

TEORÍA DE SISTEMAS

PRÁCTICA 5: ESTABILIDAD Y COMPORTAMIENTO DINÁMICO DE SISTEMAS

1. ESTUDIO DE LA ESTABILIDAD DE SISTEMAS

Teóricamente se dispone de tres medios para determinar la estabilidad de un sistema:

- A partir de su respuesta impulsional.
- A partir de la situación de los polos de la función de transferencia.
- A partir de los coeficientes del denominador de la función de transferencia (tabla de Routh).

Con Matlab podemos proceder de dos formas:

- Calcular los polos de la función de transferencia.
- Simular el comportamiento del sistema y ver si es estable ante entrada escalón (en principio también será estable ante cualquier otra entrada acotada).

Veamos esto sobre un ejemplo:

Se pretende averiguar el rango de valores del parámetro K que hacen estable el siguiente sistema:

$$G(s) = \frac{3s + 5}{s^3 + 3s^2 + 2s + K}$$

La forma de operar manualmente sería aplicar el criterio de Routh:

- En primer lugar, todos los coeficientes del denominador deben tener igual signo, con lo cual se obtiene la primera condición que debe cumplir K:

$$K > 0$$

- En segundo lugar, se debe calcular la tabla de Routh y exigir como condición que no se produzca ningún cambio de signo en la primera columna:

s^3	1	2
s^2	3	K
s^1	$\frac{6-K}{3}$	
s^0	K	

Las condiciones que se obtienen para el parámetro K son:

$$\begin{aligned} 6 - K > 0 &\Rightarrow K < 6 \\ K > 0 \end{aligned}$$

Combinando todos los resultados, se obtiene como conclusión que el sistema es estable si cumple:

$$0 < K < 6$$

A continuación se comprobarán estos resultados con Matlab. Comenzaremos por elegir un valor de K que haga al sistema estable, por ejemplo K=3.

Para este valor de K crearemos una función de transferencia como la pedida mediante la instrucción **tf** que ya conocemos:

```

» k = 3;
» sis = tf([3 5], [1 3 2 k])
Transfer function:
 3 s + 5
-----
s^3 + 3 s^2 + 2 s + 3
    
```

Lo primero que haremos será calcular la situación de los polos de la función de transferencia. Esto puede hacerse con la instrucción **pole**, tal y como se indica a continuación:

```

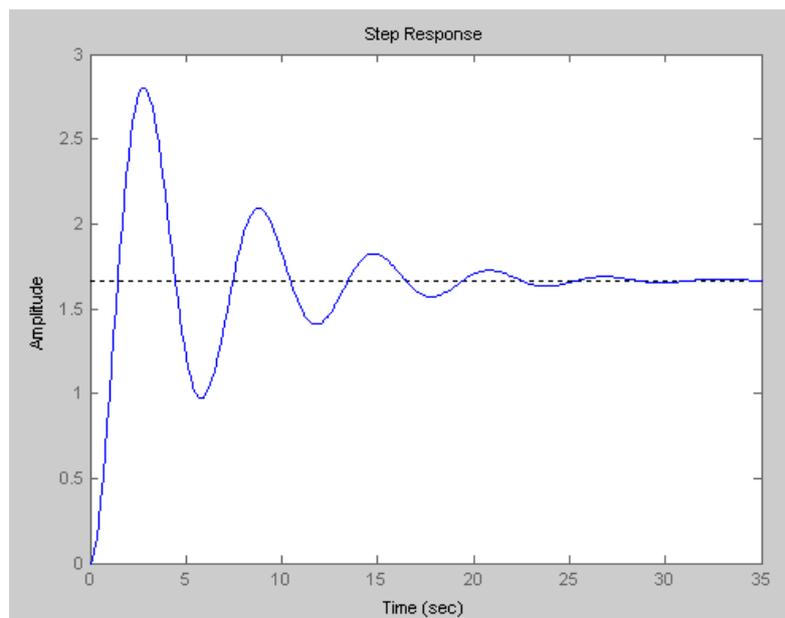
» pole(sis)
ans =
-2.6717
-0.1642 + 1.0469i
-0.1642 - 1.0469i
    
```

La instrucción nos devuelve los valores de los polos; podemos comprobar como todos ellos tienen parte real negativa, lo que corresponde a un sistema estable.

Una segunda comprobación de la estabilidad del sistema la podemos hacer calculando la respuesta que ofrecería el sistema ante una entrada tipo escalón, mediante la instrucción **step**, también conocida:

```

» step(sis)
    
```



Podemos ver sobre el gráfico devuelto por Matlab como la respuesta ante una entrada escalón es una señal que se estabiliza en un valor determinado, es acotada, por lo que podemos considerar que para cualquier otro tipo de entrada también lo será. Consideraremos comprobado que el sistema es estable.

A continuación realizaremos las mismas pruebas pero para valores del parámetro K fuera del rango de estabilidad. En primer lugar probaremos con un valor de K por encima del límite superior de estabilidad, por ejemplo K=9.

```

» k = 9;
» sis = tf([3 5], [1 3 2 k])
Transfer function:
 3 s + 5
-----
s^3 + 3 s^2 + 2 s + 9
    
```

Calculamos los polos de la función de transferencia:

```

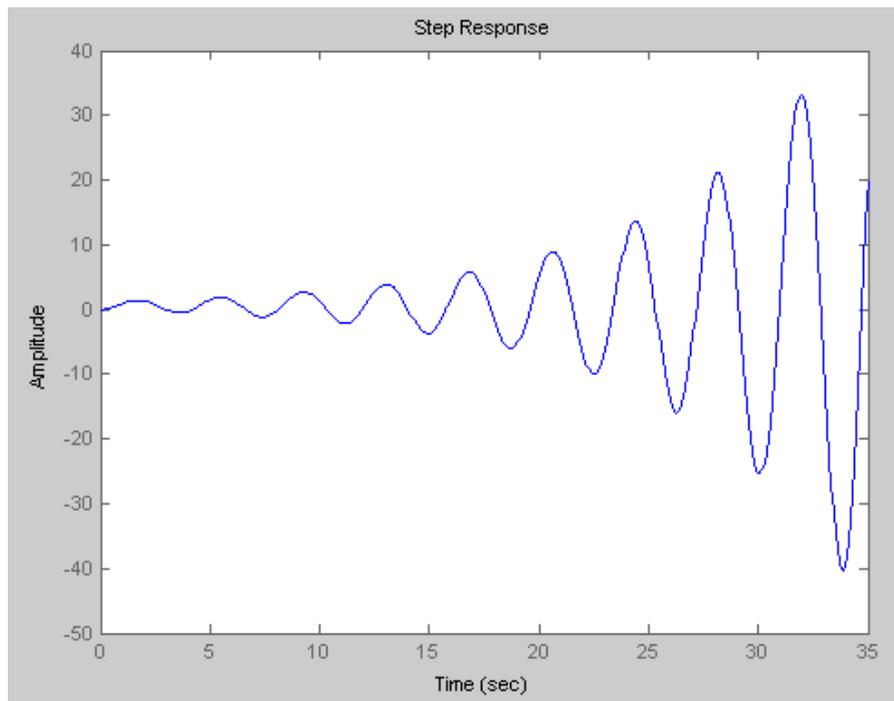
» pole(sis)
ans =
-3.2400
0.1200 + 1.6623i
0.1200 - 1.6623i
    
```

Y comprobamos como, efectivamente, dos de ellos tienen parte real positiva, lo que corresponde a un sistema inestable.

Como segunda comprobación, también obtenemos la respuesta ante entrada escalón:

```

» step(sis, 35)
    
```



Podemos ver como en este caso la señal de respuesta crece indefinidamente en amplitud, no está acotada, y por tanto debemos considerar el sistema como inestable.

Consideraremos por último el caso en el que el parámetro K se encuentra también fuera del rango de estabilidad pero en este caso por debajo del límite inferior, por ejemplo $K = -2$

```

» k=-2;
» sis = tf([3 5], [1 3 2 k])
Transfer function:
3 s + 5
-----
s^3 + 3 s^2 + 2 s - 2
    
```

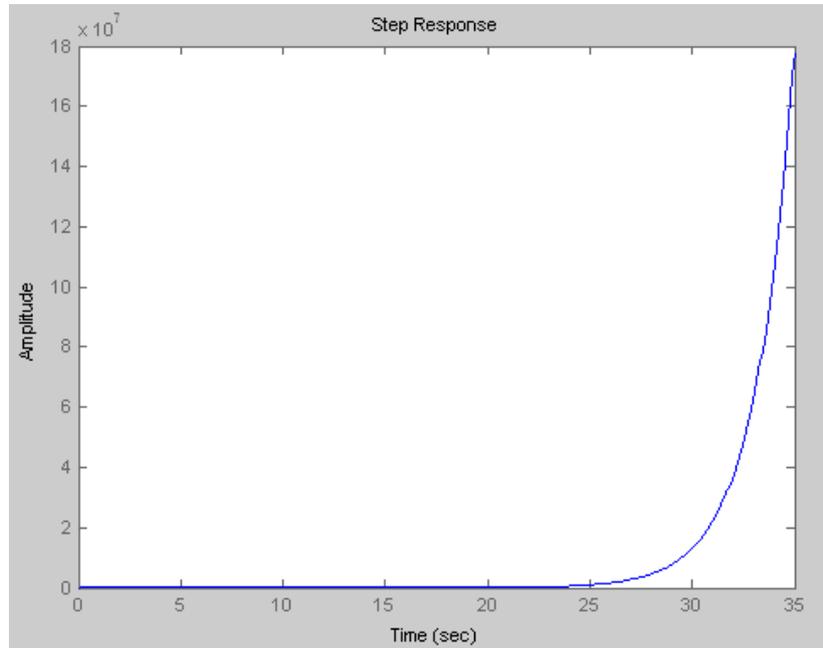
Si calculamos los polos de la función de transferencia:

```

» pole(sis)
ans =
-1.7607 + 0.8579i
-1.7607 - 0.8579i
0.5214
    
```

Podemos ver de nuevo como uno de ellos tiene parte real positiva, por tanto corresponde a un sistema inestable.

Haciendo la segunda comprobación, calcularemos la respuesta ante una entrada en escalón:
>> step(sis)



De nuevo podemos comprobar como la señal de respuesta crece indefinidamente, no está acotada, lo que corrobora que el sistema es inestable.

EJERCICIO 1: CÁLCULO DE ESTABILIDAD

Se estudiará el rango de valores de K que hacen estable el siguiente sistema:

$$G(s) = \frac{5}{Ks^3 + s^2 + 2s + K + 5}$$

Al igual que en el ejemplo, se calculará teóricamente el rango de estabilidad y a continuación se comprobarán con Matlab los resultados obtenidos.

Resultados a obtener:

- Desarrollo teórico completo (tabla de Routh) que permite determinar el rango de estabilidad del sistema planteado.
- Para un valor de K dentro del rango de estabilidad:
 - Instrucciones Matlab que permiten crear la función de transferencia para ese valor de K.
 - Valor de los polos tal y como es devuelto por Matlab.
 - Respuesta a escalón en la que se vea claramente que el sistema es estable. Para ello se deberá añadir un parámetro a la instrucción **step** de modo que se simule el comportamiento del sistema durante los 50 primeros segundos.
- Para un valor de K por encima del límite superior de estabilidad:
 - Repetir los mismos apartados del punto anterior. En este caso debe verse claramente como el sistema es inestable.
- Para un valor de K por debajo del límite inferior de estabilidad:
 - Repetir los mismos apartados del punto anterior.

Todos los gráficos deberán incluir el nombre del alumno como título.

2. ESTUDIO DEL COMPORTAMIENTO DINÁMICO DE SISTEMAS

Sabemos por teoría que el comportamiento dinámico de sistemas **en régimen transitorio** depende de la situación de los polos de la función de transferencia sobre el plano complejo. A continuación se comprobarán estos resultados teóricos mediante Matlab.

2.1 Comportamiento dinámico de sistemas de primer orden

Un sistema de primer orden responde a la siguiente función de transferencia:

$$G(s) = \frac{K}{1 + Ts}$$

Y el único polo del sistema (valor de s que anula el denominador) se encuentra en $s = -1/T$

De acuerdo con lo explicado en teoría, el sistema presentará un comportamiento en régimen transitorio más rápido cuanto más alejado se encuentre el polo del eje imaginario (cuanto menor sea la constante de tiempo T)

Comprobaremos estos resultados con 3 sistemas en los que varía únicamente la constante de tiempo T :

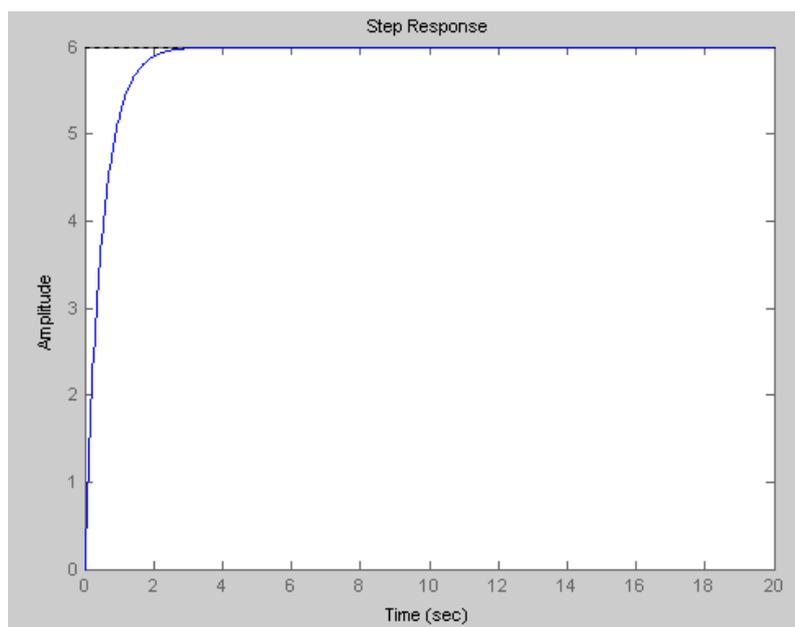
$$G_1(s) = \frac{6}{1 + 0.5s} \quad G_2(s) = \frac{6}{1 + 2s} \quad G_3(s) = \frac{6}{1 + 5s}$$

Para ello obtendremos en cada caso la respuesta a escalón mediante el comando **step** de Matlab. Para poder comparar los resultados correctamente, en todos los casos simularemos el comportamiento del sistema durante un tiempo de 20 segundos:

Para el primer sistema, tendremos:

```

» sis1 = tf(6, [.5 1])
Transfer function:
6
-----
0.5 s + 1
» step(sis1, 20)
    
```



Para el segundo sistema:

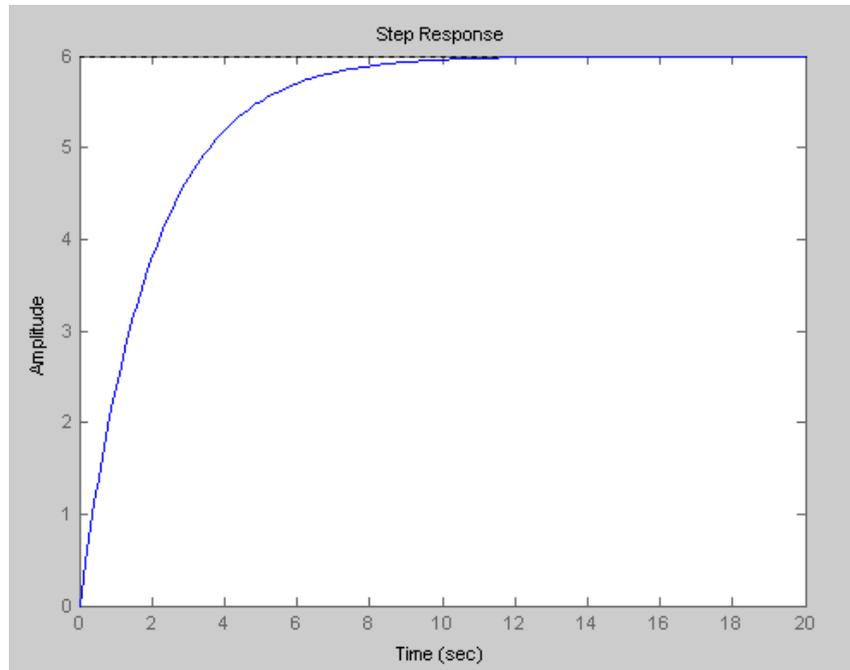
```
» sis2 = tf(6, [2 1])
```

Transfer function:

6

2 s + 1

```
» step(sis2, 20)
```



Y por último, para el tercer sistema:

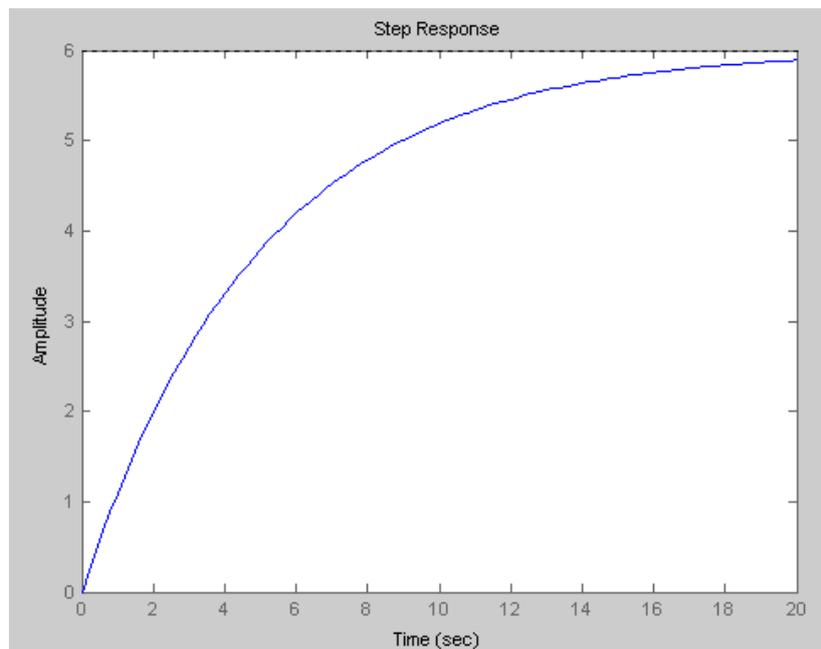
```
» sis3 = tf(6, [5 1])
```

Transfer function:

6

5 s + 1

```
» step(sis3, 20)
```

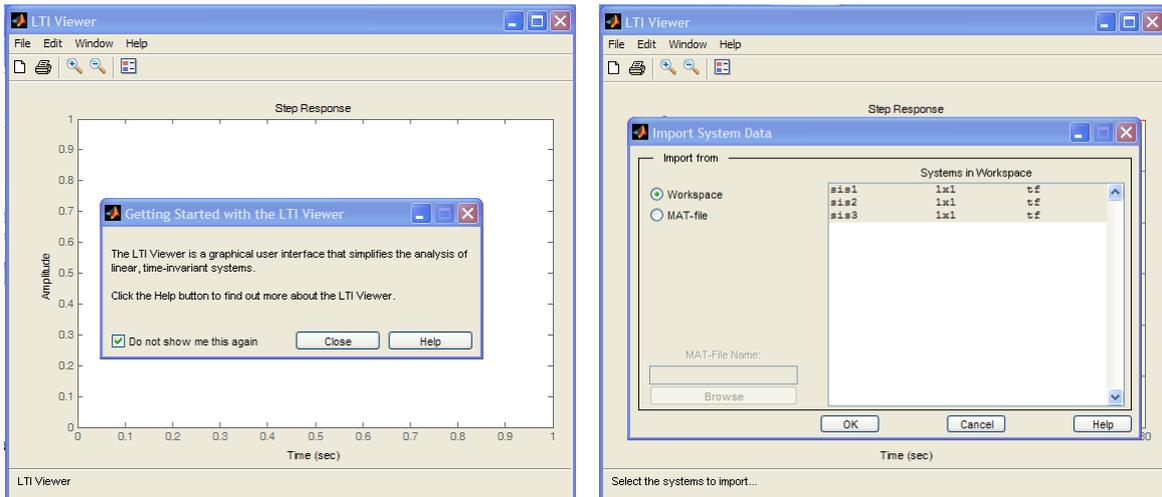


Debe quedar comprobado como al aumentar el valor de T (polos más cercanos al eje imaginario) la respuesta del sistema se hace más lenta.

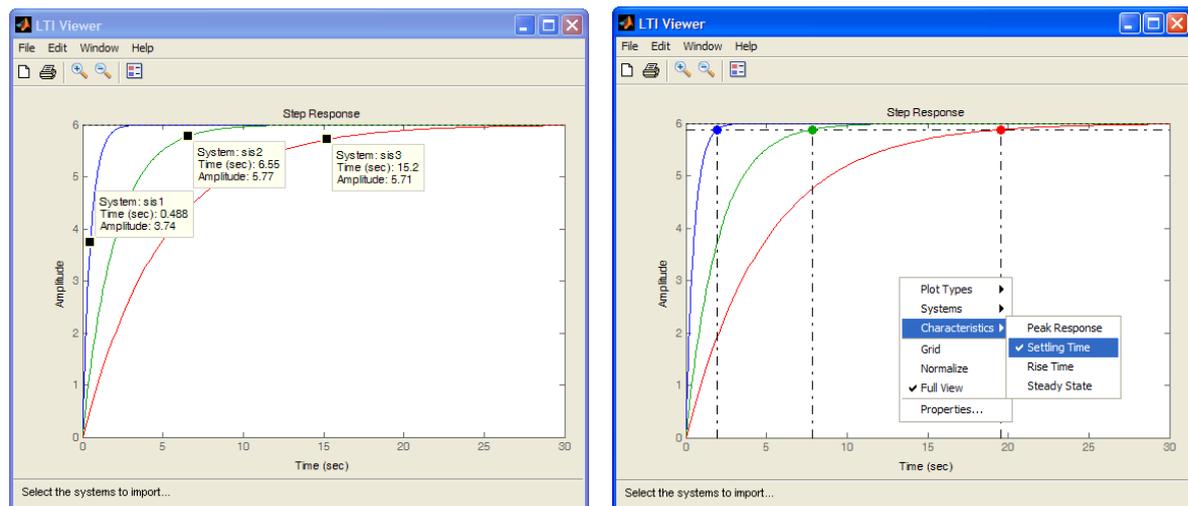
2.2 Comportamiento dinámico de sistemas de primer orden con la herramienta LTIVIEW

A continuación vamos a estudiar el manejo de la herramienta **LTIView** de Matlab para el estudio del comportamiento temporal y frecuencial de sistemas dinámicos. Para activar esta herramienta ejecutaremos el siguiente comando en la consola:

» `ltiview`



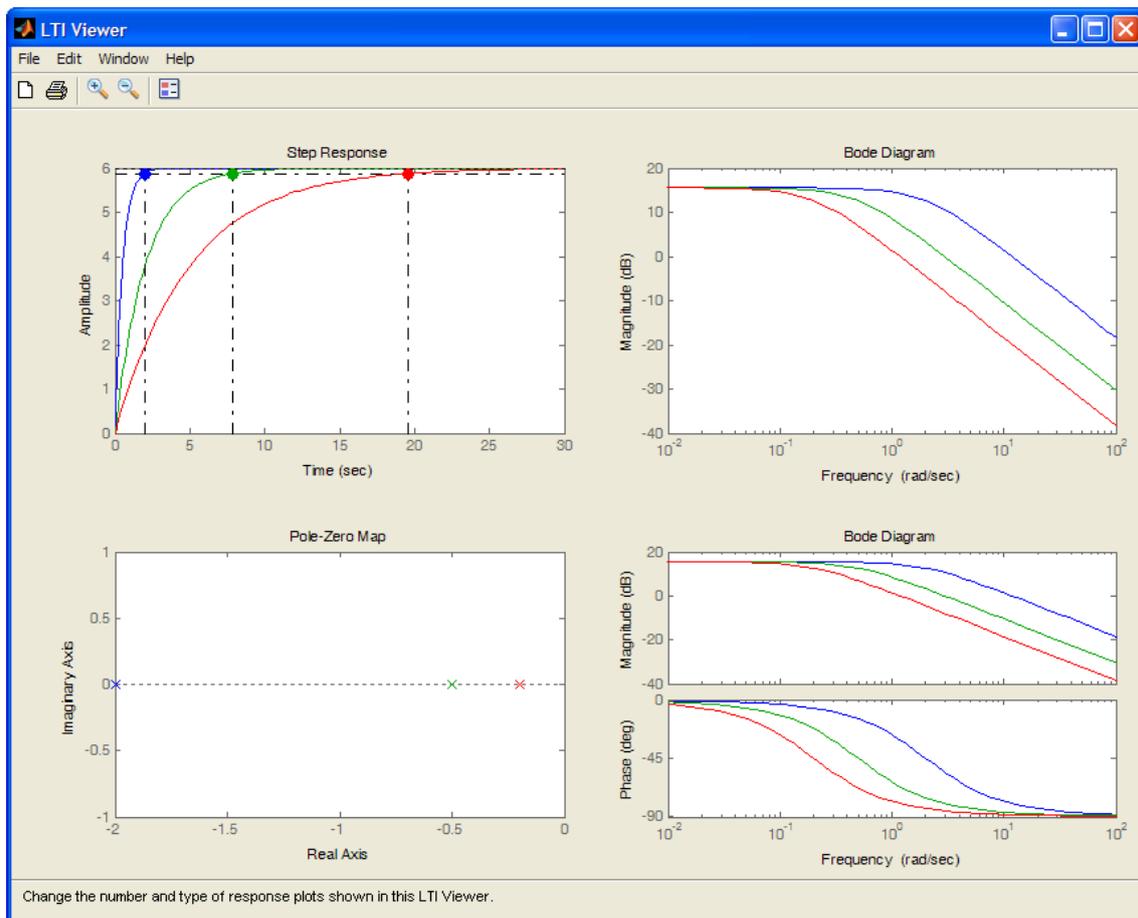
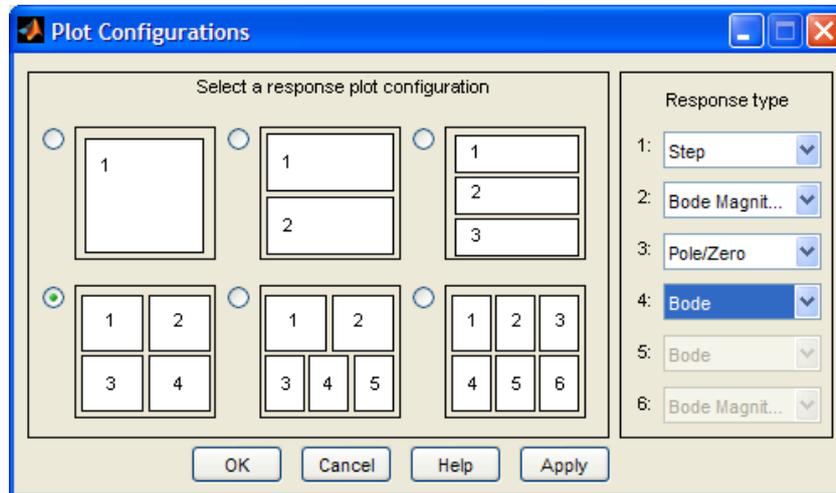
Aparecerá una nueva ventana en la pantalla. La primera vez que la ejecutamos mostrará un mensaje de ayuda que cerraremos. A continuación cargaremos desde el workspace los sistemas que queremos estudiar y que hemos implementado en el apartado anterior. En el menú superior seleccionaremos **File/Import** que desplegará una ventana donde podemos seleccionar los sistemas desde el *workspace* (*sis1*, *sis2*, *sis3*).



La aplicación LTIView nos mostrará inicialmente la respuesta temporal ante escalón de los tres sistemas. Marcando con el ratón sobre cada curva podemos visualizar el valor de amplitud para un instante determinado. Pulsando en la ventana con el botón derecho del ratón nos mostrará un menú contextual en el que podemos seleccionar la visualización de diferentes parámetros de la respuesta (activaremos el tiempo de establecimiento **Setting Time**). Podemos comparar la velocidad de la respuesta en función de la

constante de tiempo del sistema (pasaremos el ratón por encima del punto de corte para que nos muestre en valor de t_s).

A continuación vamos a añadir nuevos paneles a la respuesta temporal mostrada. Seleccionaremos la opción del menú **Edit/Plot Configurations**. Nos mostrará la siguiente ventana donde podemos seleccionar los paneles y los contenidos a visualizar. Seleccionaremos 4 paneles y los siguientes contenidos: **Step** (Respuesta ante escalón) **Pole/Zero** (Mapa de polos y ceros en el plano S), **Bode Magnitud** (Módulo del diagrama de Bode), **Bode** (Magnitud y fase del diagrama de Bode)



Con estos diagramas podemos estudiar la respuesta dinámica en función de la ubicación del polo del sistema en el plano S. Podemos ver como la frecuencia de corte de la respuesta frecuencia es inversamente proporcional a la constante de tiempo.

2.3 Comportamiento dinámico de sistemas de segundo orden

Un sistema de segundo orden responde a la siguiente función de transferencia:

$$G(s) = \frac{K \cdot \omega_n^2}{s^2 + 2\zeta \cdot s + \omega_n^2}$$

Y los dos polos del sistema se obtienen igualando a cero el polinomio del denominador (ecuación característica):

$$s_{1,2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$$

En función de los valores que tomen el coeficiente de amortiguamiento ζ y la frecuencia natural no amortiguada ω_n , el comportamiento en régimen transitorio será distinto.

Nos fijaremos en el caso más común en el que $0 < \zeta < 1$ correspondiente a sistemas subamortiguados. En este caso los polos se pueden expresar como:

$$s_{1,2} = \zeta \cdot \omega_n + j\omega_n \cdot \sqrt{1 - \zeta^2} \rightarrow s = -\sigma \pm j \cdot \omega_d \quad (0 < \zeta \leq 1)$$

Donde σ y ω_d representan el módulo de la parte real y la parte imaginaria de las raíces respectivamente.

Teóricamente se puede demostrar que el comportamiento en régimen transitorio de un sistema de segundo orden subamortiguado presenta sobreoscilaciones que se reducen en amplitud con el tiempo. Estas oscilaciones tienen una frecuencia igual a ω_d y decrecen en amplitud tanto más rápido cuanto mayor sea el valor de σ .

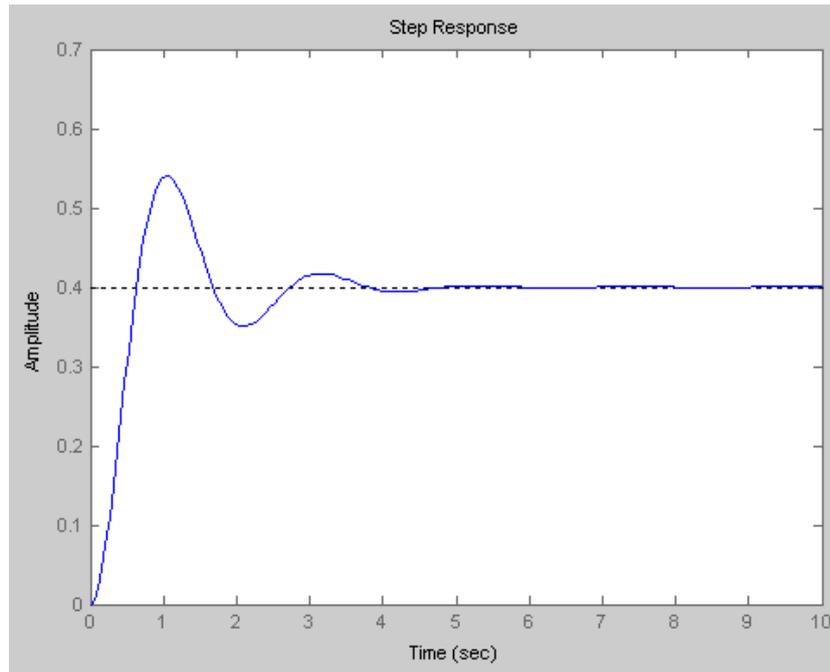
Este comportamiento lo podemos verificar sobre Matlab estudiando la respuesta a escalón de 3 sistemas en los que varían los parámetros σ y ω_d .

$$G_1(s) = \frac{4}{s^2 + 2s + 10} \quad G_2(s) = \frac{10.4}{s^2 + 2s + 26} \quad G_3(s) = \frac{10.1}{s^2 + s + 25.25}$$

Comenzando por el primer sistema, las instrucciones Matlab necesarias para obtener el valor de los polos y para comprobar la respuesta ante escalón serían las siguientes:

```

» sis1 = tf(4, [1 2 10])
Transfer function:
4
-----
s^2 + 2 s + 10
» pole(sis1)
ans =
-1.0000 + 3.0000i
-1.0000 - 3.0000i
» step(sis1,10)
    
```

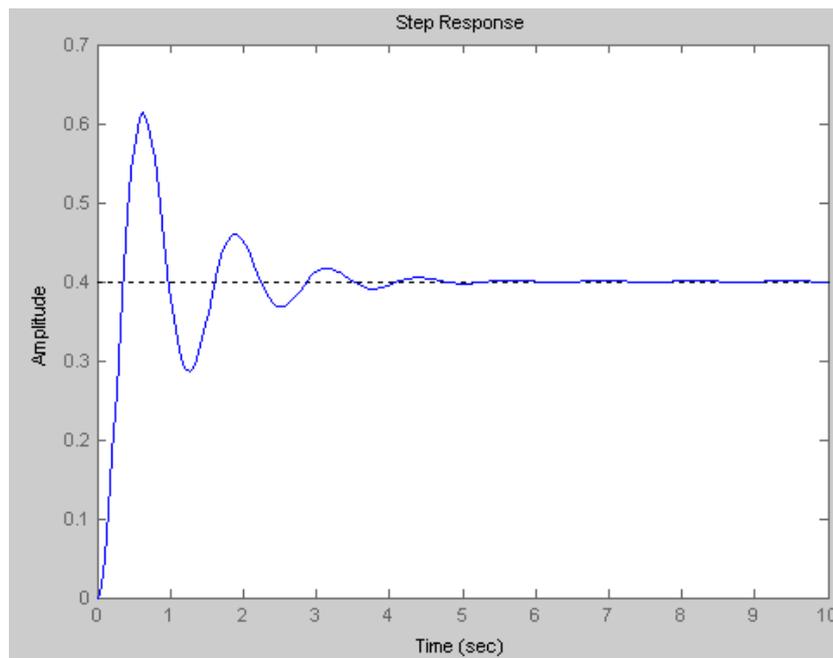


A partir de los datos devueltos por Matlab en la instrucción **pole**, podemos comprobar cómo para la función de transferencia $G_1(s)$ se obtiene $\sigma = 1$ y $\omega_d = 3$.

Haciendo las mismas operaciones con la segunda función de transferencia, se obtendrá:

```

» sis2 = tf(10.4, [1 2 26])
Transfer function:
10.4
-----
s^2 + 2 s + 26
» pole(sis2)
ans =
-1.0000 + 5.0000i
-1.0000 - 5.0000i
» step(sis2,10)
    
```

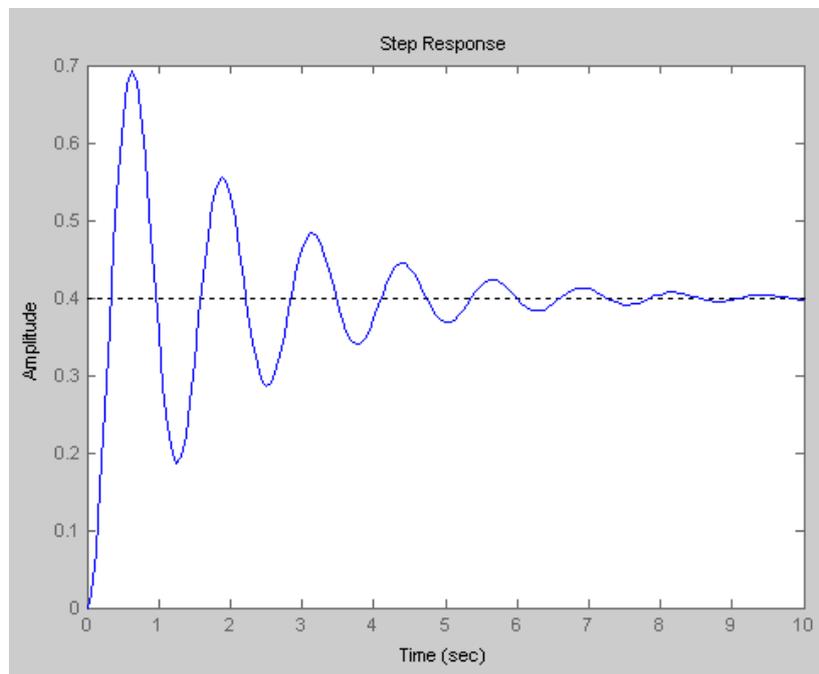


Y podemos ver como la respuesta llega al mismo valor final pero se comporta de un modo distinto en el régimen transitorio. La diferencia con los polos del sistema anterior se encuentra en el valor de la parte imaginaria, que ha aumentado hasta $\omega_d = 5$. El efecto sobre el comportamiento del sistema es el esperable, aumenta la frecuencia de las sobreoscilaciones pero la amplitud de las mismas se reduce a la misma velocidad que en el sistema anterior (s no ha variado).

Si repetimos las operaciones con la tercera función de transferencia, tendremos:

```

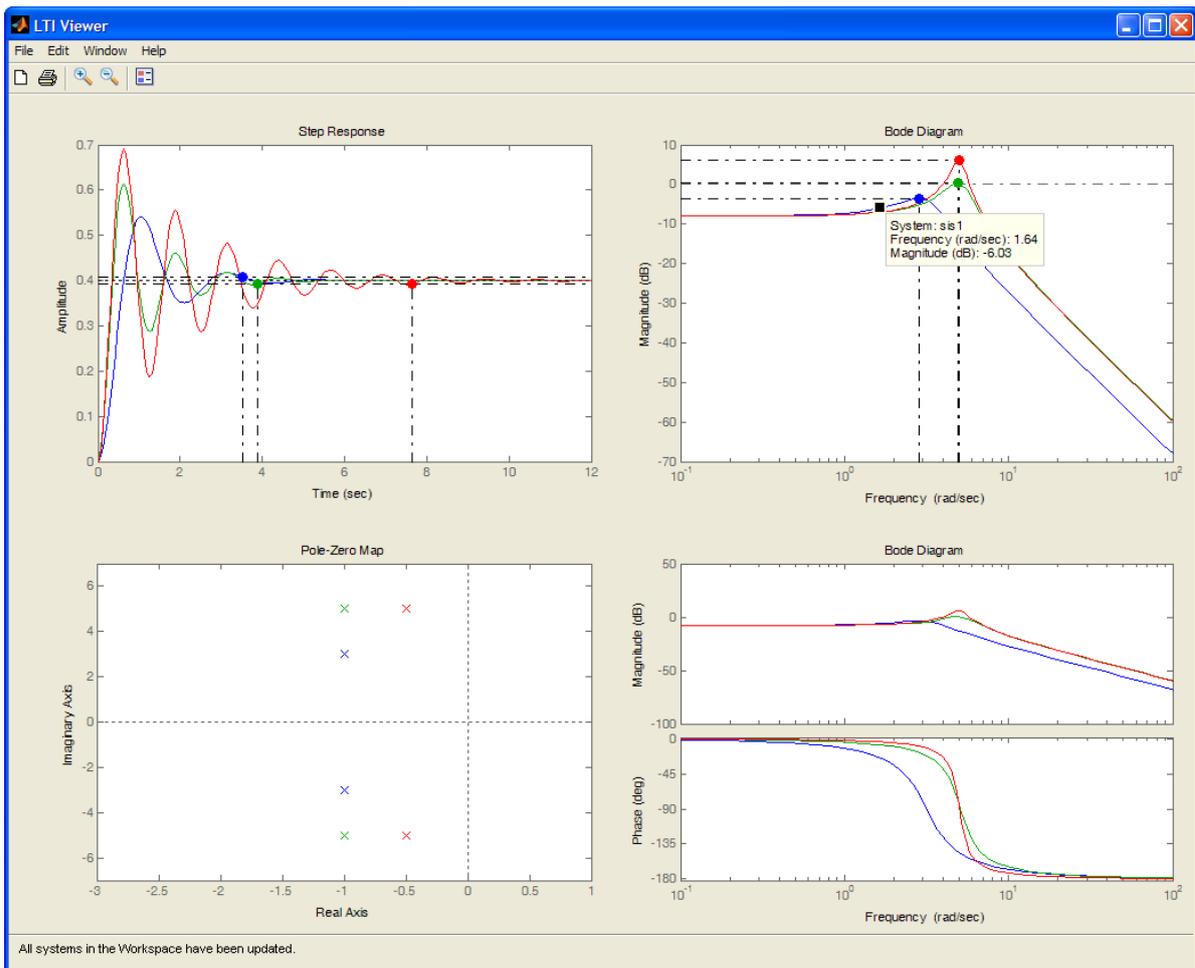
» sis3 = tf(10.1, [1 1 25.25])
Transfer function:
10.1
-----
s^2 + s + 25.25
» pole(sis3)
ans =
-0.5000 + 5.0000i
-0.5000 - 5.0000i
» step(sis3,10)
    
```



Podemos ver de nuevo como la respuesta llega al mismo valor final que en el caso anterior y presenta la misma frecuencia de sobreoscilación (ω_d no ha cambiado) pero la amplitud de la sobreoscilación decrece más lentamente, debido a la reducción del parámetro σ .

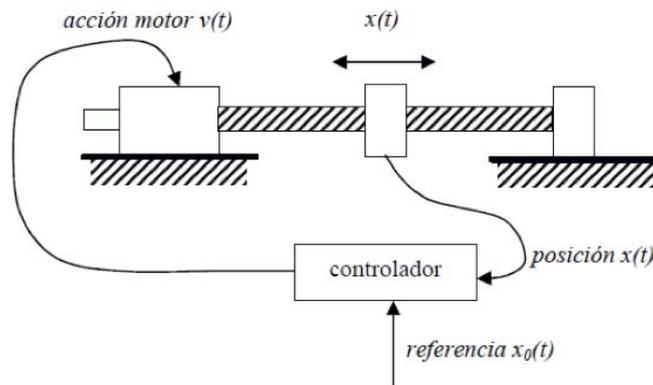
2.4 Comportamiento dinámico de sistemas de segundo orden con la herramienta LTIVIEW

Al igual que vimos con sistemas de primer orden utilizaremos la herramienta LTIVIEW para estudiar el comportamiento dinámico tanto temporal como frecuencial de los sistemas de segundo orden indicado en el apartado anterior. Destacar el pico de resonancia que se produce para sistemas poco amortiguados ($\zeta < 0.7$) a la frecuencia ω_r y produce un aumento considerable en la ganancia, pudiendo dar lugar a problemas de estabilidad en el sistema.

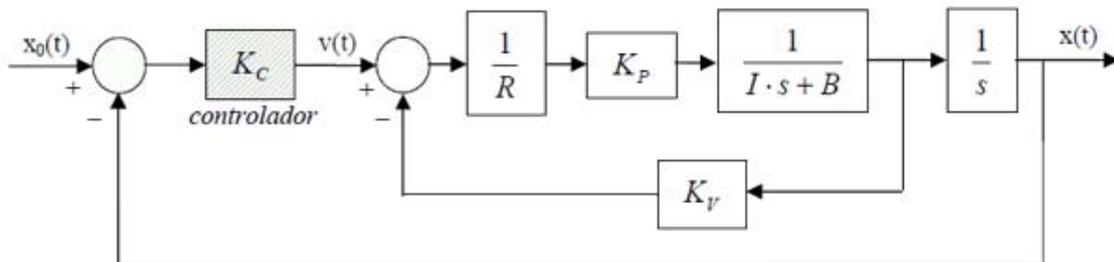


EJERCICIO 2: COMPORTAMIENTO DINÁMICO

Trabajaremos sobre un montaje ya estudiado en otra práctica: el servomecanismo de posición accionado mediante un motor eléctrico. Recordemos que el objetivo es actuar sobre el motor o aplicar al motor una tensión $v(t)$ de forma que la pieza móvil se sitúe en el lugar $x(t)$ que se le indique mediante la referencia $x_0(t)$:



El diagrama de bloques correspondiente a este montaje es el siguiente:



Donde los valores de las constantes del sistema son las siguientes:

- $R = 1.25$ (resistencia de los devanados del motor)
- $I = 0.8$ (momento de inercia del conjunto)
- $B = 0.5$ (coeficiente de rozamiento viscoso)
- $K_P = 3$ (constante de par del motor)
- $K_V = 0.01$ (constante de velocidad del motor)

La constante K_C representa el controlador empleado: la acción sobre el motor $v(t)$ será proporcional al error o diferencia entre la referencia $x_0(t)$ y la posición real $x(t)$. La constante de proporcionalidad es K_C , y **ajustar** el controlador equivale en este caso a elegir el valor de K_C que hace que el comportamiento del sistema sea el deseado.

Se pide:

- Representar el esquema anterior **en Simulink** y obtener la respuesta ante escalón de 20 unidades para los 3 valores siguientes de la constante K_C : $K_C=0.1$, $K_C=0.3$ y $K_C=0.5$. El sistema debe simularse durante **30 segundos**, y se debe ajustar el parámetro '**max step size**' a **0.01** segundos para que la representación de la respuesta sea suficientemente detallada.
- Anotar para cada valor de K_C la sobreoscilación que se produce en la señal de salida $x(t)$ (sobreoscilación = exceso porcentual sobre el valor final) y el tiempo de pico (*tiempo de pico = instante en el que se alcanza el máximo valor*). Debe comprobarse cómo la sobreoscilación aumenta al aumentar K_C y el tiempo de pico disminuye al aumentar K_C .
- Sobre Simulink, probar con distintos valores de K_C hasta encontrar uno que cumpla las siguientes especificaciones:
 - Sobreoscilación $< 28\%$
 - Tiempo de pico < 4 segundos

- Una vez elegido el valor adecuado para K_c , calcular con la ayuda de Matlab la situación de los polos del sistema resultante. Se deberán emplear las siguientes instrucciones de Matlab: **tf** para crear las funciones de transferencia; **series** y **feedback** para reducir el diagrama de bloques; y finalmente **pole** para determinar la posición de los polos en el sistema resultante.

A obtener con Matlab:

- Gráficos de la respuesta ante escalón obtenidos para los tres valores de K_c propuestos. Los gráficos deberán presentarse en formato Matlab y con el nombre del alumno como título.
- Valores de sobreoscilación y tiempo de pico obtenidos en cada caso.
- Valor de K_c que cumple las especificaciones pedidas.
- Posición de los polos en el sistema ajustado de acuerdo con las especificaciones.