Guión de la Práctica 2 Fundamentos de la Automática:

1.Modelado de sistemas LTI mediante funciones de transferencia

1.1 Creación de funciones de transferencia de forma polinómica (TF)

```
% G(s) = 2s+3 / 4s²+5s+6
numerador=[2 3]
```

```
numerador = 1×2
     2     3
```

```
denominador=[4 5 6]
```

```
denominador = 1×3
     4     5     6
```

```
G=tf(numerador,denominador)
```

```
G =

      2 s + 3
  ---------------
  4 s^2 + 5 s + 6

Continuous-time transfer function.
```

Ejercicio práctico 1. Definición de funciones de transferencia mediante TF

1. Define las tres siguientes funciones de transferencia en MATLAB:

G1 () = 1 / s^2+1

```
num_G1=1
```

```
num_G1 = 1
```

```
den_G1=[1 0 1]
```

```
den_G1 = 1×3
     1     0     1
```

```
G1=tf(num_G1,den_G1)
```

```
G1 =

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

Continuous-time transfer function.
```

2 () = s+1 / s^2-5s+6

```
num_G2=[1 1]
```

```
num_G2 = 1×2
```

```
     1     1
```

```
den_G2=[1 -5 6]
```

```
den_G2 = 1×3
     1    -5     6
```

```
G2=tf(num_G2,den_G2)
```

```
G2 =
 
      s + 1
  -------------
  s^2 - 5 s + 6
 
Continuous-time transfer function.
```

G3 () = s²+2s+3 / s+4s³+5s+6

```
num_G3=[1 2 3]
```

```
num_G3 = 1×3
     1     2     3
```

```
den_G3=[1 0 4 0 5 6]
```

```
den_G3 = 1×6
     1     0     4     0     5     6
```

```
G3=tf(num_G3,den_G3)
```

```
G3 =
 
      s^2 + 2 s + 3
  ---------------------
  s^5 + 4 s^3 + 5 s + 6
 
Continuous-time transfer function.
```

2. Una vez definidas, observa qué tipo de objeto usa MATLAB para almacenarlas (columna Class del Workspace).

Clase tf.

Ejercicio práctico 2. Definición de funciones de transferencia no polinómicas

1. Define las dos siguientes funciones de transferencia ayudándote de conv():

G4 () = 10(+ 5) (s + 1)(− 3)(2 + 3 + 6)

```
G4=tf([10 50],conv(conv([1 1],[1 -3]),[1 3 6]))
```

```
G4 =
 
             10 s + 50
  ----------------------------
  s^4 + s^3 - 3 s^2 - 21 s - 18
 
Continuous-time transfer function.
```

G5 () = s^2+3 / $(s^2+s+1)(s^3+2s+3)$

```
G5=tf([1 0 3],conv([1 1 1],[1 0 2 3]))
```

```
G5 =

                s^2 + 3
  -----------------------------------
  s^5 + s^4 + 3 s^3 + 5 s^2 + 5 s + 3

Continuous-time transfer function.
```

1.2 Creación de funciones de transferencia de forma factorizada (ZPK)

Ejercicio práctico 3. Definición de funciones de transferencia factorizadas

1. Define la función de transferencia G6(s) para un sistema que tiene una ganancia K igual a 2, un cero situado en -1 y tres polos situados en -2, -3 y -4.

```
G6=zpk(-1,[-2 -3 -4],2)
```

```
G6 =

       2 (s+1)
  -----------------
  (s+2) (s+3) (s+4)

Continuous-time zero/pole/gain model.
```

2. Repite el apartado anterior definiendo G7(s) para un sistema que tiene una ganancia K igual a 5, un cero situado en +1 y tres polos situados en -2, +3 y -4.

```
G7=zpk(1,[-2 3 -4],5)
```

```
G7 =

       5 (s-1)
  -----------------
  (s+2) (s+4) (s-3)

Continuous-time zero/pole/gain model.
```

3. Define la función de transferencia G8(s) que no tiene ceros y tiene un polo triple en -1 (con K=1).

```
G8=zpk([],[-1 -1 -1],1)
```

```
G8 =

     1
  -------
  (s+1)^3

Continuous-time zero/pole/gain model.
```

4. Define la función de transferencia G9(s) que no tiene ceros y tiene tres polos, uno en -1 y los otros dos polos complejos conjugados -1 + 2i, -1 -2i (con K=1).

```
G9=zpk([],[-1 -1+2i -1-2i],1)
```

```
G9 =

          1
  --------------------
  (s+1) (s^2 + 2s + 5)

Continuous-time zero/pole/gain model.
```

1.3 Conversión entre funciones de transferencia polinómicas y factorizadas

Ejercicio práctico 4. Conversión de funciones de transferencia

1. Convierte a forma factorizada las funciones de transferencia G1, G2, G3, G4 y G5 definidas en los ejercicios prácticos 1 y 2.

```
G1_1=zpk(G1)
```

```
G1_1 =

     1
  ---------
  (s^2 + 1)

Continuous-time zero/pole/gain model.
```

```
G2_2=zpk(G2)
```

```
G2_2 =

    (s+1)
  -----------
  (s-3) (s-2)

Continuous-time zero/pole/gain model.
```

```
G3_3=zpk(G3)
```

```
G3_3 =

                      (s^2 + 2s + 3)
  --------------------------------------------------------
  (s+0.7738) (s^2 - 1.505s + 2.244) (s^2 + 0.7312s + 3.455)

Continuous-time zero/pole/gain model.
```

```
G4_4=zpk(G4)
```

```
G4_4 =

          10 (s+5)
  --------------------------
  (s-3) (s+1) (s^2 + 3s + 6)

Continuous-time zero/pole/gain model.
```

```
G5_5=zpk(G5)
```

```
G5_5 =

             (s^2 + 3)
  ---------------------------------
```

```
  (s+1) (s^2 + s + 1) (s^2 - s + 3)

Continuous-time zero/pole/gain model.
```

2. Convierte a forma polinómica las funciones de transferencia G6, G7, G8 y G9 definidas en los ejercicios prácticos 1 y 2.

```
G6_6=tf(G6)
```

```
G6_6 =
 
          2 s + 2
  -----------------------
  s^3 + 9 s^2 + 26 s + 24
 
Continuous-time transfer function.
```

```
G7_7=tf(G7)
```

```
G7_7 =
 
          5 s - 5
  -----------------------
  s^3 + 3 s^2 - 10 s - 24
 
Continuous-time transfer function.
```

```
G8_8=tf(G8)
```

```
G8_8 =
 
            1
  ---------------------
  s^3 + 3 s^2 + 3 s + 1
 
Continuous-time transfer function.
```

```
G9_9=tf(G9)
```

```
G9_9 =
 
            1
  ---------------------
  s^3 + 3 s^2 + 7 s + 5
 
Continuous-time transfer function.
```

1.4 Forma simplificada (uso de s=tf('s'))

```
s=tf('s');
G=(s+1)/(s^3+3*s+1)
```

```
G =
 
      s + 1
  -------------
  s^3 + 3 s + 1
 
Continuous-time transfer function.
```

Ejercicio práctico 5. Uso de s = tf('s')

1. Utilizado este método simplificado para definir una función de transferencia del ejercicio práctico 1, las dos funciones del ejercicio práctico 2 y una función del ejercicio práctico 3.

```
G1_2=1/(s^2+1)
G4_2=10*(s+5)/((s+1)*(s-3)*(s^2+3*s+6))
G5_2=(s^2+3)/((s^2+s+1)*(s^3+2*s+3))
G8_2=1/(s+1)^3
```

2. Transformadas inversas de Laplace en MATLAB

2.1 Descomposición (expansión en fracciones simples)

Ejercicio práctico 6. Descomposición en fracciones simples

1. Calcula los residuos y los polos de las descomposición en fracciones simples de las siguientes funciones Y(s):

Y1 () = 32 / s^3+12s^2+32s

```
[r1,p1,k1]=residue(32,[1 12 32 0])
```

```
r1 = 3×1
     1
    -2
     1
p1 = 3×1
    -8
    -4
     0
k1 =

     []
```

2 () = $6s^2+22s+18$ / $s^3+6s^2+11s+6$

```
[r2,p2,k2]=residue([6 22 18],[1 6 11 6])
```

```
r2 = 3×1
    3.0000
    2.0000
    1.0000
p2 = 3×1
   -3.0000
   -2.0000
   -1.0000
k2 =

     []
```

Y3 () = s+7 / $s^2+10s+25$

```
[r3,p3,k3]=residue([1 7],[1 12 32 0])
```

```
r3 = 3×1
   -0.0312
```

```
   -0.1875
    0.2188
p3 = 3×1
    -8
    -4
     0
k3 =

     []
```

Y4 () = $2s^3+25s^2+95s+110$ / s^3+9s^2+50

```
[r4,p4,k4]=residue([2 25 95 110],[1 12 32 0])
```

```
r4 = 3×1
   -2.3125
   -0.1250
    3.4375
p4 = 3×1
    -8
    -4
     0
k4 = 2
```

2. Una vez descompuestas, calcula sus transformadas inversas de Laplace (tabla).

Uso de residue en el sentido inverso:

```
[num,den]=residue([1 2],[-5 5],[])
```

2.2 Transformadas simbólicas usando el Symbolic Math Toolbox

```
syms s
% Entrada en escalón unitario
X=1/s;
G=1/(s^2+3*s+2);
Y=X*G;
y=ilaplace(Y)
```

y =

$$\frac{e^{-2t}}{2} - e^{-t} + \frac{1}{2}$$

Ejercicio práctico 7. Transformadas inversas de Laplace con MATLAB

1. Utilizando un LiveScript, calcula la respuesta en el dominio del tiempo de las siguientes funciones de transferencia ante una entrada escalón unitario:

```
syms s
X_escalon=1/s
```

X_escalon =

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

1() = 1/(s+1)

```
G1_escalon=1/(s+1);
Y1_escalon=X_escalon*G1_escalon;
y1_escalon=ilaplace(Y1_escalon)
```

y1_escalon = $1 - e^{-t}$

2() = 5/(s+1)(s+5)

```
G2_escalon=5/((s+1)*(s+5));
Y2_escalon=X_escalon*G2_escalon;
y2_escalon=ilaplace(Y1_escalon)
```

y2_escalon = $1 - e^{-t}$

3() = $1/(s+1)^3$

```
G3_escalon=1/((s+1)^3);
Y3_escalon=X_escalon*G3_escalon;
y3_escalon=ilaplace(Y3_escalon)
```

y3_escalon =

$$1 - t\,e^{-t} - \frac{t^2\,e^{-t}}{2} - e^{-t}$$

4() = $1/(s^2+s+1)$

```
G4_escalon=1/(s^2+s+1);
Y4_escalon=X_escalon*G4_escalon;
y4_escalon=ilaplace(Y4_escalon)
```

y4_escalon =

$$1 - e^{-\frac{t}{2}}\left(\cos\left(\frac{\sqrt{3}\,t}{2}\right) + \frac{\sqrt{3}\,\sin\left(\frac{\sqrt{3}\,t}{2}\right)}{3}\right)$$

2. Calcula también la respuesta en el dominio del tiempo de las funciones de transferencia anteriores ante una entrada rampa unitaria.

```
X_rampa=1/s^2
```

X_rampa =

$$\frac{1}{s^2}$$

1() = 1/(s+1)

```
G1_rampa=1/(s+1);
Y1_rampa=X_rampa*G1_rampa;
```

```
y1_rampa=ilaplace(Y1_rampa)
```

y1_rampa = $t + e^{-t} - 1$

2() = 5/(s+1)(s+5)

```
G2_rampa=5/((s+1)*(s+5));
Y2_rampa=X_rampa*G2_rampa;
y2_rampa=ilaplace(Y2_rampa)
```

y2_rampa =

$$t + \frac{5\,e^{-t}}{4} - \frac{e^{-5\,t}}{20} - \frac{6}{5}$$

3() = 1/(s+1)³

```
G3_rampa=1/((s+1)^3);
Y3_rampa=X_rampa*G3_rampa;
y3_rampa=ilaplace(Y3_rampa)
```

y3_rampa =

$$t + 3\,e^{-t} + 2\,t\,e^{-t} + \frac{t^2\,e^{-t}}{2} - 3$$

4() = 1/(s²+s+1)

```
G4_rampa=1/(s^2+s+1);
Y4_rampa=X_rampa*G4_rampa;
y4_rampa=ilaplace(Y4_rampa)
```

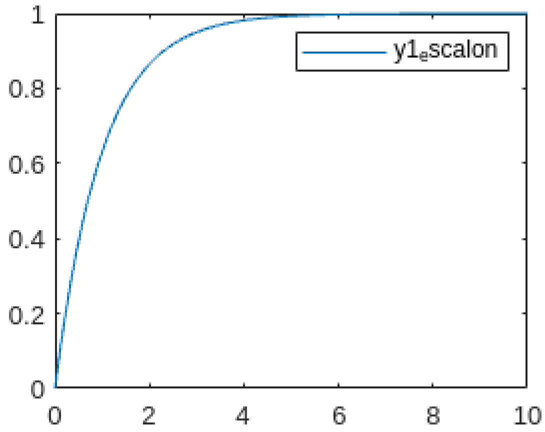
y4_rampa =

$$t + e^{-\frac{t}{2}}\left(\cos\left(\frac{\sqrt{3}\,t}{2}\right) - \frac{\sqrt{3}\,\sin\left(\frac{\sqrt{3}\,t}{2}\right)}{3}\right) - 1$$

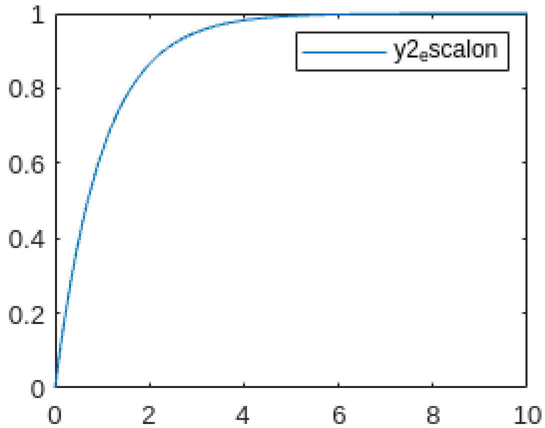
Ejercicio práctico 8. Gráficas de funciones simbólicas

1. Añade el comando fplot al LiveScript del ejercicio práctico 7 para realizar la gráfica, por separado, de cada una de las transformadas inversas de Laplace que has ido calculando, en el intervalo [0, 10].

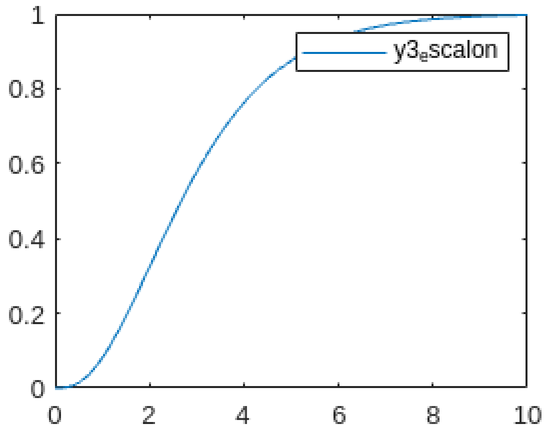
```
syms s
fplot([y1_escalon],[0,10])
legend('y1_escalon')
```



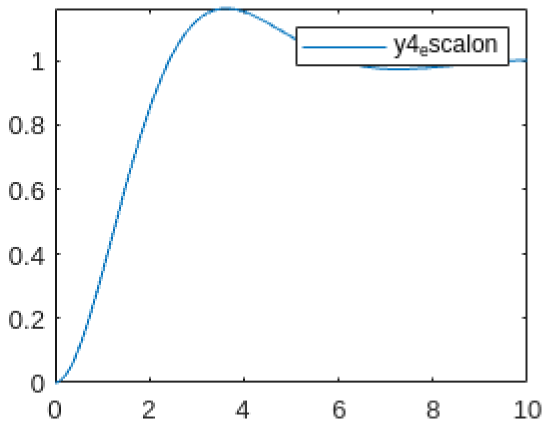

```
fplot([y2_escalon],[0,10])
legend('y2_escalon')
```



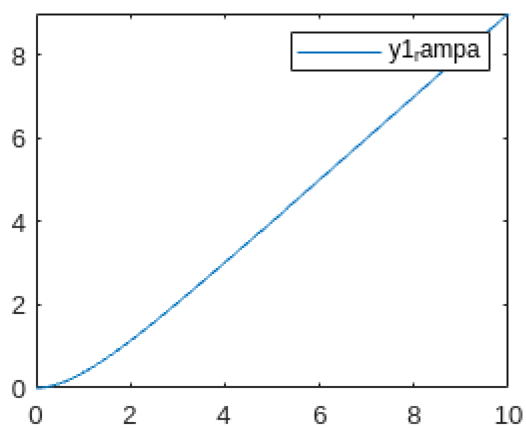

```
fplot([y3_escalon],[0,10])
legend('y3_escalon')
```



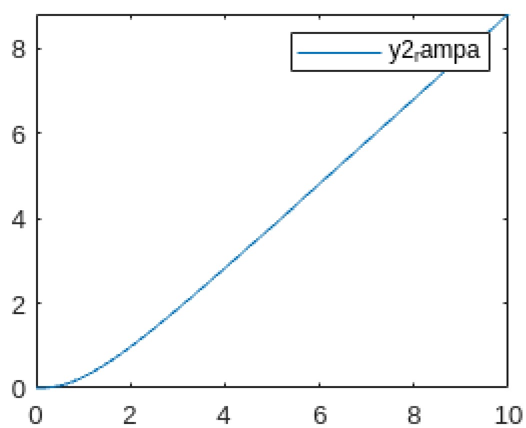

```
fplot([y4_escalon],[0,10])
legend('y4_escalon')
```



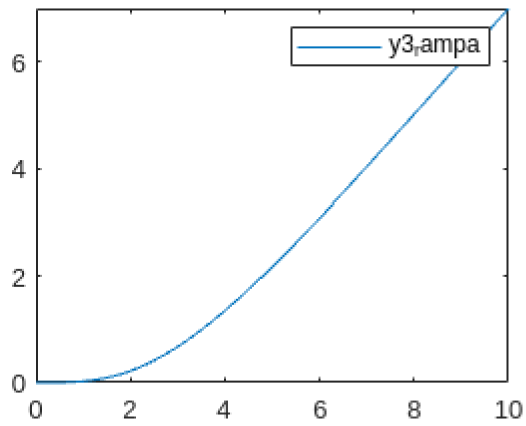

```
fplot([y1_rampa],[0,10])
legend('y1_rampa')
```



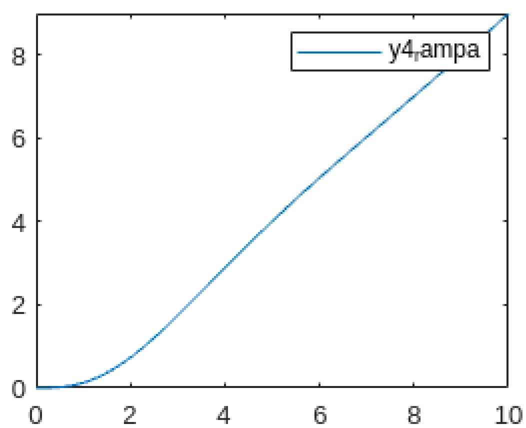

```
fplot([y2_rampa],[0,10])
legend('y2_rampa')
```




```
fplot([y3_rampa],[0,10])
legend('y3_rampa')
```




```
fplot([y4_rampa],[0,10])
legend('y4_rampa')
```



2. Realiza una única gráfica en la que aparezcan las respuestas ante una entrada escalón de las tres funciones de transferencia siguientes. ¿Observas alguna relación entre ellas?

G1() = 1/(s+1)

G2() = 5/(s+5)

G3() = 5/(s+1)(s+5)

```
X=1/s
```

X =

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

```
G1=1/(s+1);
Y1=G1*X;
y1=ilaplace(Y1)
```

y1 = $1 - e^{-t}$

```
G2=5/(s+5);
Y2=G2*X;
y2=ilaplace(Y2)
```

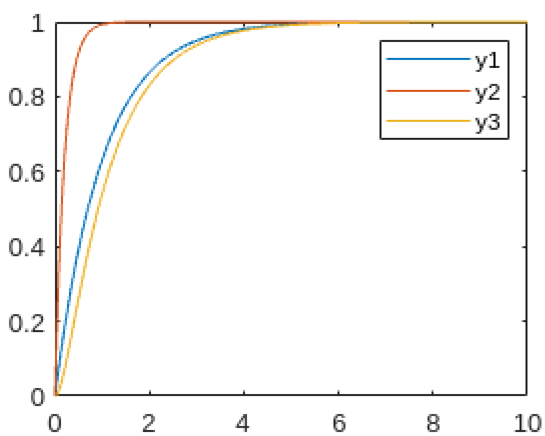
y2 = $1 - e^{-5\,t}$

```
G3=5/((s+1)*(s+5));
Y3=G3*X;
y3=ilaplace(Y3)
```

y3 =

$$\frac{e^{-5\,t}}{4} - \frac{5\,e^{-t}}{4} + 1$$

```
fplot([y1,y2,y3],[0,10])
legend('y1','y2','y3')
```



Ejercicio práctico 9. Transformadas de Laplace con MATLAB

1. Calcula las transformadas de Laplace de las siguientes funciones del tiempo:

```
syms t
```

1 () = t^2

```
y1=t^2;
Y1=laplace(y1)
```

Y1 =

$$\frac{2}{s^3}$$

2 () = cos(t)

```
y2=cos(t);
Y2=laplace(y2)
```

Y2 =

$$\frac{s}{s^2 + 1}$$

3 () = sin(2t)

```
y3=sin(2*t);
Y3=laplace(y3)
```

Y3 =

$$\frac{2}{s^2 + 4}$$

4 () = ^-2t*cos(t)

```
y4=exp(-2*t)*cos(t);
Y4=laplace(y4)
```

Y4 =

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

5 () = t²*e^-2t*sin(t)

```
y5=t^2*exp(-2*t)*sin(t);
Y5=laplace(y5)
```

Y5 =

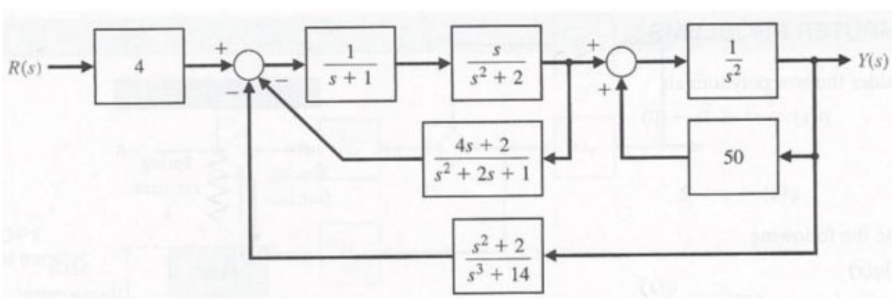
$$\frac{2\ (2\,s + 4)^2}{((s + 2)^2 + 1)^3} - \frac{2}{((s + 2)^2 + 1)^2}$$

3. Álgebra de bloques mediante MATLAB

3.1 Herramientas para la interconexión de diagramas de bloques

Ejercicio práctico 10. Reducción de diagramas de bloques

1. Utiliza los comandos de reducción de diagramas de bloques para calcular la función de transferencia en lazo cerrado del siguiente sistema de control:




```
syms s
% Definición de bloques
```

```
G1 = 4;
G2 = tf([1],[1 1]);
G3 = tf([1 0],[1 0 2]);
G4 = tf([1],[1 0 0]);
G5 = tf([4 2],[1 2 1]);
G6 = 50;
G7 = tf([1 0 2],[1 0 0 14]);
% Aplicación del álgebra de bloques
G23s = series(G2,G3);
G235f = feedback(G23s,G5);
G46f = feedback(G4,G6,1);
G23456s = series(G235f,G46f);
G234567f = feedback(G23456s,G7);
TFlc = series(G1,G234567f)
```

```
TFlc =

                             4 s^6 + 8 s^5 + 4 s^4 + 56 s^3 + 112 s^2 + 56 s
  ----------------------------------------------------------------------------------------------------
  s^10 + 3 s^9 - 45 s^8 - 125 s^7 - 200 s^6 - 1177 s^5 - 2344 s^4 - 3485 s^3 - 7668 s^2 - 5598 s - 1400

Continuous-time transfer function.
```