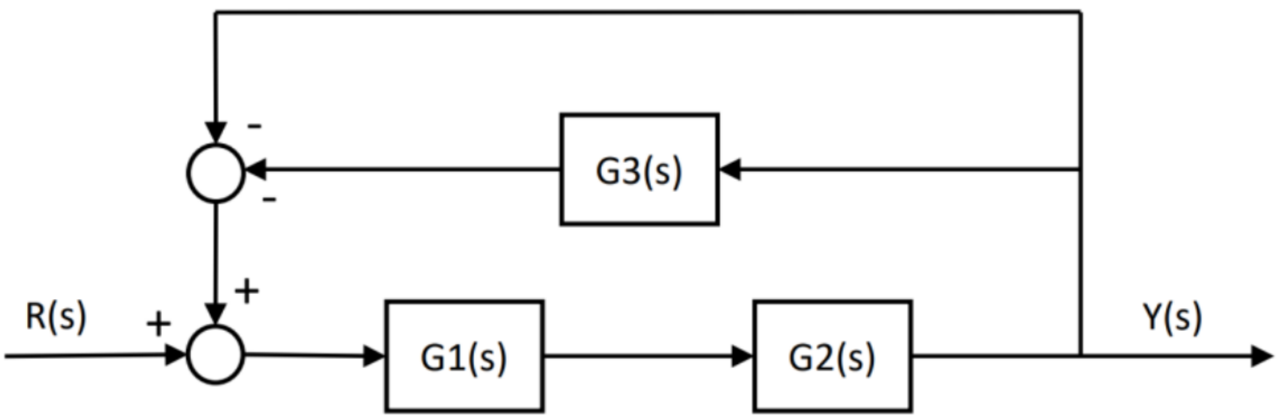
EJERCICIOS ENTREGABLES PRÁCTICA 2

MATLAB Control System Toolbox: Funciones de transferencia, transformadas de Laplace y álgebra de bloques con MATLAB

Ejercicio 1 (2,25 puntos). Generación de funciones de transferencia (tf y zpk) en MATLAB.

(i) G1(s) es una función de transferencia de primer orden con ganancia estática igual a 10 y un polo en s=-5. G2(s) tiene un cero en s=-2 y un polo triple en s=-7. G3(s) es un derivador. Calcula la función de transferencia total en lazo cerrado GLC(s) del sistema de la figura con MATLAB, mostrando el resultado con el numerador y el denominador en forma polinómica.



Se declaran las funciones de transferencia:

```
G1=zpk([],[-5],10);
G2=zpk([-2],[-7 -7 -7],1);
G3=zpk([0],[],1);
```

Se aplica el álgebra de bloques para simplificar el diagrama:

```
G12s=series(G1,G2); % G1 y G2 se encuentran conectadas en serie o cascada
G3p=parallel(G3,1); % G3 y una línea de conexión simple (1) se encuentran
% conectadas en paralelo
GTFlc=feedback(G12s,G3p); % también sería valido parallel(-G3,-1) y
% después feedback(G12s,G3p,1)
```

Se muestra el resultado con el numerador y el denominador en forma polinómica:

```
GTFlc=tf(GTFlc)
```

```
GTFlc =

                10 s + 20
  -------------------------------------
  s^4 + 26 s^3 + 262 s^2 + 1108 s + 1735

Continuous-time transfer function.
```

El resultado devuelve una función de transferencia de orden 4, con posibilidades de ser un sistema estable (criterio de Routh-Hurwitz); conteniendo además, un cero simple en s=-2 (mantiene el cero original perteneciente a G2(s)).

(ii) Se somete ahora G1(s) a una entrada rampa unitaria r(t) con condiciones iniciales iguales a cero. Calcula la expresión analítica de su salida, y(t), en función del tiempo. Realiza una gráfica de la solución en el intervalo de 0 a 5 s. Explique su forma basándose en su esbozo, y respuesta analítica.

Se declara la variable simbólica compleja del dominio de Laplace y la temporal, t:

```
syms s t
```

Se aplica la teoria simple de las funciones de transferencia, teniendo en cuenta los datos especificados en el ejercicio:

```
G1=10/(s+5);
R=1/(s^2);
Y=G1*R
```

Y =

$$\frac{10}{s^2\ (s+5)}$$

Tras mostrar por pantalla la respuesta en el dominio de s, se aplica la antitransformada de Laplace para obtener la solución en el dominio del tiempo:

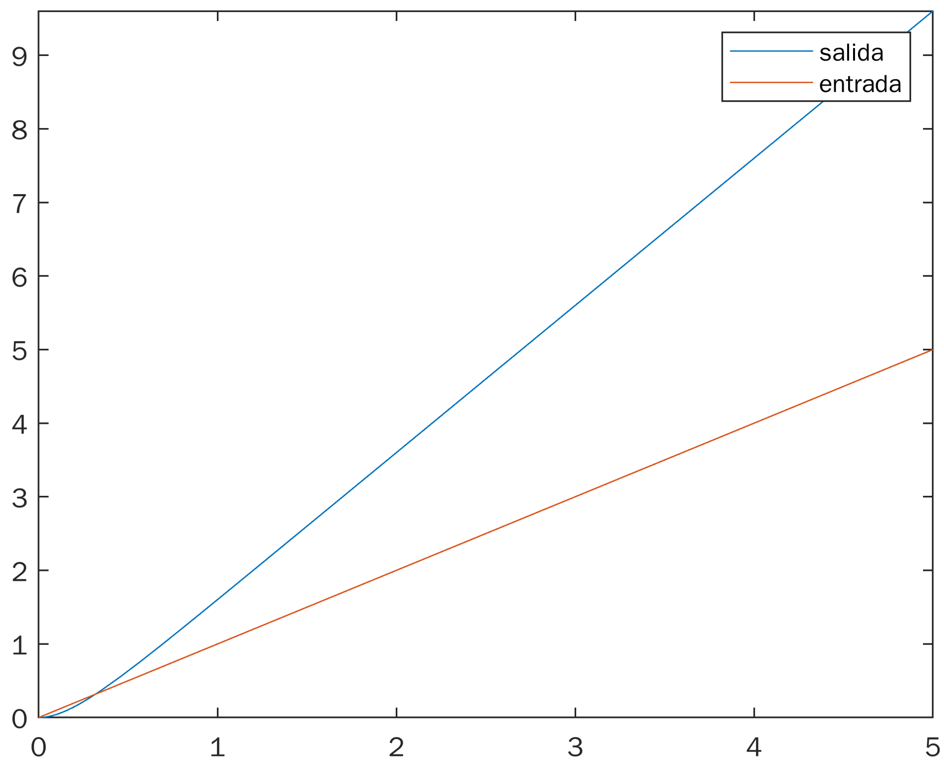
```
y=ilaplace(Y)
```

y =

$$2\,t+\frac{2\,e^{-5\,t}}{5}-\frac{2}{5}$$

En último lugar, se grafica la respuesta:

```
fplot([y t],[0 5])
legend("salida","entrada");
```



Como se puede observar, se trata de una respuesta estable (mantiene el "compromiso" entrada-salida al tener la misma naturaleza de rampa en régimen estacionario), pero poco precisa; en régimen permanente la salida tiene una pendiente cuyo valor es el doble que la entrada. El tiempo del que parte la asíntota, aproximadamente, es 0,2 segundos (constante de tiempo del sistema de primer orden).

Ejercicio 2 (1,75 puntos). Descomposición en fracciones simples con MATLAB.

Utilizando exclusivamente los comandos residue (descomposición en fracciones simples) e ilaplace (antitransformada de Laplace), obtenga la respuesta de la planta $G(s) = \dfrac{s^3 + 25s^2 + 72s + 80}{s^4 + 8s^3 + 40s^2 + 96s + 80}$ **ante una entrada en escalón unitario. Grafique la respuesta y analice su forma ayudándose de la solución analítica.**

Se declara el numerador y denominador de la función de transferencia:

```
syms s % variable compleja de Laplace
num=[3 25 72 80];
den=[1 8 40 96 80 0];
```

Se incluye la entrada 1/s en el denominador. Téngase en cuenta que la descomposición en fracciones simples se realiza sobre la salida, siendo C=G*R (C representa la salida y R es la entrada);

```
[r,p,k]=residue(num,den)
```

r = 5×1 complex

```
  -0.2812 - 0.1719i
  -0.2812 + 0.1719i
  -0.4375 + 0.0000i
  -0.3750 + 0.0000i
   1.0000 + 0.0000i
p = 5×1 complex
  -2.0000 + 4.0000i
  -2.0000 - 4.0000i
  -2.0000 + 0.0000i
  -2.0000 + 0.0000i
   0.0000 + 0.0000i
k =

     []
```

A través de un comando de flujo sofisticado, se construye, en primer lugar, la salida en el dominio de Laplace y después en el dominio del tiempo (utlizando el comando ilaplace).

```
c=0; % inicialización de la variable que almacena la salida en el dominio
% del tiempo
for i=1:length(r)
    if i>1 % condición si existen polos múltiples (en este caso, dobles)
        if real(p(i))==real(p(i-1)) & imag(p(i))==imag(p(i-1)) % si la
            % parte real e imaginaria de dos polos consecutivos, coincide;
            % se utiliza otro método de Heaviside distinto (apartado 2)
            C=r(i)/((s-p(i))^2)
        else % si no, se consideran polos simples
            C=r(i)/(s-p(i))
        end
    else
        C=r(i)/(s-p(i)); % esta afirmación se implementa para considerar
        % el primer polo bajo estudio (no puede compararse con el valor de
        % uno previo)
    end
    c=c+ilaplace(C) % se van sumando las contribuciones de las
    % "fracciones simples"
end
```

c =

$$e^{t\,(-2+4\,i)}\left(-\frac{9}{32}-\frac{11}{64}\,i\right)$$

C =

$$\frac{-\frac{9}{32}+\frac{11}{64}\,i}{s+2+4\,i}$$

c =

$$e^{t\,(-2-4\,i)}\left(-\frac{9}{32}+\frac{11}{64}\,i\right)+e^{t\,(-2+4\,i)}\left(-\frac{9}{32}-\frac{11}{64}\,i\right)$$

C =

$$-\frac{7}{16\,(s+2)}$$

c =

$$-\frac{7\,e^{-2t}}{16}+e^{t\,(-2-4\,i)}\left(-\frac{9}{32}+\frac{11}{64}\,i\right)+e^{t\,(-2+4\,i)}\left(-\frac{9}{32}-\frac{11}{64}\,i\right)$$

```
C =
```

$$-\frac{3}{8\,(s+2)^2}$$

```
c =
```

$$-\frac{7\,e^{-2t}}{16}-\frac{3\,t\,e^{-2t}}{8}+e^{t\,(-2-4\,i)}\left(-\frac{9}{32}+\frac{11}{64}\,i\right)+e^{t\,(-2+4\,i)}\left(-\frac{9}{32}-\frac{11}{64}\,i\right)$$

```
C =
```

$$\frac{1}{s}$$

```
c =
```

$$1-\frac{3\,t\,e^{-2t}}{8}-\frac{7\,e^{-2t}}{16}+e^{t\,(-2-4\,i)}\left(-\frac{9}{32}+\frac{11}{64}\,i\right)+e^{t\,(-2+4\,i)}\left(-\frac{9}{32}-\frac{11}{64}\,i\right)$$

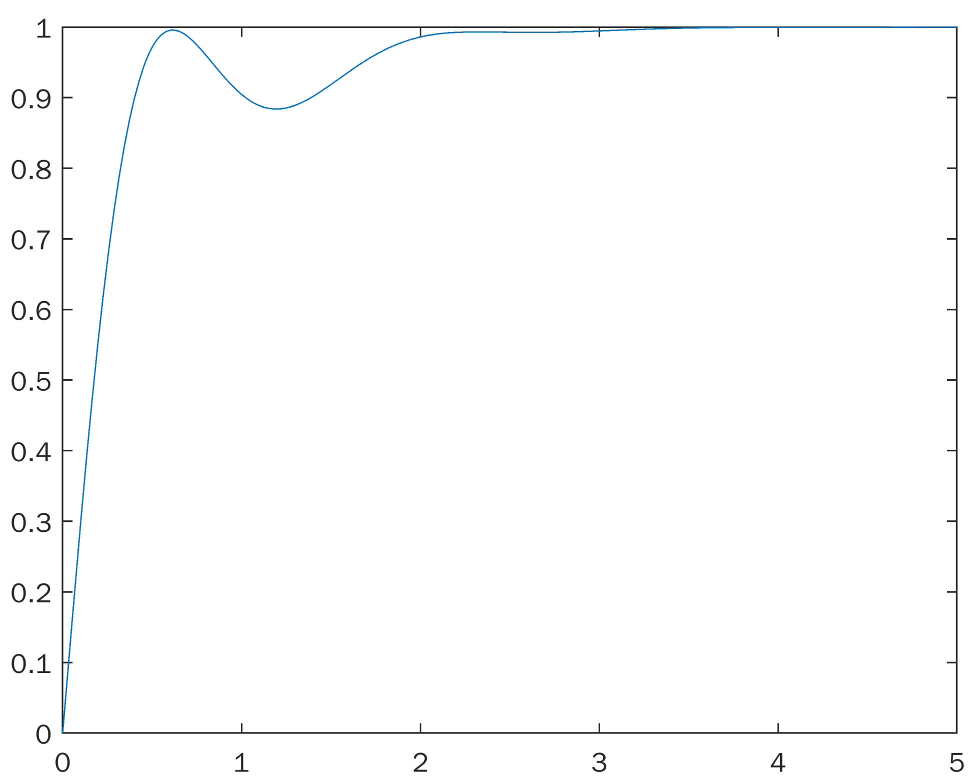
Se muestra por pantalla la solución y, posteriormente, se grafica:

```
c
```

```
c =
```

$$1-\frac{3\,t\,e^{-2t}}{8}-\frac{7\,e^{-2t}}{16}+e^{t\,(-2-4\,i)}\left(-\frac{9}{32}+\frac{11}{64}\,i\right)+e^{t\,(-2+4\,i)}\left(-\frac{9}{32}-\frac{11}{64}\,i\right)$$

```
fplot(c, [0 5])
```



La clave del ejercicio reside en el resultado de los residuos. Algunos de ellos exhiben parte compleja conjugada, por lo que es necesario operar con la expresión de C(s) antes de poder aplicar la transformada de Laplace a través de las tablas:

$$\begin{aligned}
C(s) &= \frac{3s^3 + 25s^2 + 72s + 80}{s^4 + 8s^3 + 40s^2 + 96s + 80}\frac{1}{s} \\
&= \frac{-0.2813 - j0.1719}{s + 2 - j4} + \frac{-0.2813 + j0.1719}{s + 2 + j4} \\
&\quad + \frac{-0.4375}{s + 2} + \frac{-0.375}{(s + 2)^2} + \frac{1}{s} \\
&= \frac{-0.5626(s + 2)}{(s + 2)^2 + 4^2} + \frac{(0.3438) \times 4}{(s + 2)^2 + 4^2} \\
&\quad - \frac{0.4375}{s + 2} - \frac{0.375}{(s + 2)^2} + \frac{1}{s}
\end{aligned}$$

Resultando la siguiente respuesta temporal:

$$\begin{aligned}
c(t) &= -0.5626e^{-2t}\cos 4t + 0.3438e^{-2t}\sin 4t \\
&\quad - 0.4375e^{-2t} - 0.375te^{-2t} + 1
\end{aligned}$$

La respuesta obtenida se identifica como la resultante de un sistema de orden superior. Presenta oscilaciones (términos seno/coseno), pero sin sobrepasos (que remiten con el paso del tiempo al estar multiplicados por exponenciales decrecientes) con respecto al valor final (único valor constante de la respuesta analítica obtenida, 1). Esto es así porque no se puede aplicar la aproximación por polos dominantes.

Ejercicio 3 (1,25 puntos). Transformadas de Laplace en la resolución de ecuaciones diferenciales utilizando MATLAB.

Una de las aplicaciones de la transformada de Laplace consiste en que puede ser utilizada para resolver ecuaciones diferenciales lineales de coeficientes constantes con condiciones iniciales nulas, a través de la propiedad: [d^n(x(t))/d(t^n)]=(s^n)*F(s). Se solicita resolver la siguiente ecuación diferencial; y''(t)+3y'(t)+2y(t)=e^(-t), siendo y'(0)=y(0)=0; utilizando, tan solo, los comandos laplace e ilaplace.

La solución es:

```
syms s t
y=ilaplace(laplace(exp(-t))/(s^2+3*s+2))
```

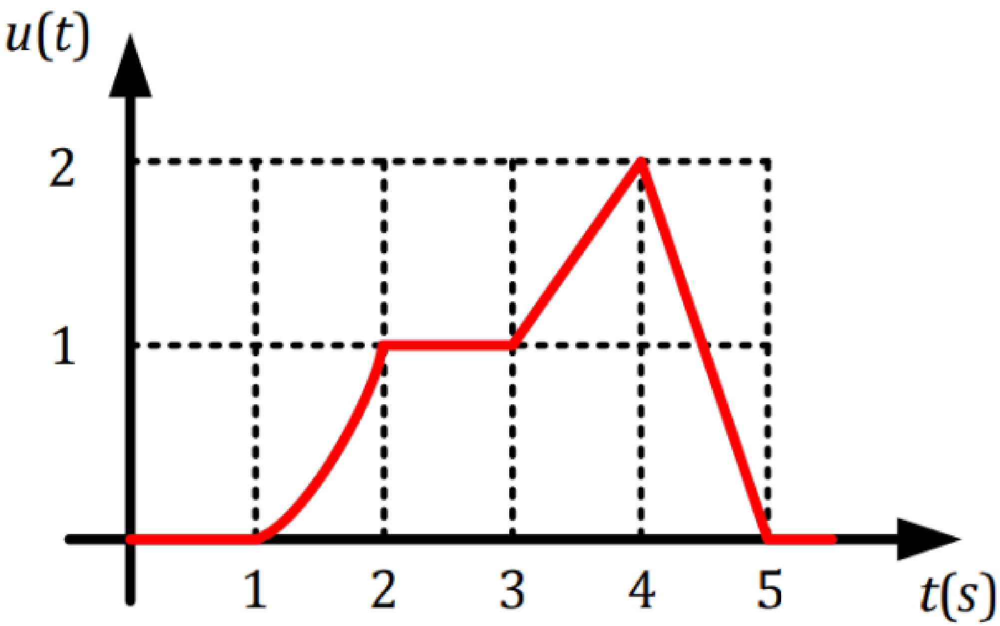
y = $e^{-2t} - e^{-t} + t\,e^{-t}$

Se podrían utilizar los siguientes comandos para indicar las condiciones iniciales nulas: subs(y,t,0) y subs(diff(y),t,0) así como la declaración de la ecuación diferencial lineal y de coeficientes constantes: diff(y,2)+3*diff(y)+2*y-exp(-t). Nótese que se trata de una ecuación diferencial lineal ya que, aunque aparezca un termino exponencial, éste sólo se aplica a la variable independiente, t.

En este caso, se obtiene una respuesta temporal con un régimen permanente consistente en un término exponencial decreciente de coeficiente característico -t. En efecto, la solución particular de la ecuación diferencial completa es un término de igual naturaleza que la entrada (miembro de la derecha de la ecuación diferencial).

Ejercicio 4 (2,25 puntos). Transformadas inversas de Laplace en MATLAB.

Obtenga la respuesta del sistema, cuyo modelo matemático se expresa a través de la función de transferencia $G(s) = \frac{2(s+1)}{s(s^2+s+2)}$**, frente a la entrada de la figura (con un tramo parabólico). Se pide mostrar en una misma gráfica la entrada y la respuesta del sistema. Analice, tramo a tramo, la solución obtenida.**



Se declara la función de transferencia utilizando la forma simbólica:

```
syms s t
G=2*(s+1)/(s*(s^2+s+2));
```

También se declara la variable temporal.

Se declara la entrada tramo a tramo, activando y desactivando intervalos de la señal a trozos utilizando la función escalón de Heaviside.

De t = 1 a 2 segundos:

```
x1=heaviside(t-1)*(t-1)^2-heaviside(t-2)*(t-1)^2;
```

Se aplica la función parabólica a partir de t=1 utilizando la función escalón u(t-1) y se anula a partir de t=2 con u(t-2).

De t = 2 a 3 segundos:

```
x2=heaviside(t-2)-heaviside(t-3);
```

El segundo tramo es una función escalón unitario simple. Se procede de igual forma que en el tramo previo.

De t = 3 a 4 segundos:

```
x3=heaviside(t-3)*(t-2)-heaviside(t-4)*(t-2);
```

En este caso, se trata de una función rampa unitaria desplazada 2 segundos (ver para los desplazamientos en el dominio de s, a través de la propiedad número 5 de las transformadas de Laplace).

De t = 4 a 5 segundos:

```
x4=heaviside(t-4)*(-2*t+10)-heaviside(t-5)*(-2*t+10);
```

Finalmente, se aplica la función rampa, teniendo en cuenta el punto inicial y la pendiente de caída, decayendo de 2 a 0 en 1 segundo.

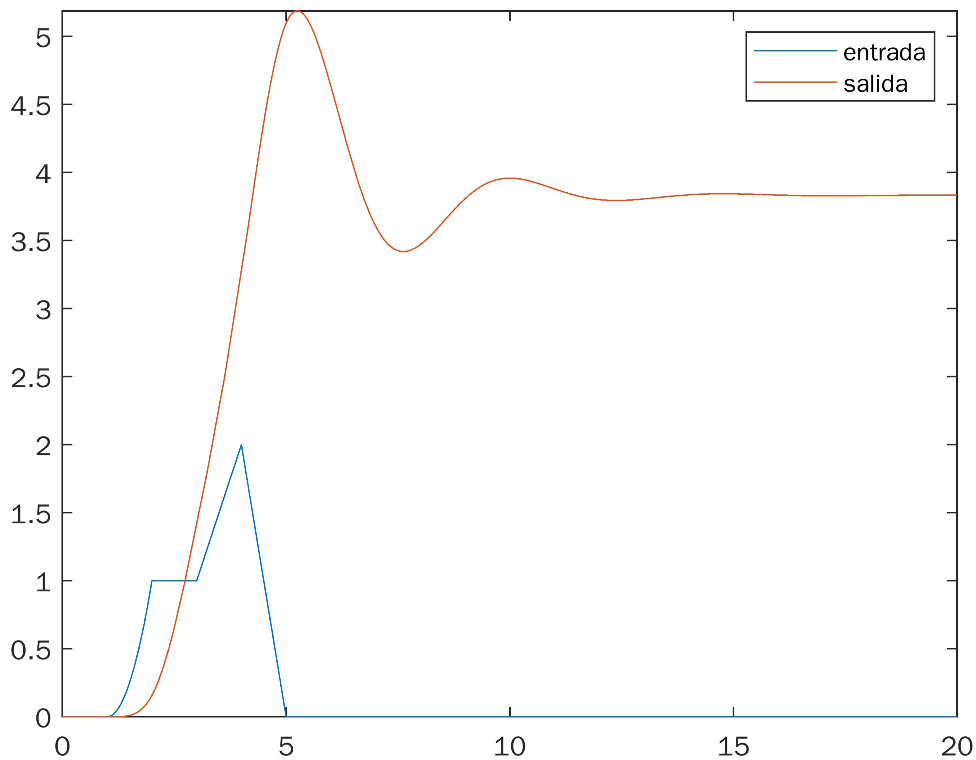
Se aplica la transformada de Laplace a la entrada construida, tras sumar todos los términos que constituyen la entrada:

```
x=x1+x2+x3+x4;
X=laplace(x);
```

Nótese que también se podría haber utilizado la función "piecewise" (función a trozos).

Se aplica la teoría simple de control clásica relativa a las funciones de transferencia y se "plotean", tanto la entrada como la salida:

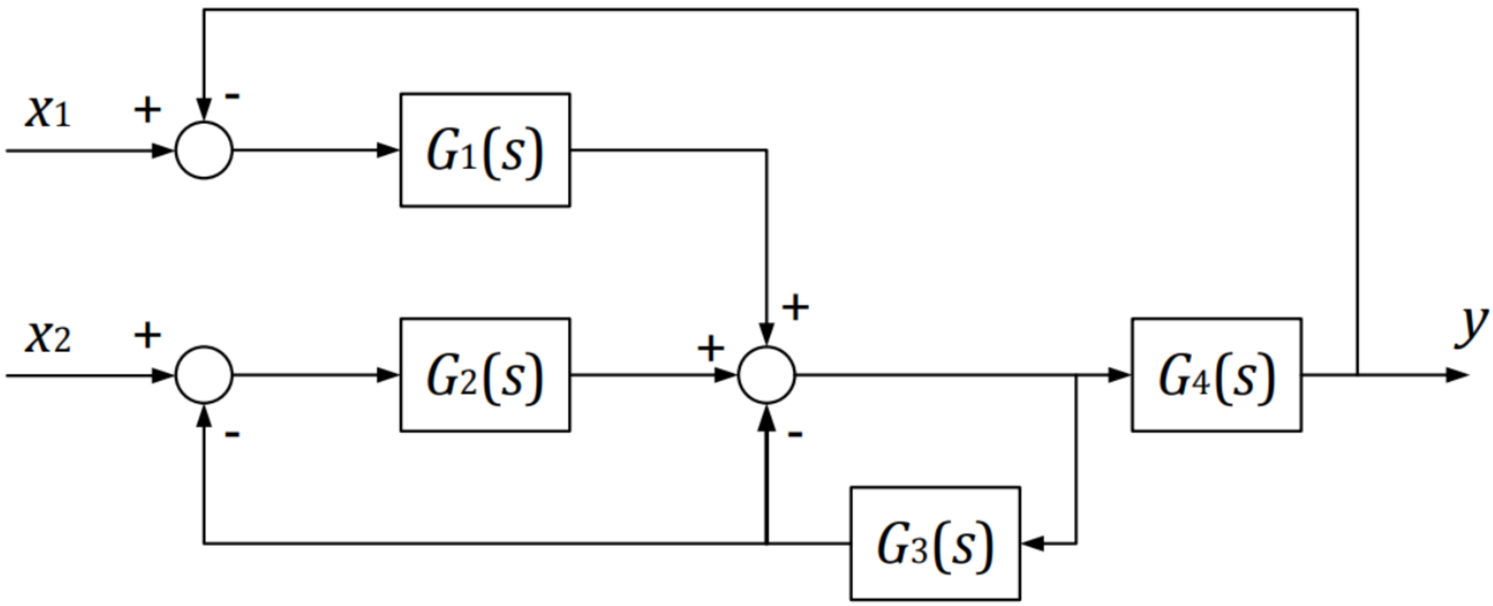
```
Y=G*X;
y=ilaplace(Y);
fplot([x y], [0 20])
legend("entrada","salida");
```



Tal y como se puede deducir de la función de transferencia, G(s), se trabaja con un sistema críticamente estable (polo en s=0). La entrada contiene tramos de velocidad y aceleración (rampa y parábola, respectivamente), por lo que aparecen errores al tratarse de entradas "tan agresivas" desde un contexto de la teoría de señales. El primer intervalo de entrada que aparece es una parábola, la cual induce un comportamiento anómalo para todo el rango temporal restante. El sistema es muy poco preciso ya que cuenta con un error de aproximadamente 400% en términos relativos.

<u>Ejercicio 5 (2,5 puntos).</u> Álgebra de bloques mediante MATLAB.

Sea el diagrama de bloques de la figura:



siendo $G_1(s) = 4; G_2(s) = \dfrac{4}{s+4}; G_3(s) = \dfrac{s+1}{s+4}; G_4(s) = \dfrac{2}{s+1}.$

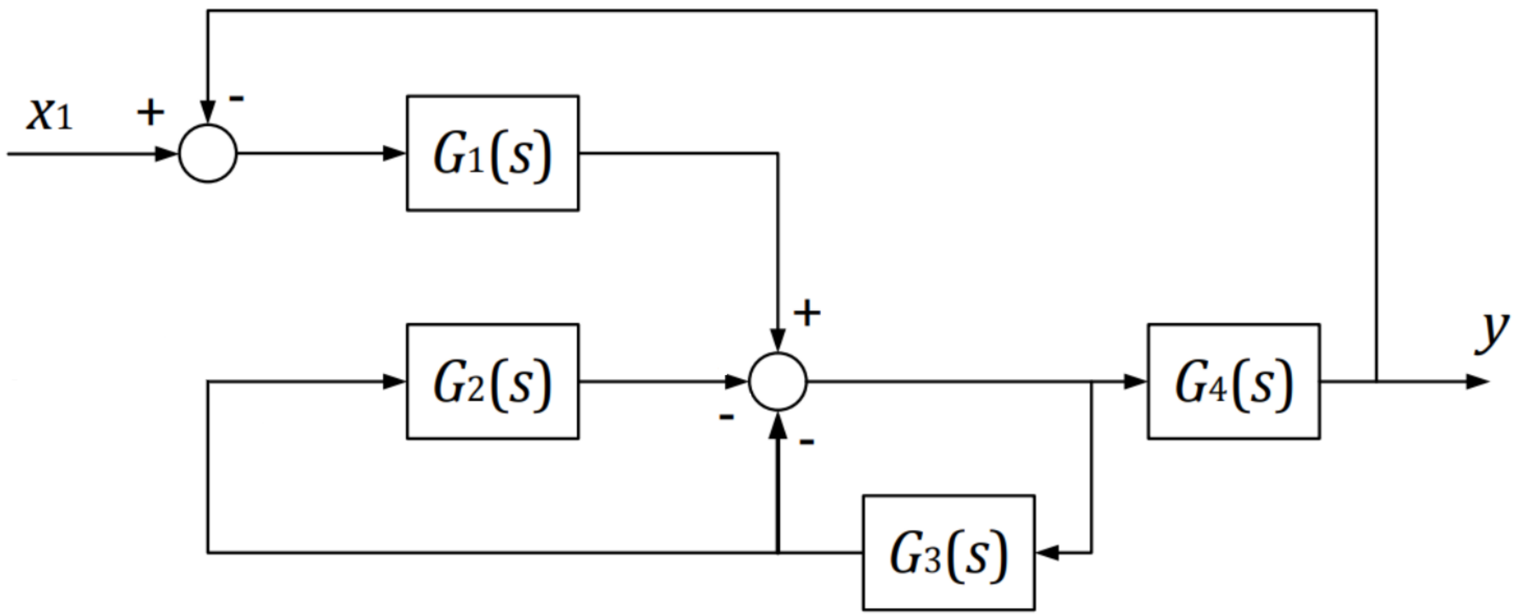
Se pide obtener las funciones de transferencia Y(s)/X1(s) y Y(s)/X2(s).

Se declaran, inicialmente, las funciones de transferencia:

```
G1=tf([4],[1]);
G2=tf([4],[1 4]);
G3=tf([1 1],[1 4]);
G4=tf([2],[1 1]);
```

A partir de aquí, se aplica el álgebra de bloques para simplificar el diagrama y obtener ambas funciones de transferencia. Además, es necesario anular o desactivar las entradas que no se estudien; es decir, x2 cuando se calcule Y(s)/X1(s) y x1 para Y(s)/X2(s).

Comenzamos con el cálculo de Y(s)/X1(s). El diagrama de bloques resultante es:



En primer lugar, se asocia G2(s) y 1 ya que ambas están en paralelo (mismo final e inicio):

```
G2p=parallel(G2,1);
```

El resultado queda en serie con G3(s) en la rama de realimentación, estando 1 en trayectoria directa. Se realizan ambas operaciones:

```
G23=series(G2p,G3);
G23f=feedback(1,G23);
```

Ya se tienen todos los bloques conectados en cascada en trayectoria directa. Por tanto:

```
G1234=series(series(G1,G23f),G4); % recuérdese que el comando "series"
% solo asocia dos bloques
```

Finalmente se aplica la operación de retroalimentación general, obteniendo, por tanto, el resultado final:

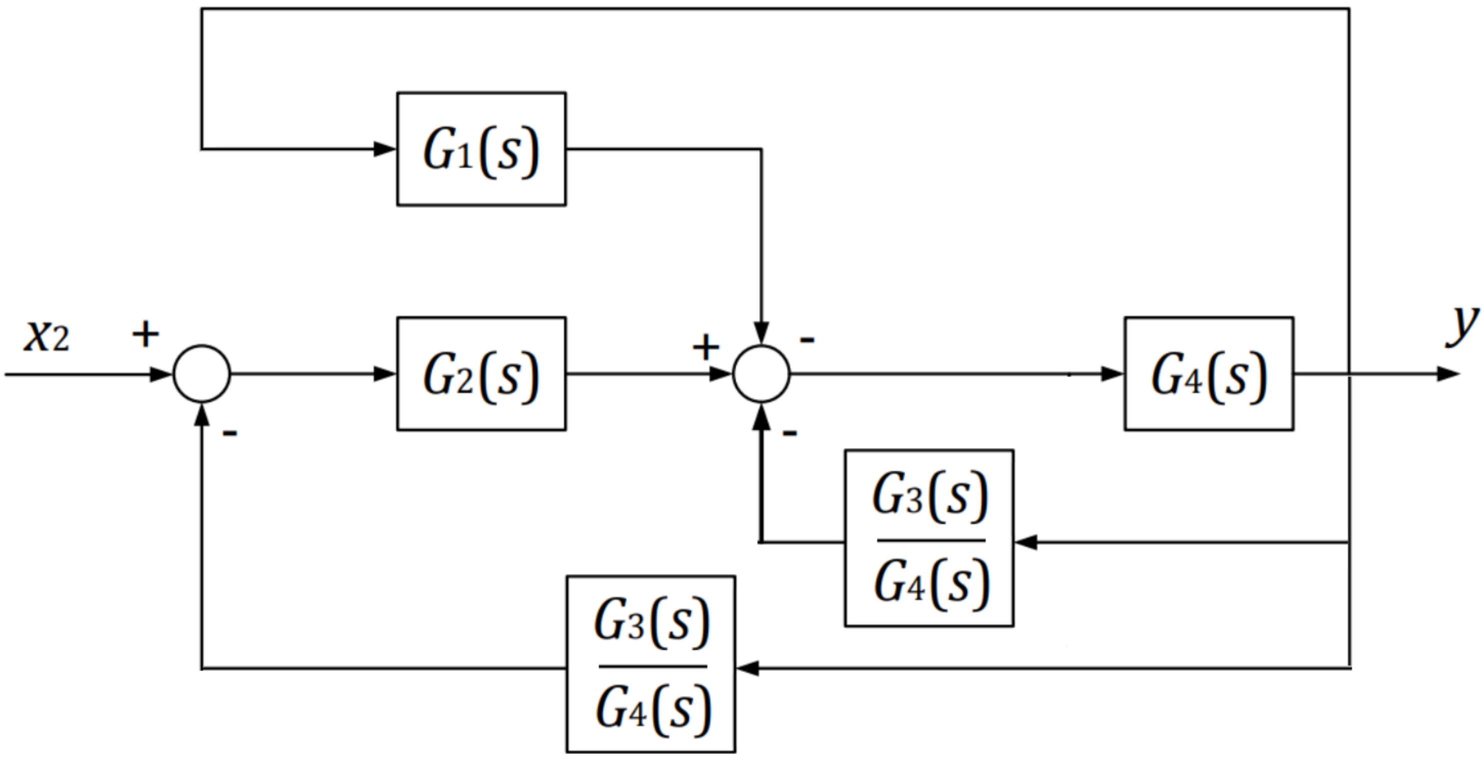
```
Gx1_y=feedback(G1234,1)
```

```
Gx1_y =

       8 s^2 + 64 s + 128
  ----------------------------
  2 s^3 + 27 s^2 + 105 s + 152

Continuous-time transfer function.
```

Por otro lado, calculamos ahora la función de transferencia Y(s)/X2(s). Antes de presentar el diagrama de bloques resultante, es necesario realizar modificaciones, ya que aparecen "nudos" en la parte inferior del sistema. La bifurcación central inferior "salta" G4(s) hacia la izquierda en retroalimentación. Por otro lado, las bifurcaciones que tiene G4(s) a la izquierda, saltan dicho bloque hacia la derecha. Tras aplicar estos pasos, se tiene:



Ahora, se procede del interior al exterior del diagrama de bloques. En primer lugar, G1(s) y G3(s)/G4(s) están en paralelo:

```
G134p=parallel(G3/G4,G1);
```

El resultado es el lazo de retroalimentación con G4(s) en trayectoria directa:

```
G134f=feedback(G4,G134p);
```

Se asocia con G2(s) ya en serie:

```
G1234s=series(G2,G34f);
```

Y, por último, se aplica el feedback del bloque resultante obtenido hasta ahora con G3(s)/G4(s), obteniendo:

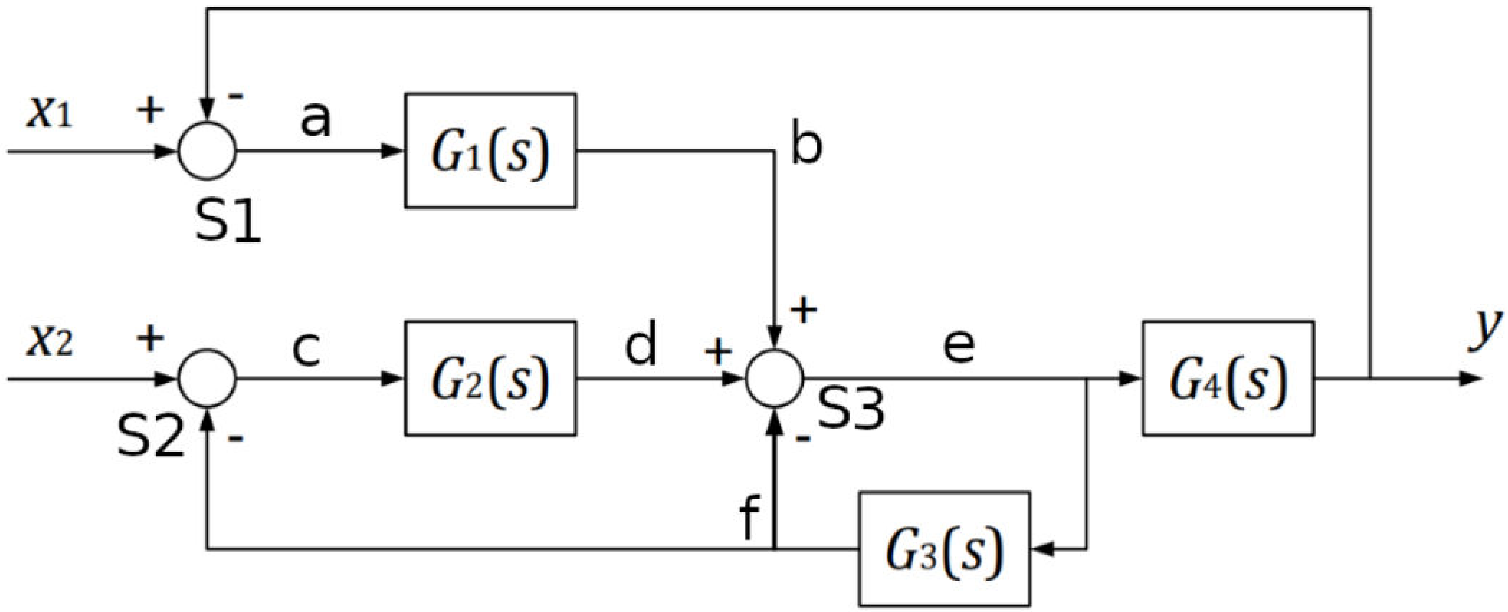
```
Gx2_y=feedback(G1234s,G3/G4)
```

```
Gx2_y =

           32 s^2 + 256 s + 512
  ---------------------------------------
  8 s^4 + 108 s^3 + 468 s^2 + 752 s + 384

Continuous-time transfer function.
```

Se pueden utilizar las funciones *connect* y *sumblk* para interconectar (mediante comandos) los bloques y así construir el diagrama solicitado. Esta constituye una solución alternativa. Para ello, se generarán inicialmente las respectivas entradas y salidas de las funciones de transferencia, asignando "un nombre" (ver figura) a cada uno de ellas y siendo coincidente cuando corresponda. Por ejemplo, si dos bloques se encuentran conectados en cascada, la salida de uno será la entrada del siguiente. Finalmente, se utlizará el comando *sumblk* para declarar los puntos de suma y resta correspondientes y *connect* para generar el diagrama de bloques en general.




```
% Declaración de entradas y salidas de las funciones de transferencia
G1.InputName='a';
G1.OutputName='b';
G2.InputName='c';
G2.OutputName='d';
G3.InputName='e';
G3.OutputName='f';
G4.InputName='e';
G4.OutputName='y';
% Declaración de los puntos de suma/resta del diagrama de bloques
S1=sumblk('a=x1-y');
S2=sumblk('c=x2-f');
S3=sumblk('e=d+b-f');
% Generación del sistema o diagrama de bloques y obtención de las
% funciones de transferencia del sistema MIMO con dos entradas,
% x1 y x2, y una salida, y
DB=connect(G1,G2,G3,G4,S1,S2,S3,{'x1','x2'},'y');
tf(DB)
```

```
ans =

  From input "x1" to output "y":
       4 s^2 + 32 s + 64
  ----------------------------
  s^3 + 13.5 s^2 + 52.5 s + 76

  From input "x2" to output "y":
           4 s + 16
  ----------------------------
  s^3 + 13.5 s^2 + 52.5 s + 76

Continuous-time transfer function.
```

Lógicamente, se obtienen los mismos resultados por ambos métodos, constituyendo estos últimos, simplificaciones de los resultados previos.