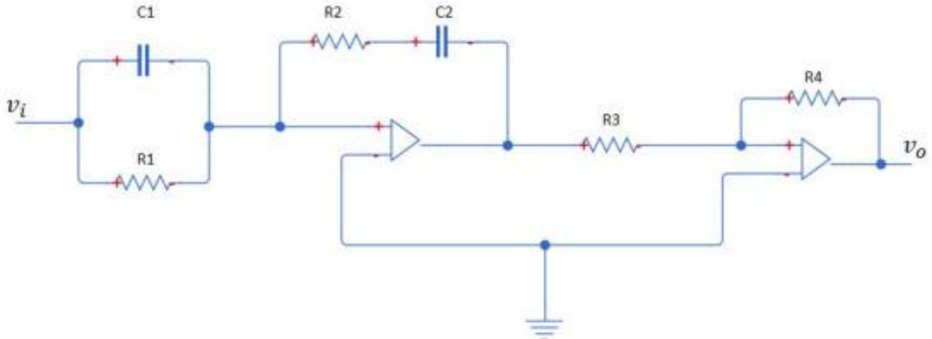
Práctica 2

Modelado y Simulación con Simulink

Juan Efraín Sánchez Díaz

P1. Para el circuito de la figura:



a) obtén la función de transferencia $G_c(s) = v_o(s)/v_i(s)$. (entregar desarrollo a mano)

b) Para valores $R_i = 50$ y $C_i = 0.01$ representa en Matlab la función de transferencia en formato `tf` y `zpk` de la forma más simplificada posible (comando minreal).

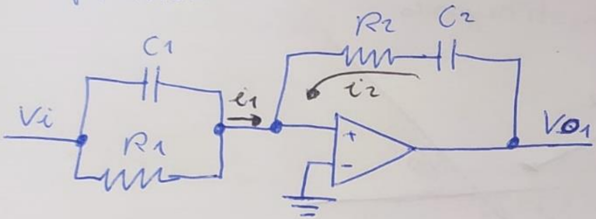
c) Considera la función de transferencia $G_p(s) = \frac{1}{s^2+2s+2}$ en serie con $G_c(s)$ y en bucle cerrado con realimentación unitaria y negativa. Obtén la función de transferencia resultante ($G_{total}(s)$ en formato tf) y la respuesta temporal ante entrada escalón unitario de $G_{total}(s)$ durante 10 segundos.

a) Resultado a mano obtenido de Gc(s)

P1. a)

- Primero dividimos el sistema en 2:

- Primera Parte:



$$\boxed{i_1 + i_2 = 0}$$

$$\boxed{i_1 = \frac{V_i}{R_1} + C_1 \cdot \frac{\partial V_i}{\partial t}}$$

$$\boxed{V_{o1} = \frac{1}{C_2}\int i_2 \, dt + i_2 \cdot R_2}$$

Antes de pasar a $\mathcal{L}$ eliminamos la integral derivando

$$V_{o1} = \frac{1}{C_2} \cdot i_2 + i_2 \cdot R_2$$

- Pasamos las 3 ecuaciones a Laplace

$$\begin{cases} I_1 = -I_2 \\ I_1 = \dfrac{V_i}{R_1} + C_1 \cdot s \cdot V_i \\ V_{o1} \cdot s = \dfrac{1}{C_2} \cdot I_2 + I_2 \cdot s \cdot R_2 \xrightarrow[a]{\text{es equivalente}} V_{o1} = \dfrac{1}{C_2 \cdot s} \cdot I_2 + I_2 \cdot R_2 \end{cases}$$

- 1º Forma de Resolverlo: Sustituyendo

$$I_1 = \frac{V_i}{R_1} + C_1 \cdot s \cdot V_i \longrightarrow I_1 = \frac{1 + R_1 \cdot C_1 \cdot s}{R_1} \cdot V_i$$

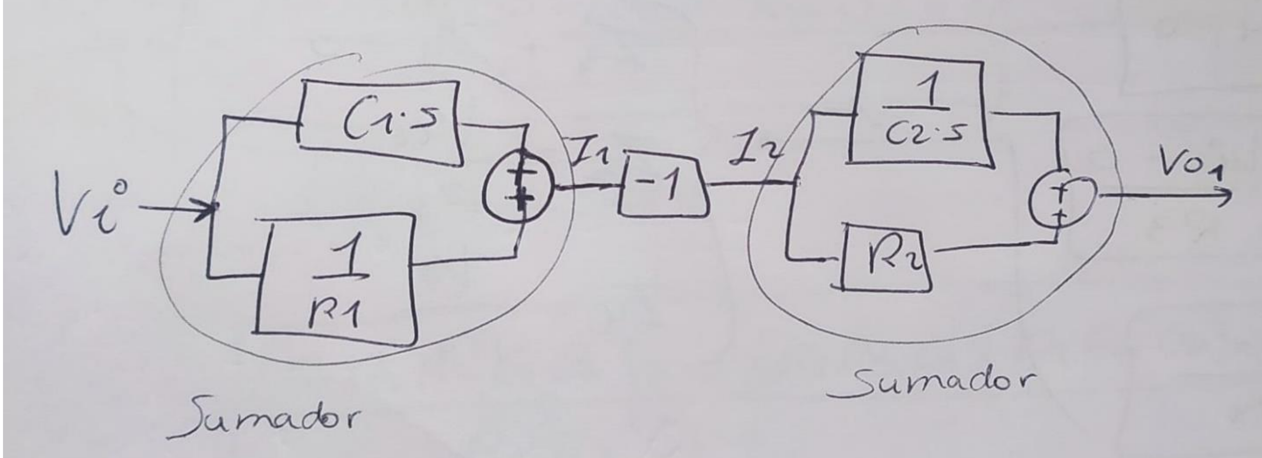
$$I_2 = \frac{V_{o1}}{\left(\frac{1 + C_2 \cdot R_2 \cdot s}{C_2 \cdot s}\right)}$$

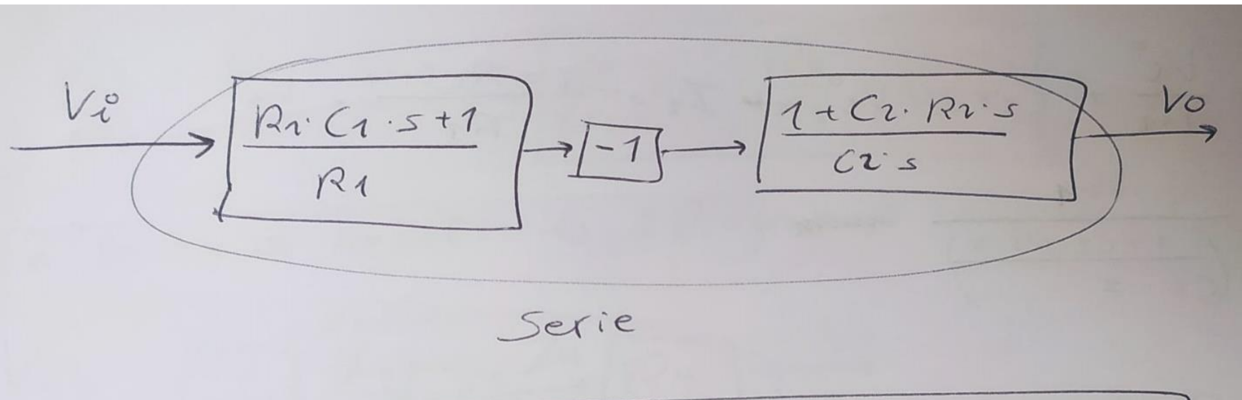
$$\frac{1 + R_1 \cdot C_1 \cdot s}{R_1} \cdot V_i = -\frac{V_{o1}}{\frac{1 + C_2 \cdot R_2 \cdot s}{C_2 \cdot s}}$$

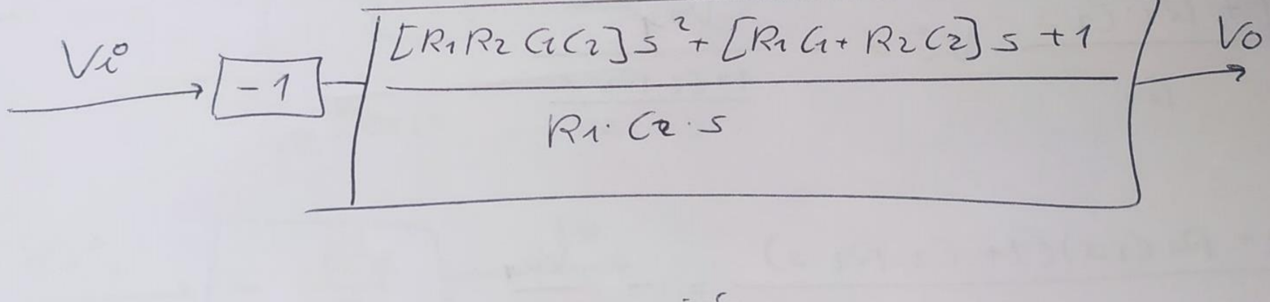
$$\frac{(1 + R_1 C_1 s)(1 + C_2 \cdot R_2 \cdot s)}{R_1 \cdot C_2 \cdot s} = -\frac{V_{o1}}{V_i}$$

$$\boxed{\frac{V_{o1}}{V_i} = G_{C1} = -\left[\frac{(R_1 R_2 C_1 C_2) s^2 + [R_1 (1 + C_2 R_2)] s + 1}{R_1 \cdot C_2 \cdot s}\right]}$$

• 2º Forma de Resolverlo: Diagrama de Bloques

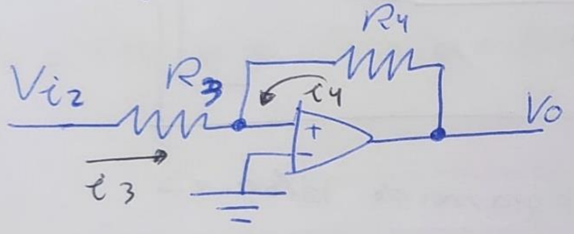






• obtenemos la misma ecuación

- Segunda Parte:

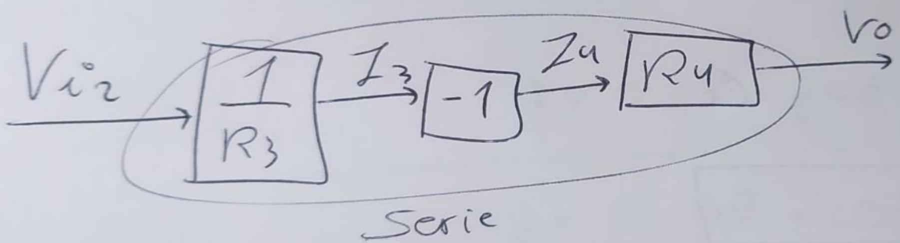


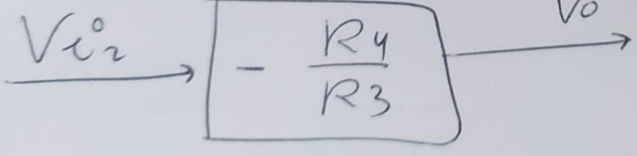
$$\left.\begin{array}{l}\boxed{i_3 + i_4 = 0} \\ \boxed{i_3 = \dfrac{V_{i2} - 0}{R_3}} \\ \boxed{i_4 = \dfrac{V_o - 0}{R_4}}\end{array}\right\} \xrightarrow{\mathcal{L}} \left\{\begin{array}{l} I_3 + I_4 = 0 \\ I_3 = \dfrac{V_{i2}}{R_3} \\ I_4 = \dfrac{V_o}{R_4}\end{array}\right.$$

- 1º Forma de Resolverlo: sustituyendo

$$\frac{V_{i2}}{R_3} + \frac{V_o}{R_4} = 0 \rightarrow \boxed{\frac{V_o}{V_{i2}} = -\frac{R_4}{R_3} = G_{c2}}$$

- 2º Forma de Resolverlo: Diagrama de Bloques





- obtenemos la misma ecuación

- Por último unimos ambas partes

son lo mismo

$$V_i \rightarrow \boxed{-\frac{[R_1 R_2 C_1 C_2]s^2 + [R_1 C_1 + R_2 C_2]s + 1}{R_1 \cdot C_2 \cdot s}} \xrightarrow{V_{o1} = V_{i2}} \boxed{-\frac{R_4}{R_3}} \xrightarrow{V_o}$$

Serie

$$V_i \rightarrow \boxed{\frac{[R_1 R_2 R_4 C_1 C_2]s^2 + [R_1 R_4 C_1 + R_2 R_4 C_2]s + R_4}{R_1 R_3 C_2 \cdot s}} \xrightarrow{V_o}$$

- Por lo tanto:

$$\boxed{G_C(s) = \frac{V_o(s)}{V_i(s)} = \frac{[R_1 R_2 R_4 C_1 C_2]s^2 + [R_1 R_4 C_1 + R_2 R_4 C_2]s + R_4}{R_1 R_3 C_2 s}}$$

```
% Práctica 2
```

b) Para valores $R_i = 50$ y $C_i = 0.01$ representa en Matlab la función de transferencia enformato `tf` y `zpk` de la forma más simplificada posible (comando minreal).

$$\boxed{G_c(s) = \frac{12'5\, s^2 + 50 s + 50}{25 s}}$$

```
% Representación en formato tf ----------------%
Gctf = tf([12.5 50 50],[25 0])
```

```
Gctf =

  12.5 s^2 + 50 s + 50
  --------------------
          25 s

Continuous-time transfer function.
```

```
minreal(Gctf)
```

```
ans =

  0.5 s^2 + 2 s + 2
  -----------------
          s

Continuous-time transfer function.
```

```
% Representación en formato zpk ----------------%

Gczpk = zpk([-2 -2], 0, 0.5)
```

```
Gczpk =

  0.5 (s+2)^2
  -----------
       s

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

```
minreal(Gczpk)
```

```
ans =

  0.5 (s+2)^2
  -----------
       s

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

c) Considera la función de transferencia $G_p(s) = \frac{1}{s^2+2s+2}$ en serie con $G_c(s)$ y en bucle cerrado con realimentación unitaria y negativa. Obtén la función de transferencia resultante ($G_{total}(s)$ en formato tf) y la respuesta temporal ante entrada escalón unitario de $G_{total}(s)$ durante 10 segundos.

```
% Creamos Gp en formato tf
Gp = tf(1,[1 2 2])
```

```
Gp =

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

Continuous-time transfer function.
```

```
% Creamos Gc en formato tf
Gc = minreal(Gctf)
```

```
Gc =

  0.5 s^2 + 2 s + 2
  -----------------
          s

Continuous-time transfer function.
```

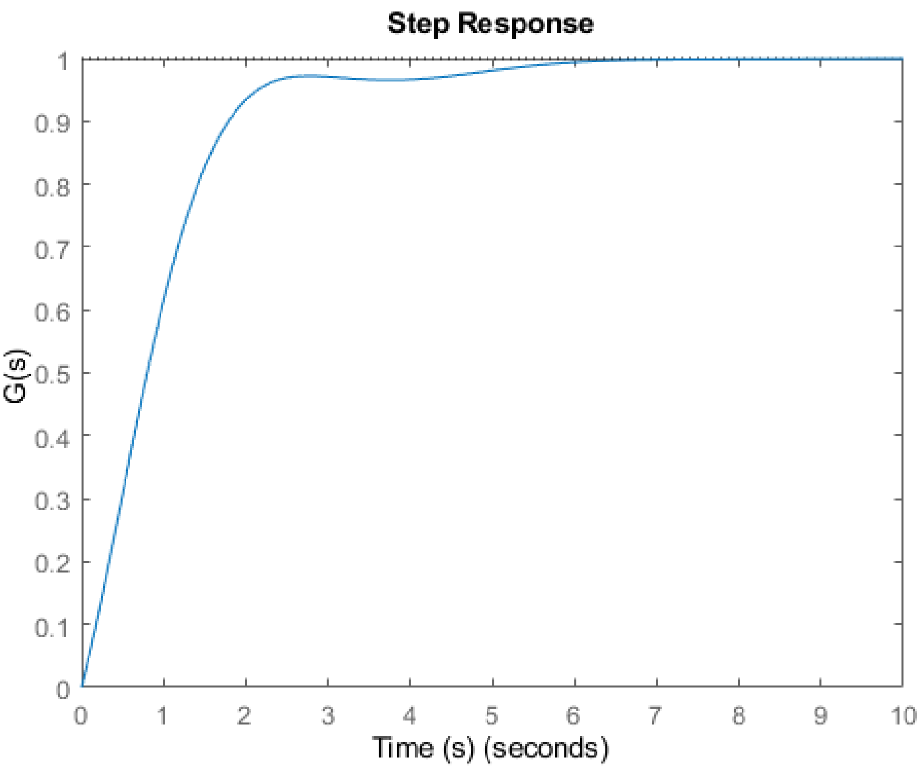
```
% Hacemos el serie de ambos
Gi = series(Gp,Gc)
```

```
Gi =

  0.5 s^2 + 2 s + 2
  -----------------
  s^3 + 2 s^2 + 2 s

Continuous-time transfer function.
```

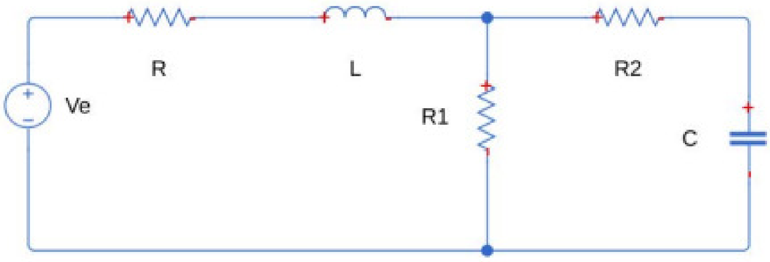
```
% Por último hacemos la retroalimentación negativa
Gtotal = minreal(feedback(Gi,1))
```

```
Gtotal =

     0.5 s^2 + 2 s + 2
  -----------------------
  s^3 + 2.5 s^2 + 4 s + 2

Continuous-time transfer function.
```

```
% Representamos la respuesta temporal ante escalon unitario durante 10
% segundos
figure();
step(Gtotal,10);
xlabel('Time (s)');
ylabel('G(s)');
```



P2. Para el circuito eléctrico de la figura:



a) obtén su función de transferencia en formato tf empleando variables simbólicas y resolución de sistemas de ecuaciones con Matlab. (recuerda usar el comando simplify).

b) posteriormente, obtén su diagrama de bloques y redúcelo mediante manipulación de bloques (será necesario entregar el desarrollo completo a mano).

c) finalmente, utilizando comandos Matlab (series, feedback, parallel, o connect y sumblk) comprueba el resultado del apartado b.

Nota-> Para la reducción del diagrama de bloques (apartados b y c), considera que las resistencias tienen un valor de 1KΩ, L=1H, C=0.001F, la entrada es $Ve = 1V$ y la salida es la caída de tensión en los bornes del condensador.

a) Primero a mano obtenemos las ecuaciones y después obtenemos su función en Matlab y resolvemos el sistema de ecuaciones.

P2.

• Malla 1:

$$\boxed{① \; V_e = R\cdot i_1(t) + L\cdot \dot{i}_1(t) + R_1(i_1(t) - i_2(t))}$$

• Malla 2:

$$\boxed{② \; R_1(i_2(t) - i_1(t)) + R_2\cdot i_2(t) + V_c = 0}$$

• Salida:

$$\boxed{i_2(t) = C\cdot \dot{V}_c \; ③}$$

• Pasamos a Laplace:

$$\begin{cases} V_e = R\cdot I_1 + L\cdot s\cdot I_1 + R_1\cdot I_1 - R_1\cdot I_2 \\ R_1\cdot I_2 - R_1 I_1 + R_2\cdot I_2 + V_c = 0 \\ I_2 = C\cdot s\cdot V_c \end{cases}$$

• Incógnitas: I_1 I_2 V_c

• Datos: R R_1 R_2 C V_e

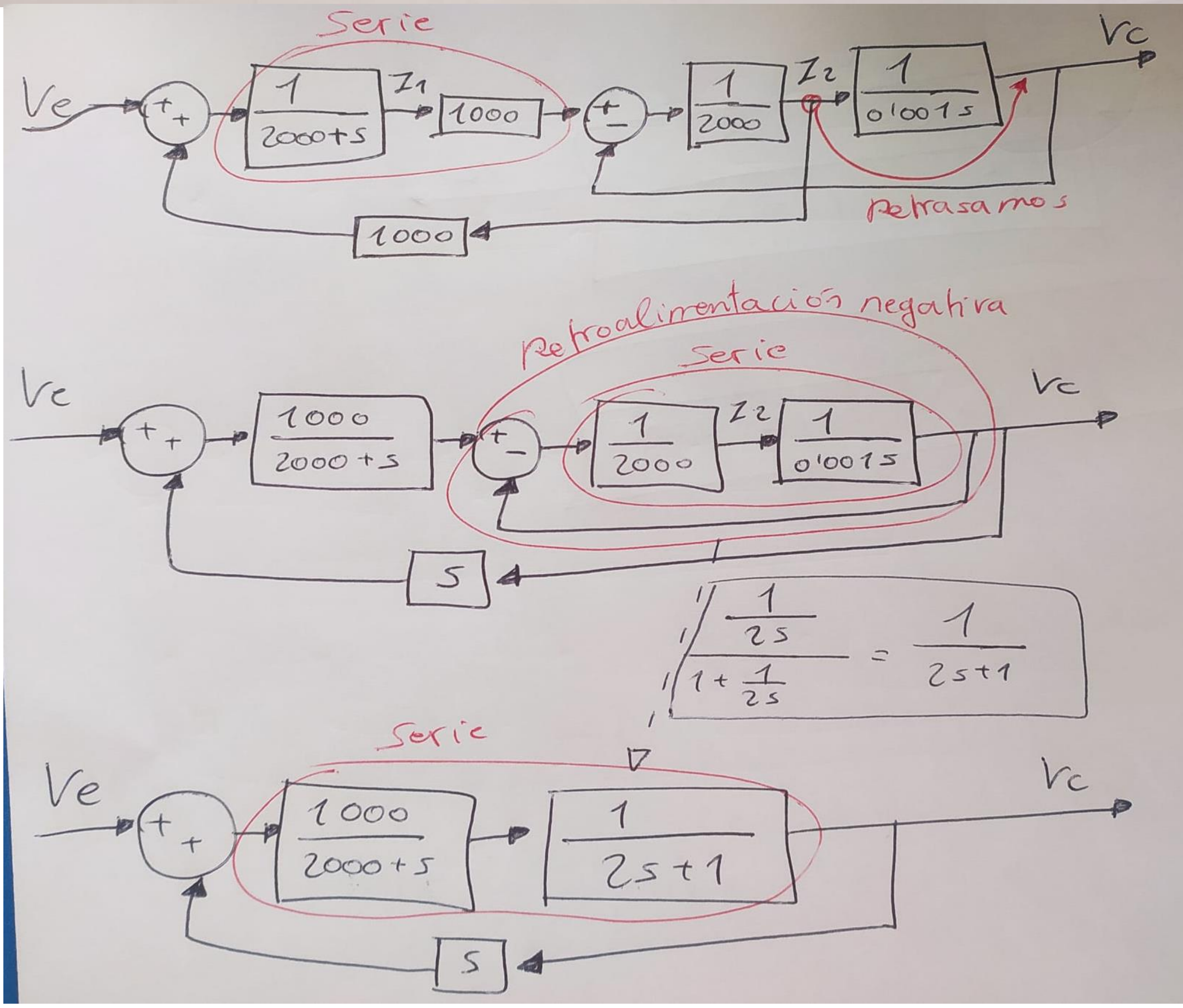
```
>> syms s;
>> syms R1 R2 R C L Ve;
>> A = [(R+L*s+R1) -R1 0; -R1 (R1+R2) 1; 0 -1 C*s]

A =

[R + R1 + L*s,      -R1,    0]
[         -R1, R1 + R2,     1]
[           0,       -1,  C*s]

>> b = [Ve; 0; 0]

b =

Ve
 0
 0

>> x = A\b;
>> pretty(x(3))
                                 R1 Ve
----------------------------------------------------------------------
                        2            2
R + R1 + L s + C L R1 s  + C L R2 s  + C R R1 s + C R R2 s + C R1 R2 s
```

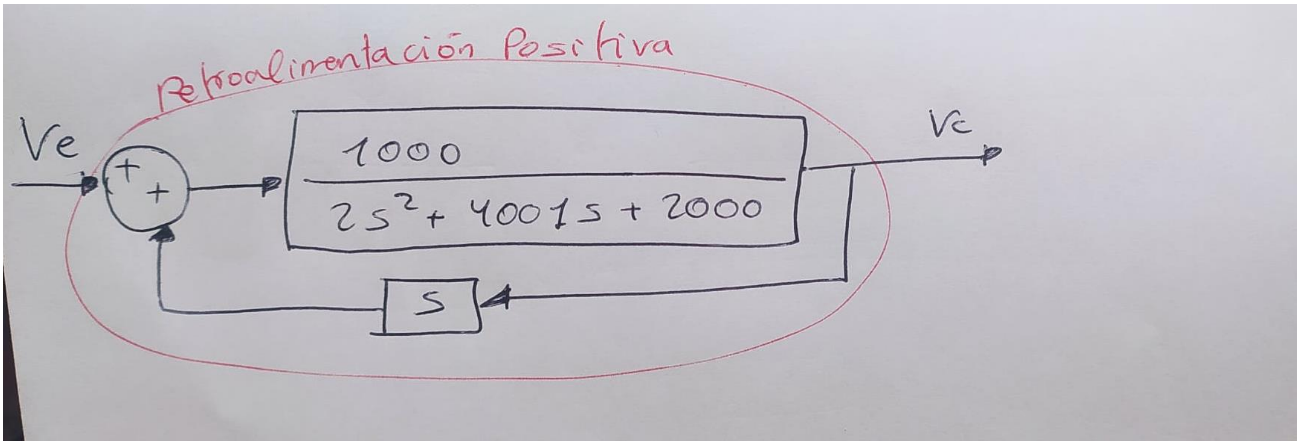
b) Creamos el diagrama de bloques usando:

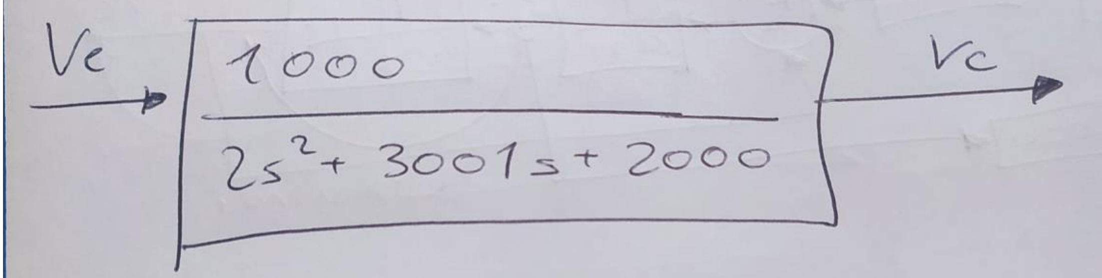
$$I_2 = \frac{R_1 \cdot I_1 - V_c}{R_1 + R_2} = \frac{1000 I_1 - V_c}{2000}$$

$$V_c = \frac{I_2}{C \cdot s} = \frac{I_2}{0'001 s}$$

$$I_1 = \frac{V_e + R_1 \cdot I_2}{R + Ls + R_1} = \frac{1 + 1000 I_2}{2000 + s}$$







$$\boxed{G(s) = \frac{1000}{2s^2 + 3001s + 2000}}$$

c) finalmente, utilizando comandos Matlab (series, feedback, parallel, o connect y sumblk) comprueba el resultado del apartado b.

```
% Primero definimos cada uno de los cuadros llamandolos Gi
G1 = tf(1,[1 2000])
```

```
G1 =

     1
  --------
  s + 2000

Continuous-time transfer function.
```

```
G2 = tf(1000)
```

```
G2 =

  1000

Static gain.
```

```
G3 = tf(1,2000)
```

```
G3 =

  0.0005

Static gain.
```

```
G4 = tf(1,[0.001 0])
```

```
G4 =

     1
  -------
  0.001 s

Continuous-time transfer function.
```

```
G5 = tf(1000)
```

```
G5 =

  1000

Static gain.
```

```
% Ahora aplicamos el serie y modificamos G5 ya que lo vamos a retrasar
a = series(G1,G2)
```

```
a =

    1000
  --------
  s + 2000

Continuous-time transfer function.
```

```
G5 = tf([1 0], 1)
```

```
G5 =

  s

Continuous-time transfer function.
```

```
% Aplicamos el serie entre G3 y G4 y la retroalimentación
b = series(G3,G4)
```

```
b =

   1
  ---
  2 s

Continuous-time transfer function.
```

```
c = feedback(b,1)
```

```
c =

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

Continuous-time transfer function.
```

```
% Hacemos series entre los dos resultado
d = series(a,c)
```

```
d =

          1000
  ---------------------
  2 s^2 + 4001 s + 2000

Continuous-time transfer function.
```

```
% Por último el feedback pero está es es positivo
Gs = feedback(d,G5,+1)
```

```
Gs =

          1000
  ---------------------
  2 s^2 + 3001 s + 2000

Continuous-time transfer function.
```

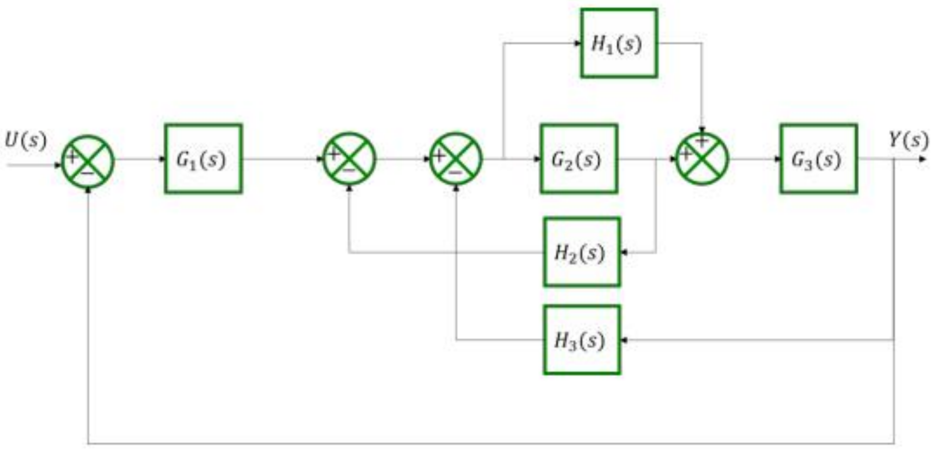
P3.- Dado el diagrama de la figura, obtén la función de transferencia $G(s) = Y(s)/U(s)$:

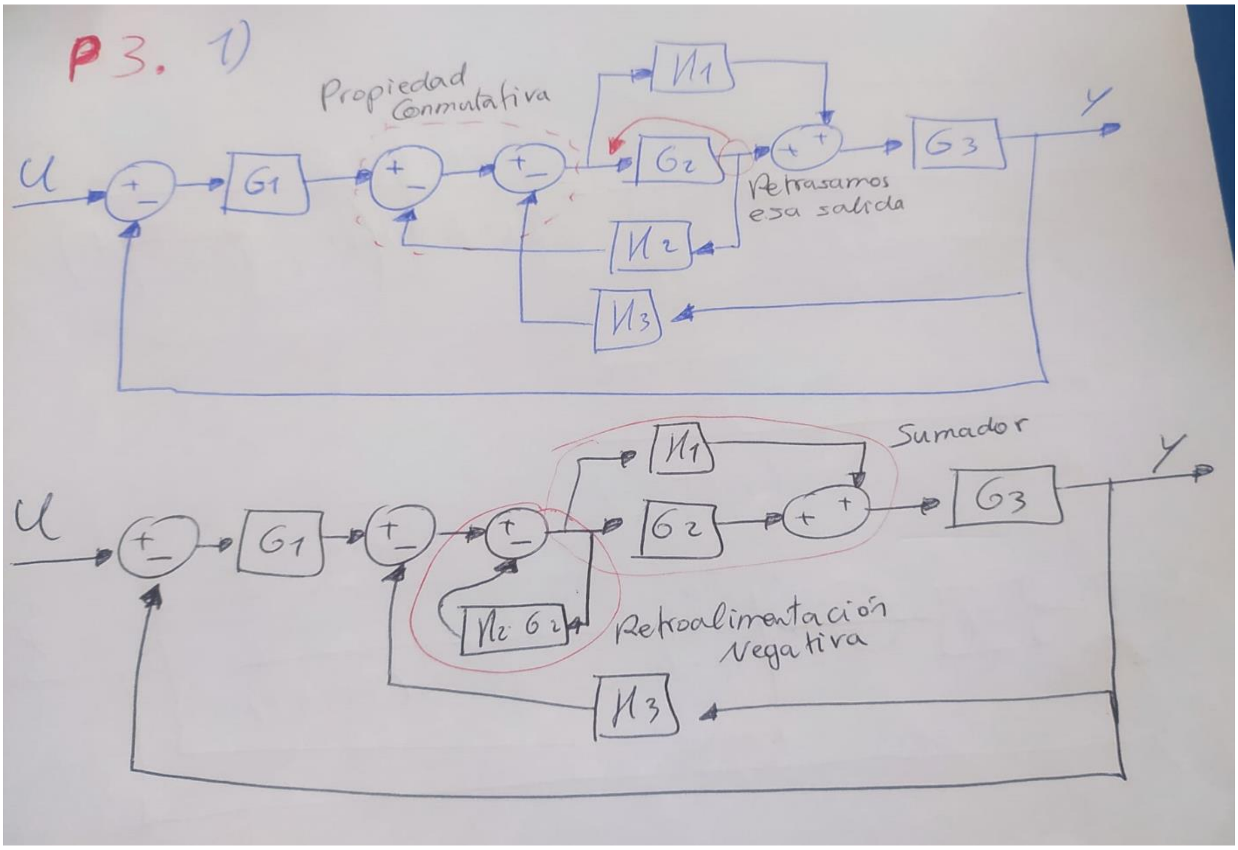
1) mediante manipulación de bloques (desarrollo a mano en función de Gi y Hi)

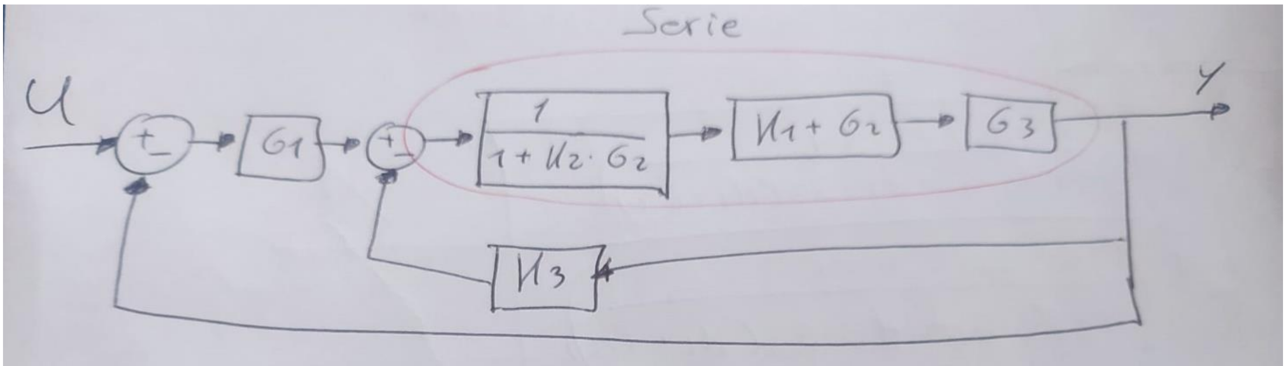
2) utilizando comandos (`series`, `feedback`, `parallel`)

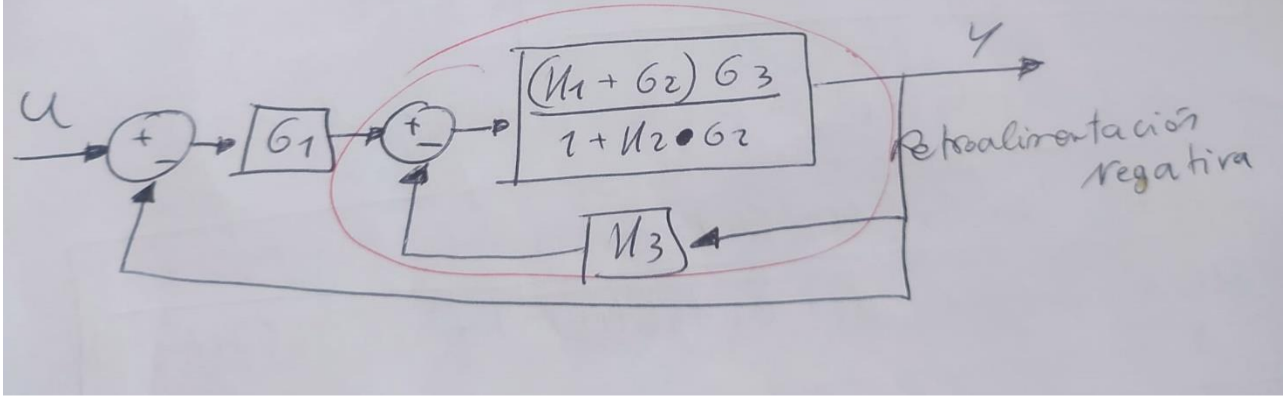
3) utilizando `connect` y `sumblk`

Nota: Para los apartados 2 y 3, considera que $G_i(s) = \frac{1}{s+i}$, $H_i(s) = \frac{1}{(s+i)^2}$





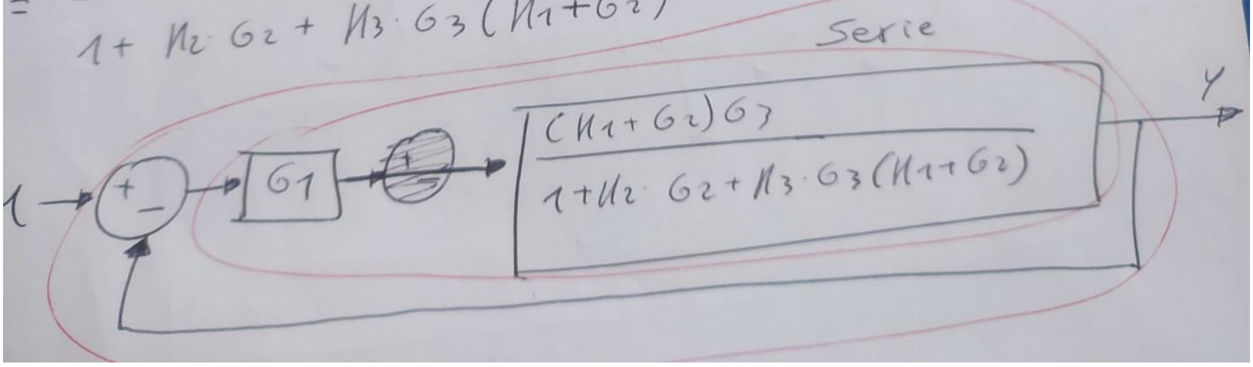




– Vamos a simplificarla:

$$\frac{\dfrac{(H_1+G_2)\,G_3}{1+H_2\cdot G_2}}{1+\dfrac{(H_1+G_2)\,G_3}{1+H_2\cdot G_2}\cdot H_3} = \frac{(H_1+G_2)\,G_3}{(1+H_2\cdot G_2)\left(1+\dfrac{(H_1+G_2)\,G_3}{1+H_2\cdot G_2}\cdot H_3\right)} =$$

$$= \frac{(H_1+G_2)\,G_3}{1+H_2\cdot G_2+H_3\cdot G_3\,(H_1+G_2)}$$



~~- simplifica~~

$$U \rightarrow \boxed{\dfrac{\left(\dfrac{(H_1+G_2)G_3\cdot G_1}{1+H_2\cdot G_2+H_3\cdot G_3(H_1+G_2)}\right)}{1+\left(\dfrac{(H_1+G_2)G_3\cdot G_1}{1+H_2\cdot G_2+H_3\cdot G_3(H_1+G_2)}\right)}} \rightarrow Y$$

- simplificando : De la misma forma que el anterior

$$\boxed{G(s)=\frac{Y(s)}{U(s)}=\frac{G_1\cdot G_3(H_1+G_2)}{1+G_2\cdot H_2+G_3H_3(H_1+G_2)+G_1\cdot G_3(H_1+G_2)}}$$

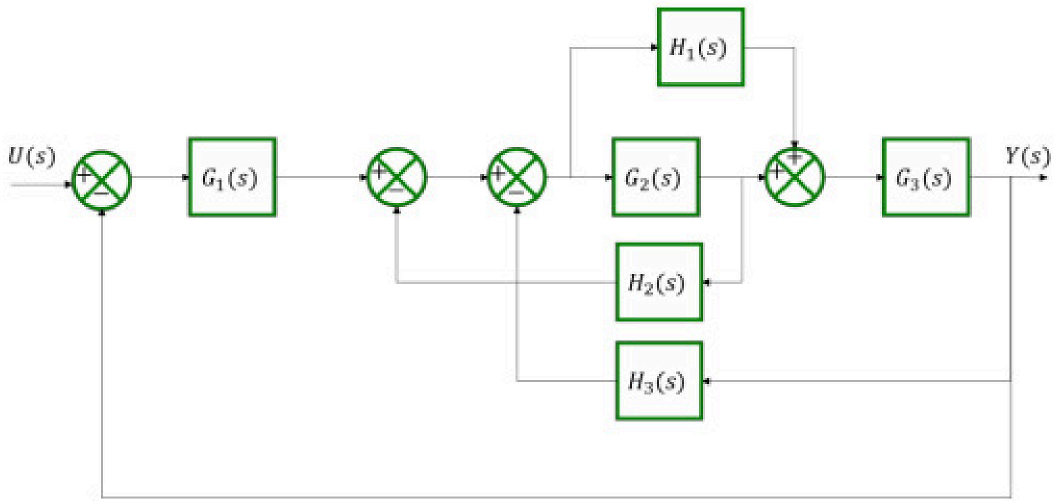
P3.- Dado el diagrama de la figura, obtén la función de transferencia $G(s) = Y(s)/U(s)$:

1) mediante manipulación de bloques (desarrollo a mano en función de Gi y Hi)

2) utilizando comandos (`series, feedback, parallel`)

3) utilizando `connect` y `sumblk`

Nota: Para los apartados 2 y 3, considera que $G_i(s) = \frac{1}{s+i}$, $H_i(s) = \frac{1}{(s+i)^2}$




```
% Comenzamos creando cada uno de las funciones
H1 = tf(1,[1 2 1])
```

```
H1 =

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

Continuous-time transfer function.
```

```
H2 = tf(1,[1 4 4])
```

```
H2 =

        1
  -------------
  s^2 + 4 s + 4

Continuous-time transfer function.
```

```
H3 = tf(1,[1 6 9])
```

```
H3 =
```

```
        1
  -------------
  s^2 + 6 s + 9

Continuous-time transfer function.
```

```
G1 = tf(1,[1 1])
```

```
G1 =

    1
  -----
  s + 1

Continuous-time transfer function.
```

```
G2 = tf(1,[1 2])
```

```
G2 =

    1
  -----
  s + 2

Continuous-time transfer function.
```

```
G3 =tf(1,[1 3])
```

```
G3 =

    1
  -----
  s + 3

Continuous-time transfer function.
```

```
% Apartado 2)
% Seguimos los mismos pasos que hice en el Apartado 1 después de retrasar
% el sumador y de aplicar la conmutatividad
```

```
% Hacemos el sumador/paralelo entre H1 y G2
a = parallel(H1,G2)
```

```
a =

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

Continuous-time transfer function.
```

```
% Y la retroalimentación negativa a H2*G2 con uno
aux = H2*G2;
b = feedback(1,aux)
```

```
b =
```

```
  s^3 + 6 s^2 + 12 s + 8
  ----------------------
  s^3 + 6 s^2 + 12 s + 9

Continuous-time transfer function.
```

```
% Hacemos el serie de los 3 bloques
aux = series(a,b);
c = series(aux,G3)
```

```
c =

             s^5 + 9 s^4 + 33 s^3 + 62 s^2 + 60 s + 24
  -----------------------------------------------------------------
  s^7 + 13 s^6 + 71 s^5 + 212 s^4 + 375 s^3 + 393 s^2 + 225 s + 54

Continuous-time transfer function.
```

```
% Hacemos la retroalimentación negativa con H3
d = feedback(c,H3)
```

```
d =

               s^7 + 15 s^6 + 96 s^5 + 341 s^4 + 729 s^3 + 942 s^2 + 684 s + 216
  ------------------------------------------------------------------------------------------
  s^9 + 19 s^8 + 158 s^7 + 755 s^6 + 2287 s^5 + 4560 s^4 + 5991 s^3 + 5003 s^2 + 2409 s + 510

Continuous-time transfer function.
```

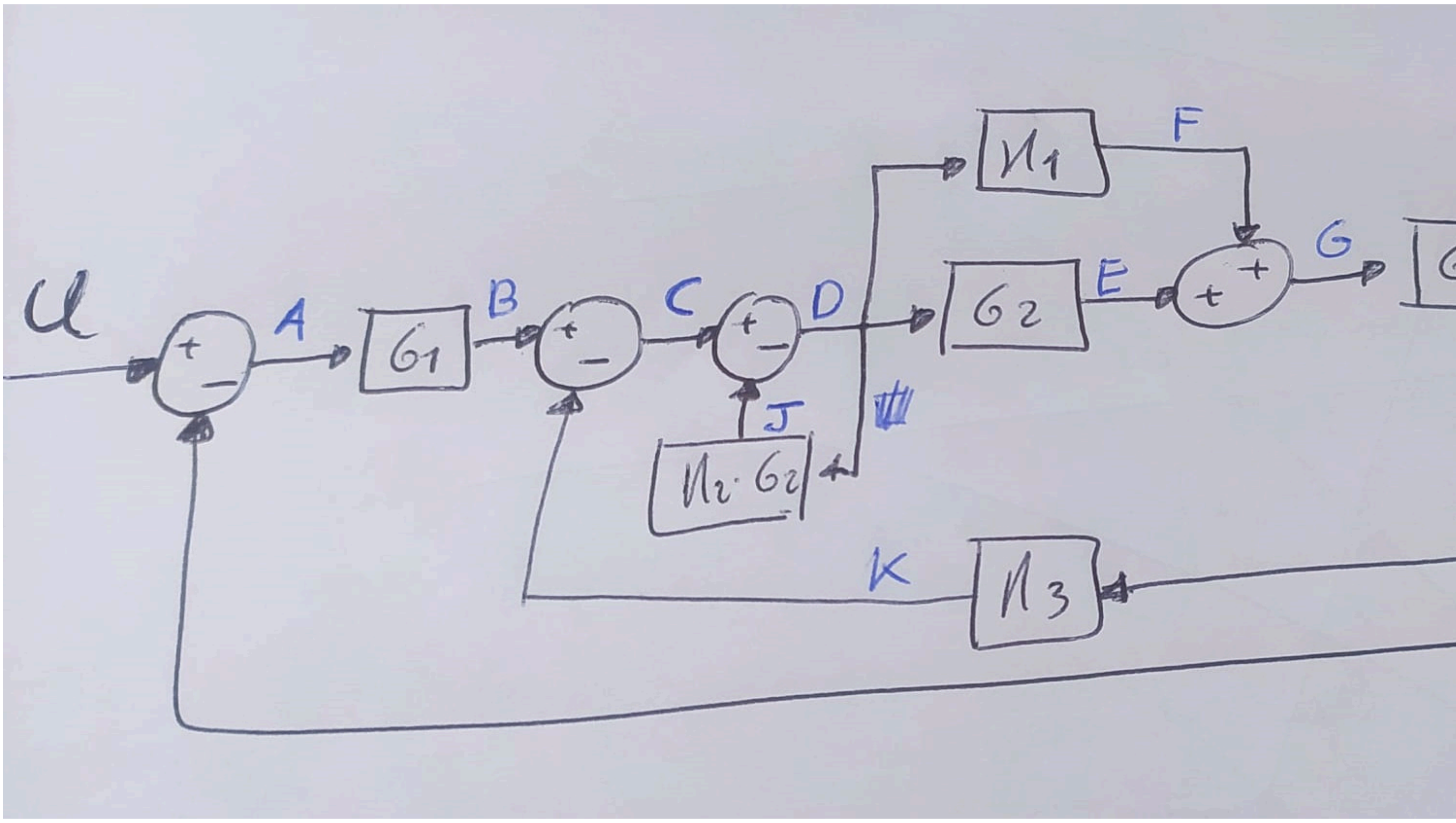
```
% Hacemos la Serie con G1
e = series(G1,d)
```

```
e =

                      s^7 + 15 s^6 + 96 s^5 + 341 s^4 + 729 s^3 + 942 s^2 + 684 s + 216
  --------------------------------------------------------------------------------------------------------
  s^10 + 20 s^9 + 177 s^8 + 913 s^7 + 3042 s^6 + 6847 s^5 + 10551 s^4 + 10994 s^3 + 7412 s^2 + 2919 s + 510

Continuous-time transfer function.
```

```
% Por último la retroalimentación unitaria
G = minreal(feedback(e,1))
```

```
G =

                  s^5 + 12 s^4 + 57 s^3 + 134 s^2 + 156 s + 72
  ---------------------------------------------------------------------------
  s^8 + 17 s^7 + 123 s^6 + 494 s^5 + 1206 s^4 + 1843 s^3 + 1745 s^2 + 959 s + 242

Continuous-time transfer function.
```

```
% Apartado 3)
% Le damos nombre a cada uno de las partes
```




```
% Definimos los sumadores
Sum1 = sumblk('A = U - Y');
Sum2 = sumblk('C = B - K');
Sum3 = sumblk('D = C - J');
Sum4 = sumblk('G = E + F');
```

```
% Ahora definimos los bloques
G1 = tf(1,[1 1]);        G1.u = 'A';      G1.y = 'B';
G2 = tf(1,[1 2]);        G2.u = 'D';      G2.y = 'E';
G3 =tf(1,[1 3]);         G3.u = 'G';      G3.y = 'Y';
H1 = tf(1,[1 2 1]);      H1.u = 'D';      H1.y = 'F';
H2G2 = tf(1,[1 6 12 8]);H2G2.u = 'D';     H2G2.y = 'J';
H3 = tf(1,[1 6 9]);      H3.u = 'Y';      H3.y = 'K';
```

```
% Ahora conectamos todos y sacamos la función de transferencia
T = connect(G1,G2,G3,H1,H2G2,H3, Sum1, Sum2, Sum3, Sum4, 'U', 'Y');
G = minreal(tf(T))
```

```
G =

  From input "U" to output "Y":
                 s^5 + 12 s^4 + 57 s^3 + 134 s^2 + 156 s + 72
  -------------------------------------------------------------------------
  s^8 + 17 s^7 + 123 s^6 + 494 s^5 + 1206 s^4 + 1843 s^3 + 1745 s^2 + 959 s + 242

Continuous-time transfer function.
```