

# **CaToKi Interfaz gráfica de identificación de sistemas en lazo abierto SISO aplicado a procesos de laboratorio.**

## **CaToKi Graphic interface for Open-Loop System Identification SISO Applied to Laboratory Processes.**

Francisco Ronay López Estrada (1)  
Tecnológico Nacional de México. Instituto Tecnológico de Tuxtla Gutiérrez.  
frlopez@ittg.edu.mx.

Álvaro Hernández Sol (2), I. T. de Tuxtla Gutiérrez, ing\_sol10@yahoo.com.mx

Joaquín Eduardo Domínguez Zenteno (3). I. T. de Tuxtla Gutiérrez, quinolalo@gmail.com

José Ángel Zepeda Hernández (4), I. T. de Tuxtla Gutiérrez, jzepeda@ittg.edu.com.

Ildeberto De Los Santos (5), I. T. de Tuxtla Gutiérrez, idlsantos@msn.com.

Raúl Moreno Rincón (6), I. T. de Tuxtla Gutiérrez, rmoreno@ittg.edu.mx

**Artículo recibido en junio 23, 2015; aceptado en septiembre 07, 2015.**

### **Resumen.**

En este trabajo se presenta el desarrollo de un prototipo tipo interfaz gráfica de usuario (GUI del inglés *Graphic User Interface*) para la identificación de sistemas tipo SISO (*single input- single output*). El GUI puede utilizarse como una alternativa a la interfaz *IDENT* de *MATLAB*. Con los algoritmos propuestos es posible identificar y estimar los parámetros de modelos matemáticos (tipo función de transferencia de orden “n”) de procesos reales tales como intercambiadores de calor, sistemas electrónicos, mecánicos, neumáticos, eléctricos, entre otros. La identificación se realiza tipo caja negra utilizando estructuras conocidas de identificación paramétrica tales como *ARX*, *ARMAX*, *Output Error* y *Box-Jenkins*. La validación de los modelos realiza mediante gráficas de respuesta temporal (respuesta al escalón, impulso y de comparación) y gráficas de tipo frecuencial (*Bode*, *LETFE* y residuos), además para verificar la estabilidad del sistema identificado se utilizan gráficas de polos y ceros. El *toolkit* se encuentra disponible de forma gratuita en los repositorios de intercambio de archivos de *mathworks.com*. Para validar la efectividad del trabajo propuesto, se analiza y desarrolla la identificación de un intercambiador de calor considerando datos reales tomados en el laboratorio de control de procesos de Instituto Tecnológico de Tuxtla Gutiérrez (ITTG).

**Palabras clave:** Estimación de Parámetros, Identificación de Sistemas, *ARX*, *ARMAX*, *Output Error*, *Box Jenkins*.

### **Abstract.**

This work proposes a prototype of a graphical user interface (GUI) for system identification of SISO (Single input-Single output) systems. The GUI is proposed as an alternative to the graphical interface *IDENT* of *MATLAB*. With the proposed algorithms it is possible to estimate and identify parameters of mathematical models (transfer functions of order “n”) of real processes. These algorithms can be used to identify real processes as heat-exchangers, electronic systems, mechanic systems, pneumatic systems, electric systems, among others. The identification is made by considering the black-box approach with standard methods as *ARX*, *ARMAX*, *Output-Error*, and *Box-Jenkins*. The model validation is done by considering transient analysis, impulse and step responses, and compare plots.

Frequency analysis is also considered as Bode, residual and LETFE plots. The Toolkit is freely available on the Mathworks file-exchange web-page. To validate the effectiveness of the proposed work, we have considered the parameter identification of a heat-exchanger system located in *the Instituto Tecnológico de Tuxtla Gutiérrez (ITTG)*.

**Keywords:** Parameter estimation, system identification, ARX, ARMAX, output error, Box Jenkins.

## 1. Introducción.

Debido a que los sistemas dinámicos abundan en nuestro medio ambiente, las técnicas de identificación de sistemas han cobrado gran relevancia en diversas áreas del conocimiento (ingenierías, economía, biotecnología, entre otros.), donde se requiere de un modelo preciso para fines de análisis, predicción, simulación, diseño y control. En particular, las técnicas de control actuales requieren de modelos matemáticos cada vez más exactos para el análisis y el diseño (Kunusch, 2003). Siendo que en muchos casos, tales modelos no pueden ser obtenidos en forma sencilla y económica a partir de las leyes que rigen cada proceso. Juega un papel decisivo la Identificación de Sistemas Dinámicos. Esta herramienta es capaz de proporcionar los métodos necesarios para obtener de manera relativamente sencilla los modelos matemáticos buscados con un alto grado de exactitud (Åström and Eykhoff, 1971). Estimar los parámetros es contar con un modelo que se comporte como el sistema real, es de vital importancia en el mundo del diseño de controladores, sobre todo aquellos basados en modelos. La *GUI* de identificación se presenta en este contexto como una ventana alternativa al *IDENT* de *MATLAB* (Ljung,2015), con una interfaz más amigable y está diseñado específicamente para identificación en lazo abierto de sistemas *SISO*. Es una herramienta para obtener modelos matemáticos de procesos reales y para facilitar la comprensión de todos los conceptos matemáticos. *CaToKi* (*Calculation Tool-Kit*) permite obtener modelos matemáticos en forma de función de transferencia, el modelo obtenido puede utilizarse para diseñar controladores y aplicarlos a los procesos. Con *CaToKi* los polinomios de la función de transferencia pueden obtenerse eligiendo una de las siguientes estructuras: *ARX*, *ARMAX*, *Output Error* y *Box-Jenkins*.

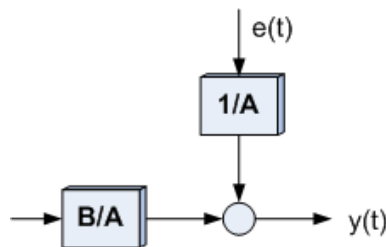
## 2. Métodos.

### Modelos Dinámicos.

El modelo matemático de un sistema consiste en una descripción conveniente de algunas de sus propiedades y de acuerdo a un propósito particular. El modelo no necesita ser una exacta descripción del sistema y el usuario debe saber esto para poder llevar a cabo su propósito. Los modelos que aquí se analizan se presentan como funciones de transferencia. Estas funciones de transferencia toman la forma de acuerdo a las siguientes estructuras de modelado:

#### A. *ARX*

Es la relación entrada-salida más simple que se puede obtener (Fig.1), el modelo *ARX*, no es el modelo más natural, ya que el ruido es sumado a la salida, luego de pasar a través del denominador del sistema dinámico.



**Figura 1.** Estructura *ARX*.

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n_a) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (2)$$

Debido a que el término de ruido blanco  $e(t)$  entra como un error directo en la ecuación en diferencias, el modelo 2 es también conocido como modelo o estructura de ecuación de error. En este caso los parámetros a ajustar serán:

$$\theta = [ a_1 \ a_2 \ a_3 \dots a_{n_a} \ b_1 \ b_2 \ b_3 \dots b_{n_b} ]$$

Donde los polinomios son:

$$A(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 q^{-1} + b_2 q^{-2} + \dots + b_{n_b} q^{-n_b}$$

Resultando:

$$y(t) = (B(q)/A(q)) + (1/A(q)) e(t) \quad (3)$$

A este modelo se lo conoce como estructura "ARX", donde "AR" hace referencia a la parte autorregresiva  $A(q)$ .  $y(t)$  y "X" a la entrada extra (*extra input*)  $B(q).u(t)$  también conocida como variable exógena. El flujo de señal de la Figura 1 nos indica que posiblemente éste no sea el modelo más natural desde un 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 primera elección en muchas aplicaciones y es que la forma del predictor define una regresión lineal.

### B. Estructura ARMAX.

La principal desventaja del modelo 2 reside en la escasez o falta de libertad en la descripción del término de perturbación. Sin embargo, es posible incorporar mayor flexibilidad al modelado si es que agregamos un término conocido como media en movimiento (*moving average*) del ruido blanco.

$$y(t) + a_1 y(t-1) + \dots + a_n y(t-n_a) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t) + \dots + c_{n_c} e(t-n_c) \quad (4)$$

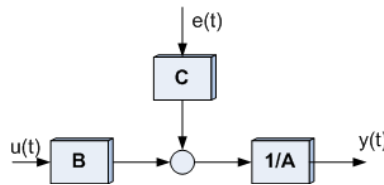


Figura 2. Estructura ARMAX.

Introduciendo un tercer polinomio.

$$C(q) = 1 + c_1 q^{-1} + c_2 q^{-2} + \dots + c_{n_c} q^{-n_c}$$

Las ecuaciones quedan:

$$y(t) = (B(q)/A(q)) + (C(q)/A(q)) e(t) \quad (5)$$

Y el vector de parámetros.

$$\theta = [ a1 \ a2 \ a3 \ \dots \ ana \ b1 \ b2 \ b3 \ \dots \ bnb \ c1 \ c2 \ \dots \ cnc ]$$

Debido al nuevo término de media en movimiento  $C(q)e(t)$ , el modelo será llamado *ARMAX* [6].

**C. Estructura *Output Error*.**

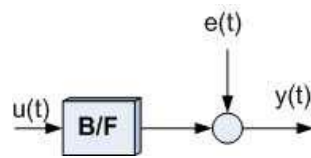
Las de modelado anteriores tienen al polinomio  $A(q)$  como factor común en sus denominadores. Desde un punto de vista físico sería más natural parametrizar estas transferencias en forma independiente. Supongamos entonces por un momento que la relación entre la entrada y una salida no perturbada  $w(t)$  puede ser representada como una ecuación en diferencias lineal, y que la perturbación consiste en ruido blanco.

$$W(t) + f_1 W(t - 1) + \dots + f_{n_f} W(t - n_f) = b_1 u(t - 1) + \dots + b_{n_b} u(t - n_b)$$

$$y(t) = W(t) + e(t) \tag{6}$$

$$F(q) = 1 + f_1 q^{-1} + f_2 q^{-2} + \dots + f_{n_f} q^{-n_f}$$

$$y(t) = (B(q) / F(q)) \cdot u(t) + e(t) \tag{7}$$



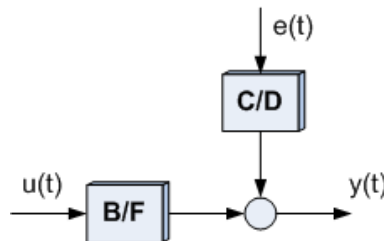
**Figura 3.** Estructura *Output Error*.

**D. Estructura *Box-Jenkins*.**

Este modelo fue presentado por Box y Jenkis (Box and Jenkins, 1970). El desarrollo del modelo de error de salida es para adicionar al modelo las propiedades del ruido en la salida. Pero si describimos esto mismo en el marco de un modelo *ARMA* llegamos al siguiente resultado:

$$y(t) = (B(q) / F(q)) u(t) + (C(q) / D(q)) e(t) \tag{8}$$

Esta es la parametrización de dimensiones finitas más natural si es que partimos del *ARX*, ya que Las funciones de transferencia  $G$  y  $H$  son parametrizadas en forma independiente como funciones racionales.



**Figura 4.** Estructura *Box-Jenkins*.

### 3. Desarrollo.

#### GUI de Identificación SISO.

El trabajo actual más importante en el campo de la identificación de sistemas es el de Ljung (Ljung, 2005), el cual sirve como teoría de base en la realización de este trabajo en conjunto con el *toolbox* (Ljung, 2015) de identificación de sistema de *MATLAB*. *CaToKi* es una interfaz gráfica la cual permite de manera interactiva encontrar, graficar y validar modelos matemáticos de procesos reales. Los modelos obtenidos por *CaToKi* son *SISOs* y lineales en forma de función de transferencia. La ventana principal se aprecia en la figura 5.

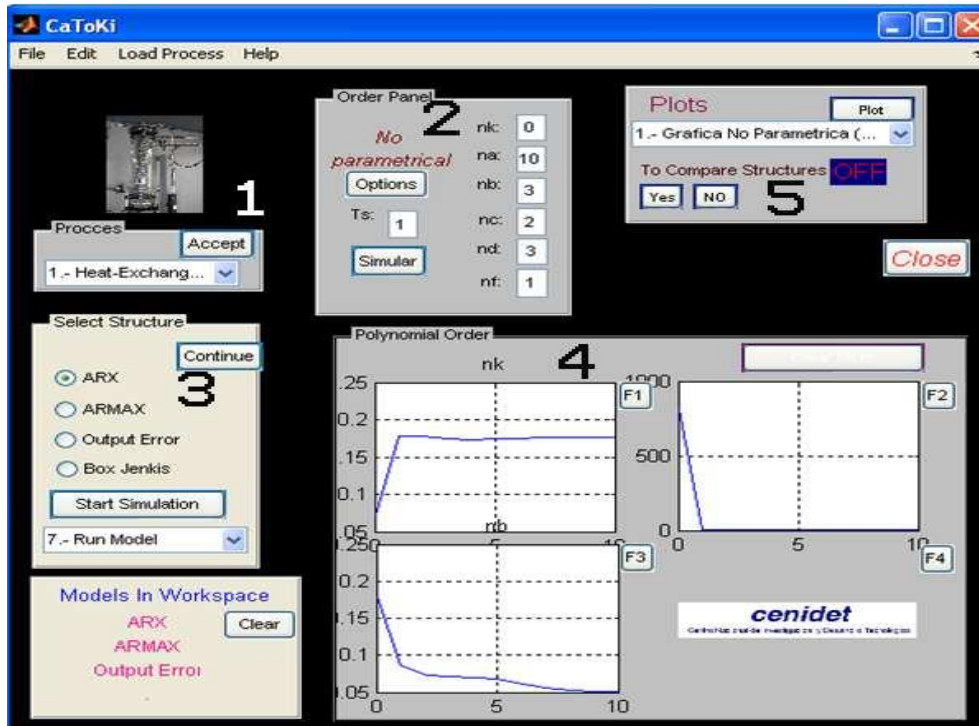


Figura 5. Panel principal.

El GUI está dividido en paneles. Estos paneles están ordenados de acuerdo a los pasos que se deben seguir para el proceso de identificación. De acuerdo a la figura 5 estos pasos son:

- 1.- Selecciona uno de los ejemplos precargados (datos tomados de Daisy data (Daisy, 2015)) o introduce los datos entrada-salida tomados del proceso por el usuario.
- 2.- El panel Orden tiene dos objetivos: a) Encontrar el modelo no paramétrico del sistema introduciendo el tiempo de muestreo con el cual se tomaron los datos. b) Introducir los órdenes de los polinomios “ $na$ ,  $nb$ ,  $nc$ ,  $nd$ ,  $nk$ ” de acuerdo a la estructura seleccionada. Éstos se calculan mediante la *Loss Function* programada por Ljung.
- 3.- Selecciona la estructura a utilizar, la cual puede ser: *ARX*, *ARMAX*, *Output Error* o *Box Jenkins*.
- 4.- Mediante la *Loss Function* se determinan los órdenes correctos de los polinomios de cualquiera de las estructuras.

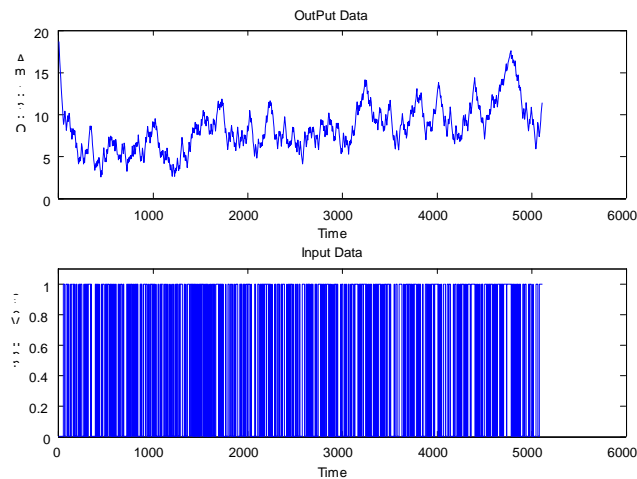
5.-Este panel permite validar el modelo encontrado mediante el uso de gráficas, tanto en respuesta temporal como frecuencial. El o los modelos obtenidos se imprimen en el *workspace* de *MATLAB*.

### A. Identificación Paramétrica

Actualmente no existe ninguna herramienta que pueda seleccionar el grado apropiado de cada uno de los polinomios y es ahí en donde entra en juego el criterio ingenieril. Los conocimientos así como también la experiencia dentro del campo de estudio nos permitirá elegir los órdenes apropiados de los polinomios. Con el *GUI CaToKi* el problema de identificación se reduce a encontrar los órdenes de los polinomios del modelo. Cuenta con una instrucción que permite calcular la función de pérdida del polinomio. En otras palabras, permite seleccionar el orden de polinomio considerando el orden del sistema y la pérdida de información respecto a éste.

### B. Resultados

Para ejemplificar el uso de *CaToKi*, se identifica un proceso real el cual se encuentra en el laboratorio de control automático de procesos del Centro Nacional de Investigación y Desarrollo Tecnológico. El proceso es un módulo de control de nivel (RCN-100). Cuenta con un sensor ultrasónico, el cual arroja datos a la salida de 4-20 mA. El sistema trabaja dentro de una zona lineal de aproximadamente 0-18 cm, y a partir de ahí tiene un comportamiento no lineal, por lo que se trabajó dentro de esa zona. En esa zona, el convertidor de corriente a voltaje nos da 0.6 V para el nivel mínimo y 1.58V para el nivel máximo de 18cm. Los datos de salida son tomados con un tiempo de muestreo de 5 segundos. El conjunto de datos entrada-salida se guardan en un archivo txt (pueden ser también \*.dat, \*.mat), y se cargan a *CaToKi*, los datos de entrada salida se aprecian en la figura 6.



**Figura 6.** Datos de entrada-salida del módulo RCN-100.

Una vez que se tienen los datos de entrada salida, se prosigue a realizar el análisis espectral. El procedimiento a seguir es simple: solo con dar *click* en la opción *BODE Y SPA* del menú graficar, el cual arroja la gráfica de la figura 6. El criterio de evaluación se basa en qué tanto difiere la *SPA* de la *ETFE* (función de transferencia empírica). Un comportamiento semejante de ambas indica que los datos tomados del proceso son buenos para realizar una identificación y se puede proseguir con los pasos siguientes. De no ser semejantes esto se puede deber a diversos factores como son: tiempo de muestreo de los datos, mala planeación en la realización del experimento o integrados o derivadores debidos a la naturaleza del sistema (como lo es un péndulo invertido el cual tiene polos en el origen). Este último factor se analiza en la sección 3-D. La elección de una estructura apropiada (por ejemplo: *ARX*, *ARMAX*, etc.) es un paso crucial en el camino de la identificación de sistemas dinámicos. Dicha elección debe estar basada tanto en el entendimiento del proceso de identificación, como en un acabado conocimiento del sistema a identificar.

Con anterioridad se ha descrito una serie de estructuras típicas para ser usadas en identificación. Lo que se propone ahora es complementar dichas listas con una discusión detallada acerca de cómo arribar a un modelo, guiados tanto por el conocimiento del sistema como por el conjunto de datos recolectados.

### C. Validación.

El modelo estimado puede llegar a ser el mejor disponible; sin embargo, más importante aún es saber a ciencia cierta si dicho modelo es suficientemente bueno para nuestros propósitos. El proceso de evaluación de un modelo para determinar si es el apropiado es conocido como validación del modelo. Para el ejemplo de control de nivel con estructura ARMAX los órdenes del modelo son:  $[na, nb, nc, nk]=[4\ 2\ 3\ 0]$ . El modelo obtenido es:

$$A(q) = 1 - 2.786q^{-1} + 3.128q^{-2} - 1.692q^{-3} + 0.3716q^{-4}$$

$$B(q) = -0.01682 + 0.0374q^{-1}$$

$$C(q) = 1 - 1.849q^{-1} + 1.198q^{-2} - 0.1909q^{-3}$$

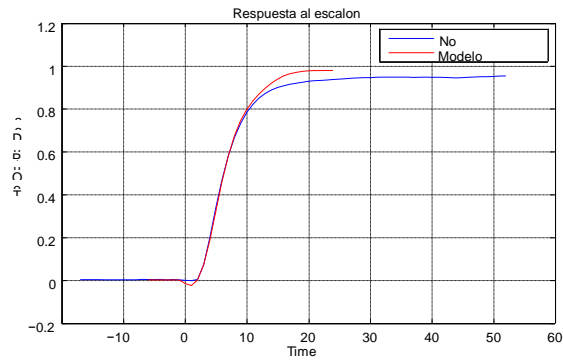
CaToKi realiza la validación del modelo en base a su respuesta temporal y de frecuencias. También se hace el cálculo de residuos del modelo y se comparan dato con dato los datos de salida del modelo aplicando la misma entrada que al proceso real. Las gráficas que se obtienen son las siguientes:

#### Gráficas de Respuesta Temporal.

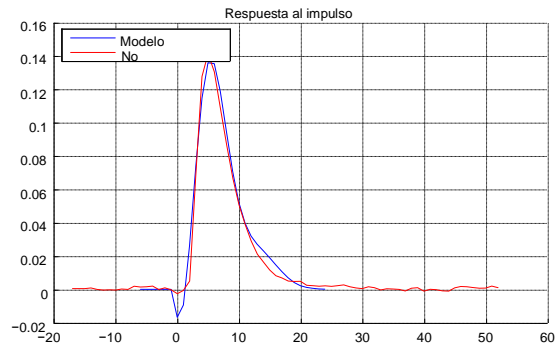
Aplicando a la entrada del modelo una señal de entrada impulso y escalón, tanto al modelo obtenido como al modelo no paramétrico, se grafican ambos y se observan si tienen un comportamiento semejante y, de ser necesario, se calcula el error. Para el ejemplo de control de nivel estas graficas se aprecian en las figura 7a y 7b.

#### Gráficas de Respuesta Frecuencial.

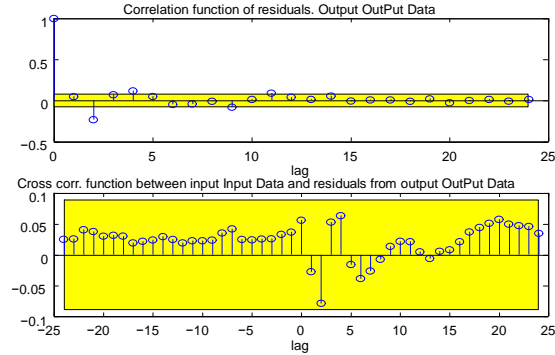
Esta es una de las pruebas principales: graficar la traza de bode del modelo y compararlo con el modelo no paramétrico para observar si tienen un comportamiento similar. Esta gráfica se puede obtener de manera fácil con CaToKi seleccionando en el menú graficar. En la figura 8a se observa que los dos trazos de bode son prácticamente idénticos, por lo que respalda la validez del modelo encontrado para el ejemplo de control de nivel.



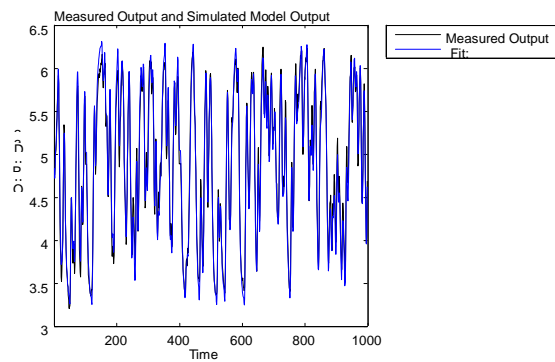
(a)



(b)

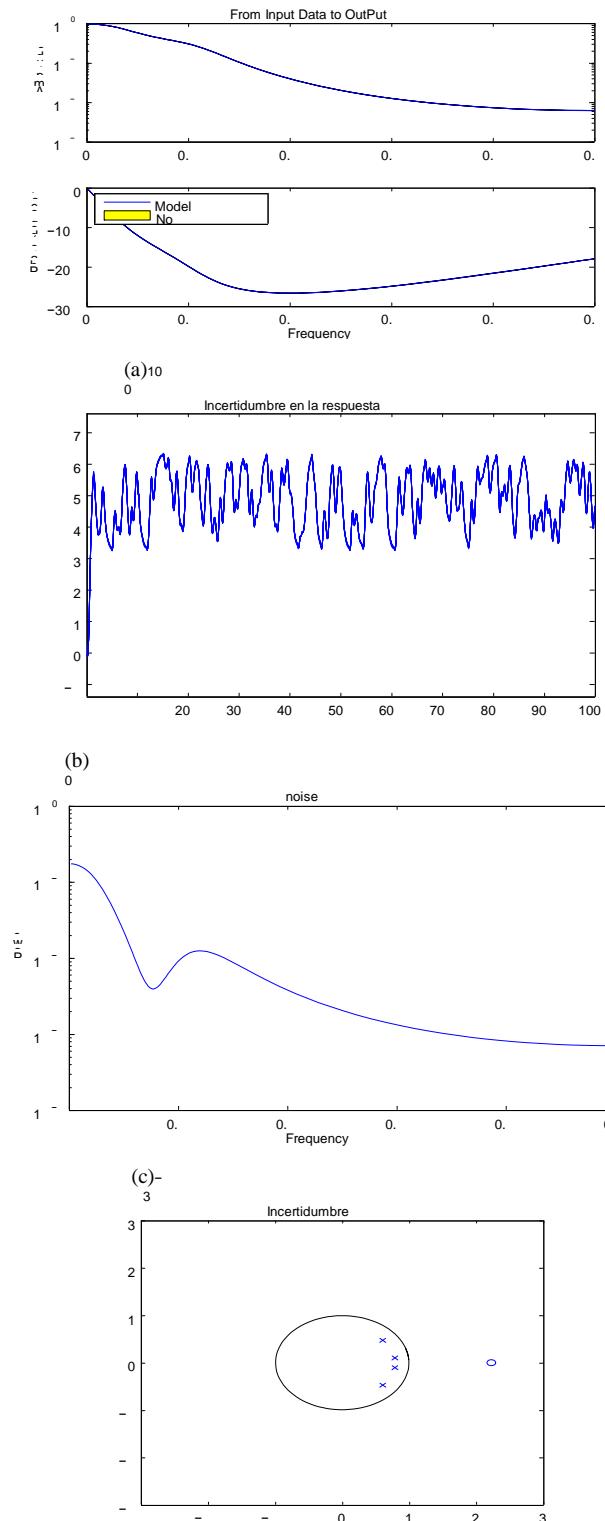


(c)



**Figura 7.** Gráficas de validación del modelo: a) Respuesta Escalón, b) Respuesta Impulso, c) Gráfica de Residuos, d) Gráfica de Comparación.





**Figura 8.** Gráficas de validación del modelo : a) Trazos de Bode, b) Incertidumbre en la Frecuencia, c) Espectro de Ruido, d) Polos y Ceros.

### Gráficas de residuo.

Los residuos asociados con los datos y el modelo dado (diferencia entre la salida real del proceso y la salida estimada con el modelo) son independientes de la entrada y tienen, idealmente, características de ruido blanco. *CaToKi* determina el error del modelo encontrado. La gráfica de residuo para el ejemplo de control de nivel se aprecia en la figura 7c.

Mediante el uso del *toolbox* de identificación de sistemas, *CaToKi* calcula los residuos y lleva a cabo los análisis de independencia y blancura. Se presenta entonces en pantalla la función de autocorrelación de  $e$  y la función de intercorrelación entre  $u$  y  $e$  hasta el retardo número 25. También son exhibidos los intervalos de confianza de estas funciones, asumiendo que  $e$  es blanco e independiente de  $u$ .

El criterio de análisis de esta gráfica es simple: la banda amarilla indica el error permitido para el modelo, por lo que si todos los residuos se encuentran dentro de esta banda es indicativo de que el error está dentro del permitido. Para este caso es de 0.2 y por ende la estimación de los parámetros del modelo es correcta y válida.

### Gráficas de Compare.

Permite saber si el modelo escogido es capaz de reproducir los datos que se observan, a partir de excitar al modelo con una cierta entrada  $u(t)$ . El *GUI* corre una simulación excitando el modelo a la entrada con los datos originales. La gráfica resultante se aprecia en la figura 7d. La gráfica en sí puede resultar un poco confusa, pero lo más importante de ésta es que nos arroja un dato numérico, el cual nos indica el ajuste entre el modelo obtenido y la función de transferencia no paramétrica.

### Otros Gráficos.

Otros gráficos que se pueden determinar con *CaToKi* son:

- Incertidumbre en la respuesta frecuencial figura 8b.
- Espectro de ruido figura 8c.
- Polos y ceros figura 8d, útil si se desea diseñar un controlador.

### D. Menú de Análisis Diferencial-Integral.

Existen ciertos sistemas que tienen un integrador o derivador a la entrada y/o salida. En otras palabras, existen sistemas que son de naturaleza inestable y pueden tener polos o ceros en el origen. A este tipo de sistemas se les tiene que hacer un tratamiento a los datos antes de continuar con el proceso de identificación. Este tratamiento se puede hacer fácilmente con el *GUI CaToKi*. Seleccionando en el botón opciones, aparece un submenú (ver figura 9), el cual permite darle el tratamiento adecuado a todo el conjunto de datos (integración o derivación del conjunto de datos).

$$d^n ( \text{datos} ) / d t^n \quad (8)$$

$$\int (\text{datos}) dt \quad (9)$$

Una vez que se le ha dado tratamiento a los datos, se realiza la estimación de los parámetros del sistema. Para el control de nivel no es necesario utilizar integradores, pero supongamos el siguiente ejemplo:



**Figura 9.** Menú de opciones.

Si se tiene un sistema con un doble integrador a la salida (por ejemplo un péndulo invertido), quiere decir que tiene dos polos en el origen. Esto indica que es marginalmente estable, por lo que para poder realizar una buena identificación antes de proceder, habría que derivar dos veces los datos de salida para poder realizar la identificación y de esta manera eliminar el efecto del doble integrador. Entonces, se puede calcular el modelo matemático. Una vez encontrado el modelo, antes de poder realizar las gráficas de validación, habría que integrar dos veces el modelo para que éste recupere el efecto del doble integrador y de esta manera el proceso de identificación sea válido.

El proceso es complicado ya que resulta más difícil integrar un conjunto de datos y comparar a estos con el modelo encontrado. Es aquí donde *CaToKi* demuestra nuevamente su utilidad, ya que para realizar este procedimiento basta con seleccionar a qué parte del conjunto de datos se desea afectar entrada, salida o el modelo. *CaToKi* realiza el procedimiento cuantas veces sea necesario de una manera sencilla, eficaz y confiable.

### **E. Análisis de Resultados.**

Este experimento se basa en una estructura *ARMAX* por lo cual deben de estimarse los parámetros de tres polinomios más un retardo. Con *CaToKi*, la parte más importante del proceso es determinar los órdenes de cada uno de estos polinomios. Para el control de nivel, el resultado es un modelo de orden 4. Una vez obtenido el modelo matemático, se le aplicaron diferentes análisis. Estos análisis permiten validar el modelo calculando el error y la estabilidad. El error permite conocer (gráficas de residuos y de *compare*) qué tanto se puede reproducir la salida real con el modelo calculado aplicando la misma entrada. La gráfica de polos y ceros determina la estabilidad útil para el diseño de controladores en línea o fuera de línea. El ejemplo de control de nivel pasa cada una de las pruebas de validación, por lo cual se concluye que la identificación fue exitosa.

### **Comparación con el Ident de Matlab.**

*MATLAB* cuenta con un *GUI* de identificación llamado *IDENT*, el cual nos permite al igual que *CaToKi* calcular los parámetros de un modelo *ARX*, *ARMAX*, *Output Error*, *Box Jenkins*. Por eso se enumeran a continuación las ventajas de utilizar el *GUI CaToKi* en comparación con el *IDENT*:

- Presenta una interfaz más amigable y está diseñado específicamente para identificación en lazo abierto de sistemas *SISOs*.
- Permite el tratamiento de los datos derivando o integrándolos según sea el caso, para aquellos que presenten polos o ceros en el origen.

- El código fuente está a disposición de la comunidad de investigadores y estudiantes (disponible en archivos compartidos en la página de MATLAB (López-Estrada, 2015)), cero costos.
- El diseño del *GUI* está basado en los primeros 14 capítulos del libro en (Ljung, 2005), por lo que es ideal para profesores y estudiantes de identificación de sistemas.

### Conclusiones.

A esta altura resulta obvio que la limitación en el uso de las técnicas de identificación está estrechamente relacionada con la disponibilidad de un buen conjunto de datos y de eficientes estructuras de modelado. Ya que, sin un razonable registro de datos no se puede hacer mucho, una razón es que la entrada no debe estar abierta para manipulaciones, las manipulaciones en la entrada podrían empeorar la relación señal a ruido y no se podría garantizar la total identificabilidad del sistema. Un ejemplo cercano es el módulo de control de nivel el cual se identificó satisfactoriamente mediante el *GUI CaToKi*. Como se demostró *CaToKi* permite identificar sistemas *SISO*, de una manera más interactiva, y creativa. De igual manera, una vez que hemos obtenido el modelo del sistema, podemos jugar con él, cambiando los órdenes en las entradas sin la necesidad de cambiar el programa ó modificar los datos. La validación se realiza mediante gráficas de respuesta temporal y en la frecuencia para cualquiera de las estructuras que se seleccione, por lo que los modelos obtenidos son confiables y aplicables en el diseño de controladores.

### Referencias Bibliográficas.

Åström, K. J., & Eykhoff, P. (1971). System identification—a survey. *Automatica*, 7(2), 123-162.

Base de datos de Daisy: [www.esat.kuleuven.ac.be/tokka/daisydata.html](http://www.esat.kuleuven.ac.be/tokka/daisydata.html).

Box, G.E.P and G.M. Jenkins (1970). *Time Series Analysis, forecasting and control*. Holden-Day. Oakland, California.

Ho, B.L. y R.E. Kalman (1965). *Effective construction of linear state-variable models from input-output functions*. *Regelungstechnik* 12, 545-548.

Kunusch C. (2003), *Identificación de sistemas dinámicos*, Universidad Nacional de la Plata, Facultad de Ingeniería Electrónica.

Ljung L. (1999), (2005), *System Identification, Theory for the User*. Cap. 1-14. Edit, Springer.

Ljung L, (2015). *System Identification Toolbox (user guide)*. The Math Works. [www.mathworks.com](http://www.mathworks.com)

López-Estrada, F. R. (2015), *System Identification Toolkit, CaToKi 1.0 y Catoki 2.1*, Matlab File-Exchange: [www.mathworks.com](http://www.mathworks.com)

### Información de los autores.



**Francisco Ronay López Estrada**, recibió su Doctorado en Ciencias en Control Automático por la universidad de Lorraine, France, en 2014, y su Doctorado en Ciencias en Ingeniería Electrónica por el Centro Nacional de Investigación (CENIDET) en 2015. Recibió el grado de maestro en Ciencias en Ingeniería Electrónica por el CENIDET en 2008. Ha sido profesor del Instituto Tecnológico de Tuxtla Gutiérrez desde 2008 adjunto al departamento de Ingeniería Electrónica. Sus intereses de investigación son: los sistemas lineales de parámetros variantes, sistemas de detección de fallas, sistemas descriptores, control de procesos y control de vehículos aéreos no tripulados.



**Álvaro Hernández Sol**, es Ingeniero en Electrónica, egresado del Instituto Tecnológico de Tuxtla Gutiérrez ITTG, en Tuxtla Gutiérrez, Chiapas. Es profesor de tiempo completo en el área de ingeniería electrónica del Instituto Tecnológico de Tuxtla Gutiérrez y es investigador desde 1997. Siendo jefe de proyectos de investigación de ingeniería electrónica desde el 2001. Fundó y asesora el “club de robótica del ITTG”. Colabora en la línea de investigación de “robótica” de Ingeniería electrónica y dirige el área de trabajo en “sistemas aéreos autónomos”.



**Joaquín Eduardo Domínguez Zenteno**, es Maestro en Ciencias en Ingeniería Electrónica egresado del Instituto Tecnológico de Toluca, Ingeniero en Electrónica con especialidad en Instrumentación y Control por el Instituto Tecnológico de Nuevo León, en Cd. De Guadalupe, Nuevo León. Es profesor de tiempo completo en el área de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez y Jefe de Proyectos de Vinculación de Ingeniería Electrónica. Es investigador desde 2009 y colabora en la línea de investigación de “Instrumentación” de Ingeniería electrónica.



**José Ángel Zepeda Hernández**, es Ingeniero Industrial en Eléctrica y Maestro en Ciencias en Ingeniería Mecatrónica, egresado del Instituto Tecnológico de Tuxtla Gutiérrez. Es profesor de tiempo completo en el área de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez e investigador desde 1999, Imparte cátedra en el área de Ingeniería Electrónica y la Maestría en Ciencias en Ingeniería Mecatrónica en el ITTG, donde imparte las asignaturas de diseño con transistores y electrónica básica respectivamente. Líder de la línea de investigación y generación del conocimiento “Instrumentación”.



**Ildberto de los Santos Ruiz**, es Ingeniero en Electrónica y Maestro en Ciencias en Ingeniería Mecatrónica por el Instituto Tecnológico de Tuxtla Gutiérrez. Es profesor a tiempo completo en el Departamento de Ciencias Básicas del ITTG desde 1995, donde investiga sobre modelado, simulación y control de sistemas dinámicos.



**Raúl Moreno Rincón**, es Maestro en Ciencias en Ingeniería Electrónica egresado del Instituto Tecnológico de Toluca, Maestro en Educación Superior por la Universidad Autónoma de Chiapas e Ingeniero en Comunicaciones y Electrónica por la ESIME-IPN, en la ciudad de México, D.F. Es profesor de tiempo completo en el área de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez y Jefe de Proyectos de Docencia de Ingeniería Electrónica. Es investigador desde 1999 y colabora en la línea de investigación de “Robótica” de Ingeniería electrónica, en donde ha realizado proyectos como: Sistema de alarma para personas con deficiencia auditiva basado en XBEE, Robot de cafetería, araña hexápoda, entre otros.