

3º INGENIERÍA DE TELECOMUNICACIÓN  
2º ITT SISTEMAS ELECTRÓNICOS  
2º ITT SISTEMAS DE TELECOMUNICACIÓN

# AUTÓMATAS Y SISTEMAS DE CONTROL

## PRÁCTICA 8 SISTEMAS. UTILIDADES MATLAB

### 1. TRANSFORMADAS Y ANTITRANSFORMADAS DE LAPLACE

Matlab permite obtener transformadas y antitransformadas de Fourier, Laplace y Z mediante su módulo de matemática simbólica.

#### Procedimiento:

- Declarar una variable simbólica con la instrucción **syms**.
- Obtener la transformada para una expresión definida utilizando la variable simbólica anterior.

#### Instrucciones de Matlab correspondientes a cada una de las transformadas:

- **fourier** transformada de Fourier
- **ifourier** transformada inversa de Fourier
- **laplace** transformada de Laplace
- **ilaplace** transformada inversa de Laplace
- **ztrans** transformada Z
- **iztrans** transformada Z inversa

En esta asignatura nos centraremos en la transformada de Laplace.

#### Ejemplo:

Obtendremos la antitransformada de Laplace de la siguiente expresión:  $F(s) = \frac{2}{s \cdot (s + 0.5)}$

Las instrucciones de Matlab que utilizaremos serán:

```
» syms s
» ilaplace (2/(s*(s+0.5)))

ans =
4-4*exp(-1/2*t)
```

Por lo tanto, hemos obtenido como respuesta:  $f(t) = 4 - 4 \cdot e^{-0.5t}$

Comprobación: haciendo el procedimiento inverso buscaremos la transformada de Laplace de f(t):

```
» syms t
» laplace (4-4*exp(-0.5*t))

ans =
4/s-4/(s+1/2)
```

Se obtiene como resultado:

$$F(s) = \frac{4}{s} - \frac{4}{(s+0.5)} = \frac{4(s+0.5)}{s \cdot (s+0.5)} - \frac{4 \cdot s}{s \cdot (s+0.5)} = \frac{2}{s \cdot (s+0.5)}$$

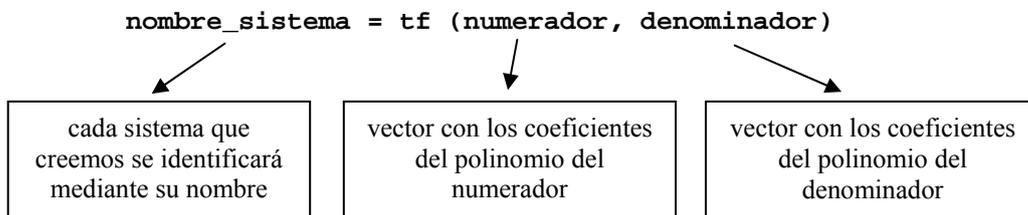
... que es la misma función F(s) de partida

## 2. REPRESENTACIÓN DE SISTEMAS CONTINUOS

Al igual que Simulink, Matlab también permite obtener la respuesta de sistemas continuos y discretos ante distintas señales de entrada. Para ello se utilizan diversas funciones del toolbox de control.

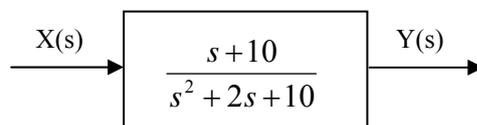
La forma de representar un sistema continuo en Matlab es mediante su función de transferencia en el dominio s de Laplace, a través de la instrucción **tf**.

### Forma de utilizar la instrucción tf:



### Ejemplo:

Queremos representar el siguiente sistema continuo:



La instrucción de Matlab a utilizar será:

```
>> sis1 = tf([1 10], [1 2 10])
```

**Transfer function:**

```

s + 10
-----
s^2 + 2 s + 10
```

A partir de este momento, la variable **sis1** representará en Matlab el sistema correspondiente a esa función de transferencia.

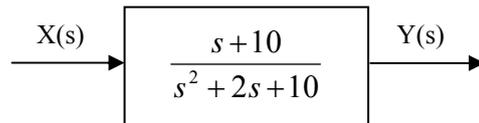
## 3. RESPUESTA A LA SEÑAL ESCALÓN

La instrucción **step** sirve para calcular la respuesta a escalón de cualquier sistema previamente definido. Caben dos posibilidades:

- Obtener la representación gráfica de la respuesta.
- Obtener los valores numéricos de la respuesta.

### Representación gráfica de la respuesta

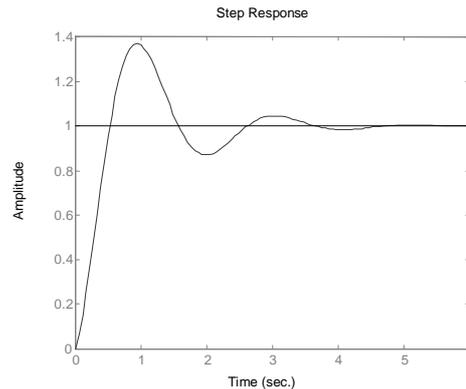
Obtendremos como ejemplo la respuesta a escalón del sistema continuo definido anteriormente:



Si suponemos que cuando creamos el sistema con la instrucción `tf` le dimos el nombre `sis1`, entonces la instrucción Matlab a teclear será la siguiente:

```
» step(sis1)
```

Y el resultado será el siguiente gráfico:



El gráfico representa el valor de la salida del sistema ante entrada escalón

### Obtención de los valores numéricos de la respuesta:

En este caso lo que deseamos no es obtener el gráfico sino descargar en una variable el valor de la respuesta en cada instante de tiempo. El formato a emplear para la instrucción es el siguiente:

```
» [y,t] = step(sis1);
```

El contenido de cada una de las variables es el siguiente:

- El vector `t` contiene los instantes de tiempo para los que se ha calculado el valor de la salida
- El vector `y` contiene los valores de la salida correspondientes a cada instante de tiempo

**NOTA:** recordemos que el punto y coma al final de la expresión sirve para que Matlab no muestre en pantalla el resultado de la operación; en otro caso habrían aparecido las ristas de valores de las variables.

Como comprobación, podemos consultar un valor cualquiera de la variable `t` y de la variable `y`; por ejemplo el valor que hace el número 25 de todos los calculados:

```
» t(25)

ans =
    1.3252

» y(25)

ans =
    1.1786
```

Vemos que el valor 25 corresponde al instante de tiempo 1.3252 segundos y que el valor de la señal de salida en ese instante de tiempo es 1.1786. Podemos comprobar estos valores aproximadamente sobre el gráfico de la respuesta que obtuvimos antes.

Disponer de los valores numéricos de los datos es útil para realizar cualquier tipo de operación matemática, como buscar el máximo, obtener el valor exacto en un instante de tiempo concreto, etc.

### **Forma de obtener la respuesta para un escalón no unitario**

Lo visto anteriormente considera que al sistema se le aplica un escalón de valor uno. Para escalones no unitarios basta con multiplicar el sistema por el valor del escalón.

Por ejemplo, para obtener la respuesta a un escalón de 5 unidades bastará con teclear estas instrucciones:

```
» step(5*sis1)
```

o bien:

```
» [y,t] = step(5*sis1);
```

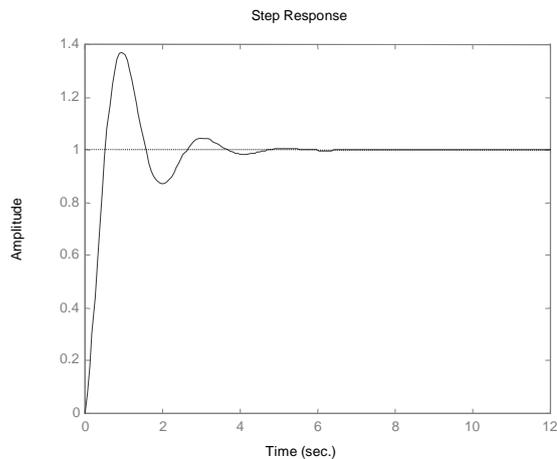
### **Forma de obtener la respuesta para instantes de tiempo posteriores**

En el ejemplo realizado, Matlab calcula la respuesta del sistema hasta el instante **t = 6 segundos** (se puede comprobar sobre el gráfico). Si se desea obtener la respuesta para instantes posteriores basta con especificar un valor para el tiempo final en la instrucción `step`.

Por ejemplo, si queremos obtener la respuesta ante escalón del sistema **sis1** no hasta el instante  $t=6$  sino hasta el instante **t=12** deberíamos teclear el siguiente comando Matlab:

```
» step(sis1, 12)
```

Y el resultado sería el que mostramos en el gráfico que aparece a continuación:



## **4. RESPUESTA A UNA SEÑAL CUALQUIERA**

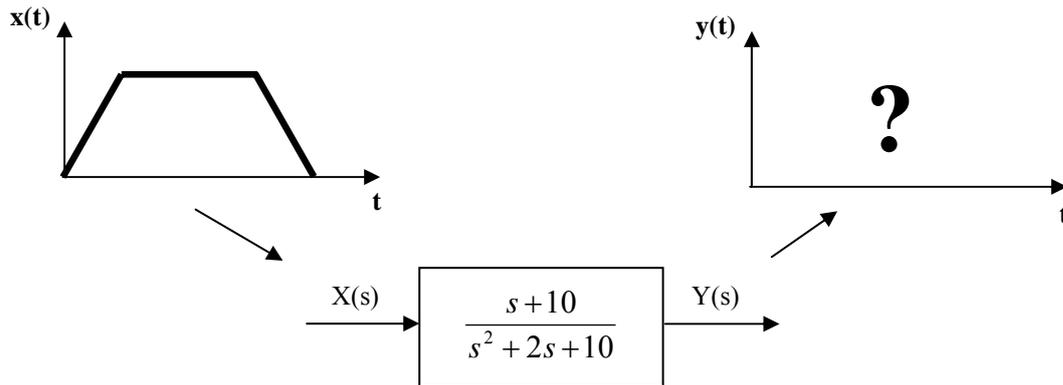
Al igual que la instrucción `step` nos ofrece la respuesta de un sistema a una señal escalón, también es posible obtener mediante Matlab la respuesta de un sistema ante una entrada cualquiera.

Para ello se utiliza la instrucción **lsim** que, al igual que la instrucción **step**, permite obtener los resultados de dos formas distintas:

- Como una representación gráfica.
- Como valores numéricos.

Hemos dicho que la instrucción **lsim** permite obtener la respuesta de un sistema ante una entrada cualquiera. El primer paso, antes de utilizar la instrucción, será definir la entrada a utilizar.

Por ejemplo, si deseamos conocer la respuesta del sistema **sis1** definido anteriormente ante una entrada  $x(t)$  como la representada en la figura, deberemos en primer lugar definir esa señal  $x(t)$ :



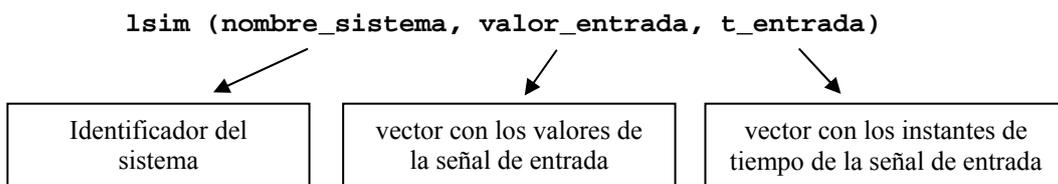
Una señal cualquiera se definirá mediante dos vectores:

- Un vector de instantes de tiempo
- Un vector de valores para la señal en cada uno de esos instantes de tiempo

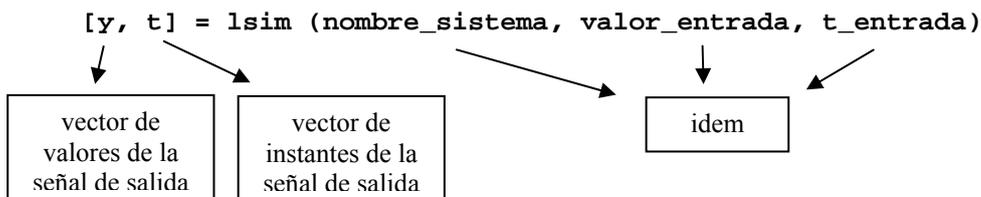
Con el objeto de representar la señal con la mayor precisión posible, es recomendable utilizar un gran número de valores o, lo que es lo mismo, hacer que los instantes de tiempo entre cada dos valores sean lo más pequeños posible.

**Formatos para la instrucción lsim:**

- Si lo que se desea es la representación gráfica de la respuesta:



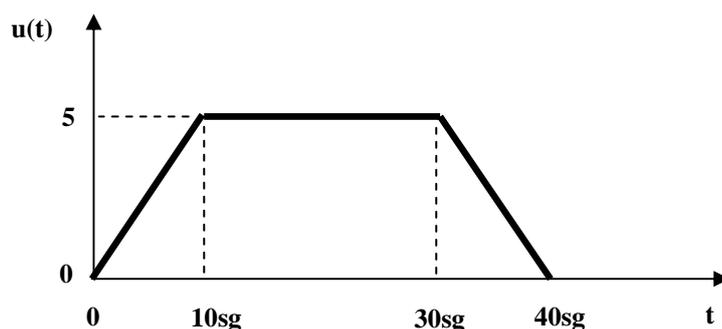
- Si lo que se desea es obtener los valores numéricos de la respuesta:



Al igual que con la instrucción **step**, los vectores **y** y **t** definen la señal de salida; de este modo es posible realizar cualquier operación matemática sobre esos datos.

**Ejemplo:**

Obtendremos la respuesta del sistema **sis1** a la siguiente señal de entrada:



Como primer paso, debemos definir mediante dos vectores **t** (tiempo) y **u** (valores) la señal de entrada. De acuerdo con lo que se ha visto en prácticas anteriores, la definición de la señal se hará con las siguientes instrucciones de Matlab:

```

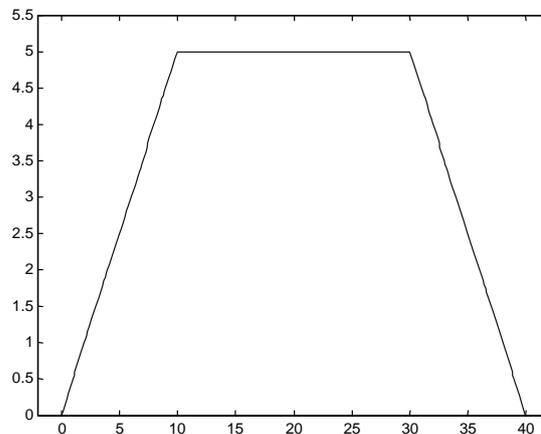
» t1 = [0:0.1:10];      % de 0 a 10 seg. a intervalos de 0.1 seg.
» u1 = 0.5*t1;          % primer tramo de la señal
» t2 = [10.1:0.1:30];  % de 10.1 a 30 seg.
» u2(1:200) = 5;        % segundo tramo de la señal
» t3 = [30.1:0.1:40];  % de 30.1 a 40 seg.
» u3 = 20 - 0.5*t3;     % tercer tramo de la señal
» t = [t1, t2, t3];     % concatenación de los vectores de tiempo
» u = [u1, u2, u3];     % concatenación de los vectores de datos

```

Es conveniente comprobar que se han definido correctamente las variables **u** y **t**. Para ello el procedimiento más inmediato es utilizar la orden **plot** de Matlab, con la que podemos representar la variable **u** (señal) en función de la variable **t** (tiempo):

```
» plot(t,u)
```

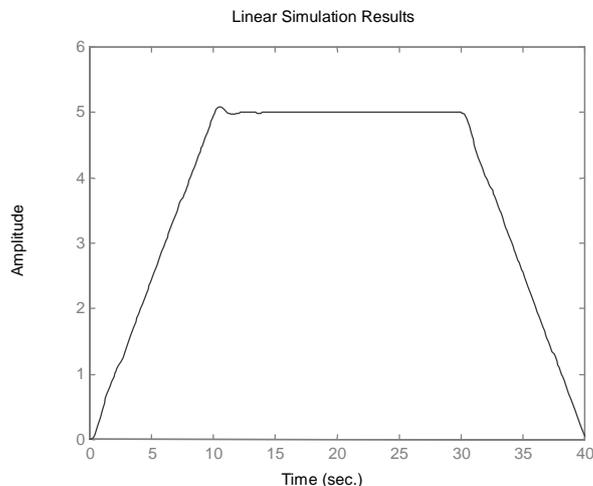
El resultado debe ser un gráfico similar al siguiente:



Una vez comprobado que los vectores se han definido correctamente, podemos lanzar la instrucción **lsim**:

```
» lsim(sis1,u,t)
```

Y el resultado será un gráfico con la respuesta que ofrece nuestro sistema **sis1** a la entrada anterior:



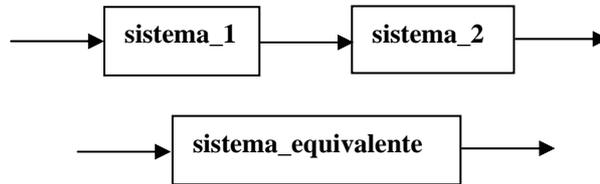
Podemos ver cómo la salida del sistema reproduce aproximadamente la entrada al mismo.

## 5. REDUCCIÓN DE DIAGRAMAS DE BLOQUES CON MATLAB

Matlab también permite obtener sistemas equivalentes para las principales combinaciones de bloques: en serie, en paralelo y en realimentación. De este modo, es posible reducir los diagramas de bloques.

### Equivalente de dos bloques en serie:

Es el producto de los dos bloques, en Matlab se obtiene con el comando **series**:

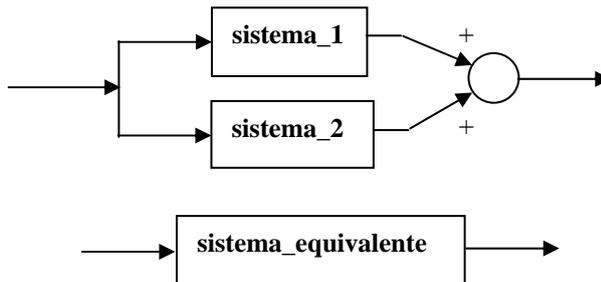


Supuestos definidos los sistemas sistema\_1 y sistema\_2, teclearíamos en Matlab:

```
» sistema_equivalente = series (sistema_1, sistema_2)
```

### Equivalente de dos bloques en paralelo:

Es la suma de los dos bloques, en Matlab se obtiene con el comando **parallel**:

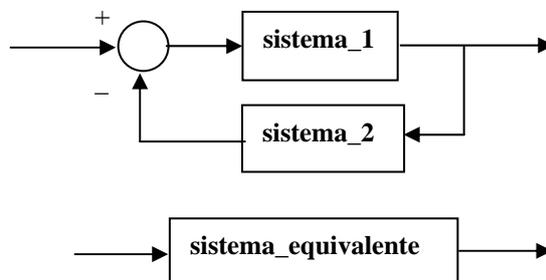


Supuestos definidos los sistemas sistema\_1 y sistema\_2, teclearíamos en Matlab:

```
» sistema_equivalente = parallel (sistema_1, sistema_2)
```

### Equivalente para un esquema de realimentación:

Se calcula en Matlab mediante el comando **feedback**:



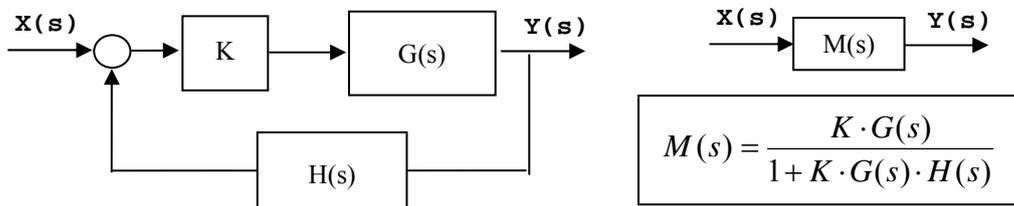
Supuestos definidos los sistemas sistema\_1 y sistema\_2, teclearíamos en Matlab:

```
» sistema_equivalente = feedback (sistema_1, sistema_2)
```

## 6. CÁLCULO DEL LUGAR DE LAS RAÍCES CON MATLAB

El toolbox de control de Matlab también dispone de comandos para el cálculo del lugar de las raíces. Las instrucciones que utilizaremos serán **rlocus** y **rlocfind**.

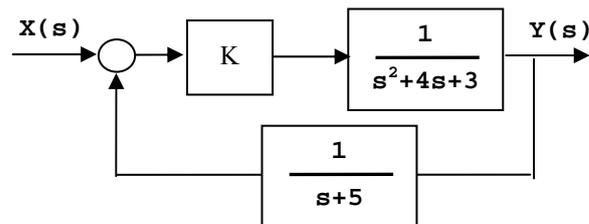
Normalmente partimos de un sistema como el siguiente:



El lugar de las raíces nos muestra la evolución de los polos del sistema realimentado o en cadena cerrada  $M(s)$  cuando el parámetro  $K$  varía desde 0 hasta  $\infty$ .

Las instrucción **rlocus** permite obtener este trazado, y utiliza como argumento el sistema en cadena abierta definido por la función de transferencia  $G(s)H(s)$ .

Ejemplo: obtendremos el lugar de las raíces para el siguiente sistema:



El primer paso es definir las funciones de transferencia  $G(s)$ ,  $H(s)$  y  $G(s) \cdot H(s)$ . Llamaremos a estas funciones **sis\_g**, **sis\_h** y **sis\_gh**, respectivamente:

```
» sis_g = tf(1, [1 4 3])
```

```
Transfer function:
```

```
      1
-----
s^2 + 4 s + 3
```

```
» sis_h = tf([1], [1 5])
```

```
Transfer function:
```

```
      1
-----
s + 5
```

```
» sis_gh = series(sis_g, sis_h)
```

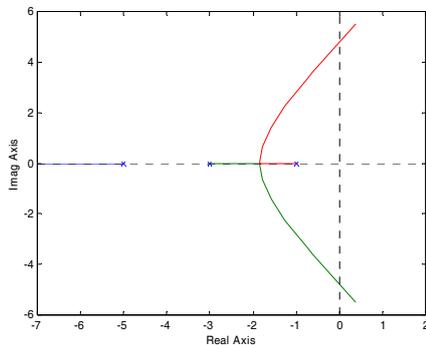
```
Transfer function:
```

```
      1
-----
s^3 + 9 s^2 + 23 s + 15
```

A continuación utilizaremos el sistema `sis_gh` recién creado como argumento para la instrucción `rlocus`:

```
» rlocus(sis_gh)
```

Matlab generará un gráfico como el siguiente:

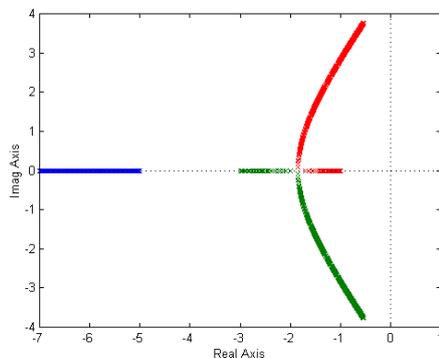


Interpretar el gráfico resultante es sencillo: muestra la situación en el plano complejo de los polos del sistema realimentado o en cadena cerrada  $M(s)$ . Cada rama representa la situación de uno de los polos; en este caso aparecen tres ramas dibujadas con tres colores distintos para mayor claridad. Los puntos de comienzo ( $K=0$ ) de cada rama coinciden con los polos en cadena abierta (cruces sobre el gráfico) y puntos de finalización ( $K=\infty$ ) de cada rama tienden a infinito en este caso.

Si no se añade ningún parámetro extra, Matlab elegirá automáticamente los valores de  $K$  entre 0 e  $\infty$  para los cuales calculará el lugar de las raíces. En determinadas ocasiones interesa elegir manualmente el rango de valores deseado para  $K$ . Para ello basta con introducir un nuevo parámetro en `rlocus`:

```
» rlocus(sis_gh, [0:.1:100]) %K de 0 a 100 a intervalos de 0.1
```

El resultado sería el siguiente:



Si no se desea un resultado gráfico, sino que se desea conocer los valores numéricos de los polos en cadena cerrada para cada valor de  $K$  la instrucción a teclear será:

```
[r, k] = rlocus(sis_gh);
```

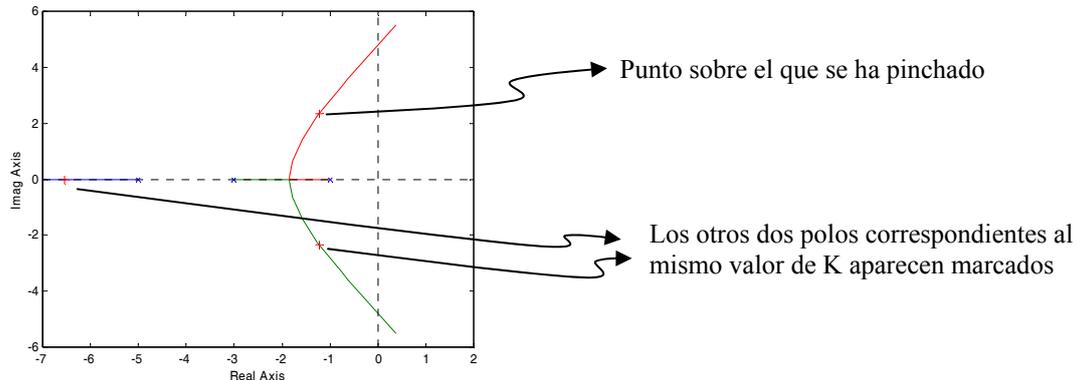
La variable `k` contendrá los valores del parámetro  $K$  utilizados para el cálculo del lugar de las raíces; la variable `r` contendrá los polos del sistema para cada valor de  $K$ .

También es posible comprobar sobre el propio gráfico los valores del parámetro  $K$  correspondientes a cada punto del lugar de las raíces. Para ello se emplea la instrucción `rlocfind`. Esta instrucción, ejecutada a continuación de `rlocus`, permite pinchar con el ratón sobre un punto cualquiera del lugar de las raíces y obtener el valor del polo más cercano al punto donde se ha pinchado, el valor de  $K$  correspondiente a ese polo y la situación del resto de polos para ese valor de  $K$  (aparecen marcados en rojo sobre el diagrama):

```
» rlocus(sis_gh)
» rlocfind(sis_gh)
```

### Select a point in the graphics window

A continuación se debe pinchar con el ratón sobre un punto cualquiera del lugar de las raíces:



La respuesta que aparece en la ventana de comandos indica el valor de  $s$  en el punto del lugar de las raíces donde se ha pinchado (selected point) y el valor de  $K$  correspondiente (ans):

```
selected_point =
  -1.2558 + 2.3509i

ans =
  30.6040
```

Tal y como indica Matlab, en este caso el punto donde se ha pinchado es  $s = -1.2558 + 2.3509j$  y el valor de  $K$  para el cual el sistema presenta ese polo es  $K = 30.604$ .

En el ejemplo, es interesante conocer el valor de  $K$  correspondiente a dos de los puntos del gráfico: el punto a partir del cual aparecen dos polos complejos conjugados y el punto a partir del cual aparecen dos polos inestables. Si buscamos estos puntos con la ayuda de **rlocfind** deberíamos obtener los siguientes resultados:

- Punto a partir del cual aparecen los polos complejos:  $K=3.08$
- Punto a partir del cual aparecen los polos inestables:  $K=192$

La forma de comprobar este comportamiento es asignar valores a  $K$  y comprobar mediante la instrucción **step** la respuesta a escalón del sistema realimentado  $M(s)$ . La forma de calcular el sistema realimentado  $M(s)$  será mediante la instrucción **feedback** vista anteriormente.

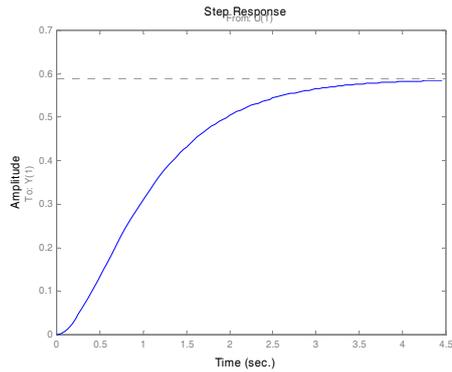
Probaremos en primer lugar un valor de  $K=2$ , de modo que los polos sean reales. En el gráfico de la respuesta a escalón obtenido se debe comprobar cómo no existe sobreoscilación:

```
>> k=2;

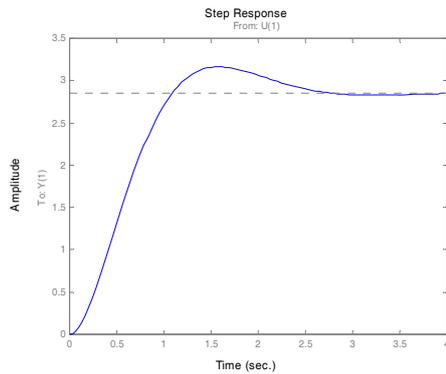
>> sis_m = feedback(k*sis_g, sis_h)

Transfer function:
      2 s + 10
-----
s^3 + 9 s^2 + 23 s + 17

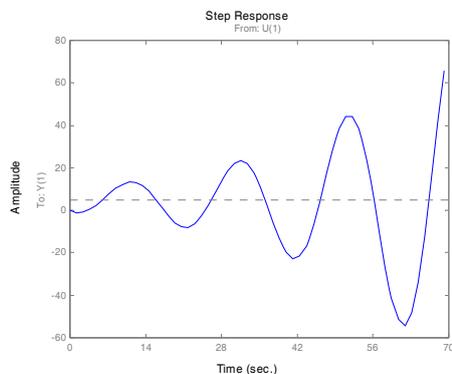
>> step(sis_m)
```



A continuación se repite la misma secuencia de comandos de Matlab, pero haciendo  $K=20$ . En este caso la respuesta a escalón sí presenta una sobreoscilación clara:



Por último, se prueba un valor  $K=200$  y se aprecia como el sistema, para ese valor de  $K$ , se hace inestable:



**NOTA:** A la hora de hacer un estudio de este tipo se debe tener en cuenta que los polos que más influirán en el comportamiento transitorio del sistema serán los más próximos al eje imaginario, lo que se ha venido denominando **polos dominantes**.

# INFORME DE LA PRÁCTICA 8

**NOMBRE:** \_\_\_\_\_ **APELLIDOS:** \_\_\_\_\_

**DNI:** \_\_\_\_\_ **TITULACIÓN:** \_\_\_\_\_

**FIRMA:** \_\_\_\_\_

1.- Como ejercicio, se obtendrán antitransformadas de Laplace para las siguientes funciones:

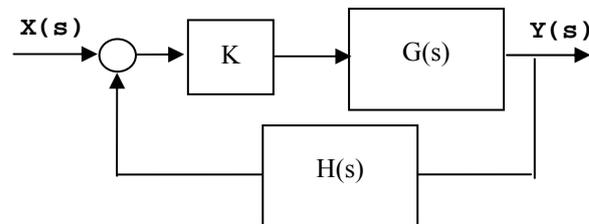
$$F_1(s) = \frac{s^2 + 2s + 3}{(s+1)^3}$$

$$F_2(s) = \frac{5s + 5}{s \cdot (s^2 + 2s + 5)}$$

$$f_1(t) = \underline{\hspace{10cm}}$$

$$f_2(t) = \underline{\hspace{10cm}}$$

2.- Partiremos del esquema utilizado a lo largo de la práctica:



Se pide:

- Crear un programa Matlab que tome como entrada dos funciones  $G(s)$  y  $H(s)$  cualesquiera y calcule el lugar de las raíces para valores de  $K$  entre 0 e  $\infty$ .

- Crear otro programa Matlab que tome como entrada dos funciones  $G(s)$  y  $H(s)$  y un valor  $K$  y devuelva un gráfico con la respuesta a escalón del sistema realimentado de la figura anterior.

- Utilizar el primero de los programas con los siguientes datos:

$$G(s) = \frac{s+3}{s^2-2s+26} \quad H(s) = \frac{s+5}{s+2}$$

...y determinar los tipos de comportamiento del sistema y en qué rango del parámetro K se producen.

Sist. Estable:  $< K <$

Sist. Intestable:  $< K <$

Sist. sobreamortiguado:  $< K <$

Sist. Críticamente amortiguado:  $< K <$

Sist. subamortiguado:  $< K <$

- Utilizar el segundo de los programas para mostrar la respuesta a escalón del sistema realimentado para un ejemplo de cada uno de los tipos de comportamiento distinto que se producen.

