

CAPÍTULO 4

PROGRAMACION DEL PROBLEMA.

4.1 OBJETIVO DEL CAPÍTULO.

En este capítulo pretendemos acercar al lector al conjunto de programas que se han desarrollado en este proyecto para cumplir el objetivo de establecer una conexión, más o menos directa, entre el usuario u operador de la planta solar y la obtención de un modelo de la misma, basado en la factorización LFMD, que sea inteligible para el programa OPTIMAX encargado de realizar el verdadero control sobre la instalación.

4.2 INTRODUCCIÓN AL PROBLEMA.

Antes de la realización de este proyecto, la relación entre el usuario y la planta solar a la hora de realizar ensayos era bastante tediosa y afectada de posibles fallos. En el caso de estar interesados en realizar una descripción en el espacio de estados, manejada por OPTIMAX, de un modelo multivariable de la planta, se hacia necesaria una cierta programación que realizara la conversión desde el espacio continuo de los datos obtenidos en los ensayos hasta el modelo CARIMA manejado por OPTIMAX.

Surge, por tanto, la necesidad de programar una serie de funciones que realicen la factorización LFMD tal y como se describe en este documento y en la reseña bibliográfica ([4]).

Una vez resuelta esta dificultad, se comentó la conveniencia de programar una GUI (Graphics User's Interface) que permitiera al operador una manipulación sencilla de estos programas alejándolo del no siempre cómodo entorno Matlab en su línea de comandos.

En último lugar, con una idea de concepción global de la toma de datos, obtención de modelos e implementación final del control, se pensó en desarrollar una función que fuera capaz de presentar un archivo que pueda ser incluido directamente a OPTIMAX.

4.3 RESUMEN DE LA SOLUCIÓN ADOPTADA.

Tal y como hemos dejado entrever en las líneas anteriores el conjunto del problema puede ser descompuesto en tres grandes bloques: Adquisición de datos, manipulación de los mismos y presentación de estos datos a OPTIMAX. Aunque estas fronteras puedan aparentar ser muy claras, es también cierto que, en aras de la flexibilidad, estas se han difuminado en algunos casos por lo que es necesaria una expresión gráfica de la solución adoptada tal y como se recoge en la figura 4.1.

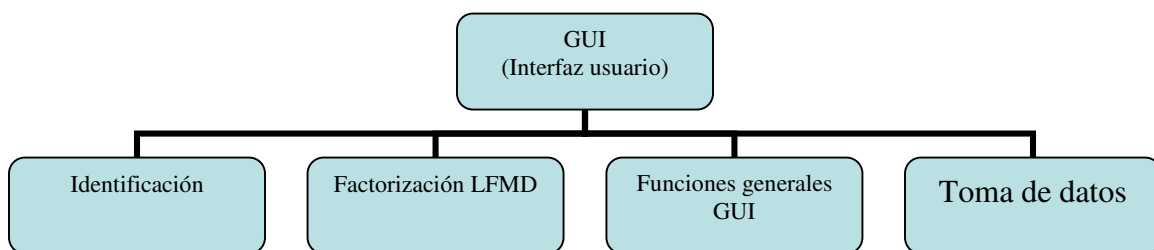


Figura 4.1 Descripción de la programación.

En los apartados posteriores se desarrollan las funciones englobadas en cada uno de estos bloques.

4.4 DESARROLLO DEL INTERFAZ DE USUARIO.

Un interfaz de usuario (GUI, del inglés Graphics User's Interface) es *“un conjunto de uno o varios paneles que están provistos de algunos de los siguientes elementos:*

- *Controles (botones, menús desplegables, editores de texto, etc.), que permitirán al usuario interactuar con el GUI y establecer el flujo de ejecución.*
- *Menús.*
- *Ejes de coordenadas, que permiten dibujar gráficos e imágenes.”([5])*

En nuestro caso vamos a utilizar para el desarrollo la herramienta que para este fin incluye el paquete informático Matlab. Esta es conocida como GUIDE (*Graphical User Interface Development Environment: Entorno de desarrollo de Interfaz Gráfico de Usuario*).

El beneficio que proporciona el uso de GUIs es evidente, ya que permiten al usuario ejecutar cómodamente código desarrollado en MATLAB sin necesidad de cumplir la incómoda sintaxis funcional necesaria cuando se trabaja desde la línea de órdenes. Además el elevado número de funciones que se maneja en la herramienta resulta, de esta forma, totalmente opaco al usuario que sólo debe preocuparse de ser capaz de invocarla desde la línea de comandos (cosa que se hace incluyendo en la misma **lfmdgui**).

A diferencia de la ejecución de funciones o scripts de MATLAB, la ejecución de GUIs no predetermina el flujo de ejecución del código. Es el usuario, a través de su interacción con el GUI, el que determina el orden en que se ejecutan las diferentes órdenes y funciones desarrolladas. Otra diferencia importante es que la ejecución no termina cuando finaliza la ejecución del script o función, sino que el GUI permanece abierto, permitiendo al usuario invocar la ejecución de ese u otro código desarrollado.

Vamos a usar el GUIDE para:

- Diseñar el aspecto del GUI.
- Programar el GUI: El GUIDE genera de forma automática un fichero .m que controla el funcionamiento del GUI. Este fichero .m inicializa el GUI y contiene un marco para todos los callbacks del GUI (las órdenes que se ejecutan cuando el usuario interactúa con un elemento del GUI). Usando el editor de MATLAB podremos añadir código a los callbacks para realizar las funciones que queramos asignarles.

El desarrollo de GUIs se realiza en dos etapas:

- Diseño de los componentes (controles, menús y ejes) que formarán el GUI.
- Codificación de la respuesta de cada uno de los componentes ante la interacción del usuario.

Para cualquier información acerca de GUIDE se recomienda la consulta de las siguientes reseñas bibliográficas ([5] y [6]). De ambos textos se ha incluido una copia en lo que hemos denominado biblioteca del proyecto por la importancia que han tenido en el desarrollo de este estudio.

El interfaz se ha configurado finalmente tal y como aparece representado a continuación, su programación se detalla en las propias funciones de Matlab y se describirán brevemente a continuación.

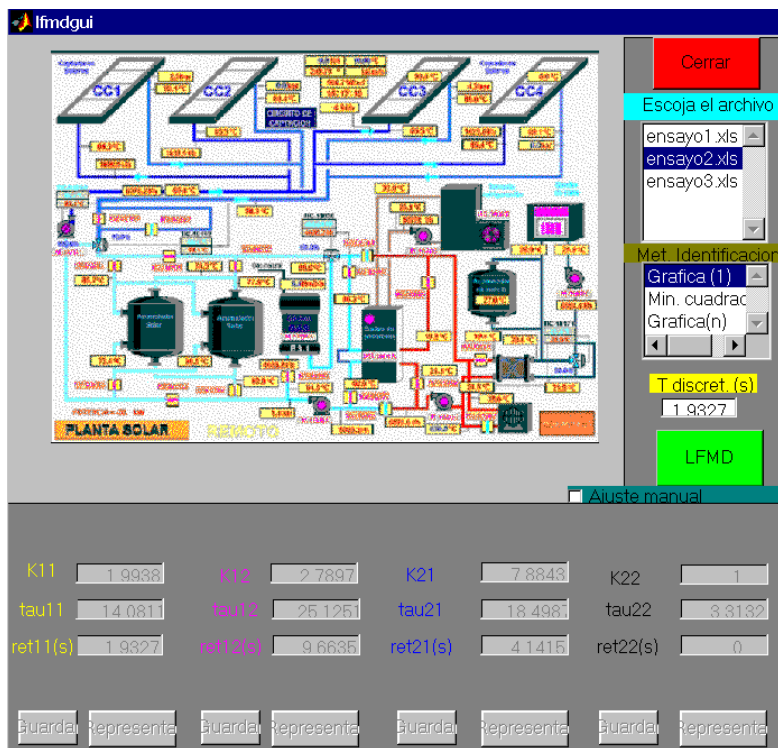


Figura 4.2-Interfaz de usuario.

4.4.1 DESCRIPCIÓN DEL PROCESO DE TOMA DE DATOS.

El proceso de toma de datos es tal vez el más sencillo conceptualmente aunque su desarrollo no sea simple. En primer lugar debemos decidir qué datos deben ser incluidos y, lo que es igualmente importante, cuando deben solicitarse al usuario. Para normalizar los datos que deben incluirse para determinar el modelo 2x2 de la planta solar (o de otro sistema a identificar) se hizo necesario definir una plantilla, en la que se incluirán los datos del ensayo. Esta plantilla se presenta en la figura 4.3.

empleado para la factorización no debe ser en ningún caso mayor que el mínimo de los retardos identificados no nulos. Dicha circunstancia queda reflejada en la siguiente figura.

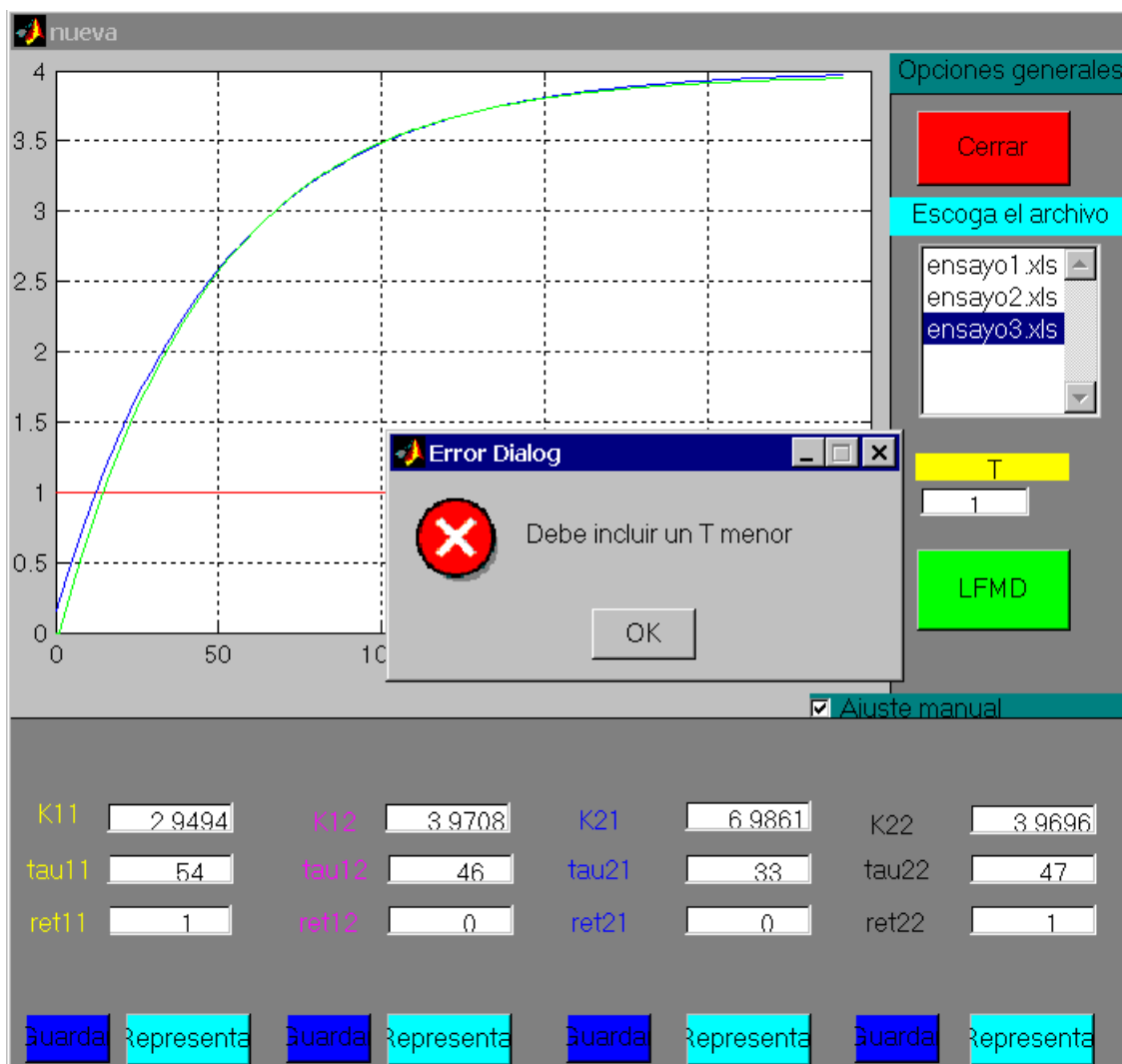


Figura 4.4 Error al incluir el tiempo de discretización al sistema.

4.4.2 PROGRAMACIÓN DE LA FACTORIZACIÓN LFMD.

Esta parte del código constituye el propio corazón de la aplicación. De hecho esta fue la primera en resolverse, el resto de aplicaciones (identificación, toma de datos...etc) se desarrollaron únicamente como una forma de simplificar el proceso que permitía el cálculo de la factorización LFMD.

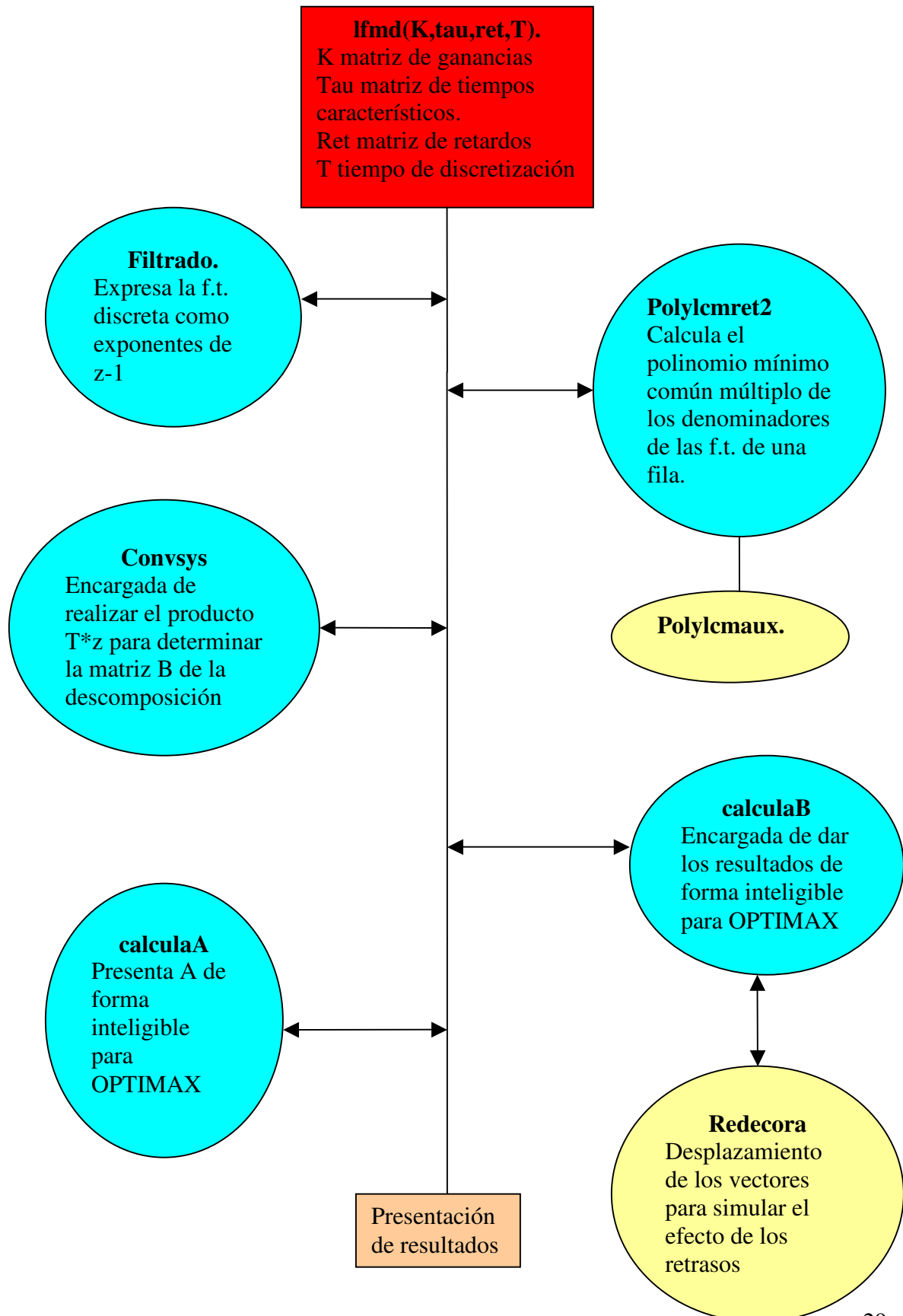
También, como consecuencia de esta circunstancia, cabe destacar que este paquete de programas puede manejar sistemas multivariables de orden $N \times N$ siempre que estos sean aproximados por un modelo en función de transferencia de primer orden (caracterizado por los valores de A , τ y el posible retardo L) o bien sistemas multivariables de dos entradas y dos salidas aproximados por modelos de orden N .

Sin embargo, para el uso del GUI se ha modificado convenientemente la presentación de datos con la idea de manejar sistemas multivariables 2×2 aproximados cada uno por modelos de primer orden.

Para cualquier duda acerca del manejo de las funciones incluidas en este paquete se recomienda al lector consulte el soporte informático de este documento en el que se incluye la programación de las mismas con sus comentarios aclarativos. También es recomendable, si el lector así lo considera, la consulta de las siguientes referencias de la bibliografía ([8],[9],[10] y [11]), algunas de ellas incluidas también en la biblioteca de proyecto.

Sin otro ánimo que explicar de forma global cómo trabajan y qué aportan cada una de las funciones que conforman el paquete para el cálculo de la LFMD se incluye un diagrama de procesos en la figura de la siguiente página.

Figura 4.5 Diagrama de proceso en factorización LFMD



4.4.3 PROGRAMACIÓN DEL PROCESO DE IDENTIFICACIÓN.

Se entiende por identificación de sistemas a la obtención de forma experimental de un modelo que reproduzca con suficiente exactitud, para los fines deseados, las características dinámicas del proceso objeto de estudio ([12]).

Tal y como vimos en el capítulo anterior para realizar la factorización LFMD es necesario disponer de un modelo de la instalación en estudio. El GUI está capacitado para la identificación de sistemas, esta identificación se realiza mediante la función **identificación**.

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

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

5. *Validación del modelo.* El último paso consiste en determinar si el modelo obtenido satisface el grado de exactitud requerido para la aplicación en cuestión. Si se llega a la conclusión de que el modelo no es válido, se deben revisar los siguientes aspectos como posibles causas:

- a) El conjunto de datos de entrada-salida no proporciona suficiente información sobre la dinámica del sistema.
- b) La estructura escogida no es capaz de proporcionar una buena descripción del modelo.
- c) El criterio de ajuste de parámetros seleccionado no es el más adecuado.

Dependiendo de la causa estimada, deberá repetirse el proceso de identificación desde el punto correspondiente. Por tanto, el proceso de identificación es un proceso iterativo, cuyos pasos pueden observarse en la figura 4.6.

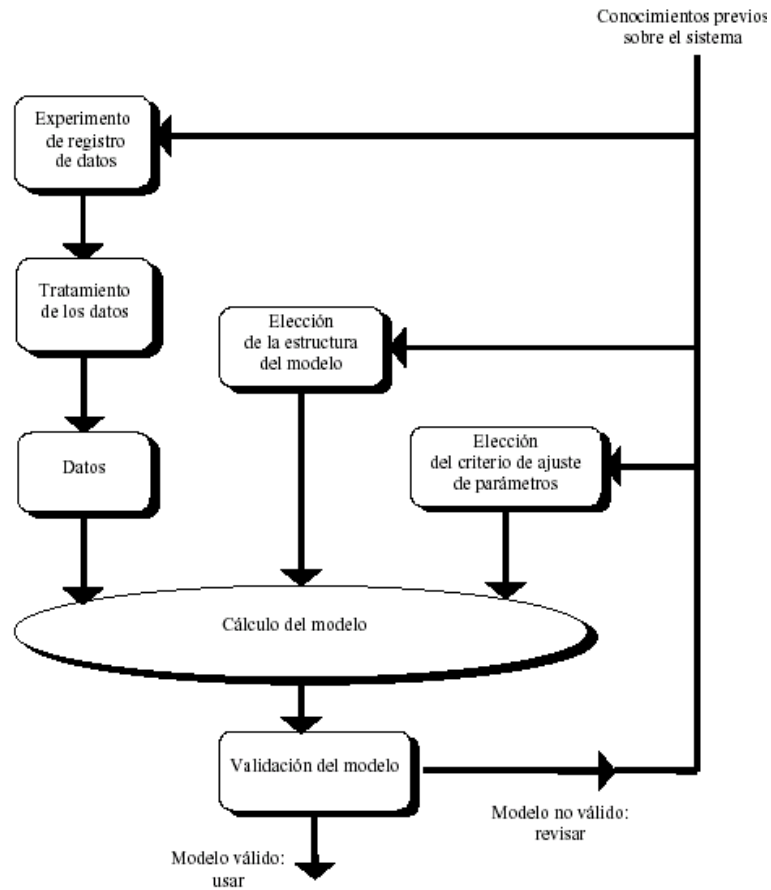


Figura 4.6 Proceso de identificación.

En muchas ocasiones es necesario realizar la identificación de un sistema del cual no se tiene ningún tipo de conocimiento previo. En estos casos, se suele recurrir a modelos estándar, cuya validez para un amplio rango de sistemas dinámicos ha sido comprobada experimentalmente. Generalmente estos modelos permiten describir el comportamiento de cualquier sistema lineal. La dificultad radica en la elección del tipo de modelo (orden del mismo, número de parámetros, etc.) que se ajuste satisfactoriamente a los datos de entrada - salida obtenidos experimentalmente. En nuestro caso hemos decidido identificar cada una de las respuestas del sistema ante la variación de una variable manipulable como un sistema de primer orden caracterizado por una ganancia A , un tiempo característico τ y un posible retardo L .

$$G(s) = \frac{A}{\tau * s + 1} e^{-Ls}$$

La estimación de los parámetros de este modelo para cada una de las señales de salida puede realizarse de muy diversos métodos, en particular, en nuestra herramienta esto es posible mediante tres técnicas distintas que comentamos brevemente a continuación.

Método gráfico 1.

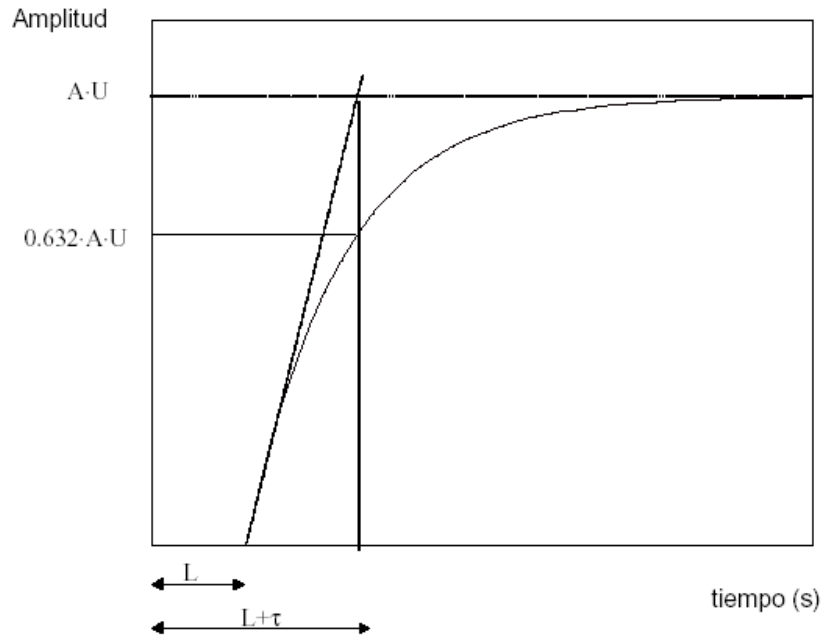


Figura 4.7 Respuesta de modelo de primer orden.

Si la respuesta al escalón del proceso tiene la forma de la figura 4.7, que responde a un sistema de primer orden, la expresión que describe la respuesta es:

$$y(t) = A * U (1 - e^{\frac{-(t-L)}{\tau}})$$

U es la amplitud del escalón aplicado a la entrada.

Basta entonces considerar un tiempo tal que cumpla la siguiente igualdad:

$$t = L + \tau$$

Para dicho tiempo la respuesta del sistema debe ser, como vemos a continuación, el 63.2% del valor que toma la salida en el régimen permanente.

$$y(L + \tau) = A * U (1 - e^{-1}) = 0.632 * A * U$$

Si suponemos, como en muchos libros especializados, que se puede aproximar el retardo de un sistema por el tiempo en que la salida tarda en alcanzar el 5% de su valor en régimen estacionario, y denominamos a este tiempo t_2 , es fácil resolver el siguiente sistema:

$$t_{0,632} = L + \tau$$

$$t_{0,05} = L$$

Lo que nos proporciona los parámetros que andamos buscando ya que la ganancia del modelo de primer orden puede determinarse a partir de:

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

Método de mínimos cuadrados (LSE).

Este método está basado en una idea que está muy extendida en el mundo del control, consiste en considerar un índice, dependiente de los parámetros que se quiere identificar, y el cual debe minimizarse. En este caso el índice es conocido como *criterio de mínimos cuadrados para una regresión lineal* y se define como sigue:

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

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

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

Para programar este método de identificación se ha recurrido a una de las funciones del toolbox de identificación denominada **arx** que es capaz de proporcionar una estimación

de un modelo ARX usando mínimos cuadrados recursivos a partir de una estructura de datos en la que se recoge la evolución tanto de la salida como de la señal manipulable.

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

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

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

El modelo ARX identificado por la función del mismo nombre es una simplificación de la estructura original en la que se consideran los polinomios de la expresión general como $C=D=F=1$. De esta forma el modelo viene expresado como:

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

La función **arx** nos devuelve los coeficientes de ambos polinomios característicos y, mediante el uso de otras funciones más avanzadas del toolbox de identificación como **thd2thc** y **th2tf**, es posible determinar los parámetros del modelo de primer orden para cada salida en cuestión.

También podría haberse resuelto a partir de las relaciones ya conocidas entre los coeficientes de A y B y los parámetros de un sistema de primer orden, a saber:

$$a = e^{-\frac{T}{\tau}}$$

$$b = A * (1 - a)$$

Esta solución no se ha adoptado ya que al hacer uso de las funciones de identificación es posible identificar el modelo como uno de mayor orden, y posteriormente reducir este de forma matemática y no por simple eliminación como sería el resultado de emplear las anteriores expresiones.

El retardo del sistema L se ha identificado haciendo uso de la función **impulse** del mismo toolbox de identificación. Para cualquier tipo de consulta sobre la programación de este método se recomienda la lectura de las siguientes referencias bibliográficas: ([12] y [13]).

Método gráfico 2.

También es frecuente el caso de que la respuesta tenga la forma mostrada en la figura 4.8. En este caso se trata de aproximar la respuesta de un sistema de orden superior mediante un modelo de primer orden. Esta aproximación se usa frecuentemente para diversos propósitos (ajustes preliminares de los parámetros de un PID, compensación feed-forward, predictor de Smith, etc.).

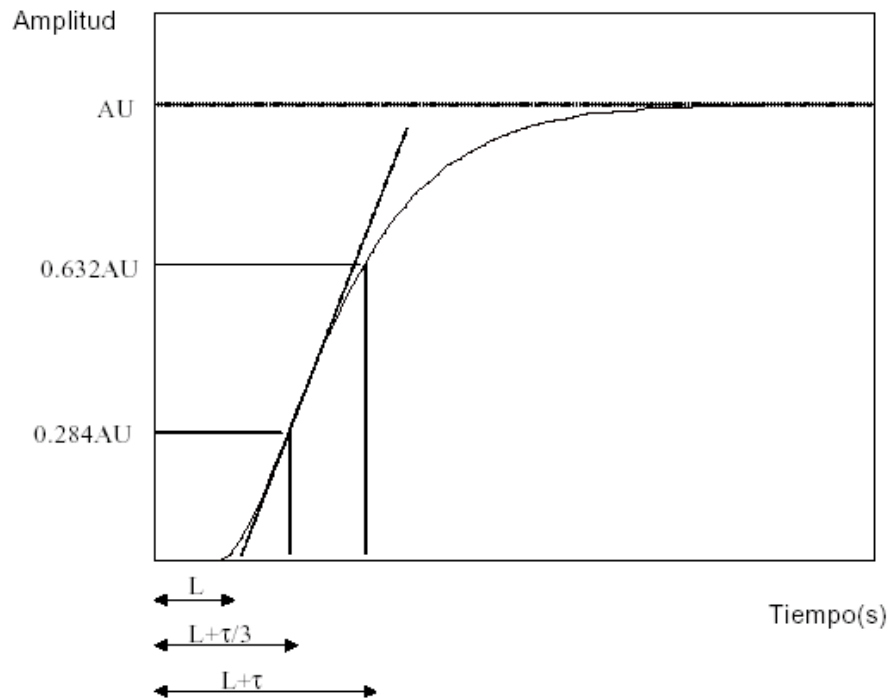


Figura 4.8 Método gráfico de identificación.

Los parámetros del modelo pueden calcularse en este caso usando la construcción gráfica mostrada en la figura y siguiendo el procedimiento que se describe a continuación:

1. Se toman dos puntos de la curva que corresponden al 63.2% y al 28.4% del valor estacionario final de la salida y se determinan $t_{0,284}$ y $t_{0,632}$, siendo estos los tiempos para los que se alcanzan estos valores.

2. Se plantea el siguiente sistema de ecuaciones:

$$t_{0,284} = L + \frac{\tau}{3}$$

$$t_{0,632} = L + \tau$$

3. Se calculan los parámetros L y τ que son solución del sistema anterior.

4. Una vez más la ganancia del sistema se calcula como:

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

La función que realiza la identificación del sistema ha sido denominada **identificación**, y entre sus parámetros de entrada (además de la señal manipulable, la respuesta del sistema y el tiempo de muestreo empleado en el ensayo), consta un número escalar cual de los tres métodos desea emplear el usuario. Esta morfología permite establecer un listbox en el GUI mediante el cual el operador fija fácilmente el método de identificación. A continuación se muestra un detalle de dicho listbox encuadrado en el interfaz de usuario programado.

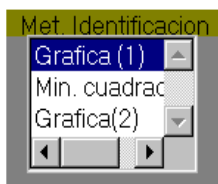


Figura 4.9 Selección del método de identificación.

Conforme al proceso de identificación que se describió al comienzo de este apartado se han realizado varios ensayos para realizar la denominada validación del modelo. Estos ensayos se recogen en el siguiente capítulo de este documento. Si bien es momento de indicar que no existe un método único que permita identificar sin ningún grado de incertidumbre el modelo de un sistema, por ello se ha dejado la libertad al usuario de realizar un ajuste manual de los parámetros inicialmente identificados por el sistema y observar, de forma gráfica, la benevolencia de las aproximaciones.

Para acceder a la zona de ajuste manual debemos picar sobre el checkbox denominado “Ajuste manual” esto provocará que los editores de texto para los parámetros se activen (lo que se traduce en que dejan de verse con el fondo gris que toman al abrirse el GUI) y los botones de representación y guardar para cada una de las funciones de transferencia estén activados.

☒ Ajuste manual

K11	2.9494	K12	3.9708	K21	6.9861	K22	3.9696
tau11	54	tau12	46	tau21	33	tau22	47
ret11	1	ret12	0	ret21	0	ret22	1

Guardar	Representa	Guardar	Representa	Guardar	Representa	Guardar	Representa
---------	------------	---------	------------	---------	------------	---------	------------

Figura 4.10 Detalle de la zona para ajuste manual de parámetros.

Mediante el botón “Representar” para cada una de las f.t. es posible ver la aproximación entre las señales reales y las que tendría el sistema identificado ante esa excitación de entrada, la representación aparecerá en unos ejes incluidos en el GUI con ese fin.

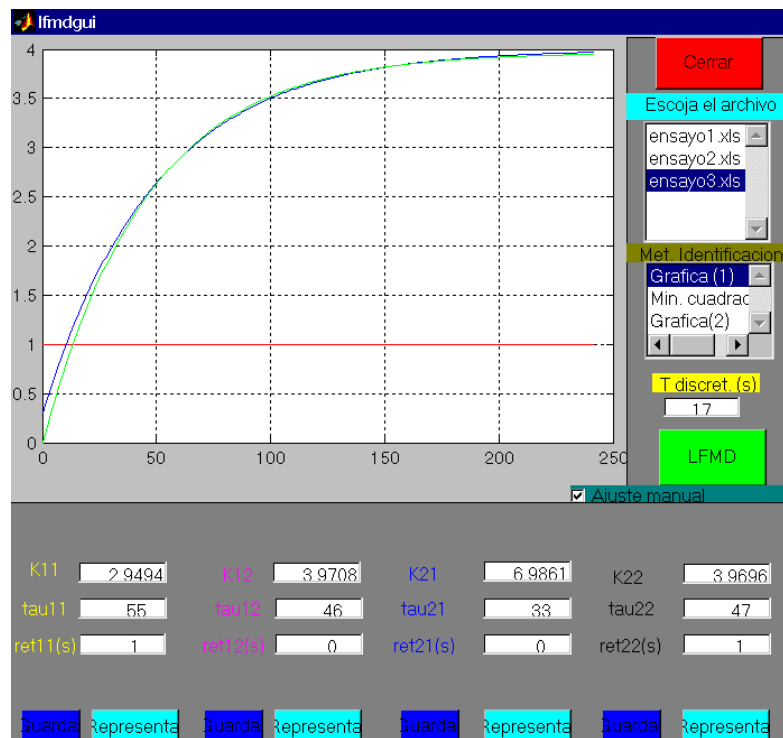


Figura 4.11 GUI mostrando sistema identificado.

4.4.4 FUNCIONES GENERALES DEL GUI.

Bajo este epígrafe se recogen todas las sentencias que, incluidas en la función **lfmdgui**, están destinadas bien a conseguir efectos de presentación, que hacen más fácil el uso de la ventana interactiva, bien a analizar los datos incluidos por el usuario comprobando que son correctos (una de estas puede ser la circunstancia ya comentada en que se incluye un tiempo T para la factorización LFMD que no es apropiado).

Se han programado mensajes de aviso que recuerdan al usuario que el valor añadido en cualquiera de los editores de texto debe ser un número como es el caso de la siguiente figura.

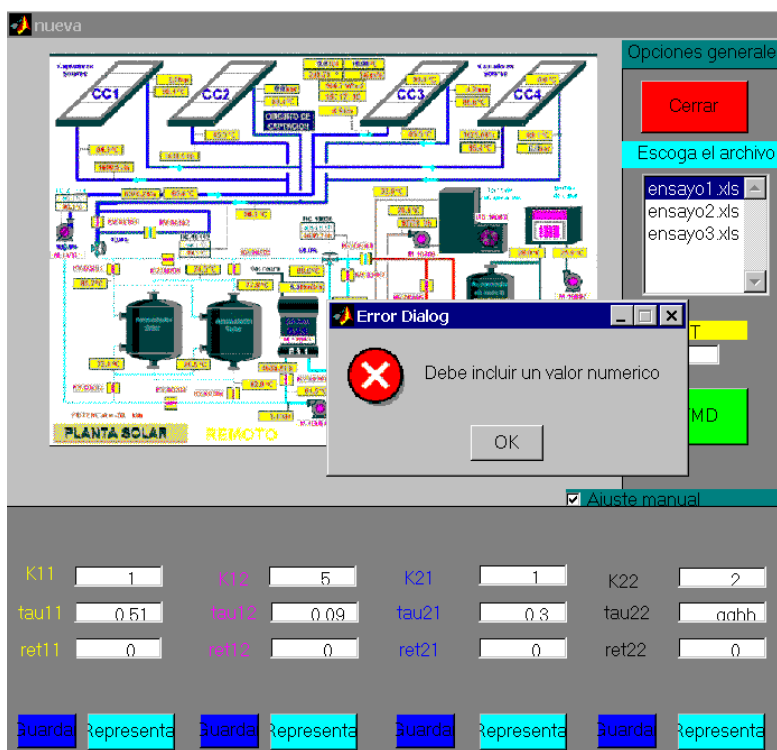


Figura 4.12 Error al incluir tau22.

En esta categoría también podría incluirse el mensaje que aparece cuando pretendemos representar alguna función de transferencia identificada para la cual no tenemos datos válidos, tal y como aparece en la siguiente figura:

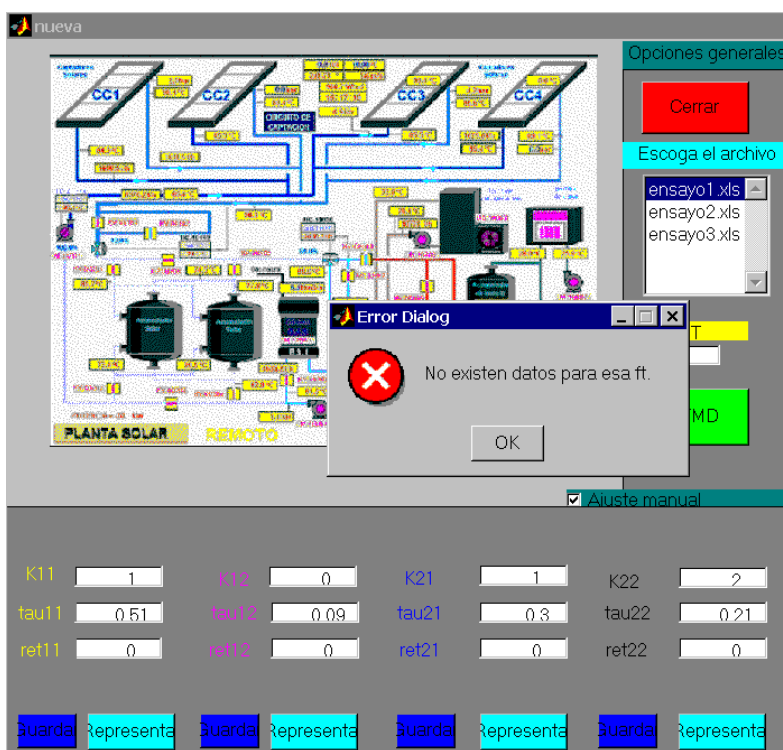


Figura 4.13 Datos no válidos / inexistentes en una f.t.

Merece así mismo la pena señalar que al pulsar el botón “Cerrar” aparece una ventana conocida como “ventana de confirmación” que pregunta al usuario si es realmente su intención salir del interfaz.

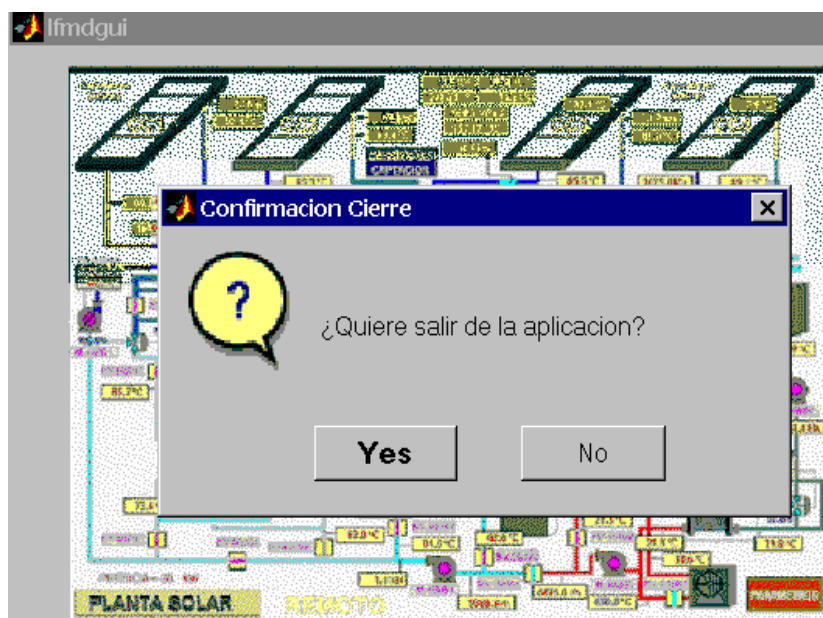


Figura 4.14 Ventana de confirmación de salida.

