EJERCICIOS ENTREGABLES PRÁCTICA 2

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

<u>Ejercicio 1 (2 puntos).</u> Generación de funciones de transferencia en MATLAB.

Se tiene la función de transferencia $G(s) = \dfrac{1}{s^2 + 2\beta s + 1}$. Se pide encontrar, utilizando comandos de control de flujo (for, while, etc.) y tf/zpk, los valores de β que conduzcan a la obtención de polos múltiples. Razone la respuesta.

Para resolver el problema se plantea un bucle for, barriendo un rango de valores razonable de, por ejemplo, -100 a 100, almacenando los resultados a través de una sentencia interior del bucle cuando se encuentre la igualdad entre ambos polos (polinomio de grado 2).

```
i=1;
for beta=-100:0.1:100
    % Se extraen los polos mediante el comando roots aplicado al
    % denominador de la función de transferencia
    poles=roots([1 2*beta 1]);
    % Si los polos son iguales, se ejecuta la sentencia de almacenamiento
    if(real(poles(1))==real(poles(2)) && imag(poles(1))==imag(poles(2)))
        soluc(i)=beta;
        i=i+1;
    end
    % Téngase en cuenta que se deben comparar la parte real e imaginaria de
    % los polos, pues si solo se compara una de ellas, el resultado sería
    % erróneo
end
```

Se muestra por pantalla los valores de beta y las función de transferencia resultantes:

```
for k=1:1:(i-1)
    soluc(k)
    G=tf(1,[1 2*soluc(k) 1])
end
```

```
ans = -1
G =

        1
  -------------
  s^2 - 2 s + 1

Continuous-time transfer function.
ans = 1
G =

        1
  -------------
  s^2 + 2 s + 1
```

```
Continuous-time transfer function.
```

<u>Ejercicio 2 (1,75 puntos).</u> Transformadas de Laplace utilizando softwares basados en métodos numéricos.

Demuestre la veracidad de las siguientes transformadas de Laplace en las que se introduce la potencia no entera de s: (i) y (ii) para a>0. Nótese que erfc (también así denotado en MATLAB) representa la función error de variable compleja.

(i) Uno puede probar diferentes números fraccionarios, por ejemplo, seleccionando γ = 1/3, se puede obtener la transformada de Laplace de la función y siendo igual que la del lado derecho:

```
% Declaración de las variables simbólicas temporales y del dominio de
% Laplace
syms s t
g=sym(1/3); % generación de una variable simbólica escalar
laplace(t^g)-gamma(g+1)/s^(g+1) % al ser ambos términos iguales
```

ans = 0

```
% según la ecuación, el resultado debe ser 0
```

(ii) La transformada de Laplace se puede evaluar directamente intentando reproducir la representación del lado derecho de la igualdad.

```
% Declaración de variables simbólicas a>0 y t
syms a positive
syms t
% Aplicación de la transformada de Laplace
laplace(1/sqrt(t)/(1+a*t))
```

ans =

$$\frac{\pi \operatorname{erfc}\left(\sqrt{\frac{s}{a}}\right) \mathrm{e}^{s/a}}{\sqrt{a}}$$

```
% El resultado debe ser el dado en el enunciado del problema
```

<u>Ejercicio 3 (2,25 puntos).</u> Descomposición en fracciones simples y transformadas inversas de Laplace utilizando MATLAB.

Para la función $F(s) = \frac{s^2 + 4s + 3}{s^5 + 7s^4 - 2s^3 - 100s^2 - 232s - 160} e^{-5s}$, se pide encontrar los polos y sus asociados residuos. Se recomienda utilizar comandos novedosos, como factor y/o limit. ¿Coinciden con el resultado que devolvería residue?. A continuación, realice la antitransformada de Laplace y razone qué representa la función F(s) haciendo hincapié en el término exponencial.

Para denominadores polinómicos, los polos se pueden encontrar mediante la técnica de factorización:

```
syms x % variable simbólica x
f=(x^2+4*x+3)/poly2sym([1 7 -2 -100 -232 -160],x)*exp(-5*x);
% También, se declara la función de transferencia simbólica, para extraer
% posteriormente el numerador y denominador de la misma
[n,D]=numden(f);
D1=factor(D) % factorización del denominador
```

D1 = $\begin{pmatrix} -1 & x+5 & x-4 & x+2 & x+2 & x+2 \end{pmatrix}$

se encuentra que el denominador factorizado es $D_1 = (x+2)^3(x+5)(x+4)$, y se ve inmediatamente que x = −5, x = 4 son polos simples mientras que x = −2 es un polo triple. Los residuos de los polos se pueden resolver como:

```
r1=limit((x+5)*f,x,-5)
```

r1 =

$$\frac{8\,e^{25}}{243}$$

```
r2=limit((x-4)*f,x,4)
```

r2 =

$$\frac{35\,e^{-20}}{1944}$$

```
r3=limit(diff((x+2)^3*f,x,2)/2,x,-2)
```

r3 =

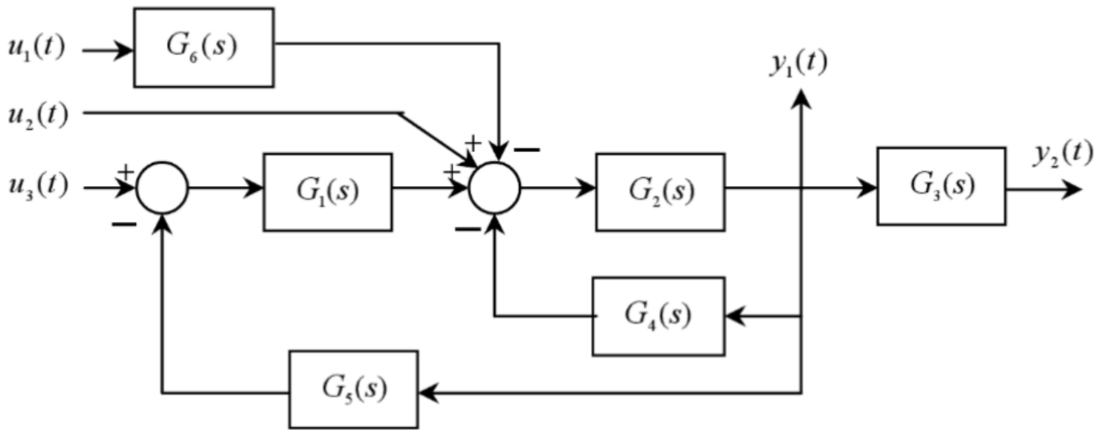
$$\frac{149\,e^{10}}{216}$$

```
% La aplicación de estos límites son la esencia de la técnica de
% descomposición en fracciones simples que, a veces, se aplica con métodos
% más simples
```

Los residuos asociados a los polos son, por tanto: $r_1 = \frac{8e^{25}}{243}$, $r_2 = \frac{35e^{-20}}{1944}$ y $r_3 = \frac{149e^{10}}{216}$.

<u>Ejercicio 4 (4 puntos).</u> Álgebra de bloques con MATLAB.

El sistema de control mostrado en la figura cuenta con dos entradas y tres salidas, siendo:



$$G_1(s)=\frac{2s}{s+1};\ G_2(s)=\frac{5}{s^2+6s+10};\ G_3(s)=\frac{s+0{,}1}{s+0{,}2};\ G_4(s)=\frac{0{,}5s+2}{s+0{,}7};\ G_5(s)=5;\ G_6(s)=6$$

Se pide obtener todas las funciones de transferencia utilizando el método convencional (comandos series, parallel y feedback) y/o el procedimiento avanzado; sumblk y connect

En primer lugar, se declaran las funciones de transferencia y sus respectivas entradas y salidas que forman parte del diagrama de bloques general:

```
G1=tf([2 0],[1 1],'InputName','e','OutputName','a');
G2=tf(5,[1 6 10],'InputName','k','OutputName','y1');
G3=tf([1 0.1],[1 0.2],'InputName','y1','OutputName','y2');
G4=tf([0.5 2],[1 0.7],'InputName','y1','OutputName','l');
G5=tf([5],[1],'InputName','y1','OutputName','o');
G6=tf([6],[1],'InputName','u1','OutputName','p');
```

A continuación, se declaran los puntos de suma:

```
sum1=sumblk('k=u2+a-l-p'); sum2=sumblk('e=u3-o');
```

Y finalmente, se conectan todos los elementos definidos.

```
T=connect(G1,G2,G3,G4,G5,G6,sum1,sum2,{'u1','u2','u3'},{'y1','y2'});
```

La solución dada por connect en el sistema MIMO (multiple inputs, multiple outputs) viene dada en espacio de estados. A continuación, hay que obtener el resultado en forma de funciones de transferencia:

```
Ttf=tf(T)
```

```
Ttf =

  From input "u1" to output...
               -30 s^2 - 51 s - 21
   y1:  --------------------------------------
        s^4 + 7.7 s^3 + 73.4 s^2 + 68.7 s + 17

                 -30 s^3 - 54 s^2 - 26.1 s - 2.1
   y2:  -----------------------------------------------------
```

```
       s^5 + 7.9 s^4 + 74.94 s^3 + 83.38 s^2 + 30.74 s + 3.4

  From input "u2" to output...
                5 s^2 + 8.5 s + 3.5
   y1:  -------------------------------------
       s^4 + 7.7 s^3 + 73.4 s^2 + 68.7 s + 17

                  5 s^3 + 9 s^2 + 4.35 s + 0.35
   y2:  ----------------------------------------------------
       s^5 + 7.9 s^4 + 74.94 s^3 + 83.38 s^2 + 30.74 s + 3.4

  From input "u3" to output...
              10 s^2 + 7 s + 4.598e-15
   y1:  -------------------------------------
       s^4 + 7.7 s^3 + 73.4 s^2 + 68.7 s + 17

                 10 s^3 + 8 s^2 + 0.7 s + 4.18e-16
   y2:  ----------------------------------------------------
       s^5 + 7.9 s^4 + 74.94 s^3 + 83.38 s^2 + 30.74 s + 3.4

Continuous-time transfer function.
```