CAPÍTULO 6

EJEMPLOS DE FUNCIONAMIENTO DE LA HERRAMIENTA

6.1.- Objeto del capítulo.

En este capítulo final lo único que se pretende es incluir distintos ejemplos de funcionamiento para la solución programada. Consideraremos los tres ensayos que hemos incluido en el interfaz y se han definido como ensayo 1, ensayo 2 y ensayo 3.

Es conveniente reseñar que ha sido necesario definir un archivo para cada uno de los ensayos según la plantilla que hemos definido en Excell. Esto no limita el uso de la herramienta a estos archivos, ya que los datos incluidos en estos archivos podrán actualizarse con datos de nuevos ensayos y obtener la descrpción que andamos buscando para el sistema.

6.2- Primer ejemplo.

El presente ejemplo viene recogido en un texto que ha sido de inestimable ayuda en la realización de la programación para la descripción LFMD, este documento no es otro que "Model Predictive Control" ([4]).

En este texto se expone el método conocido como descomposición LFMD, así como un ejemplo de realización de la misma. El sistema a considerar es un sistema multivariable 2x2 que viene descrito por el siguiente comportamiento dinámico:

$$G(s) = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix}$$

Donde cada una de las funciones de transferencia reflejadas son modelos de primer orden que aproximan la dinámica del sistema. Dichas funciones de transferencia para este caso particular son:

$$g_{11}(s) = \frac{1}{0.7s+1} \qquad g_{12}(s) = \frac{5}{0.3s+1}$$

$$g_{21}(s) = \frac{1}{0.5s+1} \qquad g_{22}(s) = \frac{2}{0.4s+1}$$

El primer paso consiste en modificar la plantilla convenientemente de forma que en la misma se incluyeron las respuestas ante escalón para cada una de estas funciones, sin olvidar incluir el tiempo de muestreo tomado en el ensayo que fue tm=0.0166 s.

Ahora estamos en condiciones de ejecutar nuestro interfaz, para ello debemos teclear **lfmdgui** en la línea de comandos de Matlab, lo que ejecutará la ventana gráfica. A continuación accionaremos el botón “LFMD” y los resultados se presentan como indica la figura siguiente.

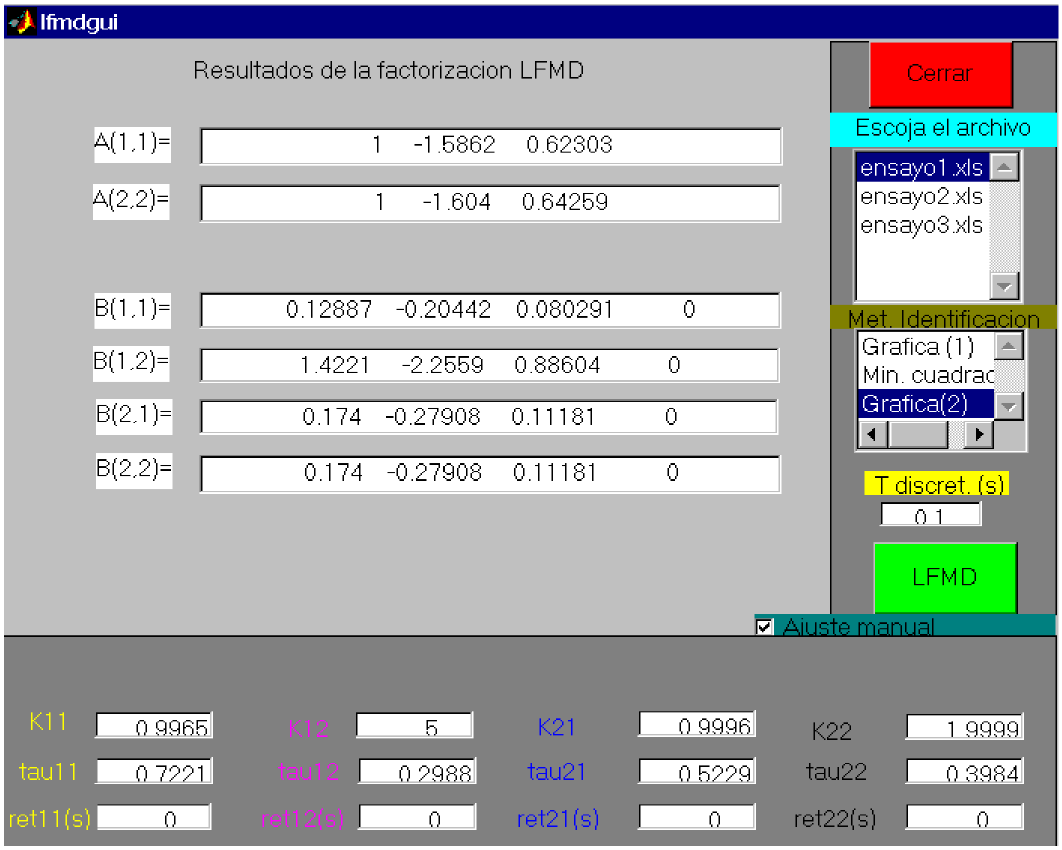

Figura 6.1 Factorización LFMD para ejemplo 1.

En la figura anterior se puede ver la presentación de resultados que proporciona el interfaz. Debemos recordar que la matriz A es diagonal, de ahí que sólo aparezcan mostrados sus elementos diagonales. Por el contrario la matriz B, por regla general, será una matriz completa de ahí que aparezcan cuatro términos.

De esta figura anterior se puede deducir el tiempo de discretización que se ha empleado en el ensayo que, en este caso ha sido de T=0.1 s, el archivo desde el que hemos tomado los datos y el método de identificación empleado.

Además de esta presentación en pantalla la función **lfmd** presenta estos datos también a través de la línea de comandos, esto se ha hecho de esta forma para conservar utilizable la capacidad de dicha función de tratar modelos multivariables mayores a 2x2. La función a la que hacemos referencia tendrá el siguiente formato:

[A,B]=lfmd(A,Tau,R,T)

donde cada uno de los términos se corresponde con:

A. Es una matriz cuadrada de orden igual al del sistema que pretendemos identificar, un término A(i,j) de la misma representará el valor del parámetro identificado como la ganancia del modelo de primer orden correspondiente a la respuesta en la salida i al darse un escalón en la entrada j y mantener el resto de entradas constantes.

Tau. De forma idéntica a la matriz anterior pero ahora el valor Tau(i,j) representa el tiempo característico identificado para el modelo de primer orden.

R. En esta matriz irán recogidos los retardos para cada uno de los modelos. En caso de tener sistemas sin retardo debe incluirse una matriz de ceros cuadrada y del orden del sistema por identificar.

T. Se corresponde con el tiempo que queremos emplear para la discretización, de forma idéntica a la aplicación con el GUI, si operamos desde la línea de comandos e incluimos un tiempo no válido, principalmente motivado porque este valor sea mayor que el mínimo de los tiempos no nulos de retardo del sistema, lo que provoca pérdida de información en el proceso.

Las variables de salida de esta función serán las matrices A y B de la factorización LFMD, de nuevo sólo aparecerán representados en la línea de comandos los elementos diagonales de la matriz A .

Si estuviésemos interesados en sistemas de orden 2x2 pero aproximados por modelos de mayor orden deberíamos emplear la función **lfmd2** incluida en el soporte informático de este documento.

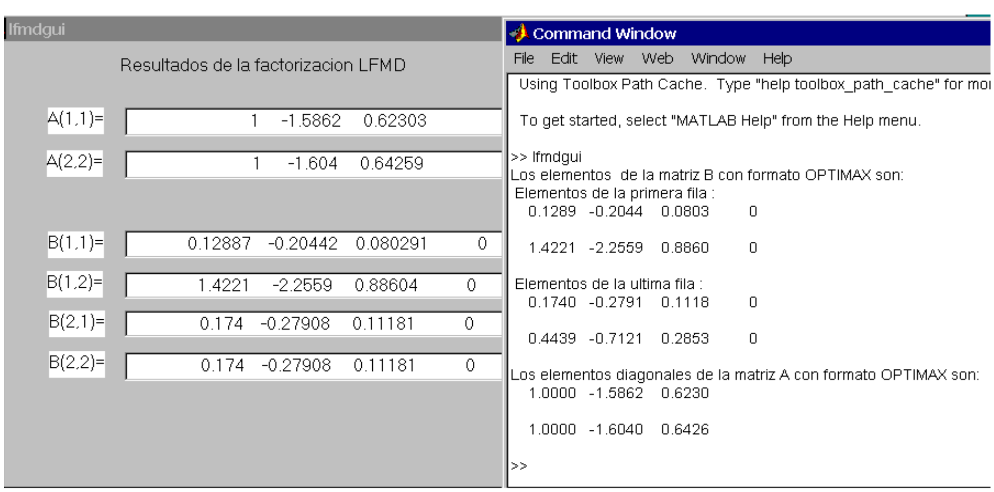

Figura 6.2 Ambos formatos de presentación de datos para el ejemplo 1.

6.3- Segundo ejemplo: Sistemas con retardo.

En este caso vamos a mostrar que la herramienta puede considerar sistemas con retardo, para este fin emplearemos los datos incluidos en el archivo ensayo2 y se corresponden con el siguiente sistema:

$$g_{11}(s) = \frac{2}{15s+1}e^{-3s} \qquad g_{12}(s) = \frac{3}{30s+1}e^{-10s}$$

$$g_{21}(s) = \frac{8}{20s+1}e^{-5s} \qquad g_{22}(s) = \frac{1}{5s+1}$$

En este caso la pantalla final presenta los siguientes resultados para la factorización LFMD.

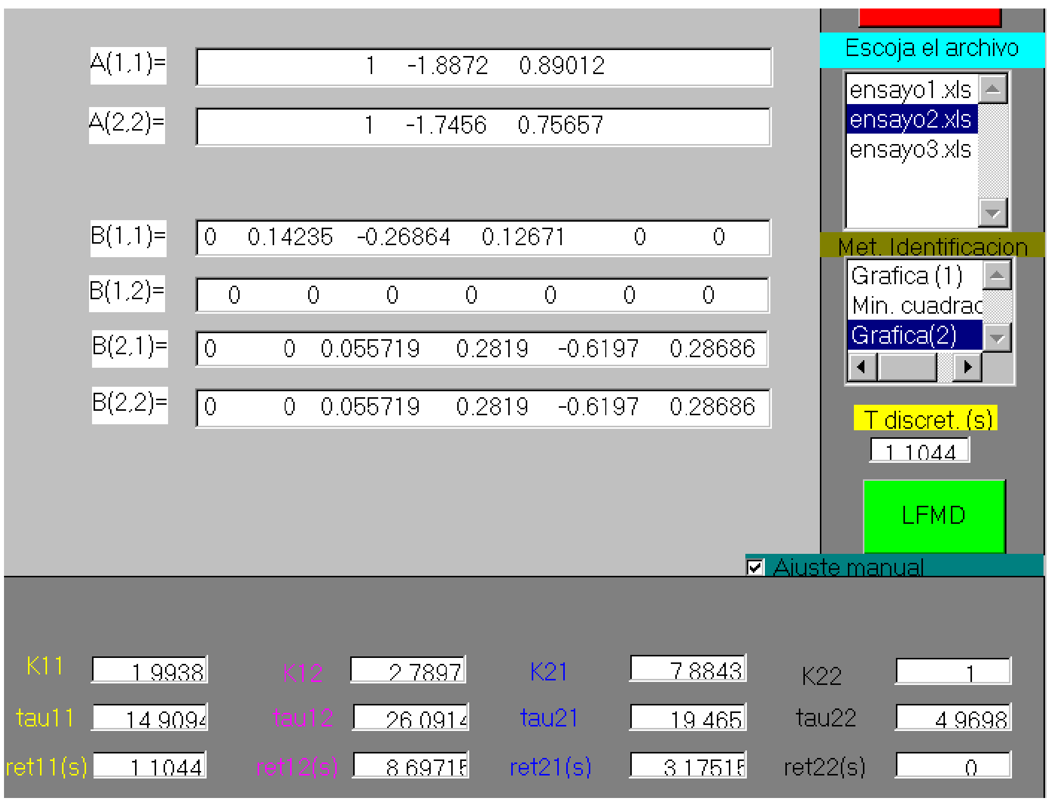

Figura 6.3 Factorización LFMD para el caso del ejemplo 2.

Se pueden observar ciertos aspectos en la solución que no ocurrían antes, el programa ha elegido de forma conveniente el valor de T, aunque se podría modificar antes de la factorización, además obsérvese que han aparecido varios términos nulos en las soluciones presentadas por la herramienta, este es el efecto del retraso. La aparición de tantos términos nulos es, tal como se explica en el anexo B, debido a la forma característica de tomar los datos que tiene OPTIMAX, según la cual los polinomios que describen el sistema deben ser todos de mismo orden, de forma que si alguno de ellos es de orden menor (potencia más negativa menor)se le añadirán ceros por detrás para que todos los polinomios tengan la misma longitud. La herramienta presenta los datos de esta manera, de forma que los polinomios se escriben en orden descendente de z^{-i} empezando a partir de z^0.

6.4- Tercer ejemplo: Una aplicación global de manejo de la herramienta.

A continuación se describe linealmente el proceso que nos ocupa en este proyecto. Trataremos de explicar el uso global de esta herramienta en relación con el entorno que constituye la planta solar y en particular el SCADA.

Figura 6.4 funcionamiento global de la herramienta.

El primer paso consiste necesariamente en la obtención de los datos a partir de la relación de históricos que han sido generados con el SCADA, para ello emplearemos la interfaz de DDCUBE tal y como se describe en el apéndice A en donde se explica cómo obtener estos históricos. A continuación seleccionamos los datos de las salidas y entradas de las variables que pretendemos identificar y pegamos los mismos en la plantilla definida previamente ya sea en el archivo “ensayo3.xls”, como es el caso, o bien sobre las plantillas “ensayo1.xls” y “ensayo2.xls”.

	INTERFAZ	PARA	DETERMINAR	MODELO	DE	LA	PLANTA	SOLAR
Tiempo de ensayo(s)	Determinación	G11	Determinación	G12	Determinación	G21	Determinación	G22
1	VM1	TT10141_PV	VM3	TT10141_PV	VM1	TT10154_PV	VM3	TT10154_P
	1	0,0000	1	0,0000	1	0,0000	1	0,000
	1	0,0000	1	0,0000	1	0,1728	1	0,000
	1	0,0000	1	0,0000	1	0,3414	1	0,000
	1	0,0000	1	0,0000	1	0,5058	1	0,000
	1	0,0000	1	0,0792	1	0,6661	1	0,000
	1	0,0496	1	0,1568	1	0,8225	1	0,000
	1	0,0984	1	0,2329	1	0,9750	1	0,079
	1	0,1463	1	0,3075	1	1,1238	1	0,156
	1	0,1935	1	0,3807	1	1,2689	1	0,232
	1	0,2399	1	0,4523	1	1,4104	1	0,307
	1	0,2855	1	0,5226	1	1,5484	1	0,380
	1	0,3304	1	0,5914	1	1,6830	1	0,452
	1	0,3745	1	0,6589	1	1,8143	1	0,522

Figura 6.5 Detalle del aspecto de la plantilla al incluir los datos propios del ensayo 3

Una vez tenemos esta plantilla debemos abrir la aplicación Matlab y teclear en la línea de comandos la orden **lfmdgui**, lo que provocará la apertura de la ventana del interfaz programado.

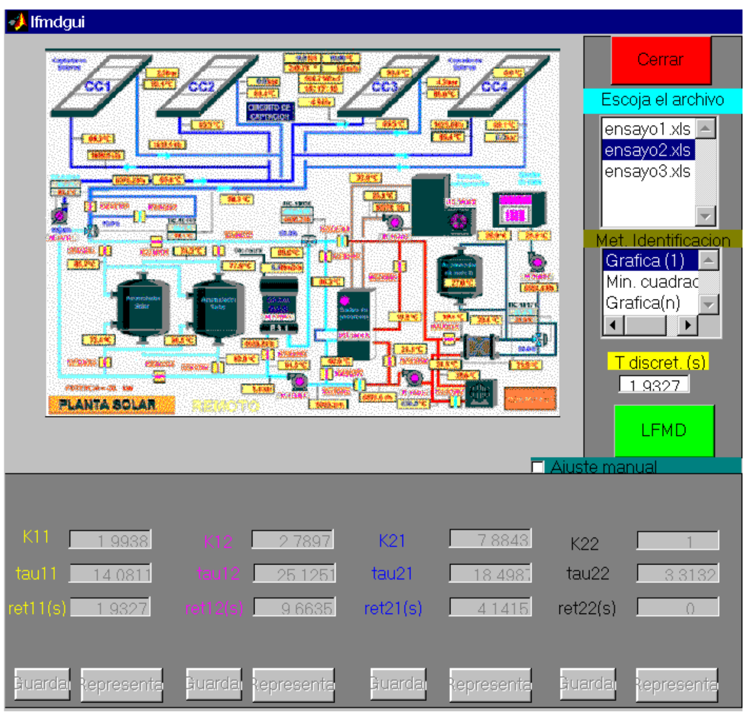

Figura 6.6 Apertura al ejecutar la orden lfmdgui.

El siguiente paso consistirá en seleccionar el archivo en el que hallamos añadido los resultados del ensayo, en nuestro caso el ensayo 3, así como el método mediante el cual queremos realizar la identificación del sistema, en nuestro caso emplearemos el método gráfico 2. Si accionamos la pestaña denominada "Ajuste manual" podremos ver de forma gráfica como ajustan los parámetros del sistema a la nube de datos introducida.

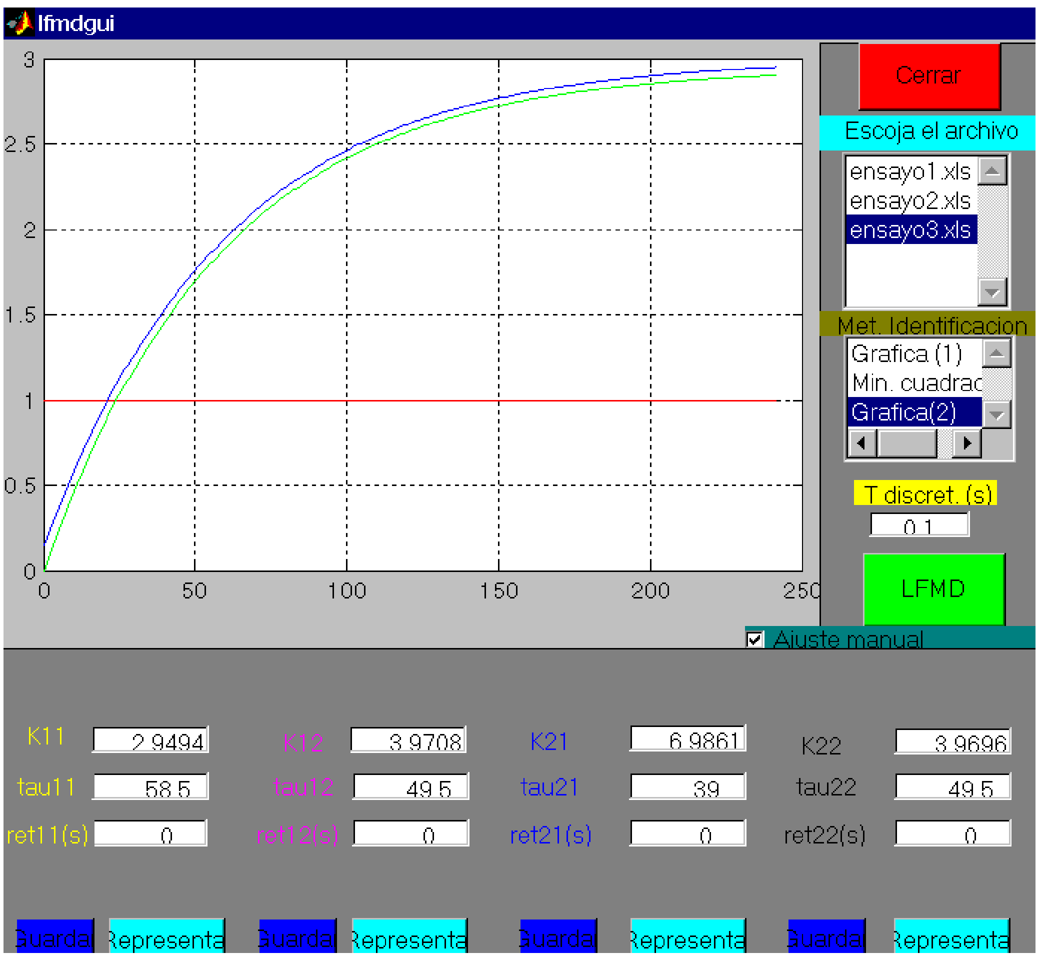

Figura 6.7 Identificación de la f.t. g11

Sobre esta ventana es posible variar cualquiera de los datos referentes a la f.t. que estamos manipulando, intentando ajustar, en caso de ser necesario, de forma manual estos parámetros. En la siguiente figura se ha manipulado el valor de k11 para poner de manifiesto esta posibilidad. Como se observa esta manipulación se ve reflejada en la representación conjunta de datos y estimación proporcionada por el GUI.

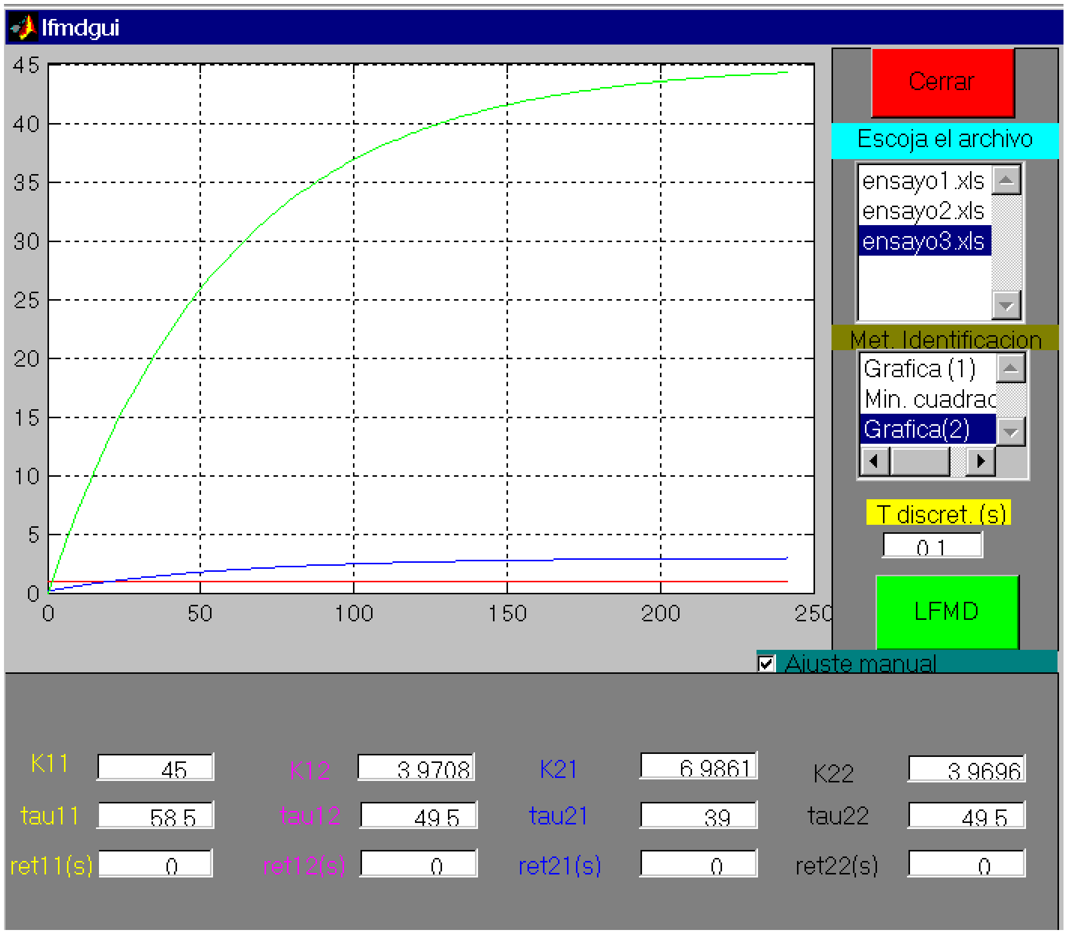

Figura 6.8 Capacidad de ajuste de los datos identificados

Una vez estemos satisfechos con la estimación de los parámetros para cada una de las funciones de transferencia con su nube de datos y habiéndolos guardado convenientemente podremos realizar la factorización LFMD programada. Para ello seleccionamos el tiempo de discretización, si creemos conveniente modificarlo, y accionamos el botón etiquetado como “LFMD” con lo que conseguiremos la descripción del modelo que buscábamos.

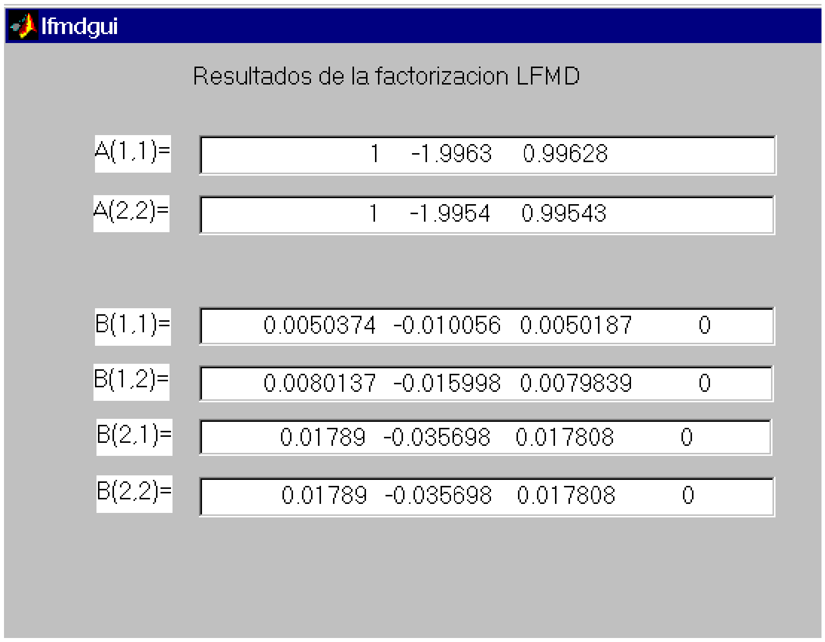

Figura 6.9 Resultados para el ensayo 3

6.4- Conclusiones.

El desarrollo de esta herramienta ha simplificado en gran medida un proceso de adquisición de datos e identificación que antes podía considerarse bastante tedioso. Según hemos visto, su manejo es harto sencillo, especialmente cuando reconocemos la serie de toolbox que se han empleado en la elaboración de la misma, estos son el de control y el identificación así como otros manuales avanzados de programación en entorno Matlab.

Si bien el uso de una ventana gráfica aleja al usuario de problemas relacionados con las órdenes internas que maneja el GUI esto no limita la capacidad de interacción que tiene el usuario con el sistema. Esto se ha puesto de manifiesto al permitir al usuario ajustar distintos parámetros de funcionamiento, para permitir, por ejemplo, una identificación bastante exacta de las funciones de transferencia que venimos manejando.

Además hemos logrado programar una función capaz de realizar la factorización LFMD evitando este paso que antes era necesario dar de forma manual. No sólo hemos programado una función capaz de manejar sistemas 2x2, sino que podría tratar con sistemas NxN, con las limitaciones expuestas en este documento. Aunque esta herramienta está pensada para un entorno dado por la planta solar es totalmente útil a la hora de calcular la identificación y modelado discreto de cualquier sistema.

Por todo lo expuesto con anterioridad consideramos que los objetivos que nos habíamos propuesto al comienzo de elaborar este proyecto han sido alcanzados e incluso superados.

D. Tomás Fco. Garrido López 20/04/05