Identificación experimental de sistemas dinámicos por métodos gráficos usando Matlab.

Introducción

Los métodos gráficos para la identificación de sistemas tienen la ventaja de que, dada su sencillez, para su aplicación sólo se necesita "lápiz y papel". Hoy en día, sin embargo, son habituales los ordenadores con posibilidades gráficas potentes, y además la existencia de programas especializados para el manejo de datos. En este contexto es importante conocer como aprovechar las potencialidades de estas herramientas computacionales para la aplicación de estos métodos de identificación.
Esta guía tiene el objetivo de brindar los procedimientos básicos a seguir para la aplicación de los métodos gráficos de identificación experimental utilizando el programa Matlab. Para ello se han desarrollado los siguientes ejemplos:

1. Obtención de un modelo de primer orden.
2. Identificación de un modelo de segundo orden con polos iguales mediante el método de Strecj.
3. Identificación de un modelo de segundo orden con polos diferentes mediante el método de gráficas logarítmicas.

Ejemplo 1

Se quiere obtener el modelo de un sistema cuya respuesta ante un escalón unitario está en el archivo *ensayo.dat*.

Cargar los datos al Workspace de Matlab:

```
>> load ensayo.dat
```

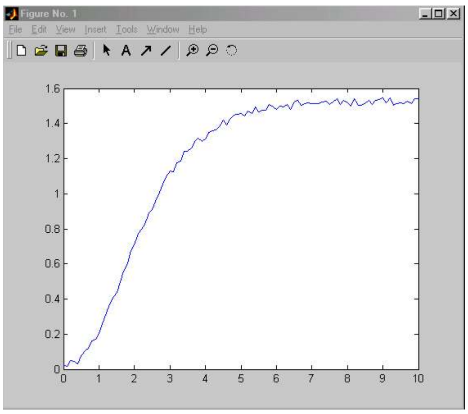
mediante la instrucción "who" podemos comprobar que los datos han sido cargados desde el archivo:

```
>> who
Your variables are:
ensayo
```

La variable *ensayo* es una matriz de 3 columnas: la primera columna contiene los valores de tiempo *t* y la segunda los valores de la entrada *u(t)*y la tercera columna los valores de la salida del sistema *y(t)*.
Obtener una gráfica de los datos experimentales:

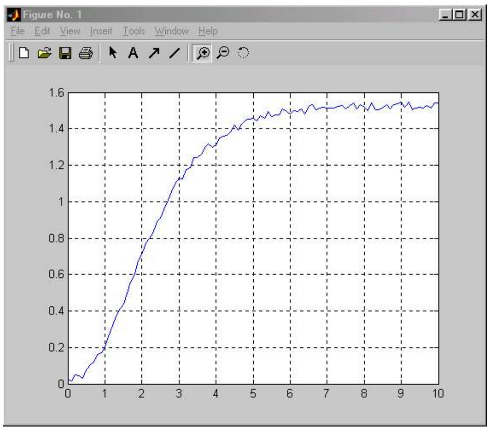
```
>> plot(ensayo(:,1),ensayo(:,3))
```

Con esta instrucción Matlab genera una figura (Figure No. 1) que contiene la gráfica de la respuesta del sistema: *y(t) vs t*



Para facilitar la búsqueda de valores en la gráfica añadir divisiones mediante la instrucción:

```
>> grid
```



Sobre esta gráfica se aplican los métodos de identificación gráficos.

Identificación de un modelo de 1^er orden

Supongamos que queremos aproximar el comportamiento por un sistema de 1 orden.

$$G(s)=\frac{k}{\tau s+1}$$

Hemos de obtener los parámetros: ganancia estática y constante de tiempo.

Ganancia estática *k*:

$$k=\frac{\Delta y(\infty)}{\Delta u}$$

de la gráfica $\Delta y(\infty)=1.5$ y conociendo que la entrada ha sido un escalón unitario $\Delta u=1$. Por tanto $k=1.5$.

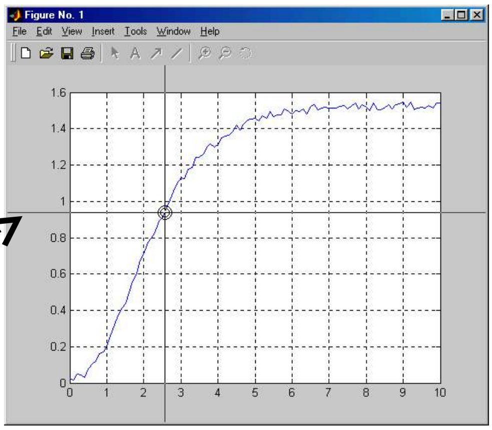
Constante de tiempo τ

Calcular el 63% del valor final de estado estable de la salida $0.63\,\Delta y(\infty)$:

```
>> y_tau=0.63*1.5
y_tau =
0.9450
```

Para este valor se obtiene de la gráfica por inspección visual el valor de τ Para obtener un valor más preciso se puede utilizar la instrucción *ginput*, con la cual es posible obtener las coordenadas (x,y) de un punto seleccionado con el ratón en un gráfico. Una vez introducida la instrucción en la línea de comando, se selecciona en el gráfico el punto del cual deseamos conocer las coordenadas:

Hacer coincidir
aproximadamente
con el valor de y_tau



hacemos clic con el ratón y aparecen las coordenadas del punto seleccionado

```
>>tau=ginput(1)
tau =
2.5461    0.9287
```

de donde tenemos que $\tau \approx 2.55$.

Verificación del modelo

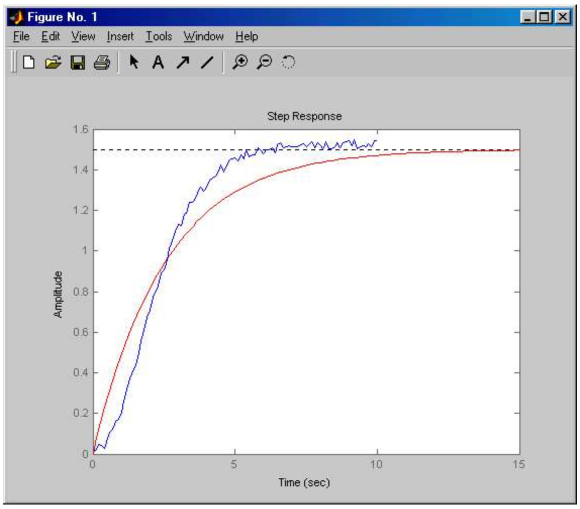
Una vez obtenidos los dos parámetros del modelo podemos comprobar la similitud de la respuesta del modelo con los datos experimentales.
Primero crearemos el modelo con los parámetros calculados con la instrucción tf, que permite definir una función de transferencia a partir de dos vectores que contienen los coeficientes de los polinomios del numerado y denominador en potencias decrecientes de *s*:

```
>> G0=tf(1.5,[2.55 1])
Transfer function:
  1.5
----------
2.55 s + 1
```

Luego generamos el gráfico de la respuesta escalón del sistema:

```
>> hold on,step(G0,'r')
```

Aquí tenemos dos instrucciones: *hold on*: mantiene en la figura la gráfica de la respuesta del sistema[1]. step(G0,'r'): traza en color rojo (argumento 'r') la gráfica de la respuesta al paso de la función de transferencia en G0.



De la figura podemos comprobar por simple inspección visual que existe una diferencia considerable entre las respuestas del modelo en *G0* (en rojo) y del sistema real (en azul). Procederemos entonces a tratar de obtener un nuevo modelo que ajuste de forma más aproximada en comportamiento del sistema real.

Identificación de un modelo de 2º orden. Método de Strecj.

Con el método de Strejc la respuesta del sistema se trata de aproximar por un sistema con polos reales múltiple:

$$G(s) = \frac{k}{(1+\tau s)^n}$$

Los valores de τ y n se busacan de la siguiente tabla, a partir de los valores de *Tu* y *Ta* que se obtiene de gráfica de respuesta del sistema.

n	T_a/τ	T_u/τ	T_u/T_a
1	1	0	0
2	2.7	0.28	0.104

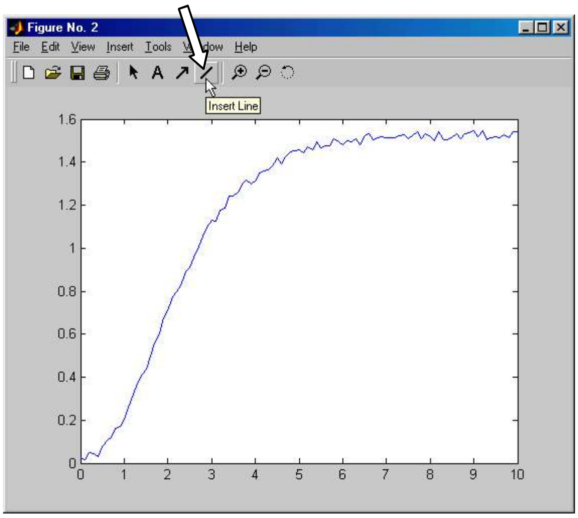
[1] Si no ponemos la instrucción *hold on* la gráfica del la respuesta experimental se borra de la figura al ejecutar la instrucción *step*.

3	3.7	0.80	0.22
4	4.46	1.42	0.32
5	5.12	2.1	0.41
6	5.7	2.8	0.49
7	6.2	3.55	0.57
8	6.7	4.3	0.64
9	7.6	5.08	0.71

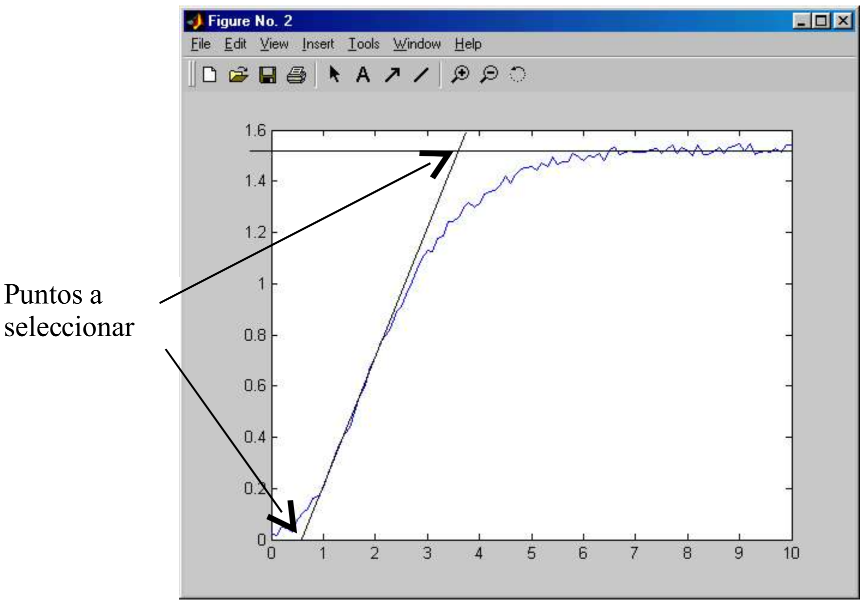
Para obtener los valores de *Tu* y *Ta* de la gráfica nos auxiliaremos de la herramienta para el trazado de líneas que aparece en la figura. Antes crearemos una nueva figura sobre la que trabajaremos:

```
>> figure,plot(ensayo(:,1),ensayo(:,3))
```

crea una nueva figura (Figure No. 2) con los datos experimentales. Minimizamos (no cerrar) la figura (Figure No. 1).
En Figure No. 2 se selecciona la herramienta para el trazado de líneas:



y se trazan las líneas necesarias para determinar *Tu* y *Ta*: la tangente al punto de máxima pendiente de la curva y la que coincide con del valor final de estado estable.
Para determinar los valores de *Tu* y *Ta* usaremos la instrucción *ginput*, seleccionando en la figura los puntos que a continuación se indican:




```
>> [Tu_Ta]=ginput(2)
Tu_Ta =
```

```
0.5876   -0.0023
3.6060    1.5181
```

de donde $Tu \approx 0.6$ y $Ta \approx 3.6 - Tu \approx 3$.
Calculando $Tu/Ta \approx 0.2$. De la tabla de Strejc se obtiene $n=3, Ta/\tau_1 = 3.7, Tu/\tau_2 = 0.8$, de donde $\tau_1 = 0.97, \tau_2 = 0.625$. Tomamos el valor como valor de τ el promedio de los dos valores anteriores: $\tau \approx 0.8$. De forma que el modelo sería:

$$G1(s) = \frac{1.5}{(0.8s+1)^3}$$

Verificación del modelo

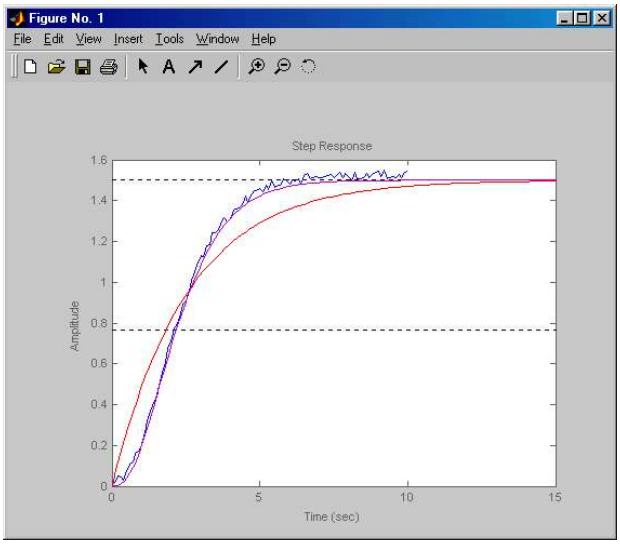
Para comprobar la exactitud del modelo, primero definamos la función de transferencia G1(s) en Matlab:

```
>> G1=tf(1.5,conv([0.8 1],conv([0.8 1],[0.8 1])))
Transfer function:
              1.5
-------------------------------------
0.512 s^3 + 1.92 s^2 + 2.4 s + 1
```

La instrucción *conv* realiza la multiplicación de dos polinomios[2].
Ahora podemos trazar la gráfica de la respuesta de G1 en la Figure No. 1. Primero hemos de seleccionar dicha figura haciendo clic en ella. Luego ejecutar la línea:

```
>> hold on,step(G1,'m')
```

aparece entonces en color magenta (argumento 'm') la respuesta del modelo G1(s) en la Figure No. 1.



La aproximación que conseguimos con este modelo a la respuesta real del sistema es mucho mejor que la conseguida con el modelo G0 de primer orden.

Ejemplo 2:
Se quiere obtener el modelo de un sistema cuya respuesta ante un escalón unitario está en el archivo *ensayo1.dat*.

Cargar los datos al Workspace de Matlab:

```
>> load ensayo1.dat
```

mediante la instrucción "who" podemos comprobar que los datos han sido cargados desde el archivo:

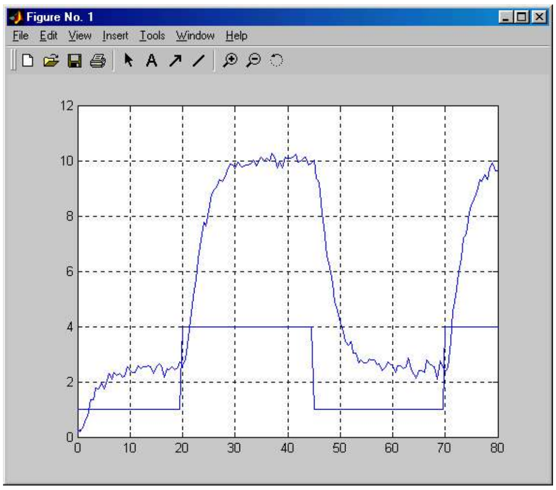
[2] Para más detalle >>help conv en Matlab.

```
>> who
Your variables are:
ensayo1
```

La variable *ensayo* es una matriz de 3 columnas: la primera columna contiene los valores de tiempo *t* y la segunda los valores de la entrada *u(t)*y la tercera columna los valores de la salida del sistema *y(t)*.

Obtener una gráfica de los datos experimentales:

```
>> plot(ensayo1(:,1),ensayo1(:,3))
>> plot(ensayo1(:,1),ensayo1(:,2)), grid
```



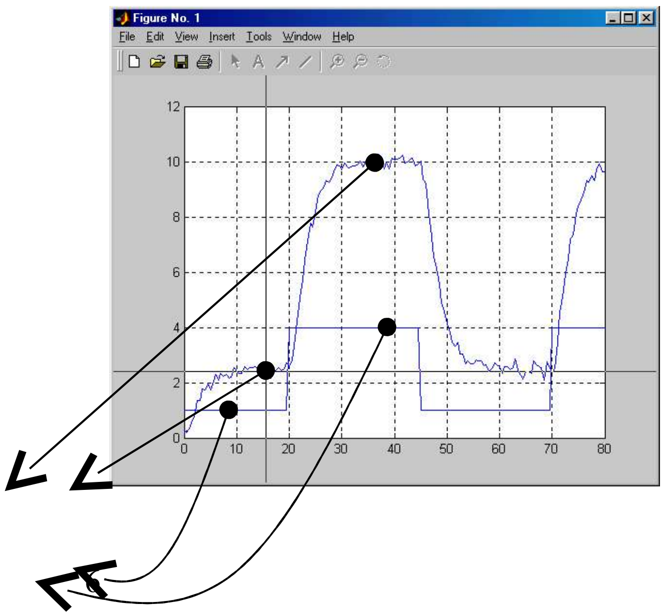
Identificación de un modelo de segundo orden con polos diferentes mediante el método de gráficas logarítmicas.

Con este ejemplo ilustraremos la aplicación de método de las gráficas logarítmicas, que se aplica a sistema con polos reales diferentes.

Consideremos un modelo con dos polos. La estructura del modelo a identificar es la siguiente:

$$G(s)=\frac{k}{(1+\tau_1 s)(1+\tau_2 s)}=\frac{k}{(1+\frac{s}{\alpha})(1+\frac{s}{\beta})}$$

Cálculo de ganancia estática:



$$k=\frac{\Delta y(\infty)}{\Delta u}=\frac{10-2.4}{4-1}$$
$$k\approx 2.5$$

Identificación de los polos del sistema

Pre-procesamiento de los datos

Como de observa en la figura, durante el experimento se han aplicado varias entradas escalón al sistema, sin embargo, para realizar la identificación sólo necesitamos la respuesta ante una única entrada. Lo primero que haremos será entonces obtener a partir de los datos de todo el experimento aquella porción que corresponda a un cambio tipo escalón en la entrada. Para ello:

Pulsar el botón en la barra de herramientas. Se abre una ventana que contiene todas las variables.
Hacer doble clic en la variable cuyo contenido deseamos visualizar, en este caso *ensayo1*.
Se abre una ventana que muestra el contenido de la variable *ensayo1* con la numeración de filas y columnas.
Buscar la fila en que la columna 2 (que contiene la entrada) cambia de un valor a otro. Anotar el número de fila: 21, como se muestra a continuación.

	1	2	3
42	17	1	2.4566
43	17.5	1	2.4505
44	18	1	2.5419
45	18.5	1	2.4252
46	19	1	2.4976
47	19.5	1	2.7255
48	20	4	2.6185
49	20.5	4	2.8353
50	21	4	3.3289
51	21.5	4	4.2448
52	22	4	5.0804
53	22.5	4	5.7327
54	23	4	6.436

Anotar este valor

Cambio en la entrada (inicio del escalón).

Buscar las filas en la que los valores de elementos de la columna 3 (que contiene la salida del sistema) permanecen prácticamente constante. Esto es indicativo de que el sistema ha alcanzado el estado estable.

	1	2	3
2	37	4	10.27
8	37.5	4	10.119
84	38	4	9.7341
85	38.5	4	9.9973
86	39	4	9.7381
87	39.5	4	10.14
88	40	4	10.052
89	40.5	4	10.087
90	41	4	10.121
91	41.5	4	10.246
92	42	4	9.9529
93	42.5	4	9.97
94	43	4	[illegible]

Anotar este valor

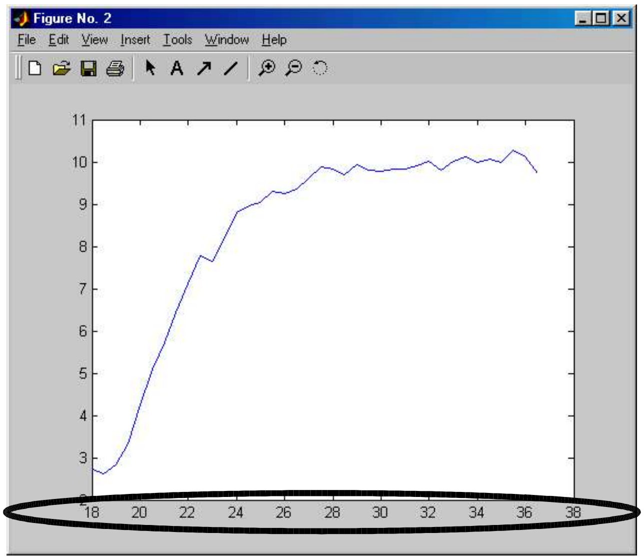
Valores de salida prácticamente constantes.

Anotar el número de una de las primeras filas donde esto ocurra. De esta forma garantizamos que sólo nos quedaremos con los datos correspondientes a la respuesta dinámica con la instrucción que a continuación.
Ahora almacenaremos en una variable los datos que están entre las filas antes anotadas: la 21 y la 84, para ello:

```
>> dato_util=ensayo1(47:84,:);
```

Haciendo un gráfico de los datos en dato_util:

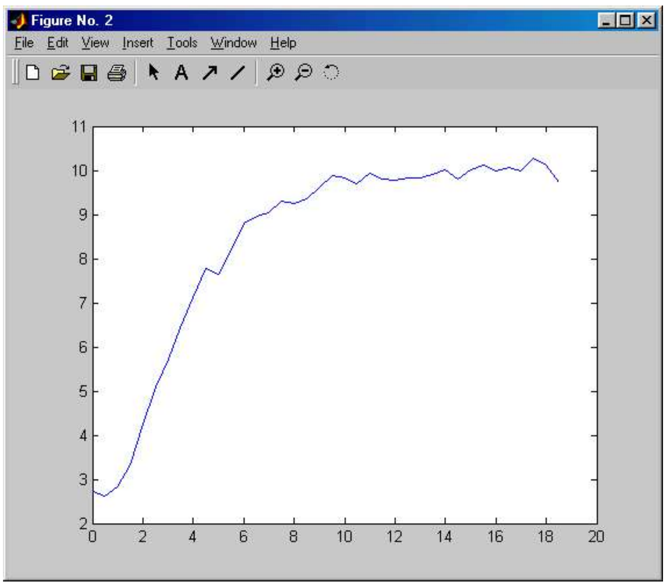
```
>> figure, plot(dato_util(:,1),dato_util(:,3))
```



podemos notar que el tiempo comienza en 18, pues hemos tomado una porción intermedia de los datos del experimento. Para hacer coincidir el tiempo inicial de esta porción de datos con t=0:

```
>> dato_util(:,1)=dato_util(:,1)-dato_util(1,1);
```

Obteniendo nuevamente el gráfico, comprobamos que el tiempo de nuestros datos comienza en t=0.



Cálculo de los polos

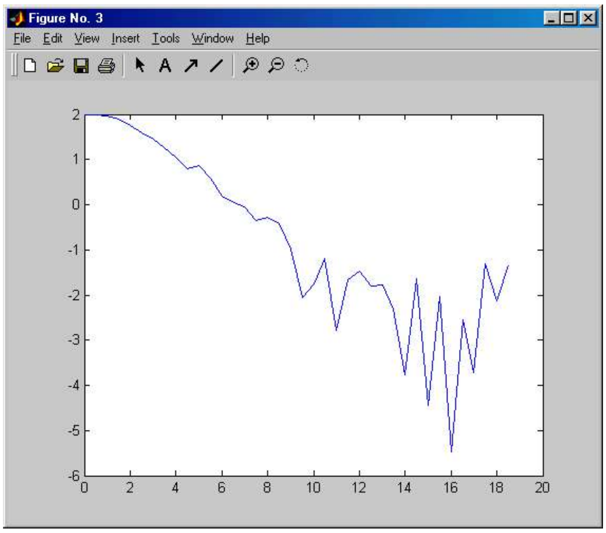
Ahora tenemos los datos listos para comenzar la identificación.

Calcular $\ln(\Delta y(\infty) - \Delta y_m(t))$:

```
>> ln_dyinf_dym=log(10-dato_util(:,3));
```

Obtener la gráfica de $\ln(\Delta y(\infty) - \Delta y_m(t))$ vs t :

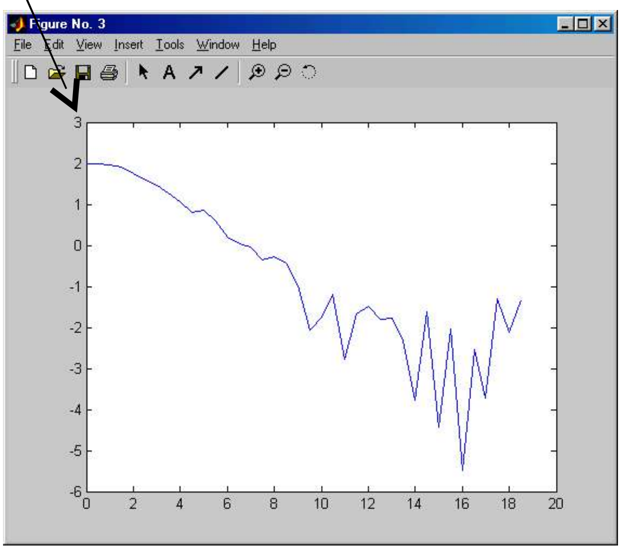
```
>> plot(dato_util(:,1),ln_dyinf_dym)
```



En la gráfica se aprecia que para *t>8* aparecen uno picos. Estos se deben que la respuesta real del sistema presenta un ruido de medida, cuyo efecto en la gráfica anterior se incrementa a mediada que la respuesta del sistema va alcanzando es estado estable. Como nuestro interés es el cálculo de los polos, con efecto sobre la respuesta transitoria, consideraremos los datos hasta $t \approx 8$.

Vemos además que el valor máximo del eje de las abscisas es 2. Para leer el valor de $\ln(-A)$ es necesario aumentar este valor, para ellos:

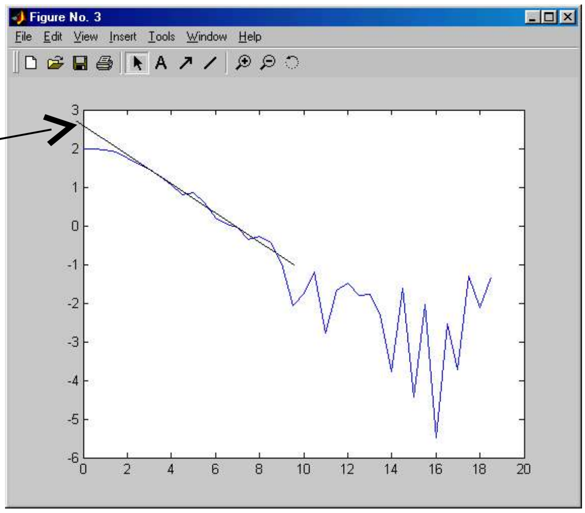
```
>> axis([0 20 -6 3])
```



Ahora estamos en condiciones de obtener el valor de $\ln(-A)$. Usando la herramienta para trazar líneas, trazamos la línea tangente a la gráfica anterior en valores de t altos y sin tener en cuenta los picos.

Para leer el valor de $\ln(-A)$ con precisión usemos la instrucción *ginput* y seleccionamos el punto:

Punto a seleccionar



obteniendo:

```
>> pA=ginput(1)
pA =
   0.0691    2.6184
```

de aquí sabemos que: $\ln(-A)=2.6$. Para obtener el valor de A:

```
>> A=-exp(pA(2))
A =
  -13.7141
```

Para calcular el valor de α (pendiente de la recta) es necesario conocer las coordenadas de otro punto cualquiera sobre la recta. Para ello, usar la instrucción *ginput*. En nuestro caso hemos seleccionadoel siguiente:

```
>> pB=ginput(1)
pB =
   7.3963   -0.2763
```

Ahora podemos calcular el valor de α :

```
>> alpha=abs((pA(2)-pB(2))/(pA(1)-pB(1)))
alpha =
   0.3951
```

De manera que:

$$\alpha \approx 0.4 .$$

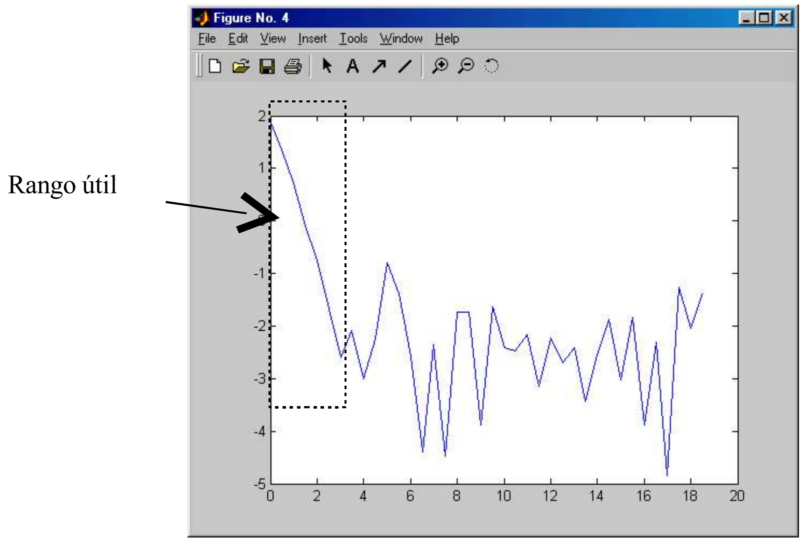
Para calcular el otro polo hay que obtener la gráfica: $\ln(-\Delta y(\infty)+\Delta y_m(t)-Ae^{-\alpha t})$ vs t

Calcular $\ln(-\Delta y(\infty)+\Delta y_m(t)-Ae^{-\alpha t})$:

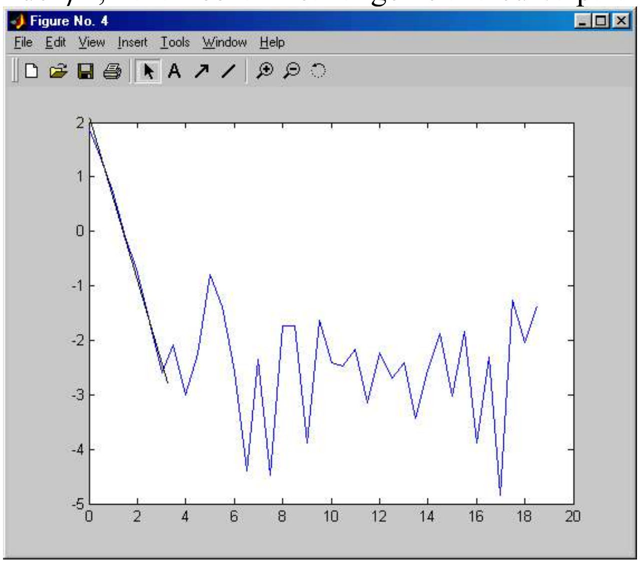
```
>> ln_dyinf_dym_Aexp=log(-10+dato_util(:,3)-A*exp(-alpha*dato_util(:,1)));
```

Obtener la gráfica:

```
>> figure,plot(dato_util(:,1),ln_dyinf_dym_Aexp)
```



Aquí, como en la gráfica $\ln(\Delta y(\infty)-\Delta y_m(t))$ vs t, aparece los picos debido al ruido de medida. En esta ocasión, y por los mismos motivos explicados anteriormente tampoco se tiene en cuenta.
Para obtener el valor de β, trazamos la línea tangente a la curva para valores alto de t:



Usando el mismo procedimiento que para el cálculo de α, calculamos β:

```
>> pA=ginput(1)
pA =
   0.0691    1.9488
>> pB=ginput(1)
pB =
   2.9724   -2.4518
>> beta=abs((pA(2)-pB(2))/(pA(1)-pB(1)))
beta =
   1.5158
```

Conocidos k, α y β sólo queda definir el modelo en Matlab y comprobar la exactitud del mismo:
Para definir el modelo:

```
>> G1=tf(2.5,conv([1/alpha 1],[1/beta 1]))

Transfer function:
```

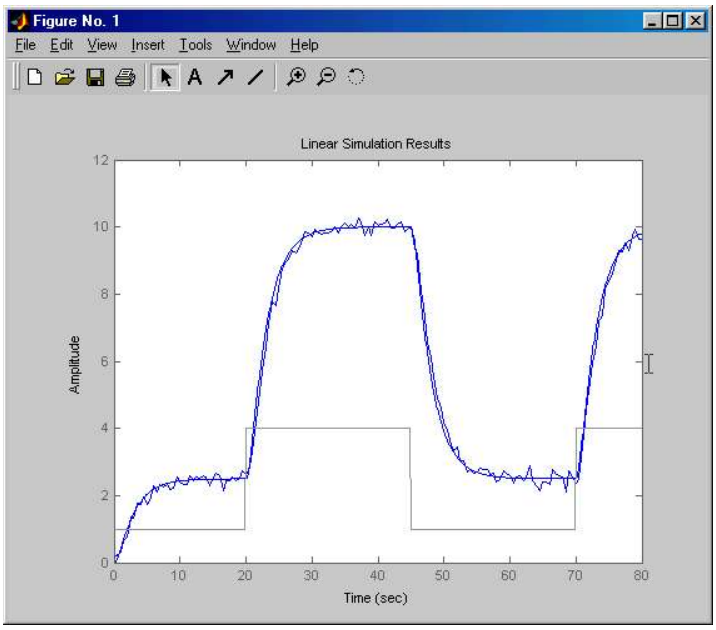
```
            2.5
----------------------
1.67 s^2 + 3.191 s + 1
```

Verificación del modelo

Para comparar la respuesta de modelos con la del proceso real:
Abrir la figura en que están representados los datos experimentales.
Obtener la gráfica de la respuesta del model G1:

```
>> hold on,lsim(G1,ensayo1(:,2),ensayo1(:,1))
```

en la misma figura aparece la respuesta del modelo para la misma entrada aplicada al experimento real (almacenada en ensayo1(:,2),).



Como se puede comprobar el modelo ajusta de forma bastante exacta al comportamiento real del sistema.