

Obtención de Modelos de los Procesos Continuos Utilizando el Toolbox de Identificación de Sistemas de Matlab

Obtaining Continuous Process Models Using Matlab Systems Identification Toolbox

Bruno Vargas Tamani

Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima, Perú

Resumen— El presente trabajo explora como el Toolbox de Identificación de Sistemas de Matlab, permite con bastante facilidad obtener un modelo de proceso en prueba para su posterior control, en base a los datos registrados de entrada y salida. Estos datos experimentales adquiridos mediante un osciloscopio o un sistema de adquisición de datos, sirven para obtener el modelo del proceso, pero antes se debe procesar mediante algoritmos de identificación de sistemas. Para la obtención de modelos continuos se ha explorado la opción: Process Models y Transfer Functions Models del Toolbox de Identificación de Sistemas de Matlab. Se ha comprobado para nuestros datos de prueba que la mejor opción resultó ser el Process Models. Adicionalmente, se obtuvo otras opciones de este Toolbox que nos ha permitido en forma segura la obtención del modelo que mejor se aproxima a la planta de prueba, siendo la validación con datos diferentes a la identificación, gráficos de entrada y salida, gráficos de entrada y salida de validación, y otros.

Abstract— The present work explore with Matlab System Identification Toolbox, it allows quite easily to obtain a model of the process to be tested for its subsequent control, based on the registered data of input and output. These experimental data acquired by an oscilloscope or a data acquisition system serve to obtain the model of the process, but must first be processed using system identification algorithms. To obtain continuous models, we have explored the option: Process Models and Transfer Functions Models of the Matlab Systems Identification Toolbox. It has been proven for our test data that the best option was Process Models. In addition, other options were obtained from this Toolbox, which has allowed us to securely obtain the model that best approaches the test plant, being the validation with data different from the identification, input and output charts, input and exit graphics validation, and others.

Palabras clave— Identification of systems, transfer function models, temporal response, poles and zeros, delay.

[†]Bruno Vargas Tamani, e-mail: bvargast@unmsm.edu.pe

Recibido: Abril 2016 / Aceptado: Junio 2016

Keys words— differential traction, kinematic model, dynamic model, equilibrium point, orientation.

I. INTRODUCCIÓN

El modelo de la planta con fines de control, es una tarea necesaria para el diseño del controlador. Existen varias formas de modelar, desde plantear ecuaciones, realizar diagramas de bloques, reducir el diagrama de bloques, espacio de estado, etc. Sin embargo, en el lugar de trabajo, es difícil utilizar estos métodos (ellos son un referente, por ejemplo, para conocer a priori el orden del sistema), por el desconocimiento de sus parámetros o porque se requiere actualizar esa información, así como, por el poco tiempo que se tiene disponible la planta en la industria. Lo más práctico en los sistemas reales de control automático, es realizar la tarea de obtener un modelo continuo del proceso, adquiriendo datos de la entrada del proceso y de su salida. Estos datos registrados se utilizan mediante una de las técnicas llamadas de identificación de sistemas, que normalmente se programa en computadora. El producto de software Matlab, permite en base a su Toolbox (existen muchos de ellos: procesamiento de señales, control, redes neuronales, lógica difusa, telecomunicaciones, control óptimo, etc.), desarrollar aplicaciones diversas. Una opción muy útil, para obtener un modelo que se aproxime a los datos registrados, es utilizar el Toolbox de Identificación de Sistemas de Matlab, el cual tiene diversas funciones para obtener el modelo de la planta, con mucha facilidad.

II. TOOLBOX DE IDENTIFICACIÓN DE SISTEMAS DE MATLAB

Para poder establecer un controlador para un proceso, se requiere conocer las características del

proceso a controlar, esto se manifiesta con un modelo continuo de ese proceso. Uno de los métodos para conocer ese proceso mediante un modelo, es registrar datos de entrada y salida de ese proceso y mediante una técnica llamada de identificación de sistemas, obtener el modelo que mejor se acerquen a los datos registrados.

El Toolbox de Identificación de Sistemas de Matlab (System Identification Tool); es una herramienta de Matlab que permite obtener varios modelos: transfer function, process control, etc. [1]. Dentro de cada uno de ellos, se tienen opciones de seleccionar sus características y realizar pruebas con ellos; por ejemplo de respuesta temporal, respuesta del modelo, etc. El usuario puede elegir diferentes configuraciones de un mismo tipo de modelo y el decide el modelo a utilizar en representación de la planta.

A. Uso del Toolbox de Identificación de Sistemas de Matlab. Acciones Previas

Previamente al uso del Toolbox de Identificación de Sistemas de Matlab, se requiere que los datos de entrada y de salida del sistema a tratar de obtener el modelo que más se asemeja a esos datos medidos, se ubiquen en el workspace de Matlab. El workspace, es la memoria de trabajo de Matlab. Ud. puede acceder a esta área, escribiendo el comando `who` en el prompt de Matlab y verá todo lo registrado en memoria como variables. Cada entrada y salida, deben estar en memoria en un vector columna (de dimensión $n \times 1$, donde n es el número de datos), ambos deben tener la misma dimensión o el número de datos [2].

Previamente, entonces se deben prepara los datos y que se ubiquen en el área de trabajo. Dependiendo del sistema de adquisición de datos, estos se registran en formatos, por ejemplo: excell, texto, etc. Además, en un archivo de esos datos se puede registrar información adicional, también es normal que se registren en dos columnas, una es el vector de tiempos y la otra es el de los datos propiamente dicho. Como se verá, se debe realizar la tarea previa de organizar los datos originales al formato que requiera el Toolbox de Identificación de Sistemas de Matlab.

Normalmente, se graba cada archivo de entrada y de salida, en n archivos que contiene una columna (utilizar el comando: `save nombre_archivo`). Así, se crea el archivo con `nombre_archivo.mat`; los archivos con extensión `mat` son propios de Matlab. Luego en cada sesión de Matlab estarán disponibles esos archivos, que solo basta cargarlos en la memoria de trabajo (workspace). Para ello, bastará al usuario con sólo invocar al comando: `load nombre_archivo`. Así, se cargará la variable en `nombre_variable_a_guardar`, en la memoria de trabajo, estando disponible para ser utilizada por el Toolbox de Identificación de Sistemas de Matlab [3].

B. Graficar datos para identificar y validar

Se han obtenido los datos de un proceso simulado, con un período de muestreo de $T = 0.1\text{msg}$. Se cargan esos datos tanto de entrada y de salida para identificarlos, así como los datos tanto de entrada y de salida para validarlos, y se han grabado en el área de trabajo, en variables de una columna llamadas: entrada, salida, entrada validar y salida validar.

Las primeras señales se utilizarán para obtener el modelo de la planta simulada y las últimas se usarán para validar el modelo obtenido. La validación, se realiza con datos diferentes a los utilizados para identificar; de tal manera que se pruebe el modelo obtenido con otros datos y se vea si ese modelo resulta en una generalización para otras señales y se pueda predecir su respuesta.

La Fig. 1, muestra la pantalla inicial de Matlab. El sector superior derecho muestra las variables que se ubican en el área de trabajo. Allí, se observan las variables entrada y salida. Para confirmar, si se trata de señales para trabajar la identificación, vamos a graficarlas. Hacer doble click en la variable entrada, se aprecia el vector correspondiente, luego seleccionar la señal y en la barra de tareas seleccionar plot mostrándose luego el gráfico de esa entrada que se usará para identificar. Ver Fig. 2.

También seleccionar desde el sector superior derecho, la variable salida. Repetir lo anterior, para ver el gráfico de la variable de salida que se usará para identificar. Ver Fig. 3.

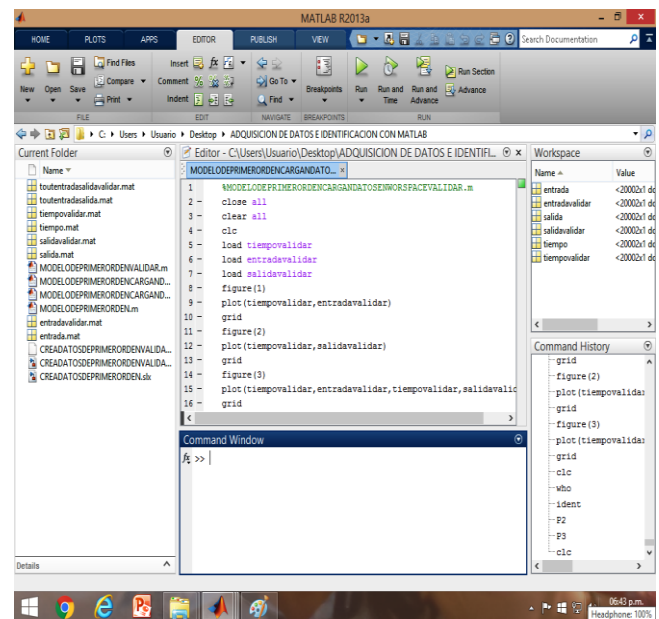


Fig. 1. Pantalla inicial de Matlab.

Realizar la misma secuencia, con las señales `entradasvalidar` y `salidasvalidar`. Se observarán los gráficos de las señales a utilizar para la validación; ver Fig. 4 y 5.

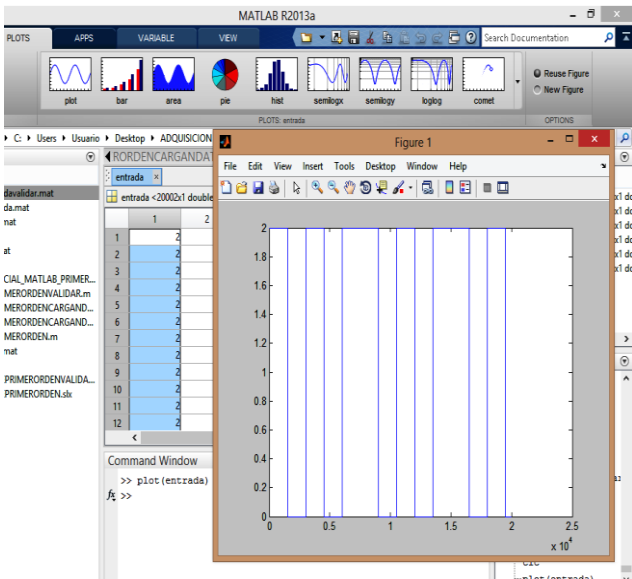


Fig. 2. Gráfico de los datos de entrada para identificar.

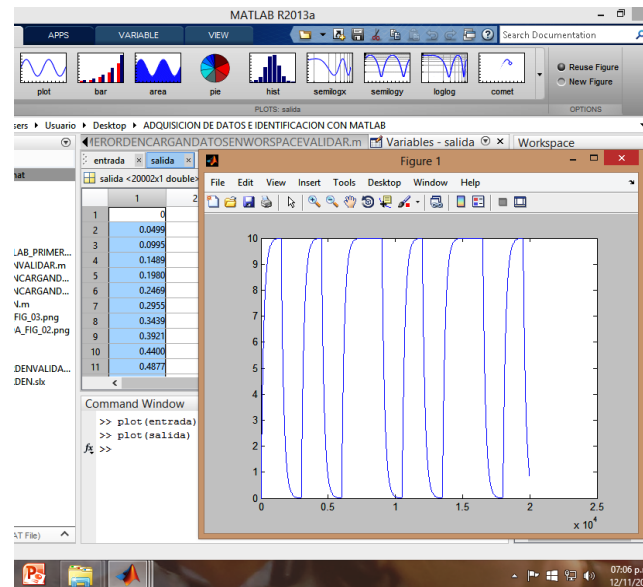


Fig. 3. Gráfico de los datos de salida para identificar.

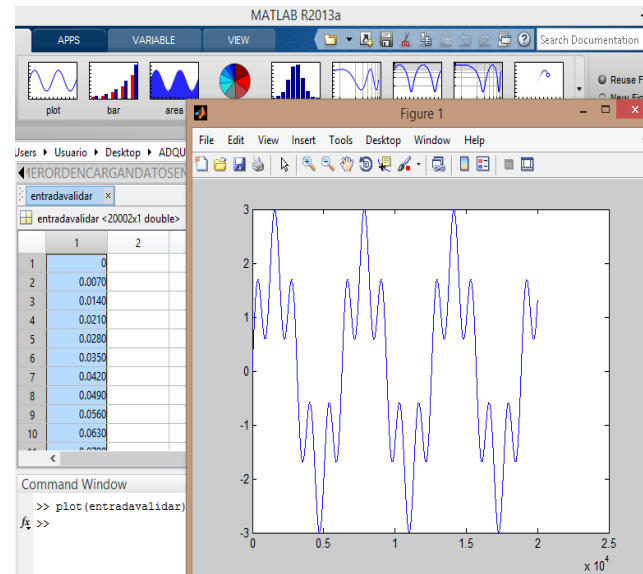


Fig. 4. Gráfico de los datos de entrada para validar.

C. Arrancar el Toolbox de Identificación de Sistemas de Matlab e Inicio de Variables

Desde el prompt de Matlab, escribir `ident` o desde la barra de tareas seleccionar APPS y dentro de ella seleccionar System Identification. Se presenta la pantalla del Toolbox de Identificación de Sistemas de Matlab. La Fig. 6, muestra esa pantalla.

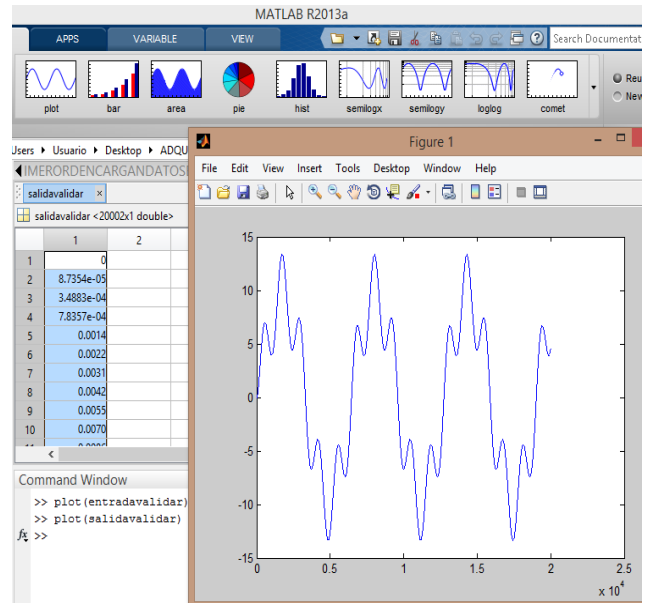


Fig. 5. Gráfico de los datos de salida para validar.

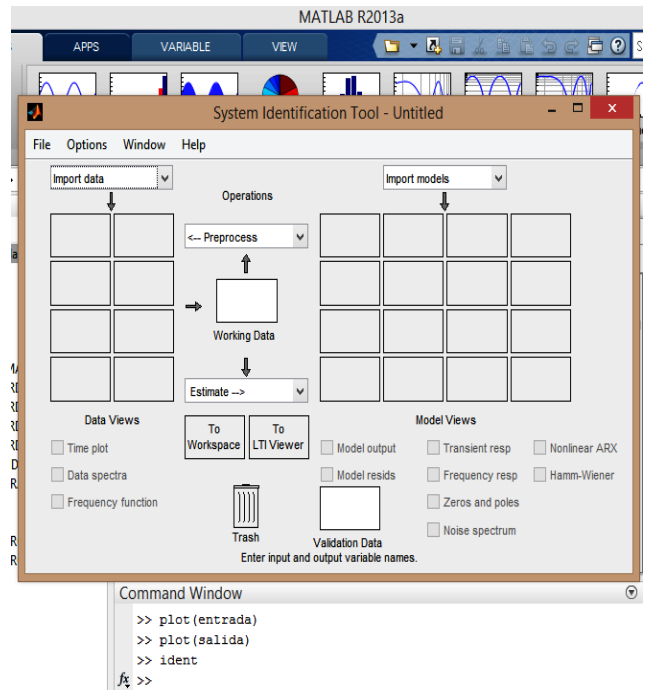


Fig. 6. Pantalla de arranque del Toolbox de Identificación de Sistemas de Matlab.

En la parte superior izquierda, seleccionar Import data, luego seleccionar Time domain data. Se abre la pantalla para seleccionar las variables de entrada y de salida para identificar. También se debe

escribir en la opción Data name: datosidentificar (que agrupa las variables de entrada y de salida), además del inicio del vector de tiempos y el período de muestreo utilizado. La Fig. 7, muestra esa pantalla de configuración de señales para identificar. La Fig. 8, muestra esa pantalla de configuración de señales para identificar, luego de seleccionar import.

Observe que, en la zona de estimación, se cargan los datos agrupados en datosidentificar, también en la zona de validación se cargan ese mismo grupo de datos.

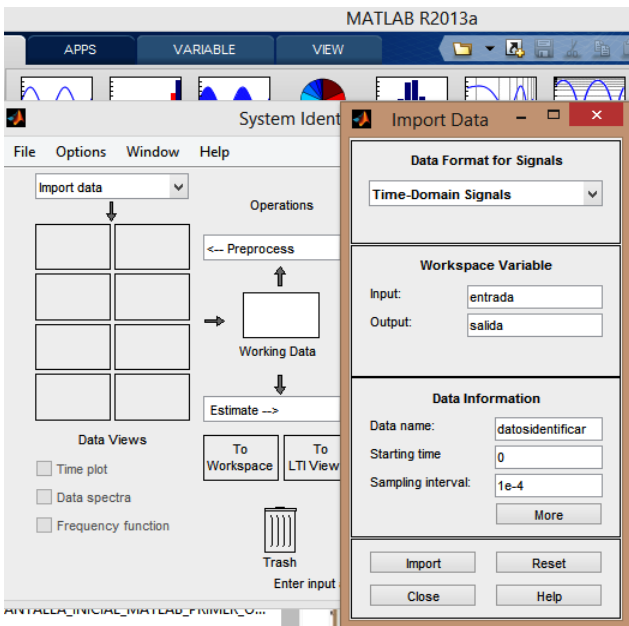


Fig. 7. Pantalla de configuración de señales para identificar.

En la pantalla del ident, se observa que en la parte izquierda abajo hay una opción: Time Plot. Seleccionar esa opción y se verán los datos de entrada y salida para identificar, en un gráfico. Ver Fig. 9.

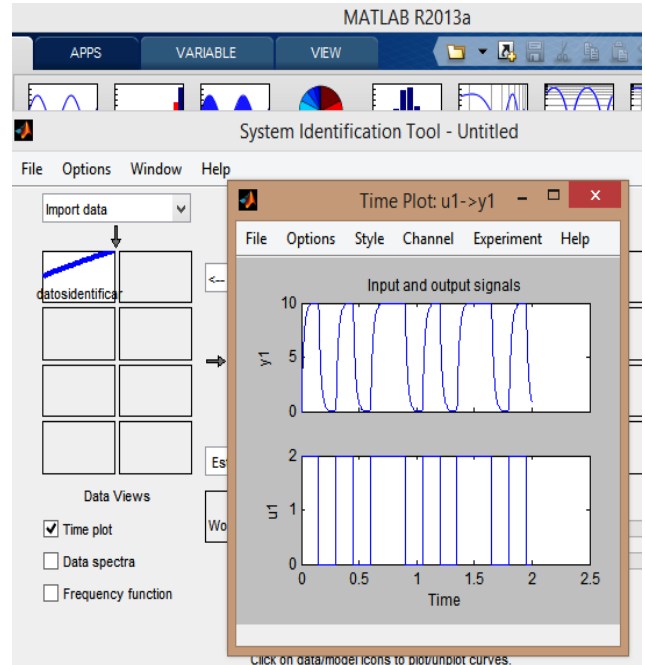


Fig. 9. Opción: Time Plot. Gráfica de los datos de entrada y salida para identificar.

D. Seleccionar el Modelo a Identificar

En la zona de estimación (parte central), seleccionar Process Models, aparece la pantalla mostrada en la Fig. 10.

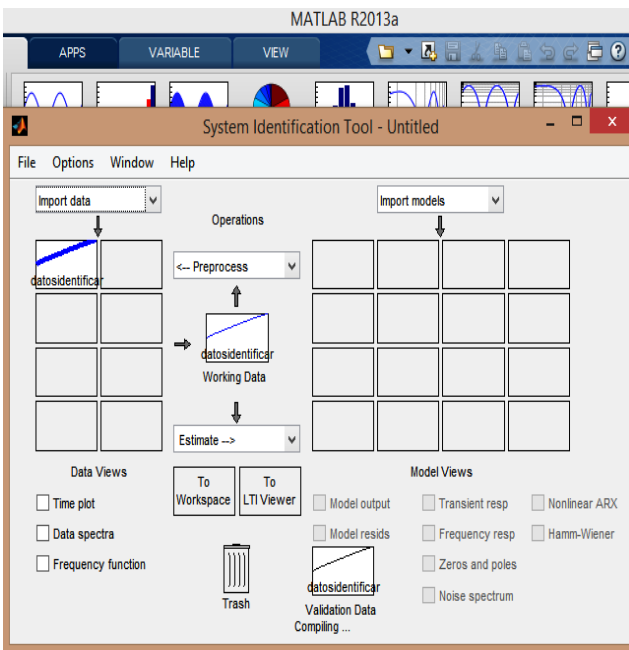


Fig. 8. Pantalla de configuración de señales para identificar, luego de seleccionar import.

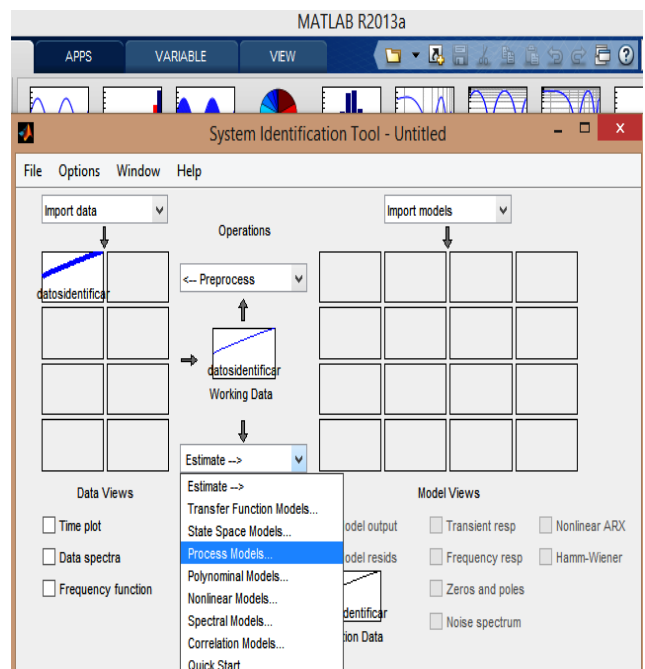


Fig. 10. Pantalla para seleccionar Process Models.

Configurar en esa pantalla, para que el modelo a identificar sea de primer orden sin retardo y sin un cero. Pulsar Estimate, en la última pantalla. La Fig. 11, muestra el resultado de esa identificación. Observe, que la pantalla advierte que el modelo será llamado P1.

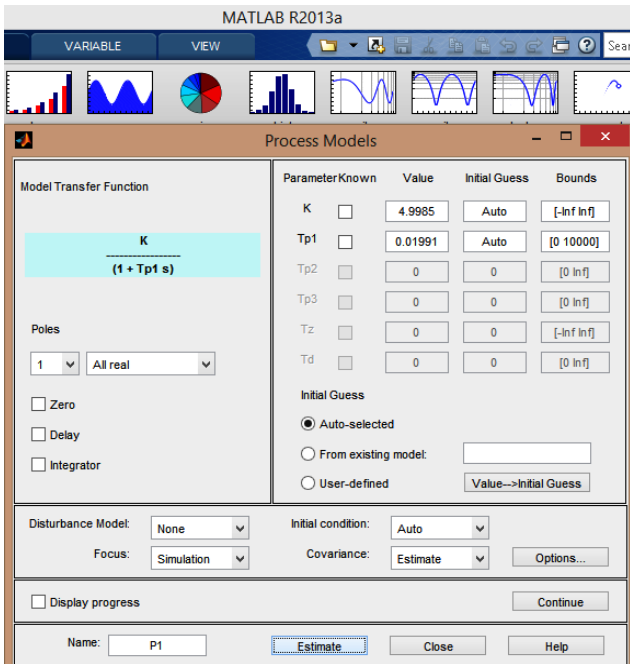


Fig. 11. Configuración de Process Models (modelo para modelo de primer orden sin retardo y sin ceros) y pantalla luego de pulsar Estimate.

Luego de estimar el modelo, en la pantalla se crea a la parte superior derecha la variable P1. Ahora seleccione en la nueva pantalla el modelo P1, pruebe el resultado de esa identificación seleccionando la opción Model output. Ver la Fig.12.

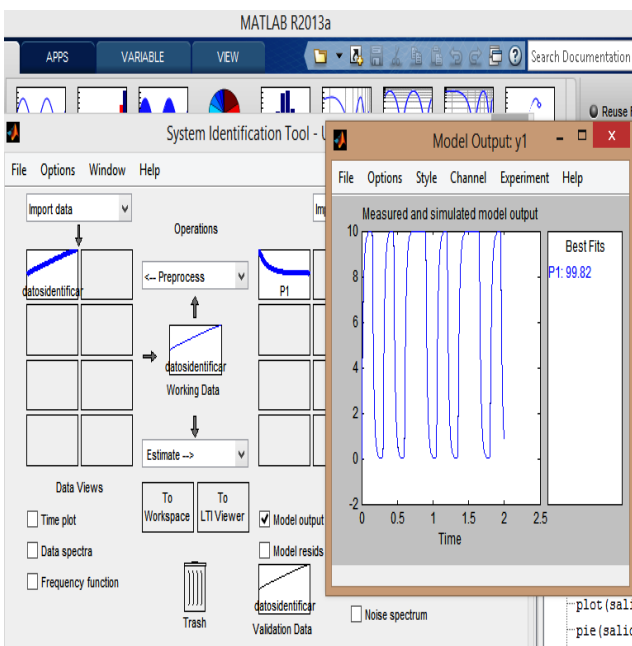


Fig. 12. Pulsar P1 y opción Model Output, para probar el modelo identificado.

Arrastre la variable P1, a la zona del workspace, para ver el modelo identificado resultado de la identificación, con la opción Process Models. Ver Fig. 13.

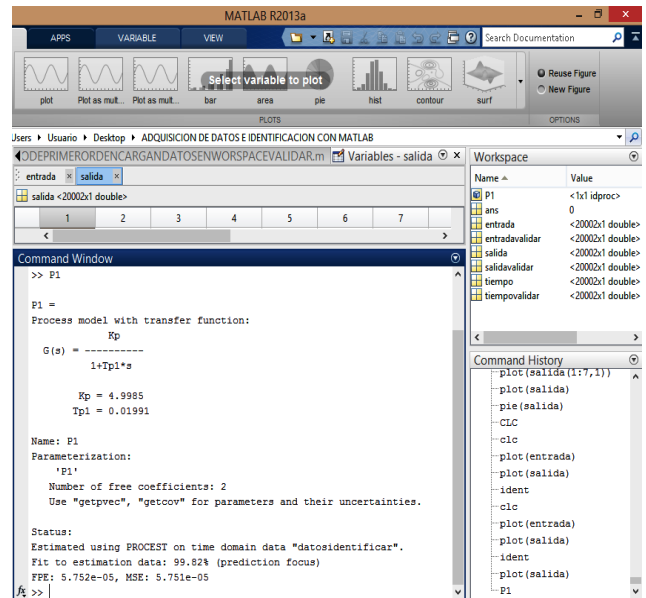


Fig. 13. Resultado de la identificación, con la opción Process Models.

E. Configuración de otros en opción Modelos con Process Models

Repetir lo anterior configurando la opción Process Models, para dos polos, tres polos y tres polos con un cero. Se crean las variables P2, P3 y P3Z, respectivamente. En la Fig. 14, se observa la prueba del modelo P2 con Models output. La Fig. 15, muestra el modelo de P2, la Fig. 16 muestra el modelo de P3 y la Fig. 17 el modelo de P3Z.

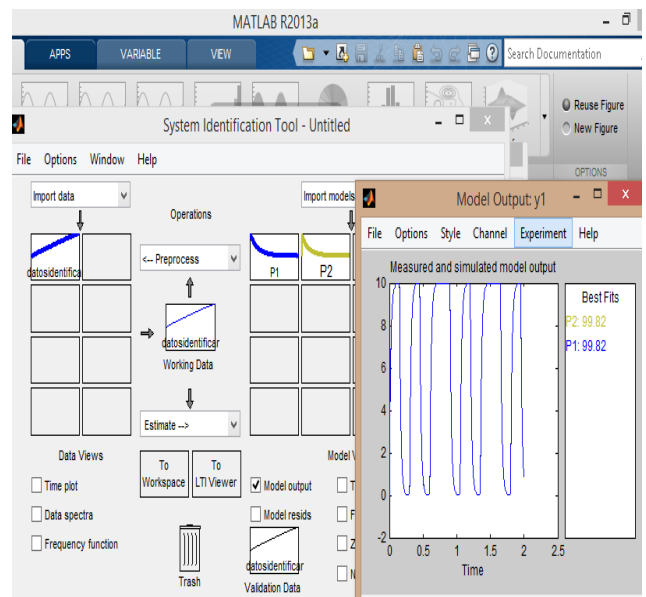


Fig. 14. Prueba de P2 con Model Output.

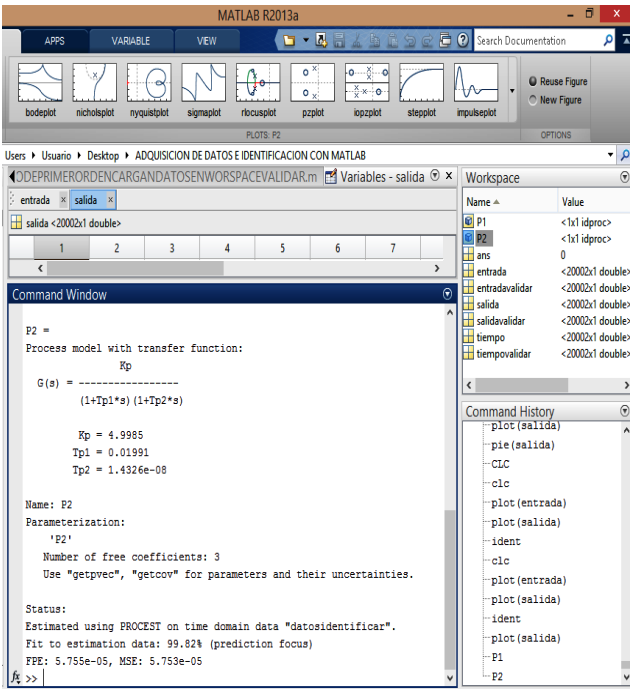


Fig. 15. Resultado del modelo obtenido P2.

F. Iniciar variables para validar

Se cargan las variables llamadas `entradavalidar` y `salidavalidar`, en el área de trabajo. En la parte superior izquierda, seleccionar `Import data`, luego seleccionar `Time domain data`. También debe escribirse en la opción `Data name: datosvalidar`, además del inicio del vector de tiempos y el período de muestreo utilizado.

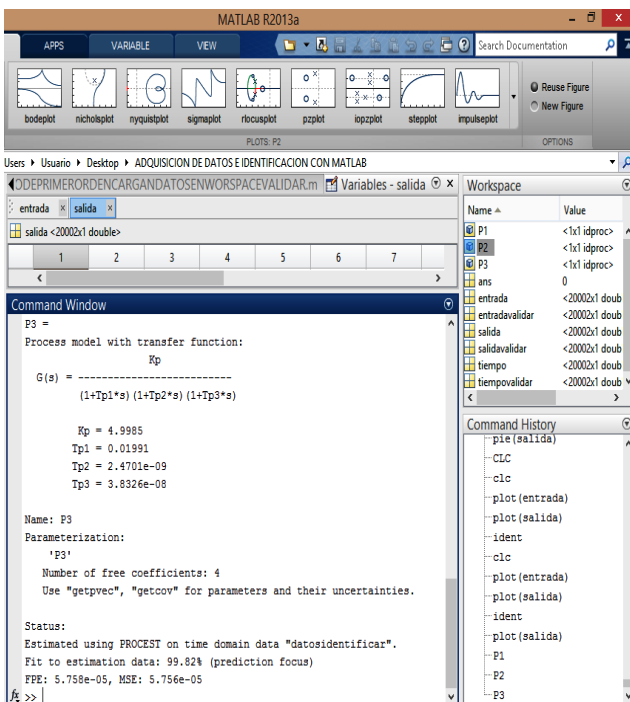


Fig. 16. Resultado del modelo obtenido P3.

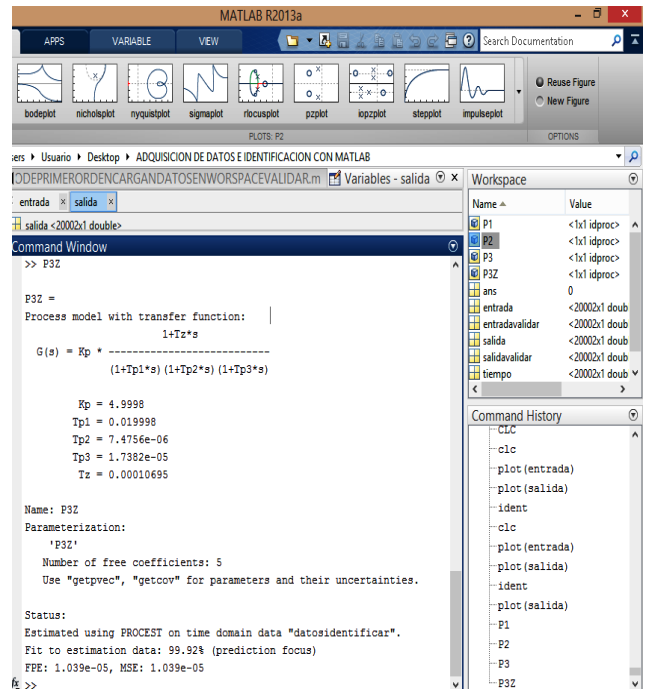


Fig. 17. Resultado del modelo obtenido P3Z.

Arrastre los datos agrupados en `datosvalidar`, a la zona de validación. Pulsar la opción `Model Output`. La Fig 18, muestra el resultado de la validación con los datos diferentes a la identificación.

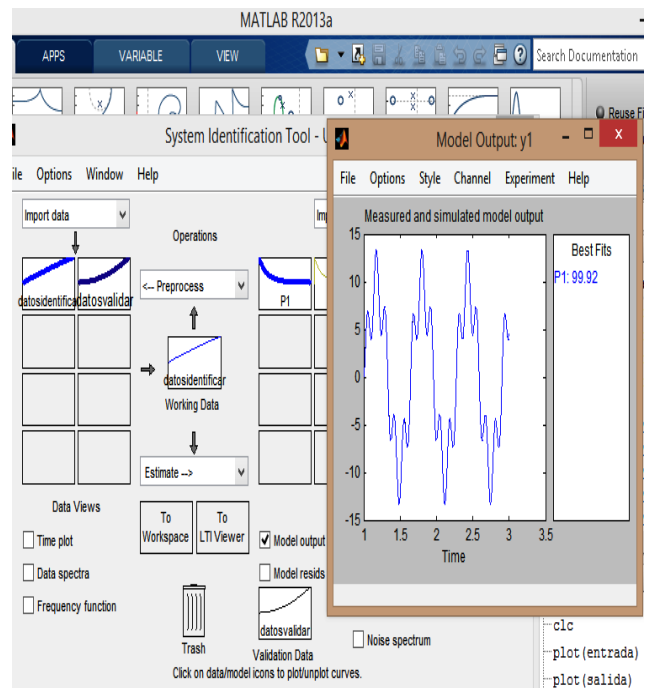


Fig. 18. Resultado de la validación con los datos diferentes a la identificación.

G. Selección de un Modelo diferente a Identificar: Transfer Functions Models.

Para el mismo grupo de datos de entrada: `datosidentificar`; se selecciona ahora otra opción de estimación del modelo, seleccionar en `estimate` la opción: `Transfer Function`

Model. Se abre la pantalla mostrada en la Fig. 19, para configurar el modelo de función de transferencia a estimar; selecciona: Continuous-time, number of poles: 1, number of zeros: 0. Se trata de un modelo de primer orden sin retardo. Observe que en la parte superior, se refiere a que la variable del modelo a crear es llamada: tf1.

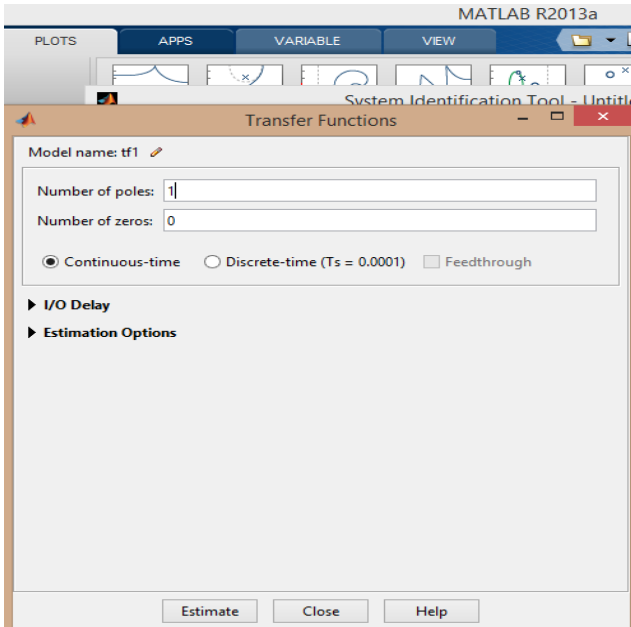


Fig. 19. Opción de Estimate, un modelo de función de transferencia, configuración del modelo de primer orden sin retardo.

Observe como automáticamente en la pantalla principal del ident se activa el resultado de la función de transferencia tf1. Selecciónese y luego pulse la activación de la opción: Model output. Así, observará el resultado de la identificación. Ver Fig. 20.

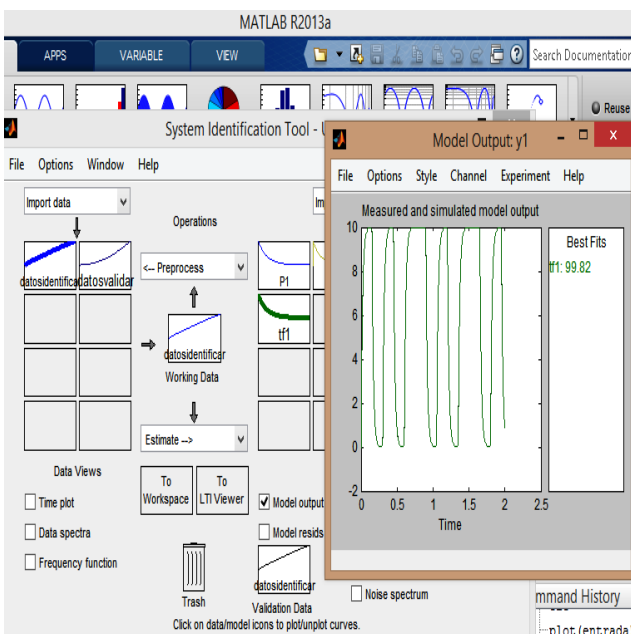


Fig. 20. Función de transferencia tf1. Resultado de la identificación realizada. Opción: Model output.

Ahora arrastre el icono de tf1 al workspace, ahora se crea en memoria esa variable. Escriba en el prompt de Matlab, tf1 y se muestra el modelo de función de transferencia y sus características adicionales. Ver Fig. 21.

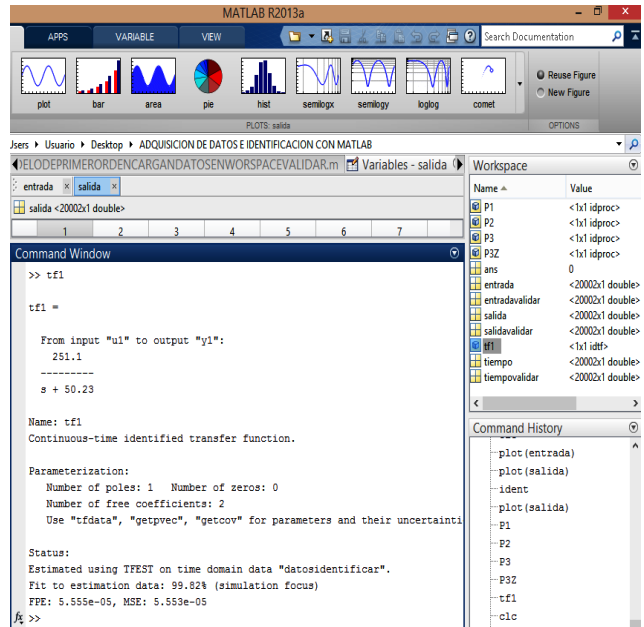


Fig. 21. Resultado del modelo de función de transferencia tf1 y sus características adicionales.

Para la validación se utilizan las mismas señales agrupadas en datosvalidar. No se olvide de arrastrar al área de validación, este grupo de datos. Pulse la opción: Model Output y verá el resultado de la validación, con los datos datosvalidar, diferentes a los de identificación. Ver Fig. 22.

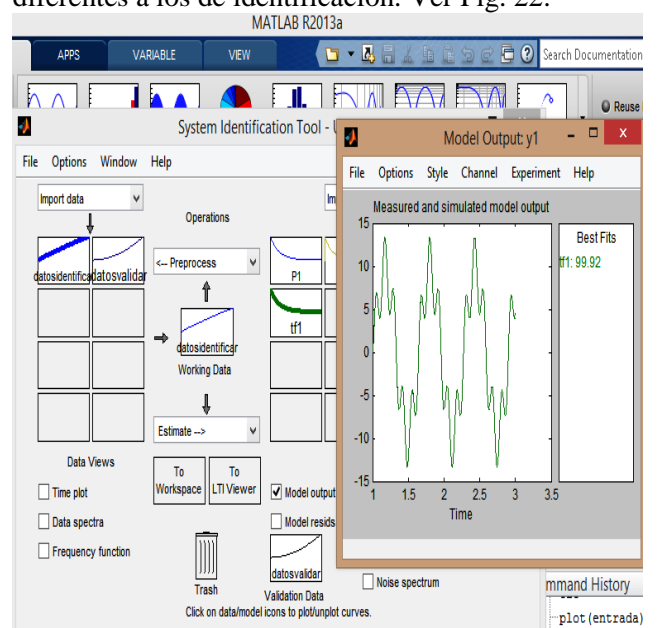


Fig. 22. Resultado de la validación para el modelo de función de transferencia tf1.

Así como hicimos con la opción de Estimate: Process Model, con la opción Transfer Function Model, también podemos probar otros modelos de función de transferencia. Probemos ahora, un modelo de función de transferencia, con dos polos (ningún cero, sin retardo). La pantalla de configuración indica que el modelo se va a guardar en la variable llamada: tf2.

Pulsar estimate, seleccionar tf2 en la pantalla ident, seleccionar la opción: Model output, para probar el modelo de segundo orden y el resultado se muestra en la Fig. 23.

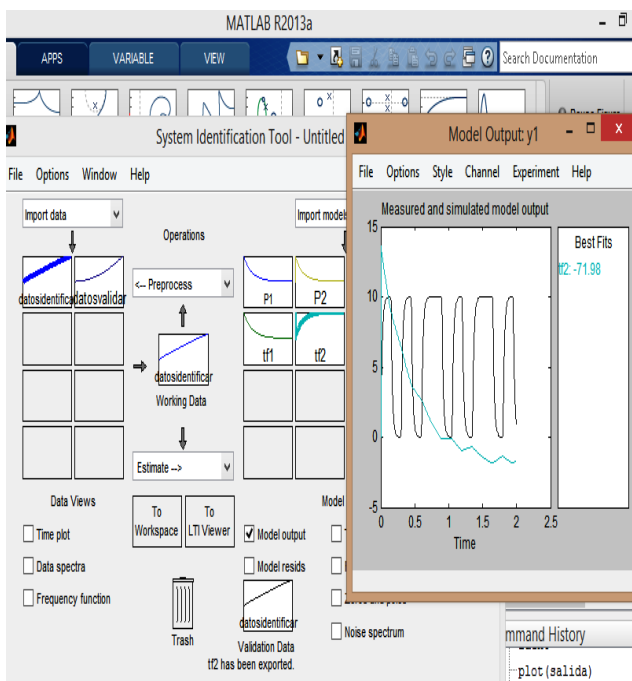


Fig. 23. Resultado de configurar un modelo de función de transferencia de segundo orden.

VI. ANÁLISIS DE RESULTADOS

Como resultados de las pruebas para la obtención del modelo mediante la opción: Process Model, reflejan que el modelo obtenido que mejor se mapean con los datos identificados es el modelo de primer orden sin retardo. Los modelos obtenidos por estos métodos de orden superior: segundo orden sin retardo, tercer orden sin retardo, tercer orden sin retardo con un cero. Crean parámetros adicionales, pero cuyas constantes de tiempo y el cero adicional son muy pequeños comparados con la constante de tiempo dominante del proceso identificado como de primer orden.

La validación del modelo de primer orden con datos diferentes a los de identificación, permite probar el modelo obtenido con señales diferentes a las de

identificación, probándose que generaliza la predicción de respuestas con otras señales.

La validación del modelo de primer orden con datos diferentes a los de identificación, permite probar el modelo obtenido con señales diferentes a las de identificación, probándose que generaliza la predicción de respuestas con otras señales.

La opción la opción: Transfer Function Model, resulta en un modelo identificado de primer orden sin retardo, que es idéntico al modelo obtenido y validado en la opción: Process Model. Sin embargo, cuando en la opción: Transfer Function Model, probamos con un modelo de segundo orden sin retardo, la identificación (así como la validación) no resulta en un adecuado modelo.

VII. CONCLUSIONES

El Toolbox de Identificación de Sistemas del programa Matlab, es una aplicación de ese programa que se puede utilizar con gran facilidad para fines de encontrar un modelo continuo, de función de transferencia de la planta, con fines de diseño de un controlador tanto analógico o digital.

Las opciones adicionales, antes de obtener el modelo identificado: time plot, validación del modelo mediante la opción: Model output (tanto con los datos de identificación, como con los datos de validación), transient resp, etc. Nos permiten asegurar que el proceso de identificación ha sido el adecuado.

REFERENCIAS

- [1] C. Osorio. "Extracción de Modelos Dinámicos Directamente de Datos Experimentales usando Identificación de Sistema". MathWorks. <https://www.mathworks.com/videos/extracting-dynamic-models-from-experimental-data-using-system-identification-spanish-100499.html?requestedDomain=www.mathworks.com>
- [2] O. F. Avilés, P. A. Niño y L. Solaque. "Identificación de parámetros de sistemas dinámico". Ciencia e Ingeniería Neogranadina, 2002, ISSN: 0124-8170. Sistema de Información Científica Redalyc Red de Revistas Científicas de América Latina y el Caribe, España y Portugal. <http://www.redalyc.org/articulo.oa?id=91101204>
- [3] C. Kunusch. "Identificación de Sistemas Dinámicos". Universidad de la Plata. Facultad de Ingeniería. Dpto. de Electrotecnia. Cátedra de Control y Servomecanismos. Año 2003. <http://catedra.ing.unlp.edu.ar/electrotecnia/cys/pdf/identificacion.pdf>