Criterios”	40
2.3.3. Panel “Estimación”	41
2.3.3.1. Validaciones del panel “Estimación”	42
2.3.4. Panel “Calcular”	44

2.3.4.1. Funcionalidad del botón “Calcular”	45
2.3.4.2. Validaciones del botón “Calcular”	48
2.4. Ventana “Resultados”, funcionalidad	48
2.4.1. Funcionalidad de la ventana “Resultados”	49
2.5. Consideraciones finales del capítulo	50
CAPÍTULO 3: Análisis de los Resultados	51
3.1. Características generales del proceso de Floculación y de los datos.....	51
3.2. Validación de la interfaz	54
3.2.1. Validación del panel “Iniciar”	54
3.2.2. Validación de los criterios de información AIC, BIC, R^2 , CME.....	55
3.3. Pruebas.....	57
3.4. Análisis económico.....	62
3.5. Consideraciones parciales del capítulo.	63
CONCLUSIONES GENERALES	64
RECOMENDACIONES	65
REFERENCIAS BIBLIOGRÁFICAS	66
ANEXOS	67

INTRODUCCIÓN

El diseño de un controlador, requiere de un modelo de la planta a controlar que caracterice su comportamiento dinámico, que a su vez permita realizar y validar, mediante simulación el ajuste de los parámetros del controlador. Uno de los métodos existentes es el de Identificación Experimental, la cual permite obtener el modelo de un sistema a partir de datos reales obtenidos de la planta bajo estudio.

Se pueden encontrar en la práctica múltiples formas de identificar un sistema y obtener el modelo correspondiente. Cada método ofrecerá un modelo diferente que puede ser validado mediante técnicas específicas. En la práctica la validación se hace por el investigador lo que lleva implícito un grado de subjetivismo al decidir si es adecuado o no para una aplicación dada.

La identificación experimental surge con el propósito de modelar sistemas complejos tanto por sus parámetros variantes e incertidumbres presentes. La obtención de los modelos trae consigo la necesidad de analizar, luego de validar los modelos, cuál de ellos es el más indicado para la aplicación específica, atendiendo a parámetros inherentes a estos modelos.

En el mundo se trabaja en torno a este tema para lograr obtener los mejores modelos tal es el caso del trabajo realizado por A. Ritter del departamento de Ingeniería, Producción y Economía Agraria de la Universidad de La Laguna en Tenerife, en colaboración con la Universidad de la Florida y el departamento de Suelos y Riegos del Instituto Canario de Investigaciones Agrarias, en el cual tratan la “Capacidad de predicción de modelos aplicados a las zonas no saturadas del suelo (ZNS): Herramientas Informáticas para la adecuada Evaluación de la Bondad de Ajuste con significación estadística”. En este se desarrolla un procedimiento práctico para la evaluación de la capacidad de predicción de modelos basado en la representación gráfica del ajuste y en el cálculo de un índice relativo y un estimador del error medio, así como instrucciones para detectar la idoneidad del valor calculado según su sensibilidad a valores atípicos en las magnitudes de los datos.

En esta rama la MSc. Sandra L. Posada y DrC. Ricardo Rosero Noguera, ambos pertenecientes al Grupo de Investigaciones en Ciencias Agrarias de la Universidad de Antioquia, Medellín, Colombia, desarrollaron el artículo “Comparación de modelos

matemáticos: una aplicación para la evaluación de comida animal”, Aquí se presentan algunos modelos matemáticos empleados para describir las curvas de producción de gases y las herramientas estadísticas que sirven para evaluar su capacidad de ajuste. Para ajustar los perfiles de producción de gases usaron dos modelos uno logístico y otro empírico, y 6 criterios para evaluar su capacidad de ajuste, entre ellos; el cuadrado medio del error, el *criterio de información de Akaike (AIC)*, el *criterio de información Bayesiano (BIC)*, el *coeficiente de determinación (R^2)*, el *análisis de los Residuos* y la *dócima de Durban-Watson (DW)*.

El grupo de investigación anteriormente mencionado, ha desarrollado otras investigaciones entre ellas “comparaciones de modelos matemáticos para describir curvas de lactancia en cabras Sannen y Alpina”, estos se evalúan en cuanto a su capacidad de cumplir con los objetivos planteados. Los criterios de evaluación de los modelos fueron: *Suma Cuadrado del Error (SQE)*, los *criterios de información de Akaike y Bayesiano (AIC y BIC)*, respectivamente), el *coeficiente de determinación (R^2)* y el *análisis de sus residuos*.

Mantener un control de la medición de pH en una industria tiene gran importancia debido a sus constantes variaciones y la necesidad de obtener un producto de elevada calidad, es por ello que Pablo Parra Rosero desarrollo una investigación, para la cual usando algoritmos en Matlab generó una serie de modelos por identificación para un proceso de medición de pH, partiendo de los datos obtenidos experimentalmente desde un simulador de donde obtuvo tres modelos *OE*, *ARMAX* y *ARX*, a estos se les realizó la validación cruzada y el análisis comparativo de los resultados para seleccionar el modelo que mostrara un mejor comportamiento dinámico del sistema. La selección se realizó partiendo de criterios comparativos expuestos por el autor.

En Argentina, específicamente en la Facultad de Ingeniería de la Universidad Nacional de Lomas de Zamora, Claudia Minnaard, trató el tema de los modelos de regresión lineales y no lineales y su aplicación en problemas de ingeniería. En este se describe una aplicación, utilizando el software XLStat, asimismo se describe el intervalo de confianza para el coeficiente de regresión en el modo lineal, para la ordenada al origen y para la imagen a través de la recta. Para los modelos lineales se prueba la bondad del ajuste realizado a través de las pruebas específicas.

J. F. Naranjo, C. A. Cuartas y H. J. Correa, de la Universidad Nacional de Colombia realizaron una comparación de cuatro modelos matemáticos para la caracterización de la cinética de degradación ruminal de algunos recursos forrajeros. Evaluando la bondad de ajuste de los cuatro modelos no lineales y basándose en los criterios de selección de suma de cuadrados del error de predicción, el cuadrado medio del error de predicción, el error medio de predicción, y el coeficiente de determinación para cada modelo, se obtuvo el modelo deseado.

En Cuba, se han realizado trabajos como es el caso de la Identificación del sistema de control del nivel en molde para el vaciado continuo de acero en la Empresa de Aceros Inoxidables ACINOX de Las Tunas, desarrollado por los Ing. Guillermo González Yero y Vladimir Góngora Carreras ambos pertenecientes a dicha empresa y con la colaboración de la DrC. Mercedes Ramírez Mendoza perteneciente al Departamento de Control Automático de la Facultad de Ingeniería Eléctrica de la Universidad de Oriente. En este trabajo se llega a una serie de modelos los cuales son analizados mediante criterios de ajuste para escoger el modelo óptimo. No obstante se debe señalar que el uso de un solo criterio impide hacer una comparación en la cual se analicen varios indicadores y así obtener un modelo lo más real posible.

El Instituto de Ciencia Animal en conjunto con la Universidad Agraria de la Habana ha trabajado en la aplicación de Criterios para la selección de modelos estadísticos en la investigación científica, desarrollado por Caridad Guerra, A. Cabrera y Lucía Fernández. Para realizar una valoración ajustaron modelos polinomiales de primer a cuarto grado, luego aplicaron criterios de ajuste como R^2 , el *método de los residuos*, mediante los cuales se llegó al modelo indicado.

En todos los casos analizados la selección del modelo más adecuado se realiza mediante el análisis de diferentes criterios individuales y en algunas ocasiones partiendo del comportamiento comparativo del modelo y del sistema real.

Después de realizada una búsqueda exhaustiva, relacionada con la identificación de sistemas, se puede apreciar, que no se ha encontrado referencias que trabajen a partir de la comparación de diferentes modelos utilizando un índice de comportamiento que pondere

figuras de merito de cada uno de ellos. Es por ello necesario acometer la tarea de encontrar relaciones que permitan determinar cuál es entre diferentes modelos de un mismo sistema el más idóneo para una aplicación específica.

Sucede que en la actualidad y asociado a *MatLab* solo se cuenta con criterios gráficos implementados en la herramienta *Ident* que tienen que ser obligatoriamente analizados por un humano como la *respuesta al paso*, *respuesta de frecuencia*, *residuos* por mencionar algunos; por lo que se hace necesario, para la obtención del índice antes mencionado, el cálculo de otros criterios con el uso de *MatLab* que midan la bondad de ajuste de las estructuras modeladas o identificadas y que puedan servir en un futuro para la definición de índice de calidad de la identificación.

Teniendo en cuenta las observaciones anteriores, se define como **situación problemática**:
¿Cómo calcular y mostrar en *MatLab* criterios que midan la bondad de ajuste de las diferentes estructuras obtenidas en un proceso de *identificación de sistemas*?

A partir de la situación planteada, surgen entonces las siguientes **interrogantes científicas**:

- ✓ ¿Cómo se realiza el proceso de identificación en la actualidad?
- ✓ ¿Cómo se lleva a cabo el proceso de selección del mejor de los modelos dentro de todos los obtenidos?
- ✓ ¿Qué criterios utilizar para medir de forma numérica la bondad de ajuste de un modelo?
- ✓ ¿Cómo lograr el cálculo de criterios que midan la bondad de ajuste de un modelo?
- ✓ ¿Cómo mostrar a través de una interface los criterios calculados?

Para dar solución a la situación problemática, se plantea el siguiente **objetivo general**, Implementar una aplicación en *MatLab*, con su respectiva interfaz gráfica, que realice el cálculo de los criterios que miden la bondad de ajuste de modelos identificados, con el uso de la identificación paramétrica y específicamente las estructuras *ARX* y *ARMAX*. Y como **objetivos específicos**:

- ✓ Estudiar y sistematizar la bibliografía existente en el tema respecto a la contribución que hacen los indicadores de calidad.

- ✓ Realizar un estudio en profundidad de los indicadores actuales que se tienen en cuenta para la selección del mejor modelo obtenido durante el proceso de identificación.
- ✓ Implementar en *MatLab* los principales criterios que miden bondad de ajustes encontrados en la bibliografía.
- ✓ Diseñar una interfaz gráfica para mostrar los resultados de la aplicación de estos criterios a las estructuras *ARX* y *ARMAX* obtenidas para un modelo.

Como **posible impacto**, se pretende que la investigación contribuir al desarrollo de un programa, aplicable a la determinación ,en un futuro, mediante un índice de calidad que pondere en función de las necesidades de un usuario, cada uno de estos criterios calculados y así seleccionar el mejor modelo obtenido para una aplicación específica. La implementación de esta herramienta, hasta el momento, permitirá a los especialistas e investigadores seleccionar la mejor de las estructuras para cada criterio.

Con la aplicación del presente trabajo se facilita, como **posible resultado**, disponer de una herramienta computacional capaz de minimizar el esfuerzo requerido para la selección del modelo más adecuado para una aplicación dada.

Por otro lado los resultados de este trabajo poseen una **aplicabilidad** en el proceso de diseño de sistemas de control, utilizando los modelos que muestren en sus relaciones las mejores características.

La **viabilidad** es apreciable debido a que solo se requerirá del uso de fuentes bibliográficas y de computadoras con software disponibles en los laboratorios de la UCLV. Se disponen de bases de datos prácticos de procesos, que pueden ser utilizados en el desarrollo del trabajo.

Organización del informe

El informe final quedara estructurado de la siguiente forma: introducción, sucedida en el mismo orden por, capitulario conclusiones, recomendaciones, referencias bibliográficas y anexos.

Los temas que se abordaran en cada capítulo son:

CAPÍTULO I: Identificación de sistemas, importancia del control de la calidad de la identificación. En este se realizara un análisis de la literatura especializada en el tema y se abordaran los principales conceptos teóricos con los que se trabajaran.

CAPÍTULO II: Criterios para el análisis de la bondad de ajuste en la identificación de sistemas. Aquí se expondrá la implementación de los criterios utilizados en el desarrollo de la aplicación; entre ellos, coeficiente de determinación R^2 , *cuadrado medio del residuo o del error CME*, *criterio de información de Akaike AIC* y *criterio de información Bayesiano BIC*.

CAPÍTULO III: Análisis de los resultados. En este capítulo se realizara la simulación y validación, para medir la efectividad obtenida con la aplicación.

CAPÍTULO 1. Identificación de Sistemas, importancia del control de la calidad de la identificación.

El funcionamiento de los procesos industriales ha cambiado drásticamente en las últimas décadas. Este cambio es debido principalmente a los adelantos en la computación. La automatización de los procesos ha propiciado un aumento de la productividad de algunos sectores industriales, provocando que las industrias se vean obligadas a incrementar su eficiencia a fin de competir y poder subsistir.

En la actualidad, cada vez más, el trabajo en el diseño, de un ingeniero en control, consiste en la realización de modelos matemáticos de procesos estudiados (Ljung, 1999). El diseño de un controlador continuo o discreto, ya sea mediante técnicas clásicas o en variables de estado, requiere de un modelo de la planta a controlar que caracterice su comportamiento dinámico. Este modelo permite al diseñador realizar y validar mediante simulación el ajuste de los parámetros del controlador que permiten obtener una respuesta que satisfaga las especificaciones de diseño (López, 2000) (Kunusch, 2003).

1.1 Modelo de un sistema.

La construcción de un modelo matemático es fundamental, un modelo es una forma muy útil y compacta para resumir el conocimiento sobre un proceso. Es también una herramienta muy eficaz para la educación y la comunicación (Amstrom, 1997)

Los modelos matemáticos describen la relación entre una o más señales de entradas medidas, una o más señales de salida medidas, $y(t)$ u $u(t)$. Sus datos pueden ser medidos en el dominio de tiempo o dominio de la frecuencia y pueden tener una sola o múltiples entradas y salidas. En los sistemas reales, hay aportes adicionales que no se puede medir, lo que afecta la salida del sistema. Tales entradas medidos se denominan perturbaciones o ruido, $e(t)$ (Ljung, 2007).

Los modelos matemáticos pueden adoptar muchas formas distintas. Dependiendo del sistema del que se trate y de las circunstancias específicas, un modelo matemático puede ser más conveniente que otros según la aplicación (Ogata, 2006).

Los modelos matemáticos son abstracciones de procesos reales. Ellos dan una posibilidad para caracterizar comportamiento de los procesos, si se conocen sus entradas. El rango e validez de los modelos determina situaciones en las que se pueden usar los modelos. Se utilizan para el control de procesos continuos. Cada proceso se determina con su naturaleza física y química que expresa su masa y los límites de energía. Investigación de cualquier proceso típico conduce al desarrollo de su modelo matemático. Esto incluye ecuaciones básicas, variables y descripción de su comportamiento estático y dinámico (Mikles and Fikar, 2007).

Cuando se hace necesario conocer el comportamiento de un sistema en unas determinadas condiciones y ante unas determinadas entradas, se puede recurrir a la experimentación sobre dicho sistema y a la observación de sus salidas. Sin embargo, en muchos casos la experimentación puede resultar compleja o incluso imposible de llevar a cabo, lo que hace necesario trabajar con algún tipo de representación que se aproxime a la realidad, y a la que se conoce como modelo (López, 2000).

1.1.1 Métodos de obtención de modelos.

Según (López, 2000) Existen dos métodos principales para obtener el modelo de un sistema:

Modelado teórico: Trata de un método analítico, en el que se recurre a las leyes de la física para describir el comportamiento dinámico de un proceso.

Identificación de sistemas: Trata de un método experimental que permite obtener el modelo de un sistema utilizando datos reales obtenidos de una planta.

En particular, las técnicas de control actuales requieren de modelos matemáticos cada vez más exactos para el análisis y el diseño. Existiendo la dificultad de que, tales modelos no pueden ser obtenidos en forma sencilla y económica a partir de las leyes que rigen cada proceso. Aquí es donde entra a jugar un papel decisivo la Identificación de Sistemas Dinámicos, que es una herramienta capaz de proporcionar los métodos necesarios para obtener de manera relativamente sencilla los modelos matemáticos buscados con un alto grado de exactitud (Kunusch, 2003).

Debido a que los sistemas dinámicos presentan gran abundancia, las técnicas de identificación de sistemas han cobrado gran relevancia en diversas áreas del conocimiento (ingenierías, economía, biotecnología, etc.), donde se requiere de un modelo preciso para fines de análisis, predicción, simulación, diseño y control (Kunusch, 2003).

En este caso solo se construirán modelos que describan el comportamiento de las entradas y salidas, obviando las leyes físicas.

1.2 Identificación de sistemas.

Se denomina *identificación* a la técnica de construir un modelo a partir de las variables medidas de un proceso: variables de entradas, variables de salidas y perturbaciones. Con el objetivo de modelar un sistema, existen tres formas de utilizar los métodos de identificación (Díaz, 2005).

- ✓ Hacer distintas aproximaciones para estructurar el problema: seleccionar las señales de interés, observar la dependencia entre ellas y estudiar el grado de linealidad del proceso.
- ✓ Construir un modelo que describa el comportamiento entre las entradas y salidas, prescindiendo del comportamiento físico. Según se considere pueden ser modelos paramétricos o modelos no paramétricos.
- ✓ Utilizar los datos para determinar los parámetros no conocidos del modelo físico obtenido mediante las leyes que rigen el proceso bajo estudio.

1.3 Pasos de la identificación.

Según (Ljung, 1999) (Amstrom, 1997) (López, 2000). Se definen 5 pasos para realizar la identificación de sistemas. Esta secuencia de pasos ha sido comúnmente utilizada en el desarrollo de las diferentes asignaturas, dada su buena adaptación a las características de los sistemas de control. Los pasos son:

1. *Obtención de datos de entrada - salida.* Para ello se excita el sistema mediante la aplicación de una señal de entrada y se registra la evolución de sus entradas y salidas durante un intervalo de tiempo.

2. *Tratamiento previo de los datos registrados.* Los datos registrados están generalmente acompañados de ruidos indeseados u otro tipo de imperfecciones que puede ser necesario corregir antes de iniciar la identificación del modelo. Se trata, por tanto, de ‘preparar’ los datos para facilitar y mejorar el proceso de identificación.
3. *Elección de la estructura del modelo.* Si el modelo que se desea obtener es un modelo paramétrico, el primer paso es determinar la estructura deseada para dicho modelo. Este punto se facilita en gran medida si se tiene un cierto conocimiento sobre las leyes físicas que rigen el proceso.
4. *Obtención de los parámetros del modelo.* A continuación se procede a la estimación de los parámetros de la estructura que mejor ajustan la respuesta del modelo a los datos de entrada-salida obtenidos experimentalmente.
5. *Validación del modelo.* El último paso consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión.

1.3.1 Métodos de identificación.

1. *Métodos no paramétricos:* Son aquellos en que no es posible definir un vector de parámetros finito para representarlo. Estas técnicas, la mayoría de ellas más fáciles o directas de aplicar o utilizar en un proceso real, aportan información muy útil para afrontar las decisiones que debe tomar el usuario ante un problema de identificación de un proceso (Díaz, 2005).
2. *Método paramétrico:* Son aquellos que quedan descritos mediante una estructura y un número finito de parámetros que permiten relacionar las señales del sistema (López, 2000).

Dadas las características de los procesos a modelar y las facilidades ofrecidas por *MatLab*, se restringirá el presente trabajo solo a la *identificación paramétrica*.

1.3.1.1 Identificación paramétrica.

La *identificación paramétrica* permite describir el comportamiento de cualquier sistema lineal. Cuando no se tiene conocimiento previo del sistema, se debe recurrir a modelos estándar. Estos requieren la elección de una estructura del modelo, de un criterio de ajuste de parámetros y de la estimación de los parámetros que mejor ajustan el modelo a los datos experimentales (López, 2000) (Ljung, 1999) (Díaz, 2005).

No obstante esta identificación presenta algunas dificultades a la hora de elegir el tipo de modelo que se ajuste satisfactoriamente a los datos obtenidos experimentalmente.

1.3.1.2 Tipos de modelos paramétricos.

Un modelo de un sistema consiste en una descripción de algunas de sus propiedades y teniendo en cuenta el propósito para el cual será utilizado. Generalmente los modelos paramétricos se describen en el dominio discreto, puesto que los datos que sirven de base para la identificación se obtienen por muestreo (López, 2000).

Los métodos de estimación paramétricos están muy relacionados con el modelo utilizado. La forma de representar la estructura de un modelo discreto es, según (Ljung, 1999).

$$y(t) = G(q)u(t) + H(q)e(t)$$

Siendo $u(t)$ y $y(t)$ los vectores de entradas y salidas respectivamente.

A diferencia de otros métodos de estimación, los errores cometidos en la modelización son incluidos en el término $e(t)$. A este término se le asocia una serie de variables aleatorias independientes uniformemente distribuidas de media nula (ruido blanco). $G(q^{-1})$ y $H(q^{-1})$ son filtros de orden finito que modelizan la parte determinista y la parte estocástica (Ljung, 1999) (Díaz, 2005) o ruido.

Existen distintas estructuras como son: *ARX*, *ARMAX*, *OE (Output Error)*, *BJ (Box Jenkins)*, *FIR*, *ARARX*, *ARARMAX*.

A continuación se describen brevemente las características de los más usados, los que serán objeto de estudio en nuestro trabajo; *ARX* y *ARMAX*.

❖ **Estructura ARX.**

La relación entrada salida más simple que se puede obtener es la proveniente de una descripción del sistema como una ecuación lineal en diferencias.

$$\begin{aligned}y(t) + a_1y(t - 1) + a_2y(t - 2) + \dots + a_nay(t - na) \\ = b_1u(t - 1) + b_2u(t - 2) + \dots + b_nbu(t - nb) + e(t)\end{aligned}$$

En este caso los parámetros a ajustar serán:

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

Atendiendo a la forma general de representar las estructuras:

$$A(q)y(t) = B(q)u(t) + e(t)$$

A este modelo se le conoce como estructura *ARX* (*Modelos autoregresivos con variables exógenas*), donde *AR* hace referencia a la parte *autorregresiva* $A(q)y(t)$ y *X* a la variable *exógena* $B(q)u(t)$. Aquí se considera que la parte determinista y la estocástica tienen el mismo denominador:

$$G(q, \theta) = \frac{B(q)}{A(q)}$$

$$H(q, \theta) = \frac{1}{A(q)}$$

 El polinomio $A(q^{-1})$ es el polinomio autoregresivo de orden na :

$$A(q) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_naq^{-na}$$

$$B(q) = b_1q^{-1} + b_2q^{-2} + \dots + b_nbq^{-nb}$$

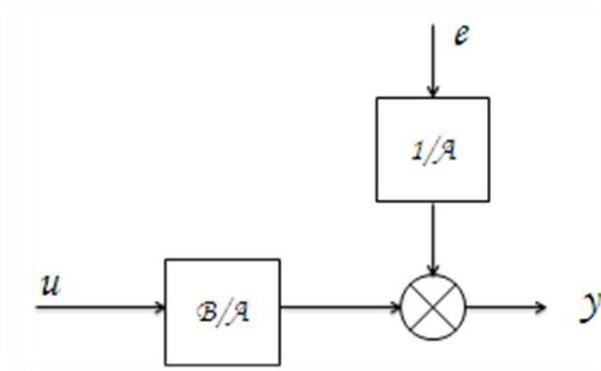


Figura 1.1. Estructura del modelo ARX.

El flujo de señal de la figura 1.1 indica que posiblemente no sea el modelo más natural desde el punto de vista físico, ya que el ruido blanco es sumado a la salida luego de pasar a través del denominador del sistema dinámico. Sin embargo, el conjunto de modelos de ecuación de error posee una propiedad importante que lo convierte en una buena primera elección en muchas aplicaciones. Y es que la forma del predictor define una regresión lineal (Ljung, 1999).

❖ **Estructura ARMAX.**

La principal desventaja del modelo ARX reside en la escasez o falta de libertad en la descripción del término de perturbación (Kunusch, 2003). Sin embargo, es posible incorporar mayor flexibilidad al modelado si se le agrega un término conocido como media en movimiento (*moving average*) del ruido blanco.

$$\begin{aligned}
 y(t) + a_1y(t - 1) + \dots + a_nay(t - na) \\
 = b_1u(t - 1) + \dots + b_nbu(t - nb) + e(t) + c_1e(t - 1) + \dots + c_nce(t - n_c)
 \end{aligned}$$

Con

$$C(q) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_nq^{-nc}$$

Por tanto la estructura resultante para los modelos *ARMAX* queda:

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

Siendo los filtros:

$$G(q, \theta) = \frac{B(q)}{A(q)}$$

$$H(q, \theta) = \frac{C(q)}{A(q)}$$

Siendo ahora el nuevo vector de parámetros:

$$\theta = [a_1 a_2 \dots a_{na} \ b_1 b_2 \dots b_{nb} \ c_1 c_2 \dots c_{nc}]^T$$

Debido al nuevo término de media en movimiento $C(q)e(t)$, el modelo será llamado *ARMAX*. Este modelo se ha convertido en una herramienta estándar en control y econometría, tanto para la descripción de sistemas como para el diseño de control.

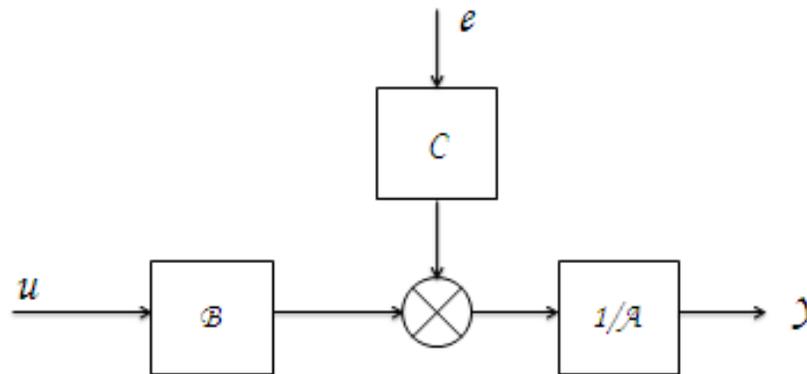


Figura 1.2. Estructura del modelo *ARMAX*.

Para elegir la estructura del tipo de modelos considerados hay que determinar el *orden* de cada uno de los polinomios anteriores, es decir na , nb , nc y el retardo entre la entrada y la salida nk . Una vez elegidos estos valores, sólo queda determinar el *vector de coeficientes* (a_i, b_i, c_i) que hacen que el modelo se ajuste a los datos de entrada-salida del sistema real. Algunos de los más utilizados, para el ajuste de los parámetros, en el campo de la

identificación son el método de mínimos cuadrados y el método de las variables instrumentales (López, 2000).

La anulación de alguno de los polinomios, resultando estructuras simplificadas, facilita el proceso de ajuste de parámetros. Cada una de las estructuras (*ARX* o *ARMAX*) tiene sus propias características y debe ser elegida fundamentalmente en función del punto en el que se prevé que se añade el ruido en el sistema. En cualquier caso, puede ser necesario ensayar con varias estructuras y con varios órdenes dentro de una misma estructura hasta encontrar un modelo satisfactorio (López, 2000).

1.3.1.3 Métodos para el ajuste de parámetros:

Luego de elegida la estructura del modelo (en el presente trabajo *ARX* o *ARMAX*, como los ordenes de cada polinomio), es necesario determinar el valor de los parámetros del mismo que ajustan la respuesta del modelo a los datos de entrada-salida experimentales (López, 2000).

Error de predicción o residuos de un modelo:

Todo modelo matemático de un sistema dinámico pretende predecir el valor de la salida del sistema en función de las entradas y salidas en instantes anteriores. Se llama error de predicción $\varepsilon(t, \theta)$ a la diferencia entre la salida real del sistema y la salida estimada por el modelo en un determinado instante de tiempo (López, 2000).

$$\varepsilon(t, \theta) = y(t) - y_e(t, \theta)$$

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

❖ Regresión lineal:

Según (López, 2000) una estructura posee *regresión lineal* cuando la salida estimada puede expresarse como:

$$y_e(t, \theta) = \varphi^T(t) * \theta$$

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

Un claro ejemplo de estructura con regresión lineal es el modelo ARX, definiendo:

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

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

❖ **Método de mínimos cuadrados (LSE):**

En el método de los *mínimos cuadrados* el problema a resolver consiste en hallar el *vector de ordenes* estimados, partiendo de las observaciones realizadas, tal que se minimice la función de costo (o, función de pérdidas),

Aplicando los criterios fijados anteriormente, la expresión del error de predicción es:

$$\varepsilon(t, \theta) = y(t) - \varphi^T(t) * \theta$$

Según (López, 2000), se define la siguiente función del error:

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} * [y(t) - \varphi^T(t) * \theta]^2$$

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

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

$$\theta_{LSE} = sol \left\{ \frac{1}{N} * \sum_{t=1}^N \varphi^T(t) * [y(t) - \varphi^T(t) * \theta] = 0 \right\}$$

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

1.3.1.4 Validación de modelos:

Una de las tareas más importantes a las que se deben enfrentar en la realización de modelos de simulación es la validación y verificación de los mismos. Entre los objetivos fundamentales del proceso de validación se encuentran: (Banks et al., 2002)

- ✓ Producir un modelo que represente el comportamiento de sistema real lo suficientemente próximo como para que el modelo pueda sustituir al sistema con el objetivo de experimentar determinados aspectos del mismo.
- ✓ Aumentar a un nivel aceptable la credibilidad del modelo, para que sea aceptado por los gestores y usuarios finales a nivel de toma de decisiones sobre los resultados proporcionados por dicho modelo.

La validación es el proceso de comparar la salida del modelo con el comportamiento del fenómeno. En otras palabras: comparar la ejecución del modelo con la realidad (física o otra cualquiera) (Banks et al., 2002).

Por otro lado existe un cierto número de problemas cuando se intenta llevar a cabo el proceso de validación, entre los que se encuentran: (Banks et al., 2002).

1. *No existe la Validación General.*

Cada modelo se valida con respecto a sus objetivos. No se puede decir que un modelo válido para un propósito lo tenga que ser necesariamente para otros. Por otro lado, la idea de la simulación es construir modelos sencillos para objetivos concretos, lo que aleja de la idea de validez general. La realidad es el único “modelo” válido de forma general.

2. *Puede no existir mundo real con el que comparar.*

En muchos casos se desarrollan, a partir de sistemas del mundo real, nuevas funcionalidades o servicios. Por lo tanto, en estos casos no se tiene la referencia del mundo real para comparar. Se puede producir el caso de un modelo válido para funcionalidades conocidas del sistema, pero esto no garantiza la validez del mismo para representar cambios sobre el sistema inicial.

3. *¿Cuál es el mundo real?*

Diferentes personas pueden tener apreciaciones diferentes del mundo real, conocidas como visiones del mundo. Un empleado del banco puede verlo como un medio para ganar dinero, mientras que un usuario de dicho banco puede considerarlo como un medio de depositar o pedir prestado dinero. Dependiendo del interlocutor se pueden encontrar diferentes

apreciaciones del propósito y operación del banco. Cada día se puede apreciar como una misma noticia es contada de diferente forma por distintos periódicos. El acontecimiento es el mismo, pero la interpretación de los periodistas es distinta.

4. *A veces los datos obtenidos del mundo real no son precisos.*

La validación supone, a menudo, una comparación del mismo aspecto del modelo frente a los datos del mundo real. El modelo se ejecuta sobre las mismas condiciones que el mundo real para ver si funciona de forma similar. Pero existen dificultades en este procedimiento, dadas porque los datos del mundo real no siempre son precisos. Si no existe certeza sobre si los datos son buenos no existen garantías para indicar que los resultados del modelo son correctos.

Luego de haber analizado los datos, es probable que se cuente con una extensa colección de modelos de diferentes órdenes y estructuras. Entonces, llegado este punto es aconsejable estar provisto de una gran variedad de herramientas que nos permitan evaluar las cualidades de los diferentes modelos. En este sentido se puede asegurar que el *Toolbox* de Identificación de *MatLab* es un poderoso instrumento capaz de llevar a cabo dichas tareas (Kunusch, 2003).

❖ **Validación en base a la aplicación del modelo**

Puesto que en la práctica es imposible determinar si un modelo responde exactamente al comportamiento de un sistema real, suele ser suficiente comprobar que el modelo es capaz de resolver el problema para el cual ha sido hallado (simulación, predicción, diseño de un controlador, etc.). Así, por ejemplo, si el controlador que ha sido ajustado por medio del modelo da buen resultado sobre el sistema real, se puede asegurar que el modelo era ‘válido’ para esta aplicación (López, 2000) (Kunusch, 2003) (Ljung, 2007).

❖ **Comprobación de parámetros físicos**

Para una determinada estructura que haya sido parametrizada en función de magnitudes físicas, un método importante de validación consiste en comparar el valor estimado de dichos parámetros y el real mediante el conocimiento previo que se tiene de la planta (López, 2000) (Kunusch, 2003) (Ljung, 2007).

❖ **Coherencia con el comportamiento de entrada-salida**

Para determinar si el comportamiento de entrada-salida está suficientemente caracterizado, puede ser necesario recurrir a diferentes métodos de identificación y comparar los resultados obtenidos. Por ejemplo, comparando los diagramas de Bode de los modelos obtenidos mediante identificación paramétrica de diferentes estructuras, por el método de variables instrumentales y por análisis espectral, se puede determinar si la dinámica del sistema ha quedado suficientemente caracterizada (López, 2000) (Kunusch, 2003) (Ljung, 2007).

❖ **Reducción del modelo**

Un procedimiento para determinar si un modelo proporciona una descripción simple y apropiada de un sistema consiste en aplicarle algún método de reducción de modelos. Si una reducción en el orden del modelo no produce alteraciones apreciables en el comportamiento de entrada-salida del mismo, entonces el modelo original era innecesariamente complejo (López, 2000) (Kunusch, 2003) (Ljung, 2007).

❖ **Intervalos de fiabilidad de parámetros**

Otro método para determinar si el modelo bajo estudio contiene demasiados parámetros consiste en comparar los parámetros estimados con su desviación estándar. Si el intervalo de confianza de un parámetro contiene el valor cero, se debe considerar la posibilidad de eliminar dicho parámetro (López, 2000) (Kunusch, 2003) (Ljung, 2007).

❖ **Simulación**

Un procedimiento muy habitual que puede ser considerado como otra técnica de validación de modelos consiste en simular el modelo con un conjunto de entradas distintas a las utilizadas para identificación, y comparar la respuesta del modelo con la obtenida del sistema real (López, 2000) (Kunusch, 2003) (Ljung, 2007).

❖ **Análisis de residuos**

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

$$\varepsilon(t) = \varepsilon(t, \theta) = y(t) - y_e(t, \theta)$$

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

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

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

$$R_{\varepsilon u} = \frac{1}{N} \sum_{t=1}^N \varepsilon(t + \tau) * u(t)$$

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

$$P_r = \frac{1}{N} \sum R_{\varepsilon}(k) * R_u(k)$$

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

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

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

- ✓ Si existe correlación para valores negativos de τ , esto indica que existe realimentación de la salida hacia la entrada, no que el modelo sea deficiente.

- ✓ Si la estimación de un modelo y su expresión de correlación entre residuos y entrada $R_{\varepsilon u}$ se han determinado utilizando los mismos datos de entrada, entonces $R_{\varepsilon u}(\tau) = 0$ para $\tau = nk, \dots, nk + nb - 1$
- ✓ Si $R_{\varepsilon u}(\tau)$ es considerablemente distinto de cero para un valor τ_0 , esto indica que el término $u(t - \tau_0)$ debería ser incluido en el modelo. Éste es un buen método para ajustar el orden más apropiado de la estructura del modelo.

Obviamente, el análisis de los residuos será un método de validación más eficaz si el conjunto de datos utilizados para realizar la correlación es distinto que el usado para la identificación del modelo (López, 2000) (Kunusch, 2003) (Ljung, 2007).

A pesar de que el uso de estos métodos propicia resultados favorables, existen otros métodos a los cuales se hará referencia en este trabajo y serán los que se tendrán en cuenta en la aplicación.

❖ **Bondad de ajuste**

Se entiende por *bondad de ajuste*, el grado de acoplamiento que existe entre los datos originales y los valores teóricos que se obtienen de la regresión. Se trata de saber si el modelo que se ha ajustado para relacionar las variables X e Y es consistente. La medida más comúnmente usada para medir el ajuste de la recta de regresión es el **coeficiente de correlación lineal (r)**:

$$r = \frac{S_{xy}}{S_x S_y}$$

Donde S_{xy} es la covarianza muestral, S_x la desviación típica muestral de la variable X y S_y la desviación típica muestral de la variable Y .

Estos datos se determinan mediante las siguientes expresiones:

$$S_{xy} = \frac{(\sum_i (x_i - \bar{x})(y_i - \bar{y}))}{n} = \left(\frac{\sum_i x_i y_i}{n} \right) - \bar{x}\bar{y}$$

$$S_x = \sqrt{\sum_i \frac{(x_i - \bar{x})^2}{n}} = \left(\sqrt{\frac{(\sum_i x_i^2)}{n - \bar{x}^2}} \right)$$

$$S_y = \sqrt{\sum_i \frac{(y_i - \bar{y})^2}{n}} = \left(\sqrt{\frac{(\sum_i y_i^2)}{n - \bar{y}^2}} \right)$$

El coeficiente de correlación lineal es un valor que cumple las condiciones siguientes:

- ✓ Toma valores entre -1 y 1 .
- ✓ Es invariante por transformaciones lineales de las variables X e Y.
- ✓ Cuanto más extremo es r (más se acerca a -1 o a 1), significa que mejor se ajusta el modelo.

Un valor cercano a 0, debe interpretarse como que no existe un buen ajuste lineal, pero ello no excluye que existan otros tipos de relaciones funcionales. De hecho, podemos hallar ejemplos de relaciones funcionales exactas, con un coeficiente de correlación 0.

Por otro lado, un coeficiente de correlación muy cercano a 1 o a -1 , no debe interpretarse como que existe una relación causa-efecto importante entre las dos variables. La relación podría deberse al efecto de otras variables no incluidas en el estudio.

El cuadrado del valor r multiplicado por 100 se denomina coeficiente de determinación y se interpreta como el porcentaje de variabilidad que explica el modelo.

❖ Coeficiente de determinación

Según (Posada and Noguera, 2007). El coeficiente de determinación  de el porcentaje de variación total en Y debido a las variables que toma el investigador Este valor se obtiene a partir de la *suma de los cuadrados del error (SCE)* y de la *suma de los cuadrados totales (SCT)*, a partir de la ecuación.

$$R^2 = 1 - \frac{SCE}{SCT}$$

La *SCE* corresponde a la suma de los cuadrados de las distancias de los puntos desde la curva de mejor ajuste, en tanto la *SCT* es la suma de los cuadrados de las distancias de los puntos desde una línea horizontal correspondiente a la media de todos los valores de *Y*.

❖ **Cuadrado medio del residuo (o del error) CME**

Según (Posada and Noguera, 2007)]. *CME* es una medida que agrupa la variabilidad de aquellos factores que no tiene en cuenta el investigador. La varianza de *n* residuales e_i se representa como:

$$CME = \frac{\sum(e_i - \bar{e})^2}{n - K} = \frac{\sum e_i^2}{n - K} = \frac{SCE}{n - K}$$

Donde, \bar{e} es la media de *n* residuales (número de observaciones), *K* es el número de parámetros estimados en el modelo y *SCE* es la suma de cuadrados de las distancias verticales de los puntos desde la curva de regresión.

Toda vez que el *CME* corresponde a la varianza residual, los modelos seleccionados por su mayor capacidad de ajuste son aquellos que expresan el menor valor en este criterio.

❖ **Criterio de información de Akaike (AIC) y criterio de información Bayesiano (BIC).**

Cuando se tiene una serie de modelos *M1*, *M2*,... con parámetros *K1*, *K2*,..., respectivamente, una metodología para compararlos corresponde a la función de máxima verosimilitud (*likelihood*). La máxima verosimilitud permite seleccionar el modelo que realiza el mejor ajuste de los datos pero no penaliza su complejidad, lo que si sucede cuando se emplean medidas de contraste como el *AIC* y el *BIC*. Ambos criterios hacen uso del *Log-likelihood* (*log Lik*), que es el logaritmo de máxima verosimilitud, y sustraen un término proporcional al número de parámetros (*K*) en el modelo, así: $\log Lik - \alpha K$, donde α corresponde a 2 para el *AIC* y a $\log(N)$ para el *BIC* (Posada and Noguera, 2007).

❖ **Criterio de información de Akaike (AIC)**

Según (Posada and Noguera, 2007). El criterio combina la teoría de máxima verosimilitud, información teórica y la entropía de información, y es definido por la siguiente ecuación:

$$AIC = -2 * \log Lik + 2K$$

Este criterio tiene en cuenta los cambios en la bondad de ajuste y las diferencias en el número de parámetros entre dos modelos. Los mejores modelos son aquellos que presentaron el menor valor de *AIC*.

Cuando los valores de *AIC* están muy cercanos, la escogencia del mejor modelo se puede realizar con base en el cálculo de la probabilidad (pesos de *Akaike*) y la probabilidad relativa (relación de evidencia), a través de las siguientes ecuaciones:

$$\text{Probabilidad} = \frac{e^{-0.5\Delta}}{1 + e^{-0.5\Delta}}$$

$$\text{Probabilidad relativa} = \frac{\text{Probabilidad de que el modelo 1 sea correcto}}{\text{Probabilidad de que el modelo 2 sea correcto}} = \frac{1}{e^{-0.5\Delta}}$$

Donde, Δ es la diferencia entre los valores de *AIC*.

❖ Criterio de información Bayesiano (BIC)

El *BIC* es calculado para los diferentes modelos como una función de la bondad de ajuste del $\log Lik$, el número de parámetros ajustados (K) y el número total de datos (N). El modelo con el más bajo valor de *BIC* es considerado el mejor en explicar los datos con el mínimo número de parámetros. El *BIC* está definido por la ecuación: (Posada and Noguera, 2007).

$$BIC = -2 * \log Lik + \log(N) * K$$

1.4 Interfaces Gráfica

Las interfaces se pueden definir de forma general como el punto de interconexión entre dos entidades, sistemas, equipos, conceptos, etc (Babylon, 2012). En la actualidad existen diferentes tipos de interfaces, estas se clasifican según su naturaleza en interfaces electrónicas (aquellas por donde se envía o recibe señales de un sistema a otro, por ejemplo, el interfaz USB), interfaces de hardware (monitor, teclado, mouse), interfaces de línea de comando, interfaces gráficas y otras.

La interfaz gráfica de usuario, conocida también como *GUI*, que surge como evolución de las interfaces de línea de comando utilizadas para manejar los primeros sistemas operativos, es el conjunto de métodos que permite lograr la fácil interactividad entre un usuario y una computadora (López, 2009). Utilizando un conjunto de imágenes y objetos gráficos para representar la información y las acciones disponibles en la aplicación, la misma proporciona un entorno visual sencillo que facilita la interacción amigable con un medio de cómputo. El usuario común, habitualmente, realiza las acciones deseadas con la manipulación directa de la interfaz, sin necesidad de conocer la programación implícita en el hecho.

1.4.1 MatLab como herramienta en el desarrollo de aplicaciones automáticas y el diseño de interfaces gráficas

MatLab es, en la actualidad, uno de los software más utilizado en el ámbito ingenieril, por múltiples universidades y centros de investigación y desarrollo. Debe su nombre a la abreviatura de Matrix Laboratory y es un programa muy potente para realizar cálculos numéricos con vectores y matrices (Escribano, 2009). Este software trabaja con el lenguaje M y permite resolver muchos problemas matemáticos, específicamente aquellos que involucran vectores y matrices, en un tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran (Esqueda, 2002).

Entre sus principales prestaciones, se destacan el desarrollo de algoritmos, cálculos numéricos, modelado, simulación y prueba de prototipos, análisis de datos, exploración y visualización, graficación de datos y desarrollo de aplicaciones que requieran de una *GUI* (Carballo, 2011).

Debido a la amplia gama de aplicaciones de tipo matemático, vinculadas al control automático (DeMoyer and Mtichel, 2002, Ong and Tan, 2000), que proporciona este software, el mismo se ha seleccionado para la realización de las principales tareas técnicas propuestas en nuestro trabajo: el diseño de compensadores a través de la Respuesta en frecuencia y la elaboración de una interfaz que facilite al usuario el rápido y fiable diseño de los compensadores anteriormente mencionados.

1.4.2 Herramienta GUIDE del MatLab

Desde los inicios de la informática la meta general de los programadores ha sido lograr que la tarea de realizar programas para ordenadores sea cada vez lo más simple, flexible y portable posible (Francisco, 2000), en base a ello, *MatLab* dispone, en la actualidad, de una herramienta adicional llamada GUIDE.

GUIDE es un entorno de programación visual capaz de crear y manipular interfaces gráficas en *MatLab* (Mathworks, 2007), presenta las características básicas de cualquier programa visual como Visual Basic o Visual C++. Esta herramienta permite al usuario ejecutar instrucciones a través del trabajo con botones, menús, sliders, cuadros de diálogo, etc. Cada uno de los elementos ya mencionados que conforman la interfaz, se pueden personalizar a conveniencia del diseñador de la misma, además, poseen una subrutina asociada que se ejecutará cuando se realice una acción determinada sobre el elemento en sí (Rodríguez, 2011a, Rodríguez, 2011b).

1.4.3 Ventajas de la herramienta GUIDE

La principal ventaja de la herramienta *GUIDE* es que permite realizar el diseño de la interfaz gráfica sin necesidad de recurrir a la utilización del método tradicional: la programación con líneas de código utilizando funciones propias del *MatLab* (Valeriano and Oria, 2012), ello conlleva a que el diseñador, mediante un ambiente amigable, con un alto nivel de interactividad, pueda realizar mayor cantidad de operaciones en un período menor de tiempo. Otra ventaja es que permite realizar y ejecutar programas que necesiten ingreso continuo de datos.

1.5 Consideraciones finales del capítulo:

La *identificación paramétrica* permite describir el comportamiento de los sistemas lineales. Cuando no se dispone del conocimiento previo del sistema, se debe partir de modelos conocidos los cuales requieren la elección de una estructura establecida, de un *criterio de ajuste de parámetros* y de la *estimación de los parámetros* que mejor puedan ajustan el modelo a los datos experimentales.

CAPÍTULO I: Identificación de Sistemas, importancia del control de la calidad de la identificación.

En el método de los *mínimos cuadrados* permite hallar el vector de parámetros estimados, partiendo de las observaciones realizadas, de forma tal que se minimice la función de pérdidas.

De los criterios de validación existentes se usaran, *Criterio de Akaike AIC*, *Criterio del Bayesiano BIC*, el *Cuadrado Medio del Error CME*, *Coefficiente de Determinación R^2* .

CAPÍTULO 2: *BondTool: Interfaz Gráfica para el Cálculo de la Bondad de Ajuste.*

Un proceso industrial puede ser modelado de varias formas obteniéndose por lo tanto diferentes modelos para describir el mismo proceso. *MatLab* en sus últimas versiones y mediante su herramienta *IDENT* permite determinar los tipos de modelo, *ARX*, *ARMAX*, *BJ*, *IV4*, *N4SID* y *OE*.

IV4 se utiliza para modelar proceso con múltiples salidas, *N4SID* ofrece la posibilidad de obtener modelos en el espacio de estado.

En aplicaciones ingenieriles los más utilizados son las estructuras *ARX* y la *ARMAX* razón por la cual serán los que se utilizaran aunque los resultados obtenidos pueden ser fácilmente extendidos a los otros tipos de estructuras de modelos.

La obtención de un modelo, utilizando *MatLab*, además de los datos reales, requiere de la especificación del orden de cada uno de los polinomios, es decir *na*, *nb*, *nc*, y el retardo *nk*, y que se conoce como *orden*, por lo que es posible obtener para un conjunto de datos y un tipo de modelo diferentes representaciones de acuerdo a los ordenes especificados de los polinomios y atrasos.

Intuitivamente se puede pensar que un modelo es más adecuado cuando mejor represente al proceso que modela de acuerdo a los fines para lo cual se ha creado, pero, como es posible saber cuál de los modelos es el que mejor representa al proceso real.

Se han desarrollado diferentes criterios, con el fin de determinar la bondad de ajuste de un modelo específico, entre ellos se encuentran el *Criterio de Información Akaike (AIC)*, el *Criterio de Información Bayesiano (BIC)*, el *Cuadrado Medio del Error (CME)* y el *Criterio del Determinante (R^2)*. Cada uno de estos criterios ofrece un valor numérico, que permite determinar dentro de un tipo de modelo que *orden* es la que más se adecua a las necesidades lo que coincide con el menor valor del criterio. Algoritmos de *MatLab* permiten, de manera directa, el cálculo de los dos primeros criterios (*AIC* y *BIC*), para el cálculo de los restantes se requiere desarrollar algoritmos que los determinen, partiendo de su formulación matemática.

Para la determinación del modelo y el cálculo de los criterios de bondad de ajuste se siguen los siguientes pasos:

1. Cargar los datos reales.
2. Determinar el número de datos.
3. Dividir los datos reales en dos partes; una destinada a la identificación y la otra a la verificación del modelo obtenido.
4. Declara el número de modelos que se desean obtener y el *ORDEN* de cada uno de ellos.
5. Iniciar contador del número de modelos obtenidos.
6. Determinar, de acuerdo al *orden*, si el modelo es *ARX* o *ARMAX*.
7. Si el modelo *ARX* se inicia contador específico para el número de esta estructura de modelo, si no es *ARX* se inicia el contador de modelos *ARMAX*.
8. De acuerdo al tipo de modelo se calcula este para el *orden* especificado.
9. Se determinan los criterios *AIC*, *BIC*, *CME* y R^2 correspondiente al modelo y al *orden* especificado.
10. Se incrementa en 1 el contador específico.
11. Se determina si el número de modelos obtenidos es igual al total de modelos que se desea obtener.
12. Si aun no se han procesado todos los modelos se incrementa en uno el número de modelos y se regresa al paso 6.
13. Si ya se procesaron todos los modelos se finaliza el proceso.

Al concluir el proceso se dispondrá de la información acerca del valor de los índices para cada estructura de modelo y *orden* especificado, pudiéndose seleccionar dentro de una estructura y criterio de bondad cual es el *orden* que mejor se ajusta, que es el que menor valor del criterio ofrece.

La determinación del mejor ajuste de todas las estructuras y *orden* será objeto de otra aplicación o un agente inteligente en el futuro.

En este capítulo se abordaran las generalidades de la aplicación, así como los componentes con que cuenta las interfaces “*BondTool*” y “*Resultados*”, además de sus funcionalidades

y las validaciones que se tuvieron en cuenta para lograr un mejor funcionamiento de la aplicación.

2.1. Generalidades de la aplicación *BondTool*.

Antes de describir el funcionamiento de la aplicación *BondTool*, se hace necesario conocer algunos aspectos de carácter general que deben ser dominados por el usuario. El propósito general que trae consigo dicho software, es la facilitar dentro del proceso de identificación de sistemas, la selección del mejor modelo para una aplicación determinada y por ende esta aplicación es destinada a usuarios que dominen los principales elementos relacionados con la identificación.

Para obtener respuestas más acertadas del software, está el referente a la adecuada selección de la estructura de los modelos. Como se explica en el capítulo 1, con el fin de la identificación de un sistema real, existen diversos tipos de estructuras de modelos, siendo necesario que el usuario determine la estructura que necesita para determinar su modelo. En esta aplicación solo se trabajaran dos estructuras, ellas son; *ARX* y *ARMAX* dado que son las más utilizadas en esta rama de la ingeniería de control según la bibliografía consultada.

Según bibliografía consultada, (Ljung, 1999) y (López, 2000), para la elección de la estructura es necesario conocer las características de estas dos estructuras.

La estructura *ARX*  define mediante la relación $A(q^{-1})y(t) = B(q^{-1})u(t) + e(t)$ como se puede apreciar para esta estructura solo se obtendrán dos polinomios, para los cuales se necesita un vector de ordenes conformado por: na (grado del polinomio A), nb (grado del polinomio B) y nk (retardo del polinomio B).

La estructura *ARMAX* se define por $A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t)$ incluye nc (grado del polinomio C) al vector, ya que esta dará como resultado tres polinomios.

Otro aspecto que resulta de vital interés señalar, es la elección de los criterios para determinar la bondad de ajuste (*AIC*, *BIC*, *CME* y R^2), implementados en la aplicación *BondTool*.

En el acápite destinado a la descripción de estos, en el capítulo anterior, se mencionan y explican los cuatro criterios que se encuentran programados en la interfaz.

La aplicación permite la selección de todos los criterios o solo algunos, pero es importante tener en cuenta, que cada criterio mide diferentes características del modelo, por lo que la selección del orden del modelo que más se adecua a la realidad solo se puede realizar dentro de los resultados obtenidos para un mismo criterio.

2.2. Componentes de la aplicación *BondTool*.

La interfaz *BondTool* está conformada por dos ventanas principales y un conjunto de ventanas secundarias, que permiten la comunicación con el usuario; como son mensajes de error, avisos, preguntas y cuadros de diálogo.

La ventana principal nombrada *BondTool*, mostrada en la figura 2.1, se considerada la más importante, ya que en ella se desarrolla casi la totalidad de códigos de la aplicación; se encuentra conformada por 19 componentes que fueron encapsulados en 3 paneles principales, para de esta forma agruparlos controles de acuerdo a su utilidad o función dentro de la interfaz.

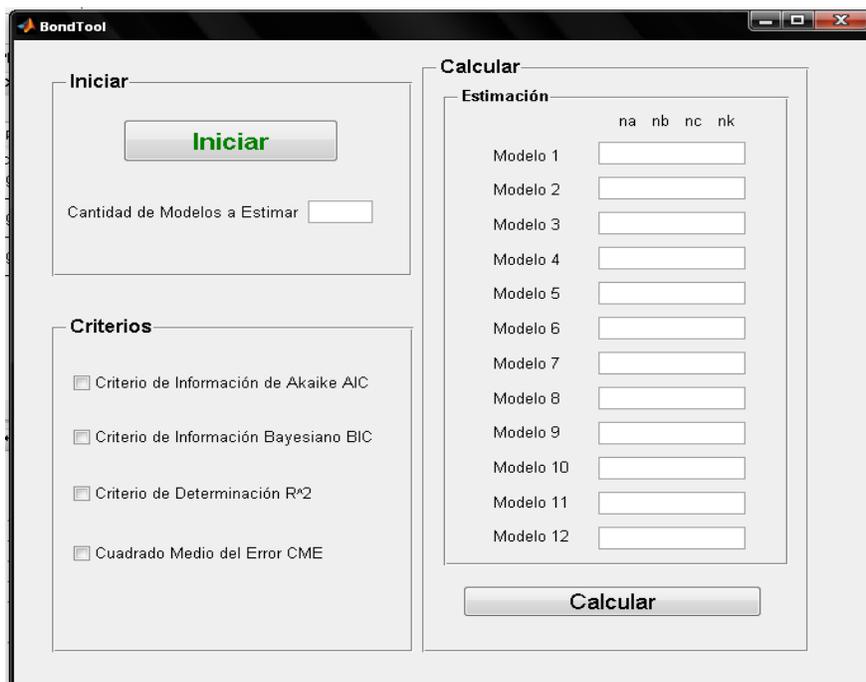


Figura 2.1. Ventana principal de la aplicación *BondTool*.

2.3. Paneles de la aplicación, componentes, funcionalidad y validaciones.

En este acápite se pretende explicar cada uno de los paneles existentes en la aplicación así como sus componentes, las funciones que realizan o la forma de intervenir en la lógica de la programación y las validaciones necesarias para su correcta funcionalidad.

2.3.1. Panel “Iniciar”.

El primer panel se muestra en la figura 2.2, este cuenta con un *push button* o botón con *string* “Iniciar” (en este caso la palabra “*string*” es utilizada por *MatLab* para nominar las etiqueta de los controles contenidos dentro del *toolbox* de diseño de interfaces). Este botón “Iniciar” será utilizado para activar el primer *edit text* (donde se determinara la cantidad de modelos que se desea estimar) y para cargar los datos: un archivo con extensión “.mat” nombrado “*planta*” que se almacena en la carpeta “*MatLab*” de Mis documentos.

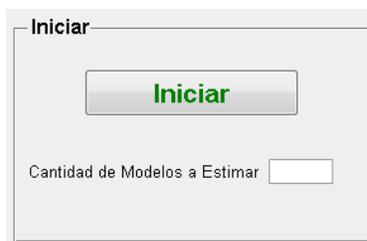


Figura 2.2. Panel “Iniciar” de la aplicación.

Al presionar el *push button* “Iniciar”, el usuario recibirá un aviso explicando que, como se había dicho anteriormente, para ejecutar correctamente la aplicación debe tener el juego de datos almacenado en un archivo con nombre “*planta*” y extensión “.mat” ubicado en la carpeta *MatLab* de Mis Documentos. Como se muestra en la figura 2.3.



Figura 2.3. Aviso obtenido al presionar el *push button* “Iniciar”.

Además en este panel se encuentra el primer *edit text* con nombre indicativo “Cantidad de Modelos a Estimar”, insertado en un *static text*, utilizado para que el usuario introduzca la cantidad de modelos que desea y a su vez para habilitar los *edit text* que serán usados para los *ordenes* de los modelos, los cuales se activaran en función de la cantidad de modelos a estimar.

El *edit text* del panel “Iniciar” está diseñado para capturar la cadena de caracteres introducida por el usuario, por lo que es válido resaltar que la inclusión de las mismas en la interfaz se produce en formato *string*. Por tanto, para poder trabajar con estos valores, se hizo necesaria la conversión a formato *double* a través de la función *str2double*¹.

2.3.1.1. Validaciones del panel “Iniciar”.

La cantidad de modelos que el usuario desee estimar, fue restringida a doce modelos, es por ello necesario realizar algunas validaciones como:

- ✓ En el *edit text* solo se admitirá valores numéricos definidos entre los rangos permisibles (1 a 12), (máximo número de modelos a estimar). En caso de insertarse algún valor que no cumpla con este rango, el usuario recibirá un aviso de error, e inmediatamente el valor erróneo desaparecerá para que este pueda introducir uno correcto.
- ✓ El *edit text* no permitirá valores numéricos negativos. Si no se cumple con dicha condición, el usuario recibirá un aviso de error, e igualmente el valor erróneo desaparecerá para que este pueda introducir uno correcto.
- ✓ No se permite que en el *edit text* se pasen letras. Por tanto en caso de insertarse algún tipo de letra, el usuario recibirá un aviso de error, y también será borrado este carácter para que pueda introducir uno correcto.

En la figura 2.4 se puede ver un ejemplo de lo que ocurre al pasar un dato erróneo.

¹ Convierte el *string* insertado a escalar siempre que sea posible.

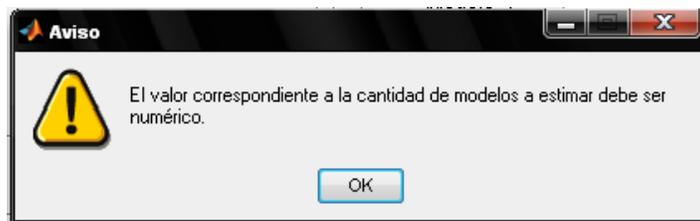


Figura 2.4. Aviso destinado a aclarar qué tipo de valores debe pasarse en el *edit text*

2.3.2. Panel “Criterios”.

El segundo panel se encarga de la selección de todos los criterios de bondad de ajuste o los que se desee dentro de los programados en la aplicación. La figura 2.5 muestra dicho panel, el cual está conformado por cuatro *check box* con los *string* “*Criterio de Información de Akaike AIC*”, “*Criterio de Información Bayesiano BIC*”, “*Criterio del Determinante R^2* ” y “*Cuadrado Medio del Error CME*”, respectivamente.

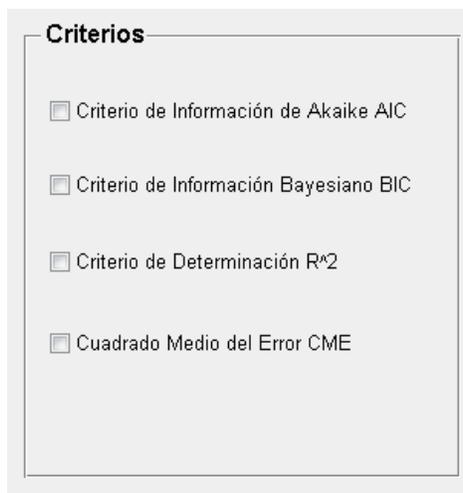


Figura 2.5. Panel “*Criterios*”

El objetivo principal de este panel (como se mencionó anteriormente), es la selección de los criterios, para ello se puede habilitar desde uno solo hasta los cuatro criterios, así el usuario podrá obtener solo los valores de bondad de ajuste que desee.

2.3.3. Panel “Estimación”.

	na	nb	nc	nk
Modelo 1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 8	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 9	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 11	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 12	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 2.6. Panel “Estimación”.

El panel representado en la figura 2.6, es el encargado de agrupar los controles donde se deben introducir los *ordenes* necesarios para la estimación de los modelos. El mismo cuenta con doce *edit text*, que se destinan a la inserción de los valores de *na*, *nb*, *nc*, *nk* necesarios en la aplicación. Dicho panel posee además trece *static text*, doce de ellos indican las casillas reservadas para cada número de modelo que el usuario va a utilizar y la otra representa los ordenes que se deben pasar para que funcione correctamente la aplicación.



En este caso, los *edit text* presentes en el panel “*Parámetros*” se encuentran diseñados para obtener las cadenas de caracteres introducidas por el usuario, por lo que es necesario resaltar que la inclusión de las mismas se produce en formato *string*. Debido a esto y a que se trata de representación de matrices, se hace necesario convertirlos a formato *num* a través de la función *str2num*².

²Convierte una matriz cadena a un arreglo numérico.

2.3.3.1. Validaciones del panel “Estimación”.

Seguidamente se explican las principales validaciones implementadas en el panel para una correcta utilización

- ✓ Los *edit text* destinados a la inserción de los ordenes estarán deshabilitados y vacíos al iniciar la aplicación; solo se utilizaran si se selecciona correctamente la cantidad de modelos a estimar.
- ✓ En los *edit text* no se permitirán valores numéricos con signo negativo. En caso de insertarse algún valor que no cumpla con esto, el usuario recibirá un aviso de error, e inmediatamente los valores erróneos desaparecerán para que este pueda introducirlos correctamente.
- ✓ No se permite que en los *edit text* se introduzcan letras. Por tanto si no se cumple la condición necesaria, igualmente, recibirá un aviso de error, y por ende los valores erróneos desaparecerán para que este pueda volver a introducirlos correctamente sin necesidad de borrarlos.
- ✓ Algo muy importante, es que en los *edit text* se adicionen los valores de los ordenes correctamente ($[na\ nb\ nc\ nk]$). En caso de que se inserte algún valor de más, o no insertarse los cuatro ordenes, el usuario recibirá un aviso de error, e inmediatamente los valores erróneos desaparecerán para que este pueda introducirlos correctamente.
- ✓ En los *edit text*, los ordenes na y nb (que son comunes a ambas estructuras) no pueden ser cero en ningún momento, de este modo en caso de que se introduzca, en el lugar de ellos un cero, se ejecutará un aviso de error, y se procederá con el mismo procedimiento de las validaciones anteriores.
- ✓ Además solo se permitirán valores numéricos enteros en los *edit text*. En caso de que se inserte un valor que no cumpla con esta característica, el usuario recibirá un aviso de error, e inmediatamente los valores erróneos desaparecerán para que este pueda introducirlos correctamente.

En el caso particular de los valores de nc y nk , es necesario mencionar que estos pueden ser cero en algunas ocasiones, debido a que si el modelo que se desea es *ARX*, entonces el valor de nc necesariamente tiene que ser cero, y para el caso de nk este puede tomar cero como valor siempre que el usuario así lo desee.

Dado que no se encontró, en *MatLab*, ninguna función que permitiera determinar si los números eran enteros, solo para los números de tipo *int8* e *int16*, mientras que los valores utilizados en esta aplicación son cadenas, debido a esto se desarrolló una subrutina utilizada con el objetivo de solucionar este problema. A continuación se muestra un ejemplo representativo de dicha subrutina:

```
elseif c1==4 && (mo1(1,1)<=0 || mo1(1,2)<=0 || mo1(1,3)<0
|| mo1(1,4)<0 || (mo1(1,1)-round(mo1(1,1)))~=0 || (mo1(1,2)-
round(mo1(1,2)))~=0 || (mo1(1,3)-round(mo1(1,3)))~=0 || (mo1(1,4)-
round(mo1(1,4)))~=0)
errordlg('Los valores del vector de parámetros deben ser números enteros
y positivos.','Error');
set (hObject,'String','');
mo1=0;
end.
```

Siendo:

- ✓ *c1*: el número de columnas que tiene el vector de ordenes
- ✓ *mo1*: vector de ordenes del modelo 1 para esta subrutina, este cambia de acuerdo al número del modelo.

En esta subrutina, como condición para ejecutar la sentencia se comprueba, si la cantidad de columnas que tiene el vector de ordenes es igual a cuatro, y si alguno de los parámetros son números negativos. Por último utilizando la función *round* (redondea los valores hacia el entero más cercano) se realiza una resta de cada uno de los valores del vector con el redondeo y se comprueba si es diferente de cero. En caso de que sea diferente de cero, entonces el valor no será un entero.

Es válido aclarar que, como se explicó en el capítulo 1, la estructura de modelo *ARX* cuenta con un vector de ordenes compuesto por [*na nb nk*] y la estructura *ARMAX* necesita la inclusión de otro orden, quedando [*na nb nc nk*]. Por tanto en la aplicación se programaron los *edit text* utilizados para la inserción de los vectores de coeficientes, para introducir los cuatro ordenes, con la aclaración, de que cuando el usuario desee obtener un modelo *ARX* el valor de “*nc*” será cero.

2.3.4. Panel “Calcular”.

El panel “*Calcular*” está constituido por el panel “*Estimación*”, explicado en el epígrafe anterior y por el *push button* con *string* “*Calcular*”. El cual será objetivo del presente epígrafe. En la figura 2.7 se muestra dicho panel.

El *push button* “*Calcular*”, es el encargado de ejecutar la aplicación una vez insertados los datos necesarios para la estimación de los modelos y el efectuar el cálculo de los criterios de bondad de ajuste.

	na	nb	nc	nk
Modelo 1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 8	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 9	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 11	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Modelo 12	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Figura 2.7. Panel “Calcular”.

2.3.4.1. Funcionalidad del botón “Calcular”.

Luego de ejecutarse el botón “*Calcular*” y verificarse que los datos se introdujeron correctamente, cuestión que será tratada en el epígrafe 2.3.4.2, la aplicación procede a estimar los modelos y calcular los criterios de bondad de ajuste, seleccionados por el usuario previamente.

Primeramente se ejecuta una subrutina que carga el juego de datos “*planta.mat*” y divide a la mitad una primera para la estimación de los modelos y la otra para realizar la validación de los mismos. Previendo que al medir el juego de datos, el valor de su longitud fuera impar, se utilizó la función *round* (redondeo al valor más próximo), sí la diferencia de la mitad es distinta de cero le restara 0.5 y así la primera mitad quedaría en un rango adecuado.

Luego chequea cuales de los *check box*, que contiene los criterios fueron seleccionados, asignándoles un “1” si están marcados, los cuales estaban inicialmente en cero.

Para la ejecución de la estimación de los modelos y el cálculo de los criterios, se crea una matriz, que tendrá cuatro columnas y el número de filas será igual a la cantidad de modelos que desee el usuario.

Como el orden “*nc*” queda en la tercera columna se comprueba, para cada uno de los modelos empezando por el primero, mediante un ciclo *for*, si es igual a cero, en caso de ser cierto, el modelo sería un *ARX* y por tanto se estimará el modelo mediante el siguiente algoritmo:

```
marx=arx(datose, [ma(i,1) ma(i,2) ma(i,4)])
```

Siendo *datose* la primera mitad del juego de datos y $[ma(i,1) ma(i,2) ma(i,4)]$ el vector de ordenes para el modelo que se esté estimando en ese momento.

En caso de que el valor del orden “*nc*” sea distinto de cero, el modelo será un *ARMAX* y se estimara mediante:

```
marmax=armax(datose, [ma(i,1) ma(i,2) ma(i,3) ma(i,4)])
```

Seguidamente el *software*, procederá a implementar el cálculo de los criterios de bondad de ajuste, para los cuales se necesitan datos provenientes de los modelos como es el caso de; la suma de los ordenes estimados del modelo, el tamaño de la muestra, la función *logLik*, el error obtenido mediante la función *pe*, los cuales fueron obtenidos por los algoritmos que se muestran a continuación en un ejemplo para el modelo *arx1*:

```
AICarx1=aic(marx1); %Cálculo AIC
darx1=ma(i,1)+ma(i,2)+ma(i,3); %suma de los ordenes del modelo
N=length(y); %Tamaño de la muestra
LLFarx1=darx1-AICarx1/2; %Obtención de la función logLik
[AICarx1,BICarx1]=aicbic(LLFarx1,darx1,N) %Cálculo del BIC
earx1=pe(marx1,planta); %Cálculo del error mediante la función pe
```

Esta secuencia se muestra como se obtienen los valores correspondientes a una parte de los criterios, para ello se encontró entre las funciones de *MatLab*, la función “*aic*”, esta permite obtener el valor del criterio *AIC* luego de estimarse el modelo. Además este coeficiente fue utilizado para a partir de las ecuaciones planteadas en el capítulo anterior, obtener el valor correspondiente a la función *logLik*, quedando como se muestra en la secuencia de códigos, esta función fue necesaria obtenerla para luego proceder a calcular el valor del *BIC*, mediante una función perteneciente al *MatLab*.

El error que se muestra es calculado por una función llamada *pe* la cual da como resultado la varianza de los residuales según la muestra, el cual se utilizará para el cálculo de la suma de los cuadrados de las distancias verticales de los puntos desde la curva de los residuales.

Esta secuencia está sucedida por un ciclo *for*, encargado de calcular la suma de los cuadrados del error (*SCE*), la que será utilizada para obtener el valor del *CME* y del R^2 . Mostrados seguidamente como continuación del ejemplo del modelo *arx1*:

```
SCEarx1=0; %inicializa SCE en cero
for v=1:1:length(y) %ciclo for para determinar SCE
    SCEarx1=SCEarx1+earx1(v)*earx1(v);
end
CMEarx1=SCEarx1/(length(y)-darx1) %Algoritmo para el cálculo del CME
ysarx1=y'-earx1; %valores de salida estimados
ymarx1=mean(y); %Valores medios de y
rvmyarx1=y-ymarx1; %
SCTarx1=0; %SCT inicializado en cero
for ns=1:1:length(y)
    SCTarx1=SCTarx1+rvmyarx1(ns)*rvmyarx1(ns); %Cálculo de SCT
end
```

$$R2arx1 = (1 - (SCEarx1 / SCTarx1)) * 100$$

%Determina el valor de R^2

Siendo:

- ✓ *SCEarx1*: Suma de los cuadrados del error, la cual será utilizada para calcular el Cuadrado Medio del Error y el criterio del Determinante.
- ✓ *CMEarx1*: Cuadrado medio del error calculado mediante la *SCE* dividido la resta del tamaño de la muestra y el número de órdenes estimados del modelo.
- ✓ *ysarx1*: salida estimada del modelo.
- ✓ *ymarx1*: valor medio de la salida real.
- ✓ *rvmyarx1*: resta de la salida real y del valor medio de la misma.
- ✓ *SCTarx1*: Suma de los cuadrados totales utilizada para el cálculo del criterio del Determinante.
- ✓ *R2*: Criterio del determinante, para el cual se utiliza la *SCE* y la *SCT*, y su resultado esta expresado en porciento.

Con el algoritmo mostrado anteriormente se completa el cálculo de los criterios de bondad de ajuste, que luego serán mostrados en la ventana “*Resultados*”

Es necesario aclarar que este ciclo se repite hasta completar la cantidad de modelos, y que el usuario puede determinar los doce modelos con una misma estructura o alternar con las dos propuestas.

Durante el tiempo de ejecución del botón “*Calcular*”, la interfaz muestra una segunda ventana con el nombre “*Procesando*”, siendo esta la encargada de explicar que el software está procesando y que debe esperar, como se muestra en la figura 2.8.

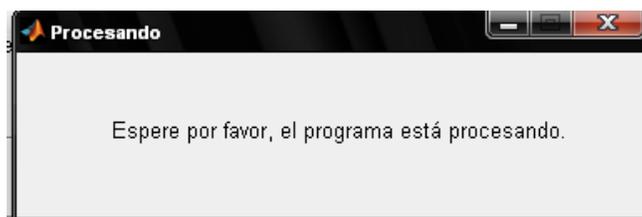


Figura 2.8. Ventana “Procesando”

2.3.4.2. Validaciones del botón “Calcular”.

- ✓ En caso de que el usuario ejecute el botón, sin antes haber entrado el *string* con la cantidad de modelos que desea estimar, se envía un mensaje de aviso explicando la necesidad de poner la cantidad de modelos.
- ✓ Si el usuario introduce en la aplicación la cantidad de modelos pero no marca ningún criterio y ejecuta el botón “*Calcular*”, entonces recibirá un mensaje de aviso pidiéndole que al menos seleccione un criterio de los cuatro propuestos.
- ✓ En caso de que introduzca bien la cantidad de modelos y seleccione al menos uno de los criterios, y ejecute el botón sin antes haber introducido los correctamente los parámetros, se ejecutará un mensaje de aviso anunciándole que debe introducir correctamente los órdenes según el número de modelos a estimar.
- ✓ Si el usuario marca correctamente los criterios y la cantidad de modelos, pero no introduce todos los datos de los modelos, recibirá un mensaje de aviso para que introduzca correctamente los órdenes según la cantidad de modelos que seleccionó.

2.4. Ventana “Resultados”, funcionalidad.

Los resultados obtenidos, luego de presionar el botón son enviados a la segunda ventana principal, nombrada “*Resultados*”, mediante una llamada a la interfaz “*Resultados*”, pasándole el propio nombre de esta, realizándose desde la ventana “*BondTool*”. Dicha ventana se muestra en la figura 2.9.



Figura 2.9. Segunda ventana principal “Resultados”.

Esta ventana está compuesta por 96 *edit text*, dedicados a la organización de los resultados y 34 *static text*, de ellos 8 para mostrar las casillas en las que se colocarán los criterios, 24 para el número de modelo y 2 para dividir los resultados de los modelos una parte para los modelos *ARX* y la otra para los modelos *ARMAX*.

2.4.1. Funcionalidad de la ventana “Resultados”

Para que la ejecución de la ventana se realice, es necesario que la ventana secundaria “Procesando” se cierre  para ello se programo mediante la función *delete*, en el *opening* del fichero “Resultados.mat”, el cierre de la ventana “Procesando”.

Seguidamente se declararon las variables globales, destinadas a la obtención de los valores calculados para cada criterio. Estas variables también fueron declaradas en el fichero “BondTool.m”, en ellas se guardan los valores de los criterios de información al ser calculados, para luego poder obtenerlas desde cualquier interfaz con solo hacer una llamada a la variable global.

Entre las variables globales declaradas se encuentran los cuatro *check box*, para determinar si fueron seleccionados los cuatro criterios o solo alguno de ellos, y además para cada uno

de los modelos se utilizaron las variables con los nombres de cada criterio según el modelo, donde fueron guardados los resultados calculados en la ventana principal.

Se debe  señalar que los *edit text* dedicados a mostrar los resultados, se inicializaron cero y se les paso una cadena en blanco para que se encuentren limpios los que no se necesiten.

Al abrir la interfaz se toman todos los valores, ya sean los calculados como los que no se calcularon y se convierten de número a cadena utilizando la función *num2str* y seguidamente utilizando la función *strcmp* (compara dos cadenas) se comparan los valores obtenidos de las variables con los valores de los *edit text* y si estos no son iguales, se procede a mostrar los que fueron calculados previamente.

2.5. Consideraciones finales del capítulo.

Se ha trabajado con las estructuras *ARX* y *ARMAX* sin embargo los resultados alcanzados se pueden extender a otras estructuras programadas en *MatLab*.

Los criterios de validación programados son los más utilizados en aplicaciones técnicas.

La aplicación permite la selección de diferentes criterios de validación, pero es importante tener en cuenta, que cada criterio mide diferentes características del modelo, por lo que la selección del orden del modelo que más se adecua a la realidad solo se puede realizar dentro de los resultados obtenidos para un mismo criterio.

La Interfaz Gráfica nos permite determinar hasta 12 modelos diferentes y hasta 4 criterios de validación de cada uno de ellos, de esta manera se facilitar el análisis del especialista de acuerdo a las exigencias del proceso bajo estudio

CAPÍTULO 3: Análisis de los Resultados.

Luego de describir la forma de obtención de los criterios de validación *AIC*, *BIC*, *CME* y R^2 , además de la estructura de la aplicación *BondTool*, se procede a exponer, en el presente capítulo una serie de pruebas con el objetivo de demostrar el buen desempeño del software y la correcta utilización de los criterios implícitos para validar modelos obtenidos mediante la identificación de sistemas. Con este propósito se utilizara un juego de datos procedente del proceso de floculación de una planta de tratamiento de residuales de la Refinería “*Camilo Cienfuegos*”.

3.1. Características generales del proceso de Floculación y de los datos.

Las refinerías de petróleo están conformadas por diferentes plantas de transformación, que tienen en cuenta las características del crudo que reciben y de los productos finales necesarios, según las demandas del mercado.

Por lo general se producen gasolina, gas licuado, bencinas especiales y fuel-oil.

En las diferentes etapas de los procesos de transformación se utiliza agua y vapor de agua. Estas al entrar en contacto con el petróleo se mezcla con hidrocarburos y otras sustancias del petróleo tales como compuestos sulfurados, oxigenados y nitrogenados; en estos procesos el agua se contamina y pasa a ser denominada como: aguas residuales. Estas aguas deben ser tratadas para devolverlas al medio ambiente ya descontaminadas.

Entre los procesos que generan aguas residuales se encuentran, el de desalación del petróleo crudo, la destilación, la separación de gases, la del craqueo térmico, craqueo catalítico, la purificación de gases.

Las aguas residuales de las refinerías se recolectan separadas, de acuerdo al punto donde se genera y son llevadas a las plantas de tratamiento, generalmente se deben diferenciar entre, aguas libres de petróleo, aguas que pueden contener petróleo y aguas que han estado en contacto con el petróleo.

Por lo general se practican las siguientes etapas de tratamiento:

Etapas 1: Los destiladores que eliminan sustancias muy volátiles por medio del vapor de agua.

Etapas 2: Se separa el petróleo del agua en los separadores y el petróleo que flota en la superficie del agua se recupera para incorporarlo de nuevo al proceso de transformación.

Etapas 3: Mediante procesos químico-físicos se eliminan las partículas de petróleo que están finamente distribuidas dentro del agua. Este proceso se apoya en la flotación.

Etapas 4: Las sustancias que aún quedan en el agua, después de estos pasos, se eliminan en buena medida por procesos biológicos, en caso de que sean biodegradables o que puedan someterse a una eliminación físico-química por ejemplo por absorción o adsorción.

Si es necesario, como etapa biológica avanzada se puede descargar posteriormente en una laguna secundaria o una laguna aireada.

Los tratamientos químico y físico pueden abarcar, fundamentalmente los siguientes procedimientos:

- ❖ Floculación
- ❖ Separación de sólidos por sedimentación
- ❖ Flotación
- ❖ Filtración

Este caso trata específicamente del procedimiento de floculación.

El proceso de floculación de dispersiones de petróleo en agua puede lograrse añadiendo un agente floculante, como es el caso de los iones de metal relativamente grandes de Hierro y Aluminio.

El petróleo disuelto en agua forma una solución coloidal, caracterizadas por estar constituidas por una dispersión de partículas sólidas o líquidas en agua y que son difíciles de eliminar utilizando procedimientos como pueden ser gravedad, filtración y otros.

El fenómeno mediante el cual se logra desestabilizar el coloide y su aglomeración posterior, es lo que se conoce como coagulación y floculación, de hecho es un proceso en dos etapas.

En la primera etapa de coagulación se elimina la doble capa eléctrica que caracteriza a los coloides, y la floculación se da a continuación y consiste básicamente en la aglomeración de los coloides mediante la atracción de las partículas con el aglutinamiento que se logra por la presencia de sustancias conocidas como floculantes.

Las operaciones que se llevan a cabo en el proceso físico-químico para el tratamiento de una solución coloidal son:

- ❖ Mezclado
- ❖ Coagulación
- ❖ Floculación
- ❖ Separación

Cada una de estas operaciones tiene sus variables de diseño u operación definidas y para lograr un buen resultado, es necesario que las variables estén dentro de los límites que establece el proceso.

Mediante el mezclado y la coagulación y la floculación se forman los denominados floculos o aglomeración de los sólidos disueltos, que mediante la separación, permiten liberar los sólidos de la solución coloidal cumpliendo así una fase importantísima en la purificación de las aguas residuales.

Los datos obtenidos para realizar la validación están conformados por una muestra de 165 valores, donde la salida es el flujo de recirculación a flotadores y la entrada su valor deseado (señal PRBS). Los que pueden ser observados en el Anexo I

3.2. Validación de la interfaz

Antes de aplicar los datos y comprobar los resultados de la validación de los modelos correspondientes a las distintas pruebas realizadas, es importante desarrollar las pruebas para comprobar la detección de posibles errores en la ejecución. Con este fin se somete la interfaz a entradas erróneas mostrándose los mensajes emitidos.

3.2.1. Validación del panel “Iniciar”.

Con el propósito de lograr un desempeño favorable de la aplicación, al iniciarse, solo podrá ejecutar el botón que permite iniciar la aplicación, este permite activar el *edit text* de la cantidad de modelos, los *check box* para seleccionar los criterios de bondad de ajuste y el botón “*Calcular*”.

Presionando inicialmente el botón “*Calcular*”, sin haber introducido la cantidad de modelos y los criterios que se desean calcular, recibimos el siguiente mensaje mostrado en la figura 3.1.

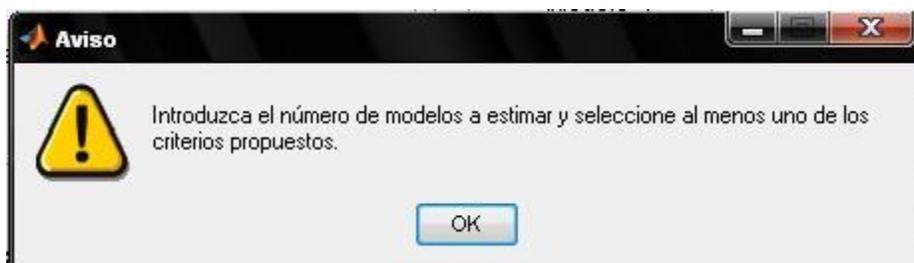


Figura 3.1. Aviso de error en el panel iniciar.

Al introducir en el *edit text* de cantidad de modelos un valor que no sea un entero positivo, por ejemplo 7.1, -4 , o similares que no estén en el rango de uno a doce, o letras, tales como A, b, Z etc., se ejecuta el mensaje de aviso y se limpia nuevamente el cuadro de datos correspondiente, para que se introduzca el valor correcto. Como ejemplo de esto se puede ver en la figura 3.2.

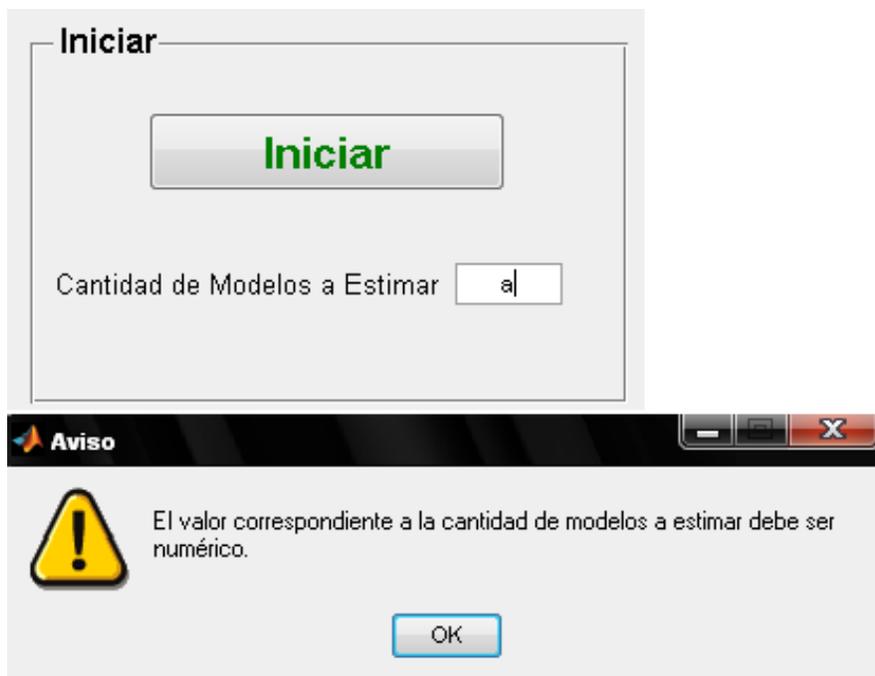


Figura 3.2. Aviso de error.

3.2.2. Validación de los criterios de información AIC, BIC, R^2 , CME.

El cálculo de los criterios de bondad de ajuste, requiere una buena identificación y por ello una adecuada selección de la estructura del modelo (en este caso *ARX* y *ARMAX*), acompañado de la correcta inserción del vector de ordenes, que se encarga de determinar el orden del modelo, según los requerimientos específicos de un proceso dado.

Al seleccionar el botón “*Calcular*”, sin antes haber introducido los vectores de ordenes de los modelos, como se muestra en la figura 3.3 se recibió el mensaje de aviso que se muestra en la figura 3.4.

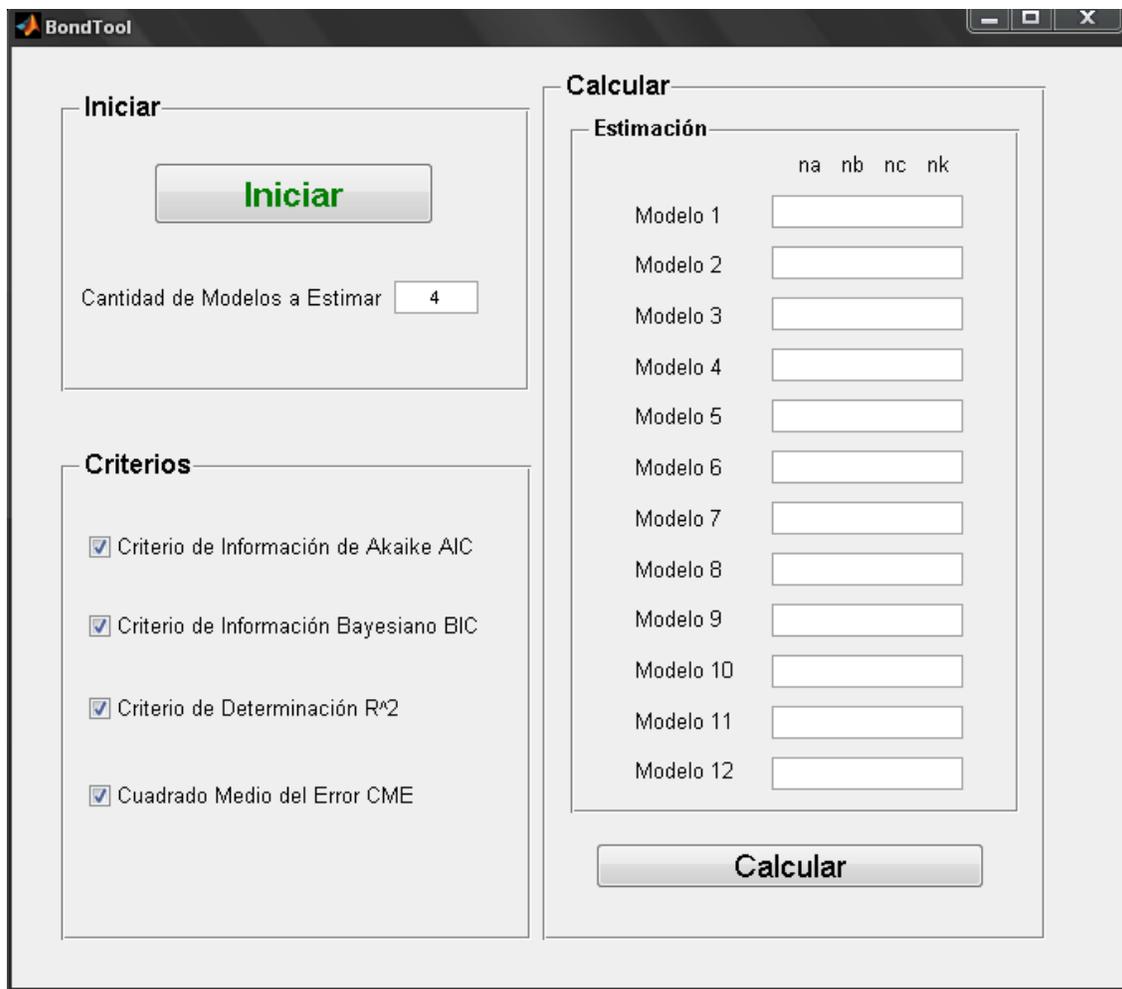


Figura 3.3. Ventana Principal sin introducir los vectores de ordenes.

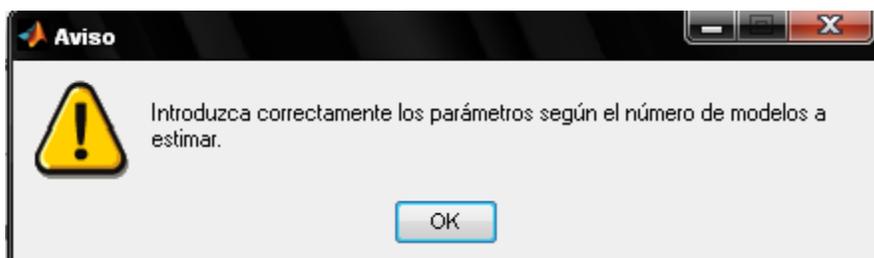


Figura 3.4. Aviso enviado para que el usuario introduzca los datos que faltan.

Debido a esto, se realizaron validaciones en los *edit text* de los ordenes de los modelos para que solo se puedan introducir valores numéricos, enteros y positivos, por tanto si se introducen valores no enteros negativos como se muestra a continuación, la aplicación se encarga de enviar el mensaje que se muestra en la figura 3.5.

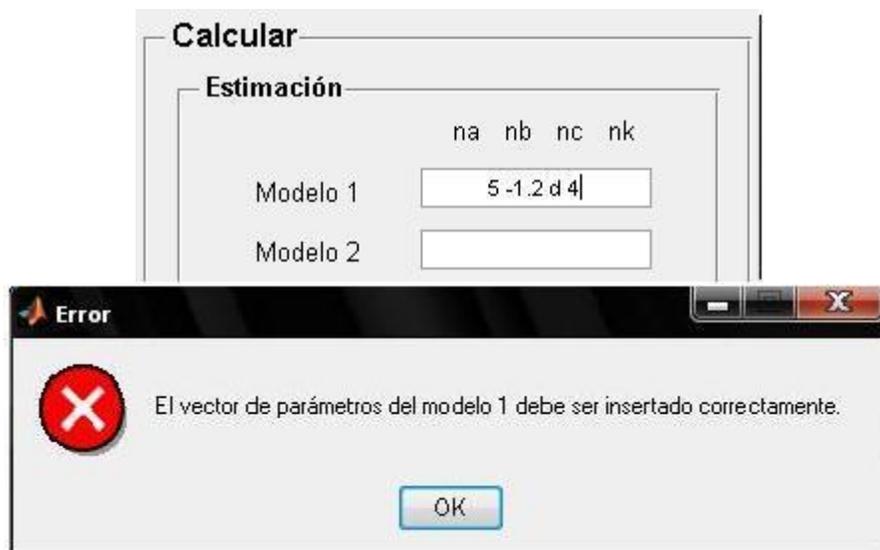


Figura 3.5. Error debido a la incorrecta inserción de los ordenes.

Finalmente, luego de introducir todos los datos y seleccionar los criterios de bondad de ajuste que se muestran en los *check box* del panel “Criterios”, se procede a realizar la estimación de los modelos y el cálculo de los criterios, para ellos se tomo el juego de datos, mencionado anteriormente.

3.3. Pruebas

❖ Primera prueba.

Para la primera prueba se estimaron ocho modelos, de ellos cuatro presentan estructura *ARX* y los restantes *ARMAX*. Los datos necesarios para la obtención de los resultados se muestran en la figura 3.6

Para iniciar la prueba, se comprobó la veracidad de la estimación de los modelos, para ello se estimó el primer modelo *ARX* utilizado en la interfaz, por la herramienta *ident* y por la interfaz *BondTool*, obteniéndose los siguientes resultados.

❖ Estimación utilizando la interfaz *BondTool*.

Discrete-time IDPOLY model: $A(q)y(t) = B(q)u(t) + e(t)$

$$A(q) = 1 - 1.097 q^{-1} + 0.09954 q^{-2} + 0.4763 q^{-3} - 0.4619 q^{-4}$$

$$+ 0.214 q^{-5} + 0.06129 q^{-6} - 0.185 q^{-7} + 0.09486 q^{-8}$$

$$B(q) = -0.002198 q^{-1} + 0.2367 q^{-2} + 0.1632 q^{-3} + 0.05097 q^{-4}$$

Estimated using *ARX* from data set *datose*

Loss function 0.0480689 and FPE 0.0643175

Sampling interval: 0.01

❖ **Estimación utilizando el *ident*.**

Discrete-time IDPOLY model: $A(q)y(t) = B(q)u(t) + e(t)$

$$A(q) = 1 - 1.097 q^{-1} + 0.09954 q^{-2} + 0.4763 q^{-3} - 0.4619 q^{-4}$$

$$+ 0.214 q^{-5} + 0.06129 q^{-6} - 0.185 q^{-7} + 0.09486 q^{-8}$$

$$B(q) = -0.002198 q^{-1} + 0.2367 q^{-2} + 0.1632 q^{-3} + 0.05097 q^{-4}$$

Estimated using *ARX* from data set *mydatade*

Loss function 0.0480689 and FPE 0.0643175

Sampling interval: 1

Con este resultado se demuestra que la estimación realizada por la interfaz *BondTool*, está calculada exactamente como en el *Toolbox "Ident"*

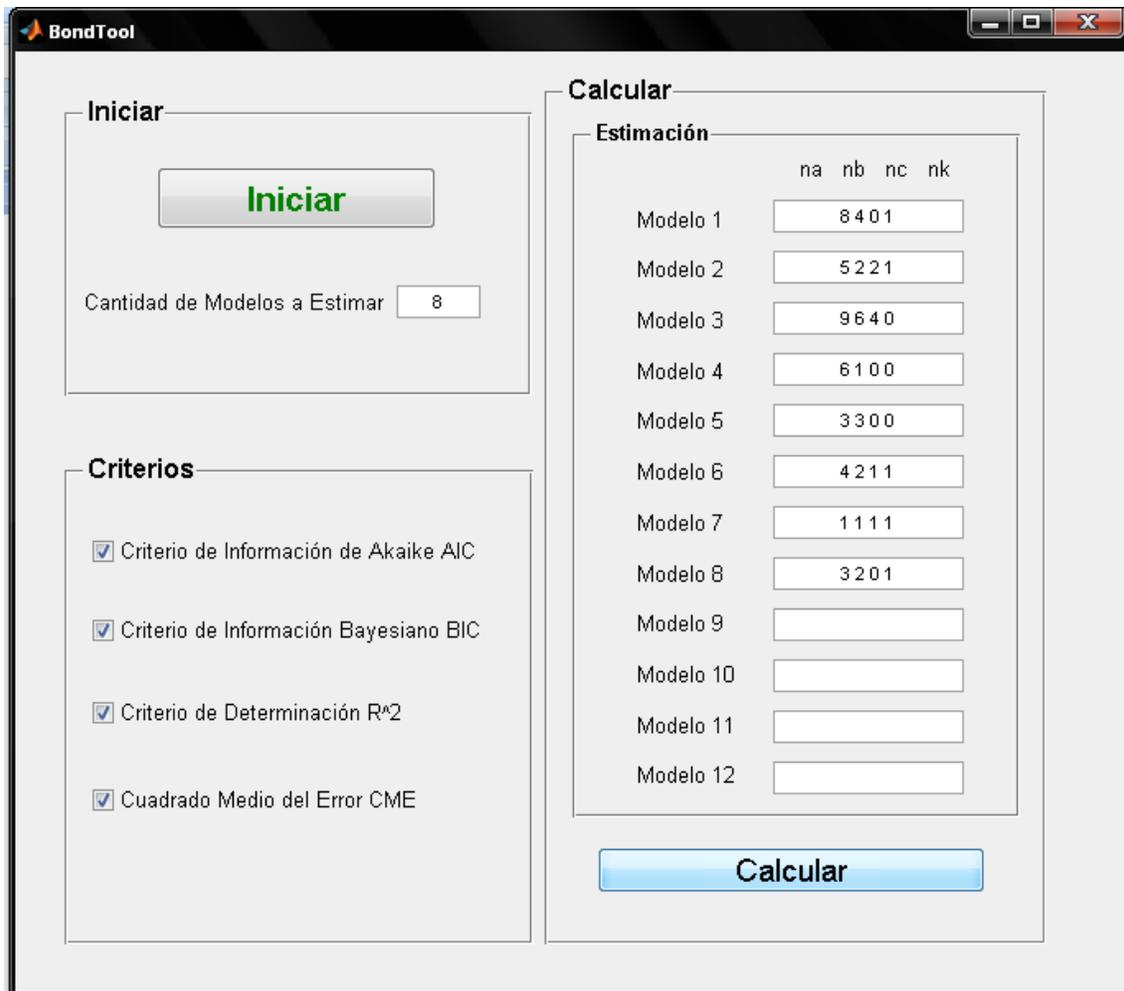


Figura 3.6. Ventana principal con los datos necesarios para calcular los criterios.

En esta ventana se puede observar cómo se seleccionan los criterios de bondad de ajuste, en este ejemplo los cuatro. Además de mostrar la forma en que deben ser introducidos correctamente los vectores de ordenes de los modelos. Por último, en la figura 3.7, se muestra una porción de la ventana “Resultados”, la cual se puede observar en su totalidad en el Anexo III .

APLICANDO ARX							
	1	2	3	4	5	6	7
AIC	-2.8368	-1.4198	-2.3666	-2.3909			
BIC	34.4345	20.3219	16.269	13.1388			
R ²	0.63459	0.88844	0.83739	0.83738			
CME	48.7066	12.6929	19.0788	18.9605			

APLICANDO ARMAX							
	1	2	3	4	5	6	7
AIC	-2.5443	-2.0971	-2.3196	-0.2529			
BIC	25.4093	56.9159	19.4221	9.0649			
R ²	0.81663	0.73279	0.73473	0.80699			
CME	22.1681	34.789	32.8469	22.5821			

Figura 3.7. Ventana “Resultados”

Como se explicó, en el capítulo anterior, el análisis del mejor de los modelos no puede resumirse a una simple comparación entre estos criterios, debido a que cada uno de ellos mide distintos aspectos. Por tanto, solo fue posible analizar y llegar a conclusiones teniendo en cuenta cada criterio por separado y así determinar cual de las estructuras es la más idónea en cada caso.

Teniendo en cuenta lo anteriormente expresado se puede decir que:

- ✓ Los modelos que mejor se ajustan a la aplicación, según cada criterio, son los que se encuentran encerrados en rojo en la figura 3.7.

❖ **Segunda prueba.**

Se desea determinar cuáles son los mejores modelos según los criterios de AIC y R^2 de un total de cuatro modelos que presentan estructura $ARMAX$.

Para ello se seleccionan los criterios e introducen los datos, como en la primera prueba, con la diferencia de que solo se marcan dos criterios y se introducen solamente cuatro vectores de ordenes. En la figura 3.8 se muestran los resultados.

APLICANDO ARMAX				
	1	2	3	4
AIC	-2.7815	-2.7652	-2.769	-2.7899
BIC				
R ²	0.70993	0.59164	0.57107	0.65307
CME				

Figura 3.8. Resultados de la segunda prueba.

La ventana principal, con los datos necesarios para realizar la prueba puede ser observada en su totalidad en el Anexo IV.

❖ **Tercera prueba.**

Por último se realizó una última prueba para cuatro modelos con estructura ARX , para ser validados utilizando los criterios BIC y CME . Los resultados obtenidos se muestran en la figura 3.9, de forma parcial. En el Anexo V, se muestra la ventana con los datos de los vectores de ordenes que se utilizaron para desarrollar la tercera prueba.

APLICANDO ARX				
	1	2	3	4
AIC				
BIC	40.5646	32.7469	6.2923	28.2548
R ²				
CME	59.2582	56.3921	20.2122	33.7485

Figura 3.9. Resultados de la tercera prueba.

3.4. Análisis económico.

Debido a que se está en presencia de una investigación cuyo mayor aporte está en el campo teórico y no se ha desempeñado con ella un proceso de aplicación práctica estable, el análisis económico se realizará teniendo en cuenta los costos por concepto de adquisición de los softwares y hardware que en ella se utilizaron.

Teniendo en cuenta los anteriormente dicho, los costos de la investigación están dados por la compra de la licencia de *MatLab*, que incluya el *Toolbox "Ident"*, además de las funciones necesarias para el trabajo. El valor es de 2480 cuc, según el listado de precios consultado (ENERG, 2009). Se incluyen además los costos de la obtención de una PC, monitor, mouse óptico, teclado, UPS, llegando a un costo total de 648 cuc (CIMEX, 2009). En otro orden se puede analizar que el salario de un profesor universitario es de 675.00 cup, por lo que si mensualmente consume el 30 % de su trabajo en el mes en función de la investigación estaríamos hablando por concepto de salario de 236.00 cup (9.44 cuc) que totalizado en los 4 meses de trabajo sería 945.00 cup (37.8 cuc), para finalmente tener un costo total de la investigación de 3165.8 en cuc.

3.5. Consideraciones parciales del capítulo.

- ❖ Los datos utilizados son reales y caracterizan un importante proceso dentro del tratamiento de residuales de una refinería de petróleo, no obstante el programa puede ser utilizado con datos de otros procesos.
- ❖ Las pruebas realizadas en la introducción de datos así como en el orden en que se introducen muestran los avisos previstos, garantizando la correcta ejecución del programa.
- ❖ En las pruebas realizadas con diferentes estructuras y órdenes, los resultados del cálculo de los criterios de validación seleccionados se caracterizan por la exactitud y rapidez.
- ❖ La interfaz da la posibilidad de utilizar diferentes juegos de datos, establecer el orden a voluntad del especialista y obtener los criterios de validación de los modelos de regresión correspondientes.

CONCLUSIONES GENERALES

- ❖ La modelación de procesos o sistemas a partir de datos reales es una necesidad de la ingeniería de control moderna ya que permite la prueba de diferentes estrategias de control sin la necesidad de utilizar el proceso o sistema real.
- ❖ Los modelos de regresión son de amplia utilización y se encuentran programados en paquetes profesionales tales como *MatLab*. Se pueden seleccionar estructuras y ordenes obteniéndose por lo tanto diferentes modelos que pueden caracterizar un mismo proceso.
- ❖ Mediante criterios de ajuste es posible determinar, de todos los modelos, cuales se adecuan más a la realidad del sistema o proceso. Cada criterio es capaz de medir características específicas del modelo por lo cual no es correcto comparar los modelos a partir de diferentes criterios de bondad.
- ❖ La interfaz confeccionada, sobre la plataforma que nos ofrece *MatLab*, ha mostrado la utilidad en la obtención de modelos, a partir de datos reales, con diferentes estructuras y ordenes determinando los valores de los criterios de bondad con lo que el especialista podrá seleccionar el que más se ajuste a sus requerimientos.

RECOMENDACIONES

- ❖ Aplicar los resultados en investigaciones futuras donde a partir de diferentes funcionales puedan ser determinados los mejores modelos teniendo en cuenta los diferentes criterios de bondad determinados mediante esta interfaz.
- ❖ Extender los resultados a la obtención de otras estructuras de modelos como ejemplo *BJ* y *OE* entre otras.

REFERENCIAS BIBLIOGRÁFICAS

- AMSTROM, K. J. (1997) *Computer Controlled Systems. Theory and Design.*, Prentice Hall.
- BABYLON (2012) "Definición de Interfaz".
- BANKS, J., CARSON, J. S. & NELSON, B. L. (2002) Verificación y Validación.
- CARBALLO, C. R. (2011) Interfaz Gráfica para la Identificación de la Plataforma de 2 Grados de Libertad., UCLV.
- CIMEX (2009) Listado de Precios. Disponible en: <http://tecun.cimex.com.cu> ed.
- DEMOYER, R. & MTICHEL, E. E. (Eds.) (2002) *Use MatLab Graphical User Interface Development enviroment for some control system applications.*
- DÍAZ, J. (2005) Introducción a la Identificación de Sistemas Dinámicos.
- ENERG (2009) MATLAB Campus License Pricing.
- ESCRIBANO, J. F. (2009) Desarrollo de una Interfaz Gráfica en MatLab para la aplicación de modelos de Regresión Local Polinómica. Madrid, Universidad Carlos III.
- ESQUEDA, J. J. (2002) MatLab e interfaces gráficas. . México.
- FRANCISCO, M. (2000) Introducción a la OOP. IN EIDOS, G. (Ed.).
- KUNUSCH, C. (2003) Identificación de Sistemas Dinámicos.
- LJUNG, L. (1999) *System Identification Theory for the User*, Prentice Hall.
- LJUNG, L. (2007) *System Identification Toolbox 7. User`s Guide*, The MathWorks, Inc.
- LÓPEZ, E. (2009) Interfaz Genérica para Sistemas de Adquisición basada en Software Libre., UCLV.
- LÓPEZ, M. E. (2000) Identificación de Sistemas. Aplicación al modelado de un motor de continua.
- MATHWORKS (2007) GUIDE toolbox. <http://www.mathwork.com/help/techdoc/matlabproductpage.html> ed.
- MIKLES, J. & FIKAR, M. (2007) *Process Modeling, Identification, and Control*, Springer Berlin Heidelberg New York.
- OGATA, K. (2006) *Ingeniería de Control Moderna*, Prentice Hall.
- ONG, E. K. & TAN, F. L. (2000) Menu driven graphical interface for MatLab Control Design. *International Journal Engineering Education.*
- POSADA, S. L. & NOGUERA, R. R. (2007) Comparación de modelos matemáticos: una aplicación en la evaluación de alimentos para animales. *Revista Colombiana de Ciencias Pecuarias.*
- RODRÍGUEZ, S. (2011) Evaluación de dosis radiológicas mediante técnicas de procesamiento digital de imágenes. *CEETI*. Santa Clara, Cuba, Universidad Central "Marta Abreu" de Las Villas.
- VALERIANO, Y. & ORIA, H. J. (2012) VASmodel. Interfaz gráfica para modelo dinámico de un AUV. *VI Conferencia UCIENCIA 2012*. Universidad de Ciencias Informáticas, Habana, Cuba.

ANEXOS

Anexo I: Datos de la planta.

Entrada PRBS	Salida flujo de recirculación a flotadores
-0,1	0
-0,1	0,083437651
-0,1	0,083437651
-0,1	0,072477289
-0,1	0,072477289
-0,1	0,055833895
-0,1	0,098241538
-0,1	0,134021908
-0,1	0,170391545
-0,1	0,187856928
-0,1	0,174797982
-0,1	0,165783405
-0,1	0,161561072
-0,1	0,123850368
-0,1	0,152941495
-0,1	0,120924264
-0,1	0,071514055
-0,1	0,055603188
-0,1	0,056574881
-0,1	0,093073636
-0,1	0,055710267
-0,1	0,041486327
-0,1	0,058264051
-0,1	0,043533397
-0,1	0,117299877
-0,1	0,089937426
-0,1	0,062972337
-0,1	0,068662949
-0,1	0,050806701
-0,1	0,050067037
-0,1	0,072765589
-0,1	0,051634025
-0,1	0,059694264
-0,1	0,032717761
-0,1	0,055839866
-0,1	0,084432594
-0,1	0,076749079
-0,1	0,095279269

ANEXOS

-0,1	0,05891085
-0,1	0,031492244
-0,1	0,054481581
-0,1	0,02971884
-0,1	0,018673753
-0,1	0,010822703
-0,1	0,031815674
-0,1	0,064176351
-0,1	0,044008419
-0,1	0,03807772
-0,1	0,044522922
-0,1	0,051216751
10	0,026911924
10	0,016910033
10	1,419467568
10	5,690325737
10	10,73308563
10	15,49450207
10	18,29346848
10	20,27998161
10	21,36013031
10	22,39792252
10	22,89549446
10	23,06914711
10	23,19935799
10	23,34850502
10	23,51872444
10	23,52942848
10	23,29836082
10	23,25537872
10	23,41269875
10	23,55482101
-0,1	23,61941147
-0,1	23,44628334
-0,1	19,88837433
-0,1	15,39073181
-0,1	10,36756897
-0,1	7,253555775
-0,1	4,03324461
-0,1	2,493487597
-0,1	1,871989846
-0,1	1,243091345
-0,1	0,849529564

ANEXOS

-0,1	0,559703052
-0,1	0,484443486
-0,1	0,430838943
-0,1	0,447675526
-0,1	0,3932257
-0,1	0,337296695
-0,1	0,330316782
-0,1	0,367379755
-0,1	0,307557076
10	0,296028793
10	0,312172443
10	2,396435976
10	6,164606094
10	11,81780052
10	16,29794693
10	18,8725338
10	20,70899963
10	21,87607956
10	22,55501556
-0,1	23,08005142
-0,1	23,19221878
-0,1	19,48604584
-0,1	14,47544193
-0,1	10,84659386
-0,1	7,608232975
-0,1	4,911119461
-0,1	3,306224585
-0,1	2,322966814
-0,1	1,403148532
10	0,972662866
10	0,626317859
10	2,426110744
10	6,630412579
10	11,48007584
10	15,35776424
10	18,81150246
10	20,58694839
10	22,00175858
10	22,6421032
10	23,30513382
10	23,70137596
10	23,62082672
10	23,56492233

10	23,57911682
10	23,64101219
10	23,60725975
10	23,48132324
10	23,44054794
10	23,49955559
-0,1	23,61227417
-0,1	23,42806244
-0,1	19,79892921
-0,1	14,1971035
-0,1	10,34166145
-0,1	6,992571831
-0,1	4,65237093
-0,1	2,953194857
-0,1	1,896676898
-0,1	1,280676126
10	0,890970469
10	0,69909519
10	2,401872158
10	6,71236372
10	11,94264221
10	15,86165047
10	19,67446327
10	21,43455124
10	22,67834282
10	23,25519371
10	23,70491982
10	24,02350807
10	23,91971397
10	23,97410774
10	24,1492691
10	24,36916542
10	24,51059341
10	24,57595634
10	24,68298149
10	24,77156639
10	24,80916214
10	24,83934784
10	24,79515076
10	24,76144409
10	24,66693115

Anexo II: Ventanas de la aplicación BondTool.

The screenshot shows the main interface of the BondTool application. It is divided into two main sections: 'Iniciar' and 'Calcular'.

Iniciar: Contains a large green 'Iniciar' button and a text input field labeled 'Cantidad de Modelos a Estimar'.

Calcular: Contains a sub-section 'Estimación' with a table of input fields for 12 models. The columns are labeled 'na', 'nb', 'nc', and 'nk'. Below the table is a 'Calcular' button.

Criterios: A section with four checkboxes for selection criteria: 'Criterio de Información de Akaike AIC', 'Criterio de Información Bayesiano BIC', 'Criterio de Determinación R²', and 'Cuadrado Medio del Error CME'.

a) Ventana Inicial

The screenshot shows the 'Resultados' window, which displays a grid for applying ARX and ARMAX models. The grid is organized into two main sections: 'APLICANDO ARX' and 'APLICANDO ARMAX'.

APLICANDO ARX: A grid with 4 rows (AIC, BIC, R², CME) and 12 columns (1-12). Each cell contains an empty input field.

APLICANDO ARMAX: A grid with 4 rows (AIC, BIC, R², CME) and 12 columns (1-12). Each cell contains an empty input field.

b) Ventana "Resultados".

Anexo IV: Datos de la segunda prueba.

The screenshot shows the BondTool software interface. It is divided into two main sections: 'Iniciar' and 'Calcular'.

Iniciar

There is a large green button labeled 'Iniciar'. Below it, there is a label 'Cantidad de Modelos a Estimar' followed by a text input field containing the number '4'.

Calcular

Under the 'Calcular' heading, there is a sub-section titled 'Estimación'. It contains a table with 12 rows and 4 columns. The columns are labeled 'na', 'nb', 'nc', and 'nk'. The rows are labeled 'Modelo 1' through 'Modelo 12'. The data for the first four models is as follows:

	na	nb	nc	nk
Modelo 1	5	2	4	1
Modelo 2	7	3	2	0
Modelo 3	6	4	2	1
Modelo 4	5	5	1	0
Modelo 5				
Modelo 6				
Modelo 7				
Modelo 8				
Modelo 9				
Modelo 10				
Modelo 11				
Modelo 12				

At the bottom of the 'Calcular' section, there is a blue button labeled 'Calcular'.

Anexo V: Datos de la tercera prueba.

The screenshot shows the BondTool software interface. The window title is 'BondTool'. The interface is divided into two main sections: 'Iniciar' and 'Calcular'.

Iniciar

There is a large green button labeled 'Iniciar'. Below it, there is a label 'Cantidad de Modelos a Estimar' followed by a text input field containing the number '4'.

Calcular

Under the 'Calcular' section, there is a sub-section titled 'Estimación'. It contains a table with 12 rows and 4 columns. The columns are labeled 'na', 'nb', 'nc', and 'nk'. The rows are labeled 'Modelo 1' through 'Modelo 12'. The data for the first four models is as follows:

	na	nb	nc	nk
Modelo 1	10	4	0	2
Modelo 2	6	5	0	3
Modelo 3	1	1	0	0
Modelo 4	7	3	0	1
Modelo 5				
Modelo 6				
Modelo 7				
Modelo 8				
Modelo 9				
Modelo 10				
Modelo 11				
Modelo 12				

Below the table is a large button labeled 'Calcular'.

Criterios

Under the 'Criterios' section, there are four checkboxes:

- Criterio de Información de Akaike AIC
- Criterio de Información Bayesiano BIC
- Criterio de Determinación R²
- Cuadrado Medio del Error CME

Anexo VI: Fragmento de Código de los edit text del panel “Estimación”.

```
m1=get(hObject,'String')
if ~isempty(m1) %Pregunta si no está vacío m1, si
contiene algo escrito lo convierte
    m1=str2num(m1); %a doble, mide el vector para saber
la cantidad de columnas que tiene
    [f1,c1]=size(m1);
    if isempty(m1)|| c1 ~= 4 %Si esta vacío o la cantidad de
columnas es diferente de 4 da un error y limpia el edit.
        errordlg('El vector de parámetros del modelo 1 debe ser insertado
correctamente.','Error');
        set(hObject,'String','');
        m1=0;
    elseif c1==4 && (m1(1,1)<=0 ||m1(1,2)<=0 ||m1(1,3)<0
||m1(1,4)<0|| (m1(1,1)-round(m1(1,1)))~=0|| (m1(1,2)-
round(m1(1,2)))~=0|| (m1(1,3)-round(m1(1,3)))~=0|| (m1(1,4)-
round(m1(1,4)))~=0)
        errordlg('Los valores del vector de parámetros deben ser números
enteros y positivos.','Error');
        set(hObject,'String','');
        m1=0;
    end
elseif isempty(m1)
    m1=0;
end
handles.mod1=m1;
guidata(hObject,handles);

m2=get(hObject,'String');
if ~isempty(m2)
    m2=str2num(m2);
    [f2,c2]=size(m2);
    if isempty(m2)|| c2 ~= 4
        errordlg('El vector de parámetros del modelo 2 debe ser insertado
correctamente.','Error');
        set(hObject,'String','');
        m2=0;
    elseif c2==4 && (m2(1,1)<=0 ||m2(1,2)<=0 ||m2(1,3)<0
||m2(1,4)<0|| (m2(1,1)-round(m2(1,1)))~=0|| (m2(1,2)-
round(m2(1,2)))~=0|| (m2(1,3)-round(m2(1,3)))~=0|| (m2(1,4)-
round(m2(1,4)))~=0)
        errordlg('Los valores del vector de parámetros deben ser números
enteros y positivos.','Error');
        set(hObject,'String','');
        m2=0;
    end
elseif isempty(m2)
    m2=0;
end
handles.mod2=m2;
guidata(hObject,handles);
```

Anexo VII: Código del botón “Iniciar”.

```
warndlg('Para ejecutar correctamente la aplicación debe tener el juego de  
datos almacenado en un archivo con nombre ``planta`` y extensión ``.mat``  
ubicado en la carpeta MATLAB de Mis Documentos. En dicho archivo deben  
estar almacenadas las variables de entrada y salida en los arreglos ``u``  
y ``y`` respectivamente.','Aviso');  
set(findobj('Tag','cantidad'),'Enable','on')  
set(findobj('Tag','estimar'),'Enable','on')  
set(findobj('Tag','checkbox1'),'Enable','on')  
set(findobj('Tag','checkbox2'),'Enable','on')  
set(findobj('Tag','checkbox3'),'Enable','on')  
set(findobj('Tag','checkbox4'),'Enable','on')
```

Anexo VIII: Código del edit text “Cantidad de modelos”.

```
ca=get(hObject,'String')
if ~isempty(ca)
    can=str2double(ca);
    if isnan(can)
        warndlg('El valor correspondiente a la cantidad de modelos a
estimar debe ser numérico.','Aviso');
        set(hObject,'String','');
        can=0;
        set(findobj('Tag','modelo1'),'Enable','inactive');
        set(findobj('Tag','modelo2'),'Enable','inactive');
        .
        .
        .
        set(findobj('Tag','modelo1'),'String','');
        set(findobj('Tag','modelo2'),'String','');
        set(findobj('Tag','modelo3'),'String','');
        .
        .
        .
        handles.mod1=0;
        handles.mod2=0;
        .
        .
        .
    elseif
can~=1&&can~=2&&can~=3&&can~=4&&can~=5&&can~=6&&can~=7&&can~=8&&can~=9&&c
an~=10&&can~=11&&can~=12
        warndlg('El valor correspondiente a la cantidad de modelos a
estimar es un entero que se encuentra en el rango de 1 a 12. ','Aviso');
        set(hObject,'String','');
        can=0;
        set(findobj('Tag','modelo1'),'Enable','inactive');
        set(findobj('Tag','modelo2'),'Enable','inactive');
        set(findobj('Tag','modelo3'),'Enable','inactive');
        .
        .
        .
        set(findobj('Tag','modelo1'),'String','');
        set(findobj('Tag','modelo2'),'String','');
        set(findobj('Tag','modelo3'),'String','');
        .
        .
        .
        handles.mod1=0;
        handles.mod2=0;
        handles.mod3=0;
        .
        .
        .
    elseif can==1
        set(findobj('Tag','modelo1'),'Enable','on');
        set(findobj('Tag','modelo2'),'Enable','inactive');
        set(findobj('Tag','modelo3'),'Enable','inactive');
```

ANEXOS

```
set(findobj('Tag','modelo4'),'Enable','inactive');
set(findobj('Tag','modelo5'),'Enable','inactive');
set(findobj('Tag','modelo6'),'Enable','inactive');
.
.
.
set(findobj('Tag','modelo2'),'String','');
set(findobj('Tag','modelo3'),'String','');
set(findobj('Tag','modelo4'),'String','');
set(findobj('Tag','modelo5'),'String','');
.
.
.
handles.mod2=0;
handles.mod3=0;
handles.mod4=0;
handles.mod5=0;
handles.mod6=0;
handles.mod7=0;
.
.
.
elseif can==2
    set(findobj('Tag','modelo1'),'Enable','on');
    set(findobj('Tag','modelo2'),'Enable','on');
    set(findobj('Tag','modelo3'),'Enable','inactive');
    set(findobj('Tag','modelo4'),'Enable','inactive');
    set(findobj('Tag','modelo5'),'Enable','inactive');
    set(findobj('Tag','modelo6'),'Enable','inactive');
    .
    .
    .
    set(findobj('Tag','modelo3'),'String','');
    set(findobj('Tag','modelo4'),'String','');
    set(findobj('Tag','modelo5'),'String','');
    .
    .
    .
    handles.mod3=0;
    handles.mod4=0;
    handles.mod5=0;
    .
    .
    .
    handles.cant=can;
guidata(hObject,handles);
```

Anexo IX: Código del botón**Calcular”.**

```
a=handles.mod1;
b=handles.mod2;
c=handles.mod3;
d=handles.mod4;
e=handles.mod5;
f=handles.mod6;
g=handles.mod7;
h=handles.mod8;
k=handles.mod9;
l=handles.mod10;
m=handles.mod11;
o=handles.mod12;
p=handles.cant;
global check1
global check2
global check3
global check4
check1=0
check2=0
check3=0
check4=0
global AICarmax1
global AICarmax2
global AICarmax3
global AICarmax4
global AICarmax5
global AICarmax6
global AICarmax7
global AICarmax8
global AICarmax9
global AICarmax10
global AICarmax11
global AICarmax12
global BICarmax1
global BICarmax2
global BICarmax3
global BICarmax4
global BICarmax5
global BICarmax6
global BICarmax7
global BICarmax8
global BICarmax9
global BICarmax10
global BICarmax11
global BICarmax12
global CMEarmax1
global CMEarmax2
global CMEarmax3
global CMEarmax4
global CMEarmax5
global CMEarmax6
global CMEarmax7
global CMEarmax8
global CMEarmax9
global CMEarmax10
global CMEarmax11
global CMEarmax12
global R2armx1
global R2armx2
global R2armx3
global R2armx4
global R2armx5
```


ANEXOS

```
datosv=datos((mitad+1):length(planta));
if p==0 &&
~get(findobj('Tag','checkbox1'),'Value')&&~get(findobj('Tag','checkbox2')
,'Value')&&~get(findobj('Tag','checkbox3'),'Value')&&~get(findobj('Tag','
checkbox4'),'Value')
    warndlg('Introduzca el número de modelos a estimar y seleccione al
menos uno de los criterios propuestos.','Aviso')
    return
end
if p==0
    warndlg('Introduzca primeramente el número de modelos a
estimar.','Aviso')
    return
end
if
~get(findobj('Tag','checkbox1'),'Value')&&~get(findobj('Tag','checkbox2')
,'Value')&&~get(findobj('Tag','checkbox3'),'Value')&&~get(findobj('Tag','
checkbox4'),'Value')
    warndlg('Seleccione al menos uno de los criterios
propuestos.','Aviso')
    return
end
if get(findobj('Tag','checkbox1'),'Value')
    check1=1
end
if get(findobj('Tag','checkbox2'),'Value')
    check2=1
end
if get(findobj('Tag','checkbox3'),'Value')
    check3=1
end
if get(findobj('Tag','checkbox4'),'Value')
    check4=1
end
if p==2&&(all(a==0)||all(b==0))
    warndlg('Introduzca correctamente los parámetros según el número de
modelos a estimar.','Aviso')
    return
elseif p==2
    Procesando
    ma=[a;b]
    k=0;
    l=0;
    for i=1:1:2
        if ma(i,3)==0
            k=k+1
            if k==1
                marx1=arx(datose,[ma(i,1) ma(i,2) ma(i,4)])
                AICarx1=aic(marx1);
                darx1=ma(i,1)+ma(i,2)+ma(i,3);
                N=length(y);
                LLFarx1=darx1-AICarx1/2;
                [AICarx1,BICarx1]=aicbic(LLFarx1,darx1,N)
                earx1=pe(marx1,planta);
                SCEarx1=0;
                for v=1:1:length(y)
```

```

        SCEarx1=SCEarx1+earx1(v)*earx1(v);
    end
    CMEarx1=SCEarx1/(length(y)-darx1)
    ysarx1=y'-earx1;
    ymarx1=mean(ysarx1);
    rvmyarx1=y-ymarx1;
    SCTarx1=0;
    for ns=1:1:length(y)
        SCTarx1=SCTarx1+rvmyarx1(ns)*rvmyarx1(ns);
    end
    R2arx1=1-(SCEarx1/SCTarx1)
elseif k==2
    marx2=arx(datose,[ma(i,1) ma(i,2) ma(i,4)])
    AICarx2=aic(marx2);
    darx2=ma(i,1)+ma(i,2)+ma(i,3);
    N=length(y);
    LLFarx2=darx2-AICarx2/2;
    [AICarx2,BICarx2]=aicbic(LLFarx2,darx2,N)
    earx2=pe(marx2,planta);
    SCEarx2=0;
    for v=1:1:length(y)
        SCEarx2=SCEarx2+earx2(v)*earx2(v);
    end
    CMEarx2=SCEarx2/(length(y)-darx2)
    ysarx2=y'-earx2;
    ymarx2=mean(ysarx2);
    rvmyarx2=y-ymarx2;
    SCTarx2=0;
    for ns=1:1:length(y)
        SCTarx2=SCTarx2+rvmyarx2(ns)*rvmyarx2(ns);
    end
    R2arx2=1-(SCEarx2/SCTarx2)
end
elseif ma(i,3)>=0
    l=l+1
    if l==1
        marmax1=armax(datose,[ma(i,1) ma(i,2) ma(i,3) ma(i,4)])
        AICarmax1=aic(marmax1);
        darmax1=ma(i,1)+ma(i,2)+ma(i,3);
        N=length(y);
        LLFarmax1=darmax1-AICarmax1/2;
        [AICarmax1,BICarmax1]=aicbic(LLFarmax1,darmax1,N)
        earmax1=pe(marmax1,planta);
        SCEarmax1=0;
        for v=1:1:length(y)
            SCEarmax1=SCEarmax1+earmax1(v)*earmax1(v);
        end
        CMEarmax1=SCEarmax1/(length(y)-darmax1)
        ysarmax1=y'-earmax1;
        ymarmax1=mean(ysarmax1);
        rvmyarmax1=y-ymarmax1;
        SCTarmax1=0;
        for ns=1:1:length(y)
            SCTarmax1=SCTarmax1+rvmyarmax1(ns)*rvmyarmax1(ns);
        end
        R2armax1=1-(SCEarmax1/SCTarmax1)
    elseif l==2

```

```

marmax2=armax(datose,[ma(i,1) ma(i,2) ma(i,3) ma(i,4)])
AICarmax2=aic(marmax2);
darmax2=ma(i,1)+ma(i,2)+ma(i,3);
N=length(y);
LLFarmax2=darmax2-AICarmax2/2;
[AICarmax2,BICarmax2]=aicbic(LLFarmax2,darmax2,N)
earmax2=pe(marmax2,planta);
SCEarmax2=0;
for v=1:1:length(y)
    SCEarmax2=SCEarmax2+earmax2(v)*earmax2(v);
end
CMEarmax2=SCEarmax2/(length(y)-darmax2)
ysarmax2=y'-earmax2;
ymarmax2=mean(ysarmax2);
rvmyarmax2=y-ymarmax2;
SCTarmax2=0;
for ns=1:1:length(y)
    SCTarmax2=SCTarmax2+rvmyarmax2(ns)*rvmyarmax2(ns);
end
R2armax2=1-(SCEarmax2/SCTarmax2)
end
end
end
if p==1&&(all(a==0))
    warndlg('Introduzca correctamente los parámetros según el número de
modelos a estimar.','Aviso')
    return
elseif p==1
    Procesando
    if a(1,3)==0
        marx1=arx(datose,[a(1,1) a(1,2) a(1,4)])
        AICarx1=aic(marx1);
        darx1=a(1,1)+a(1,2)+a(1,3);
        N=length(y);
        LLFarx1=darx1-AICarx1/2;
        [AICarx1,BICarx1]=aicbic(LLFarx1,darx1,N)
        earx1=pe(marx1,planta);
        SCEarx1=0;
        for v=1:1:length(y)
            SCEarx1=SCEarx1+earx1(v)*earx1(v);
        end
        CMEarx1=SCEarx1/(length(y)-darx1)
        ysarx1=y'-earx1;
        ymarx1=mean(ysarx1);
        rvmyarx1=y-ymarx1;
        SCTarx1=0;
        for ns=1:1:length(y)
            SCTarx1=SCTarx1+rvmyarx1(ns)*rvmyarx1(ns);
        end
        R2arx1=1-(SCEarx1/SCTarx1)
    elseif a(1,3)>=0
        marmax1=armax(datose,[a(1,1) a(1,2) a(1,3) a(1,4)])
        AICarmax1=aic(marmax1);
        darmax1=a(1,1)+a(1,2)+a(1,3);
        N=length(y);
        LLFarmax1=darmax1-AICarmax1/2;

```

ANEXOS

```
[AICarmax1,BICarmax1]=aicbic(LLFarmax1,darmax1,N)
earmax1=pe(marmax1,planta);
SCEarmax1=0;
    for v=1:1:length(y)
        SCEarmax1=SCEarmax1+earmax1(v)*earmax1(v);
    end
    CMEarmax1=SCEarmax1/(length(y)-darmax1)
    ysarmax1=y'-earmax1;
    ymarmax1=mean(ysarmax1);
    rvmyarmax1=y-ymarmax1;
    SCTarmax1=0;
    for ns=1:1:length(y)
        SCTarmax1=SCTarmax1+rvmyarmax1(ns)*rvmyarmax1(ns);
    end
    R2armax1=1-(SCEarmax1/SCTarmax1)
end
end

handles.AICarmax1=AICarmax1
handles.AICarmax2=AICarmax2
handles.AICarmax3=AICarmax3
handles.AICarmax4=AICarmax4
handles.AICarmax5=AICarmax5
handles.AICarmax6=AICarmax6
handles.AICarmax7=AICarmax7
handles.AICarmax8=AICarmax8
handles.AICarmax9=AICarmax9
handles.AICarmax10=AICarmax10
handles.AICarmax11=AICarmax11
handles.AICarmax12=AICarmax12
handles.BICarmax1=BICarmax1
handles.BICarmax2=BICarmax2
handles.BICarmax3=BICarmax3
handles.BICarmax4=BICarmax4
handles.BICarmax5=BICarmax5
handles.BICarmax6=BICarmax6
handles.BICarmax7=BICarmax7
handles.BICarmax8=BICarmax8
handles.BICarmax9=BICarmax9
handles.BICarmax10=BICarmax10
handles.BICarmax11=BICarmax11
handles.BICarmax12=BICarmax12
handles.CMEarmax1=CMEarmax1
handles.CMEarmax2=CMEarmax2
handles.CMEarmax3=CMEarmax3
handles.CMEarmax4=CMEarmax4
handles.CMEarmax5=CMEarmax5
handles.CMEarmax6=CMEarmax6
handles.CMEarmax7=CMEarmax7
handles.CMEarmax8=CMEarmax8
handles.CMEarmax9=CMEarmax9
handles.CMEarmax10=CMEarmax10
handles.CMEarmax11=CMEarmax11
handles.CMEarmax12=CMEarmax12
handles.R2armax1=R2armax1
handles.R2armax2=R2armax2
handles.R2armax3=R2armax3

handles.R2armax4=R2armax4
handles.R2armax5=R2armax5
handles.R2armax6=R2armax6
handles.R2armax7=R2armax7
handles.R2armax8=R2armax8
handles.R2armax9=R2armax9
handles.R2armax10=R2armax10
handles.R2armax11=R2armax11
handles.R2armax12=R2armax12
handles.AICarx1=AICarx1
handles.AICarx2=AICarx2
handles.AICarx3=AICarx3
handles.AICarx4=AICarx4
handles.AICarx5=AICarx5
handles.AICarx6=AICarx6
handles.AICarx7=AICarx7
handles.AICarx8=AICarx8
handles.AICarx9=AICarx9
handles.AICarx10=AICarx10
handles.AICarx11=AICarx11
handles.AICarx12=AICarx12
handles.BICarx1=BICarx1
handles.BICarx2=BICarx2
handles.BICarx3=BICarx3
handles.BICarx4=BICarx4
handles.BICarx5=BICarx5
handles.BICarx6=BICarx6
handles.BICarx7=BICarx7
handles.BICarx8=BICarx8
handles.BICarx9=BICarx9
handles.BICarx10=BICarx10
handles.BICarx11=BICarx11
handles.BICarx12=BICarx12
handles.CMEarx1=CMEarx1
handles.CMEarx2=CMEarx2
handles.CMEarx3=CMEarx3
handles.CMEarx4=CMEarx4
handles.CMEarx5=CMEarx5
handles.CMEarx6=CMEarx6
handles.CMEarx7=CMEarx7
```

ANEXOS

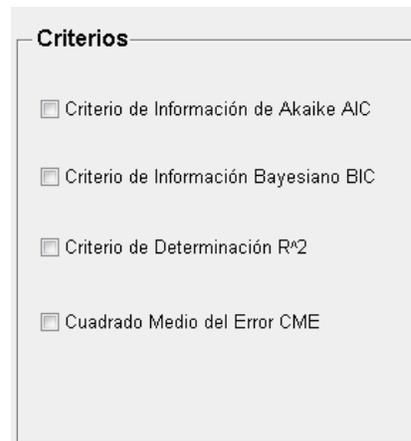
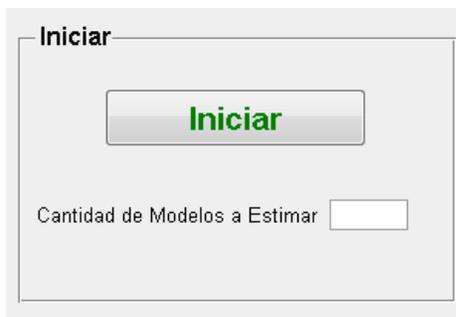
```
handles.CMEarx8=CMEarx8
handles.CMEarx9=CMEarx9
handles.CMEarx10=CMEarx10
handles.CMEarx11=CMEarx11
handles.CMEarx12=CMEarx12
handles.R2arx1=R2arx1
handles.R2arx2=R2arx2
handles.R2arx3=R2arx3
handles.R2arx4=R2arx4
handles.R2arx5=R2arx5
```

```
handles.R2arx6=R2arx6
handles.R2arx7=R2arx7
handles.R2arx8=R2arx8
handles.R2arx9=R2arx9
handles.R2arx10=R2arx10
handles.R2arx11=R2arx11
handles.R2arx12=R2arx12
guidata(hObject,handles);
Resultados
```

Anexo X: Ayuda de la aplicación BondTool.

En este documento se presentan una serie de indicaciones y pasos a seguir, con vista a lograr una mejor utilización de la aplicación *BondTool*.

1. Para el correcto funcionamiento de la interfaz debe tener los datos con los que trabajara, almacenados en un fichero con extensión “.mat” y nombre “planta”, el mismo debe contener como variables de entrada y salida a “u” y “y” respectivamente.
2. En el panel nombrado “Iniciar”, presione el botón que contiene dicho nombre.
3. Seguidamente debe introducir la cantidad de modelos que desea estimar en el *edit text* con dicho nombre. Este valor debe ser un número entero en el rango de 1 a 12.



4. Seleccione al menos uno de los criterios de bondad de ajuste mostrados en el panel “Criterios”, para obtener un mejor resultado se le sugiere marcar los cuatro criterios.
5. Según la cantidad de modelos que desee, en el panel “Parámetros”, introduzca los vectores de parámetros $[na \ nb \ nc \ nk]$, teniendo en cuenta la estructura de modelos que desea; en caso de que sea ARX debe introducir en el lugar de “nc” debe pasar como valor un cero. Además no puede introducir valores negativos, letras o números que no sean enteros.
6. Presione el botón “Calcular” para ejecutar la aplicación y pasar a la interfaz de visualización.