

Identificación de sistemas

IDENTIFICACIÓN DE SISTEMAS.....	1
1. INTRODUCCIÓN	2
2. IDENTIFICACIÓN DE SISTEMAS.....	3
3. IDENTIFICACIÓN UTILIZANDO LA FUNCIÓN DE MATLAB “IDENT”	4
4. IDENTIFICACIÓN MEDIANTE LA RESPUESTA AL ESCALÓN Y ARX.....	8

1. Introducción

La identificación de sistemas tiene por objeto el cálculo del modelo matemático de un sistema físico. Estas técnicas de identificación pueden ser clasificadas en dos grandes grupos:

1. La identificación mediante la aplicación de señales especiales de excitación que permiten obtener gráficas de las que extraer el modelo de la FDT (obteniendo la FDT de sistemas continuos) y,
2. Las técnicas basadas en la identificación de parámetros mediante la minimización de errores (obtenido sistemas FDT de sistemas discretos).

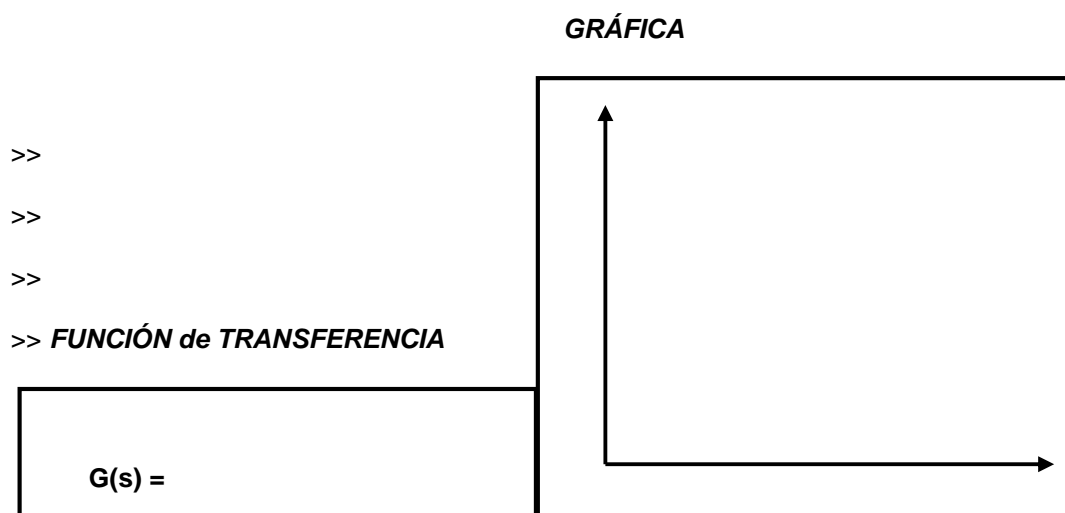
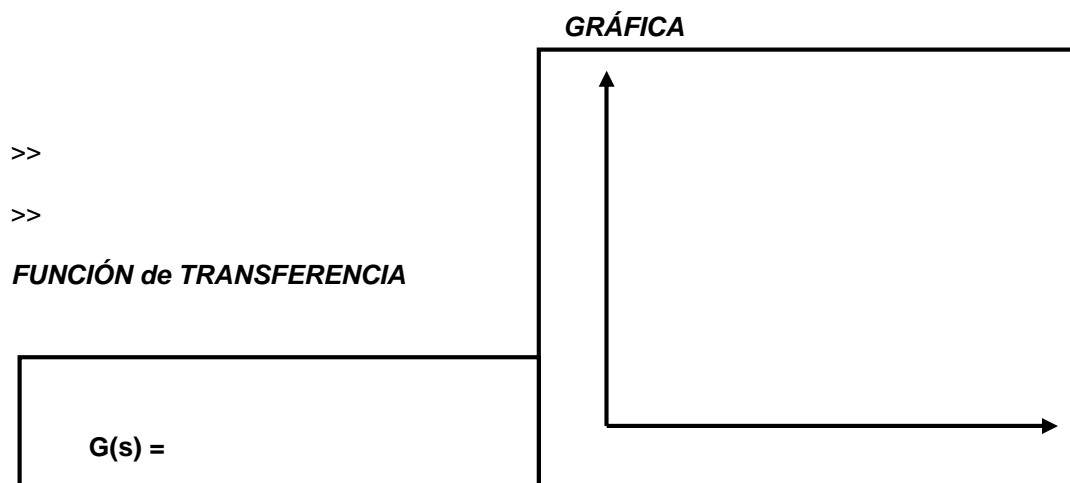
MatLab® tiene funciones que permiten, tanto la representación de gráficos con los datos de salida y entrada (`plot`), como el procesamiento de datos según diferentes algoritmos de identificación (`arx`, `armax`, `bj`, `oe`). Estas últimas son más complejas y precisan de definir los datos de forma conveniente para la correcta estimación de parámetros.

2. Identificación de sistemas

1. Cargar el fichero **datosG?.mat** donde ? corresponde con el número del grupo de prácticas al que perteneces. En este fichero encontrarás las variables *tiempo (t)*, *amplitud (y)* correspondientes a la respuesta ante entrada escalón unitario de dos sistemas.

2. Dibujar las gráficas en MATLAB y, a la vista del resultado, **identificar numéricamente** la correspondiente función de transferencia (puedes emplear cualquiera de los métodos que te hayan contado en clase). Define en MATLAB las correspondientes funciones de transferencia para su comprobación.

Nota: Utilice los comandos load, plot, step, tf, zpk. Si no sabe cómo funciona alguno de ellos, consulte el "help" del software a través de la línea de comandos. (>> help load)



3. Identificación utilizando la función de Matlab “ident”

Con el programa MATLAB abierto importamos desde *workspace* los datos que representan las variables (**t**, **y**) del apartado anterior. Se genera un vector de datos correspondiente a una entrada escalón.

Una vez que tenga las tres variables en el *workspace* procedemos a llamar a la interfaz gráfica de usuario (GUI) con la siguiente instrucción:

```
>>ident
```

En el *command window* y aparece la ventana de interfaz gráfica GUI del sistema de identificación como se muestra en la figura 1

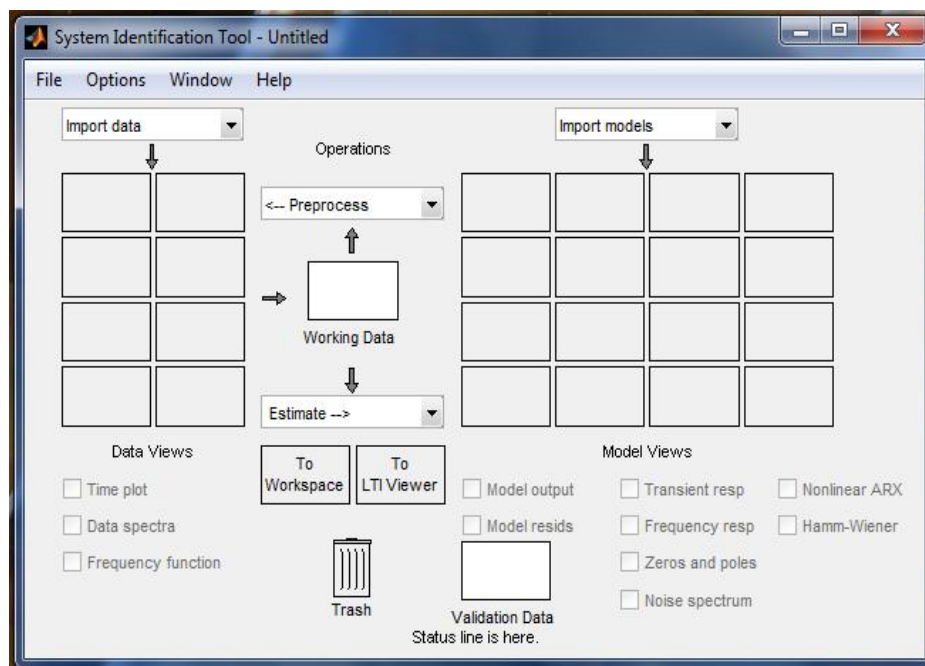


Figura 1: Interfaz Gráfica de la toolbox de identificación

En el desplegable *Import data* se selecciona *Time domain data* y aparece la ventana *Import Data* que se observa en la figura 2.

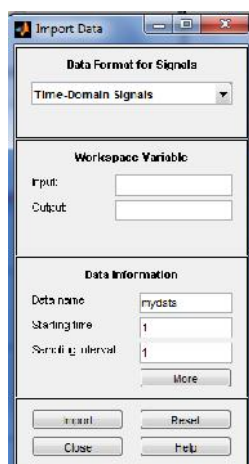


Figura 2: Interfaz gráfica para la importación de datos.

En el casillero *Input* introducimos la variable de entrada del sistema, que en nuestro caso es la variable u y en el casillero *Output* la variable de salida que para el caso es y , en la sección *Data Information* le damos un nombre a los datos en *Data name* e iniciamos el tiempo en cero en *Starting time* y por último en *Sampling interval* introducimos el tiempo de muestreo, para este ejemplo los datos se muestrearon cada medio segundo, la Figura 3 nos muestra los casilleros llenos.

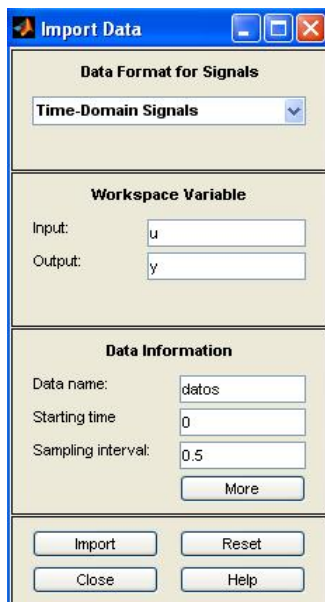


Figura 3: Interfaz gráfica para la importación de datos.

Una vez llenados los casilleros damos click en el botón *Import* y los datos se los puede ver en la *Data Views*, del GUI también están habilitados los casilleros *Time plot*, *Data spectra* y *Frequency function*, como ejemplo al hacer click en el casillero *Time Plot* se grafican las dos señales, como se ve en la Figura 4

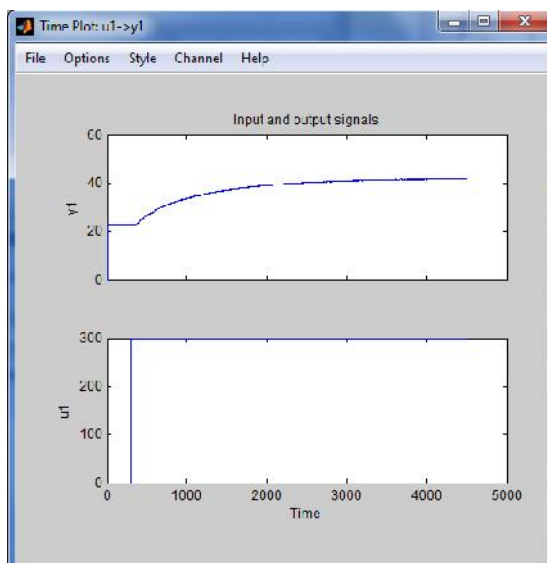


Figura 4: Respuesta del Sistema Identificado

En el desplegable *Estimate* que se encuentra en la zona del *Working Data* escogemos *Process models* y aparece la ventana de la Figura 5

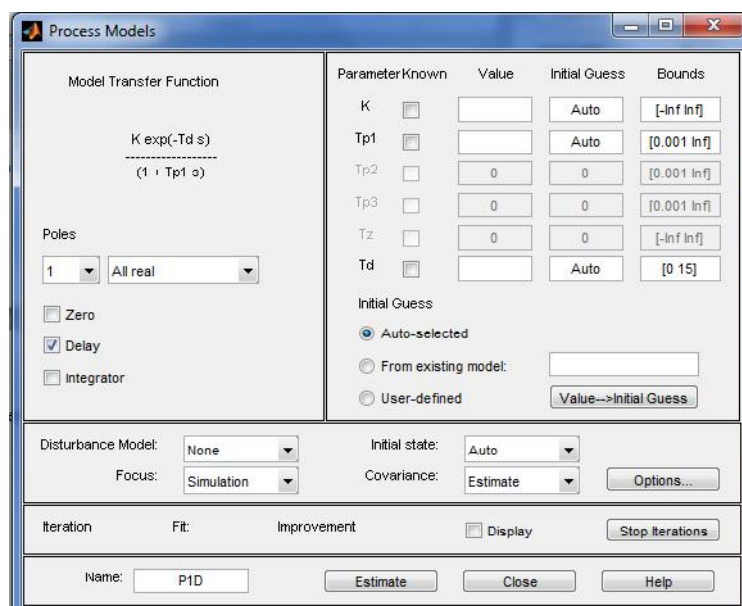


Figura 5. Ventana para determinar el modelo

En esta sección se define un modelo del sistema.

Polos (Poles)

Se selecciona el número de polos, y se elige si deben ser reales o complejos

Retardo (Delay)

Indicar si en el modelo se debe incluir un retardo

Ceros (Zero)

Indicar si en el modelo se debe incluir un cero

Nombre del modelo (Model Name)

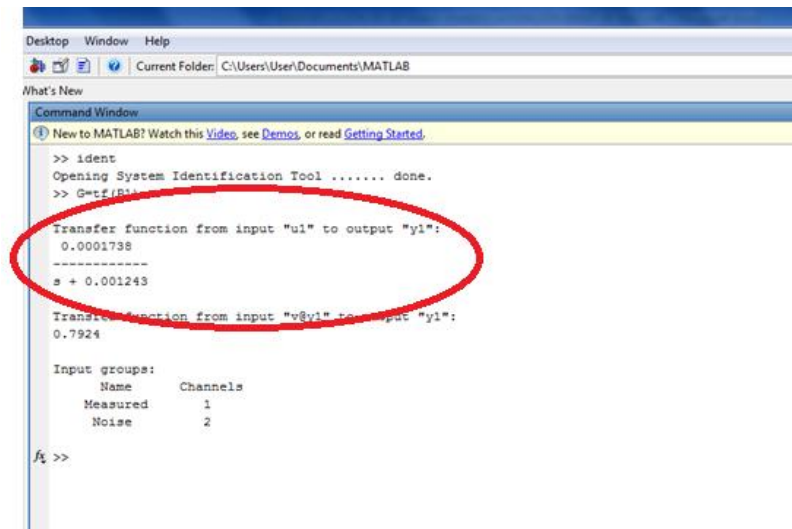
Por defecto se asigna un nombre al modelo, pero se lo puede cambiar escribiendo en la casilla correspondiente

Finalmente hacemos click en el botón *Estimate* y se obtiene la estimación del modelo que aparece en el tablero de modelos (*Model Views*).

Para una visualización del modelo hay que arrastrarlo al casillero *To Workspace* del GUI, para llamarlo desde el *command window* se escribe la siguiente instrucción

```
>> G=tf(P1)
```

P1 es el nombre por defecto y significa que la función de transferencia tiene un polo, el resultado se muestra en la Figura 6



```

Desktop Window Help
Current Folder: C:\Users\User\Documents\MATLAB

What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> ident
Opening System Identification Tool ..... done.
>> G=tf(81)

Transfer function from input "u1" to output "y1":
0.0001738
-----
s + 0.001243

Transfer function from input "v@y1" to output "y1":
0.7924

Input groups:
      Name      Channels
Measured        1
Noise           2

f1 >>
  
```

Figura 6: Obtención de la FdT.

Ejercicios

1. Comparar los resultados de identificación obtenidos en los apartados 1 y 2.
2. Cada grupo utiliza los datos de los ficheros G? y generar un vector de una entrada escalón para identificar el modelo del sistema utilizando la herramienta IDENT.
3. Elabore un informe con "SUS" resultados y conclusiones.

4. Identificación mediante la respuesta al escalón y arx

La respuesta al escalón permitirá aplicar técnicas de identificación que se basan en ajustar dicha respuesta a unos modelos preestablecidos (primer orden con o sin retardo, segundo orden con o sin retardo, orden superior, etc.). Las técnicas de identificación por minimización de errores requiere los datos numéricos obtenidos durante el ensayo realizado en *Simulink*; posteriormente en *Matlab* se obtendrán y validarán los modelos mediante los siguientes comandos:

```
>> g=tf([1 0.5],[1 -1.5 0.7],1)           % sistema original
>> [y t]=step(g)                         % generación vector salida y vector tiempo
>> u=ones(34,1)                          % generación vector entrada (escalón entrada)
>> datos=iddata(y,u,1)                   %generación objeto datos Entrada/salida
>> modelo=arx(datos,[2 2 0])             %generación modelo ARX con retraso 0
>> garx0=tf([0.75 0.75 0],[1 -1.5 0.7],1) %considerando u(t-nk) en el modelo
>> step(g,garx0)
```

Ejercicio

1. Cada grupo propone otra función de transferencia e identifica el modelo utilizando ARX.
2. Elabore un informe con "SUS" resultados y Conclusiones.