Caso de estudio motor eléctrico: modelado, modelado de orden reducido, respuesta temporal Laplace ante escalón, y ante tren de escalones

Presentaciones en vídeo:

http://personales.upv.es/asala/YT/V/motccmodel.html ,

http://personales.upv.es/asala/YT/V/motcctfstep.html ,

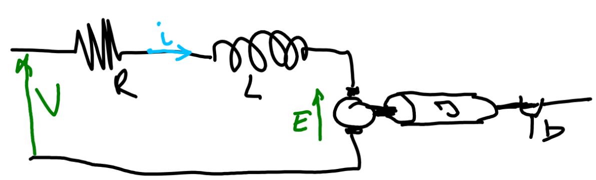
http://personales.upv.es/asala/YT/V/motccLapl.html .

Este código funcionó correctamente en **`Matlab R2022b`** (Linux)

Objetivos: modelar un motor eléctrico de CC, haciendo girar una carga de cierta inercia. Obtener su respuesta temporal ante escalón y ante un tren de pulsos, usando método de Transformada de Laplace.

Tabla de Contenidos

Modelado



Modelo completo (con dinámica parte eléctrica)

Teniendo en cuenta que en un motor eléctrico de corriente continua, en unidades internacionales la tensión generada es $E = k \cdot \omega$, y el par motor mecánico es $T = k \cdot i$, las ecuaciones de la física serán:

$$\frac{d\theta}{dt} = \omega$$

$$\frac{d\omega}{dt} = \frac{1}{J} T_{res} = \frac{-b}{J}\omega + \frac{k}{J} i$$

$$\frac{di}{dt} = \frac{1}{L} V_L = \frac{1}{L}(V - iR - k\omega)$$

Representación interna matricial normalizada (se trata de un sistema lineal) $\dot{x} = Ax + Bu$, $y = Cx + Du$:

```
syms k R L b J real
A=[0 1 0;0 -b/J k/J;0 -k/L -R/L];
B=[0;0;1/L];
C=[1 0 0;0 1 0];
D=[0;0];
```

donde hemos tomado posición y velocidad angular como salidas de interés para construir C y D.

Modelo simplificado (sin dinámica en parte eléctrica, L=0)

Entonces ya el sistema es de segundo orden, no de tercero... no podemos "dividir por L" (que sería "infinito") en la tercera ecuación, por lo que escribimos $L\frac{di}{dt} = V_L$... o sea $V_L = 0 = V - iR - k\omega$.

Las ecuaciones quedan:

```
syms theta w i V real
syms dthetadt dwdt real
ModeloSimplificado=  [dthetadt;dwdt;0] == [w; -b/J*w+k/J*i ; -k*w-R*i+V]
```

ModeloSimplificado =

$$\begin{pmatrix} \text{dthetadt} = w \\ \text{dwdt} = \frac{i\,k}{J} - \frac{b\,w}{J} \\ 0 = V - R\,i - k\,w \end{pmatrix}$$

pero este modelo no está en representación interna: hay que eliminar la variable i usando la tercera ecuación, y sustituirla en la segunda... se lo dejamos a Matlab:

```
solsimple=solve(ModeloSimplificado,{dthetadt;dwdt;i})
```

```
solsimple = struct with fields:
    dthetadt: w
        dwdt: -(w*k^2 - V*k + R*b*w)/(J*R)
           i: (V - k*w)/R
```

```
EcEstadoSimple=[solsimple.dthetadt;solsimple.dwdt]
```

EcEstadoSimple =

$$\begin{pmatrix} w \\ -\frac{w\,k^2 - V\,k + R\,b\,w}{J\,R} \end{pmatrix}$$

```
Asimple=jacobian(EcEstadoSimple,[theta,w])
```

Asimple =

$$\begin{pmatrix} 0 & 1 \\ 0 & -\frac{k^2 + R\,b}{J\,R} \end{pmatrix}$$

```
Bsimple=jacobian(EcEstadoSimple,V)
```

Bsimple =

$$\begin{pmatrix} 0 \\ \frac{k}{J\,R} \end{pmatrix}$$

```
Csimple=[1 0;0 1]; Dsimple=[0;0]; %queremos posición y velocidad, o sea TODO el estado.
```

Matriz de transferencia con symbolic toolbox

```
syms s
Estado=[theta;w;i]; %del modelo 3er orden
ModeloLaplace= s*Estado==A*Estado+B*V
```

ModeloLaplace =

$$\begin{pmatrix} s\,\theta = w \\ s\,w = \frac{i\,k}{J} - \frac{b\,w}{J} \\ i\,s = \frac{V}{L} - \frac{R\,i}{L} - \frac{k\,w}{L} \end{pmatrix}$$

```
sol=solve(ModeloLaplace,{theta,w,i})
```

```
Warning: Solutions are only valid under certain conditions. To include parameters and conditions in
the solution, specify the 'ReturnConditions' value as 'true'.
sol = struct with fields:
    theta: (V*k)/(s*(R*b + k^2 + J*R*s + L*b*s + J*L*s^2))
        w: (V*k)/(R*b + k^2 + J*R*s + L*b*s + J*L*s^2)
        i: (V*(b + J*s))/(R*b + k^2 + J*R*s + L*b*s + J*L*s^2)
```

Lo que multiplica a V en cada vble de estado sería la matriz de transferencia de entrada a estados:

```
collect(diff([sol.theta;sol.w;sol.i],V),s)
```

ans =

$$\begin{pmatrix} \dfrac{k}{(J\ L)\ s^3 + (L\ b + J\ R)\ s^2 + (k^2 + R\ b)\ s} \\ \dfrac{k}{\sigma_1} \\ \dfrac{J\ s + b}{\sigma_1} \end{pmatrix}$$

where

$$\sigma_1 = (J\ L)\ s^2 + (L\ b + J\ R)\ s + k^2 + R\ b$$

Las FdT de posición y velocidad angular (salidas de interés) serían las dos primeras.

Con la fórmula "matricial" teórica sale, obviamente, lo mismo:

```
MdT=collect(C*inv(s*eye(3)-A)*B+D,s) %no nos interesa i(s) en la matriz de transferenci
```

```
MdT =
```

$$\begin{pmatrix} \dfrac{k}{(J\ L)\ s^3 + (L\ b + J\ R)\ s^2 + (k^2 + R\ b)\ s} \\ \dfrac{k}{(J\ L)\ s^2 + (L\ b + J\ R)\ s + k^2 + R\ b} \end{pmatrix}$$

En el modelo "simplificado" sin autoinduccion de armadura:

```
MdT_simple=Csimple*inv(s*eye(2)-Asimple)*Bsimple+Dsimple
```

```
MdT_simple =
```

$$\begin{pmatrix} \dfrac{k}{s\ (k^2 + R\ b + J\ R\ s)} \\ \dfrac{k}{k^2 + R\ b + J\ R\ s} \end{pmatrix}$$

Coincide con hacer $L = 0$ en la MdT, como era de esperar... aunque, como hemos visto, no se podia hacer $L = 0$ en la representacion interna normalizada. En la MdT como no salia L "dividiendo" a nada, habriamos podido llegar al modelo simple de forma facil, al menos en este ejemplo.

Sustituyendo parámetros constantes por sus valores:

```
R=1.11; L=0.01; k=0.7; b=0.01; J=0.05; %parámetros constantes
```

Evaluemos en qué quedan las expresiones simbólicas:

```
MdT=simplify(eval(MdT))
```

```
MdT =
```

$$\begin{pmatrix} \dfrac{7000}{s\,(5\,s^2+556\,s+5011)} \\ \dfrac{7000}{5\,s^2+556\,s+5011} \end{pmatrix}$$

```
MdT_simple=simplify(eval(MdT_simple))
```

```
MdT_simple =
```

$$\begin{pmatrix} \dfrac{7000}{s\,(555\,s+5011)} \\ \dfrac{7000}{555\,s+5011} \end{pmatrix}$$

Introducción modelo en Control Systems Toolbox y función (matriz) de transferencia

Trabajamos ahora en numérico

```
A=eval(A)
```

```
A = 3×3
         0    1.0000         0
         0   -0.2000   14.0000
         0  -70.0000 -111.0000
```

```
B=eval(B)
```

```
B = 3×1
     0
     0
   100
```

```
sys=ss(A,B,C,D);
sys.InputName='V'; sys.OutputName={'\theta','\omega'};
tf(sys)
```

```
ans =

  From input "V" to output...
                     1400
   \theta:  ------------------------
            s^3 + 111.2 s^2 + 1002 s

                   1400
   \omega:  --------------------
            s^2 + 111.2 s + 1002

Continuous-time transfer function.
```

```
Asimple=eval(Asimple);Bsimple=eval(Bsimple);
sys_simple=ss(Asimple,Bsimple,Csimple,Dsimple);sys_simple.InputName='V';sys_simple.Outp
tf(sys_simple)
```

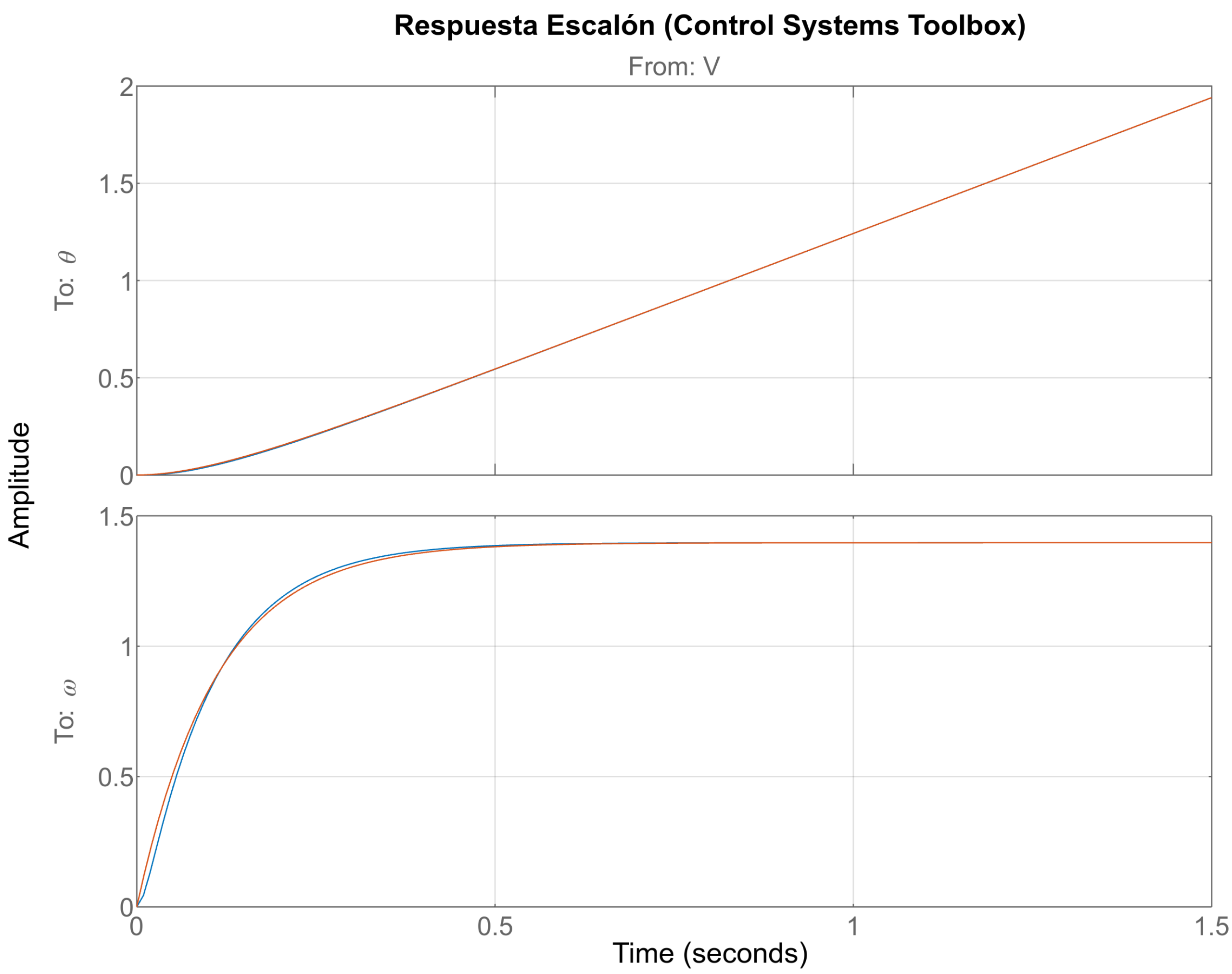
```
ans =

  From input "V" to output...
```

```
               12.61
   \theta:  -------------
            s^2 + 9.029 s

             12.61
   \omega:  ---------
            s + 9.029

Continuous-time transfer function.
```

Respuesta temporal ante escalón

Representación gráfica con Control Systems Toolbox

```
step(sys,sys_simple,1.5), grid on %sólo 1.5 segundos.
title("Respuesta Escalón (Control Systems Toolbox)")
```



Cálculos teóricos (Laplace, Symbolic Toolbox)

Las exponenciales de la respuesta libre son los valores propios de A:

```
eig(A)' %modelo orden 3
```

```
ans = 1×3
         0   -9.8927 -101.3073
```

```
eig(Asimple)' %modelo sin dinámica eléctrica.
```

```
ans = 1×2
         0   -9.0288
```

Coinciden con las raíces del denominador de las funciones de transferencia; por ejemplo, el modelo "completo"

```
[Num,Den]=numden(MdT)
```

Num =

$$\begin{pmatrix} 7000 \\ 7000 \end{pmatrix}$$

Den =

$$\begin{pmatrix} s\ (5\,s^2+556\,s+5011) \\ 5\,s^2+556\,s+5011 \end{pmatrix}$$

```
vpa(solve(Den(1)==0),6) %en simbólico
```

ans =

$$\begin{pmatrix} 0 \\ -101.307 \\ -9.89267 \end{pmatrix}$$

```
vpa(solve(Den(2)==0),6) %en numérico
```

ans =

$$\begin{pmatrix} -101.307 \\ -9.89267 \end{pmatrix}$$

Estas exponenciales estarán presentes (salvo casos especialmente preparados) también en la respuesta forzada.

Salida (en dominio de Laplace)

```
Y_s=MdT*1/s
```

Y_s =

$$\begin{pmatrix} \dfrac{7000}{s^2\ (5\,s^2+556\,s+5011)} \\ \dfrac{7000}{s\ (5\,s^2+556\,s+5011)} \end{pmatrix}$$

```
partfrac(Y_s) %no me vale, porque intenta factorizar en el dominio de los números racio
```

```
ans =
```

$$\begin{pmatrix} \dfrac{\frac{19460000\,s}{25110121}+\frac{1988567000}{25110121}}{5\,s^2+556\,s+5011}-\dfrac{3892000}{25110121\,s}+\dfrac{7000}{5011\,s^2} \\ \dfrac{7000}{5011\,s}-\dfrac{\frac{35000\,s}{5011}+\frac{3892000}{5011}}{5\,s^2+556\,s+5011} \end{pmatrix}$$

```
pfy=vpa(partfrac(Y_s,FactorMode="real"), 6) %ahora sí sale lo que buscamos
```

```
pfy =
```

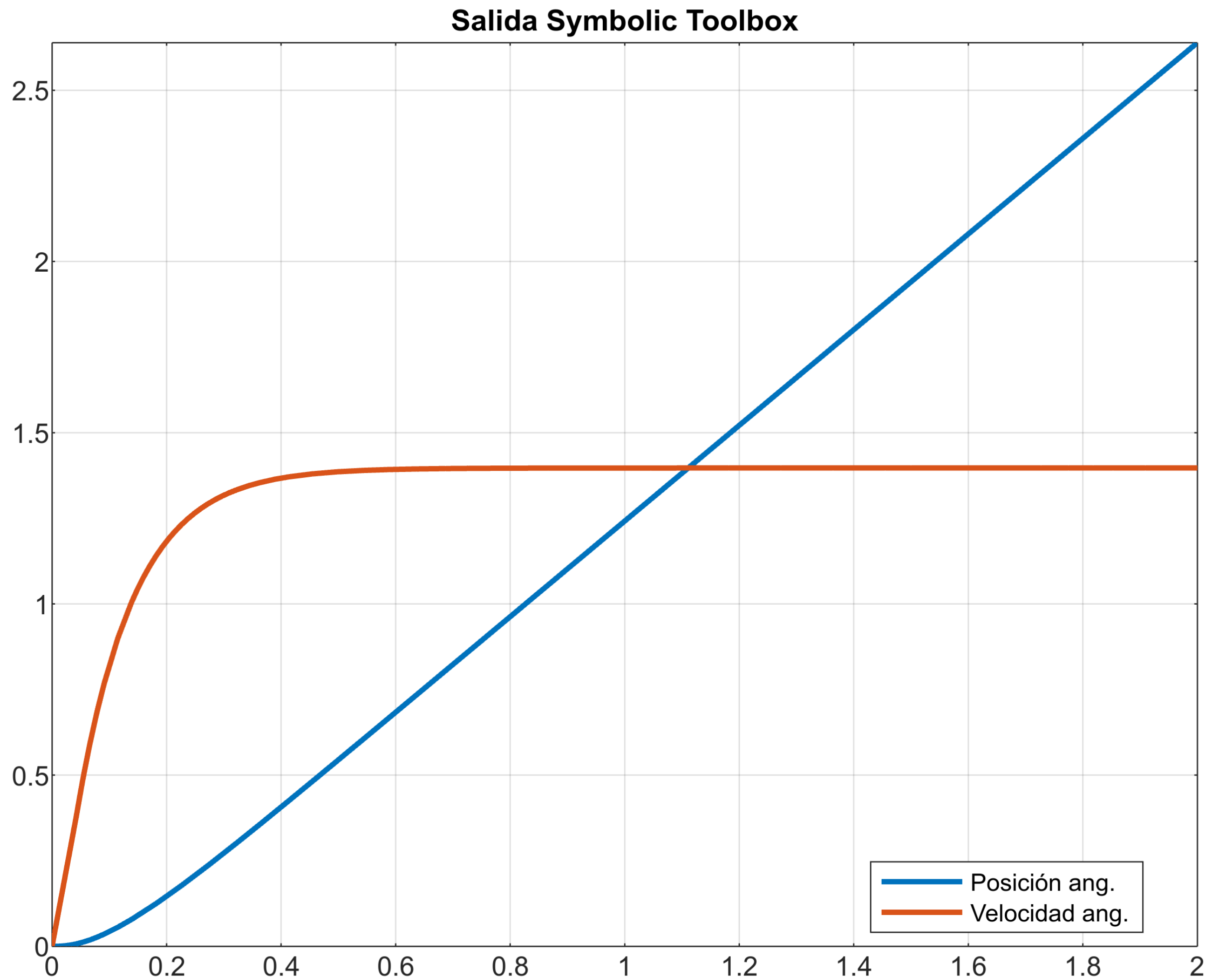
$$\begin{pmatrix} \dfrac{0.156489}{s+9.89267}-\dfrac{0.00149221}{s+101.307}-\dfrac{0.154997}{s}+\dfrac{1.39693}{s^2} \\ \dfrac{0.151172}{s+101.307}-\dfrac{1.5481}{s+9.89267}+\dfrac{1.39693}{s} \end{pmatrix}$$

```
resp_escalon_sym=vpa(ilaplace(pfy),6)
```

```
resp_escalon_sym =
```

$$\begin{pmatrix} 1.39693\,t+0.156489\,\mathrm{e}^{-9.89267\,t}-0.00149221\,\mathrm{e}^{-101.307\,t}-0.154997 \\ 0.151172\,\mathrm{e}^{-101.307\,t}-1.5481\,\mathrm{e}^{-9.89267\,t}+1.39693 \end{pmatrix}$$

```
fplot(resp_escalon_sym,[0 2],LineWidth=2), grid on, legend("Posición ang.","Velocidad a
```



El modelo "simplificado" tendrá respuesta ante escalón dada por:

```
Y_s2=MdT_simple*1/s
```

```
Y_s2 =
```

$$\begin{pmatrix} \dfrac{7000}{s^2\ (555\ s+5011)} \\ \dfrac{7000}{s\ (555\ s+5011)} \end{pmatrix}$$

```
PF2=vpa(partfrac(Y_s2,FactorMode="real"),6)
```

```
PF2 =
```

$$\begin{pmatrix} \dfrac{0.154718}{s+9.02883}-\dfrac{0.154718}{s}+\dfrac{1.39693}{s^2} \\ \dfrac{1.39693}{s}-\dfrac{1.39693}{s+9.02883} \end{pmatrix}$$

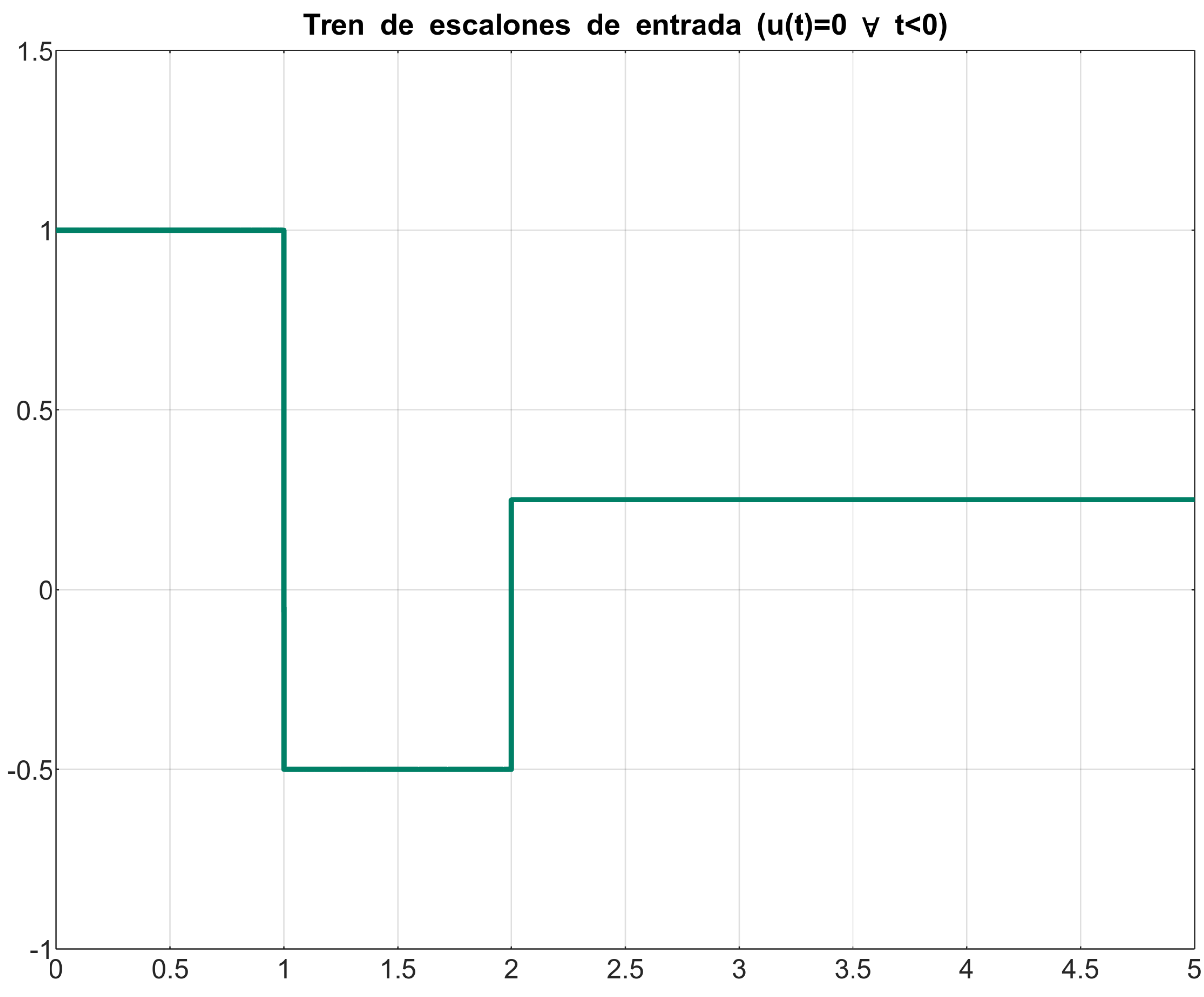
```
Y_t_simple=vpa(ilaplace(PF2),6)
```

```
Y_t_simple =
```

$$\begin{pmatrix} 1.39693\, t + 0.154718\, e^{-9.02883\, t} - 0.154718 \\ 1.39693 - 1.39693\, e^{-9.02883\, t} \end{pmatrix}$$

Respuesta temporal ante tren de escalones

```
u_s=1/s*(1-1.5*exp(-s)+0.75*exp(-2*s)); %tren de pulsos
fplot(ilaplace(u_s),[0 5], LineWidth=2, Color=[0 0.5 0.4]), grid on, ylim([-1 1.5])
title("Tren de escalones de entrada (u(t)=0 \forall t<0)")
```



Como hemos calculado la respuesta ante escalón pero "$u(t)$" es una superposición de escalones retrasados, no necesitamos hacer nada más en Transformada de Laplace:

El modelo completo con $L \neq 0$ da como resultado, si llamamos $\phi(t)$ a la respuesta ante escalón unitario (ya calculada):

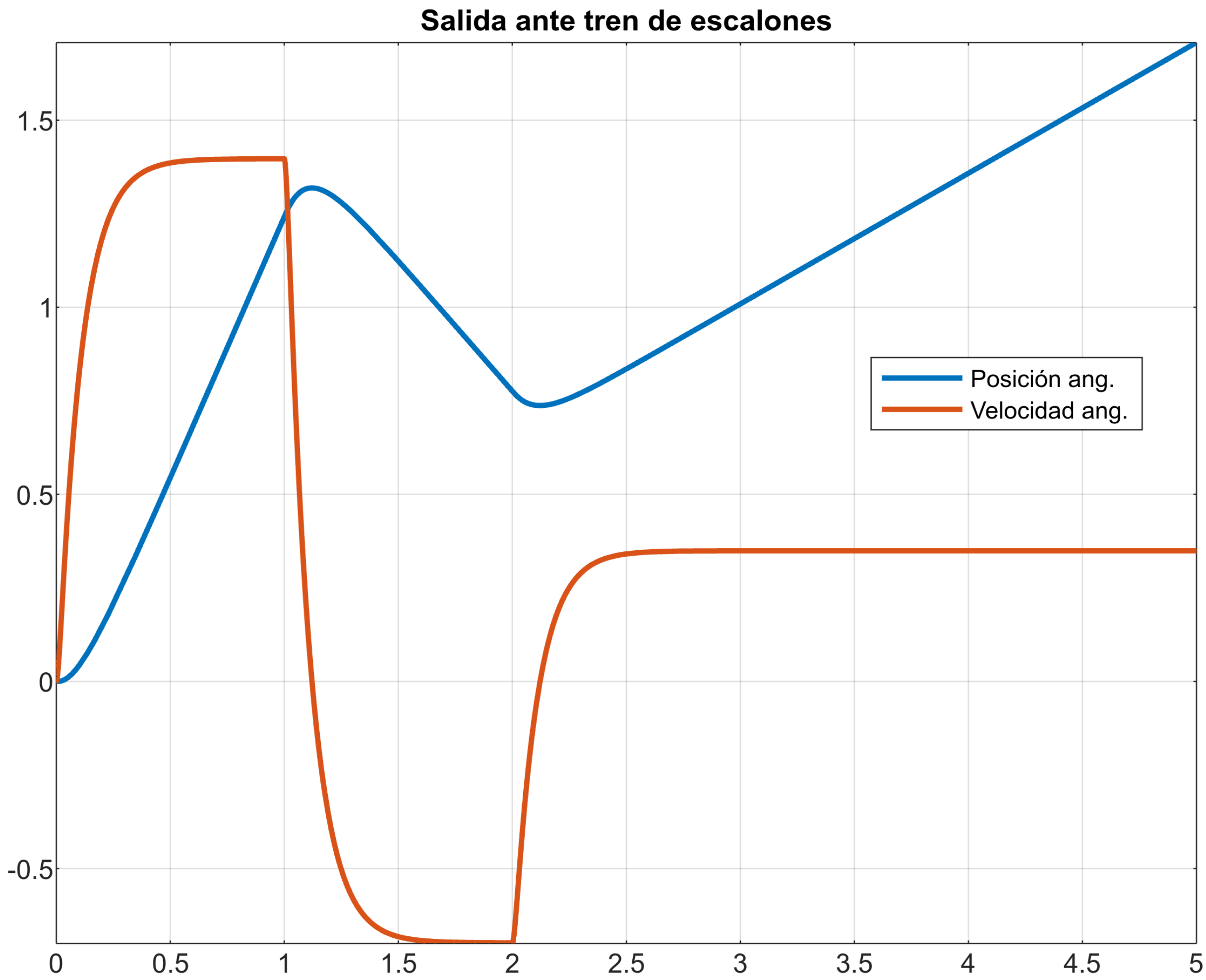
$$y(t) = \phi(t) \;+\; \begin{cases} 0 & 0 \le t < 1 \\ -1.5\phi(t-1) & 1 \le t \end{cases} \;+\; \begin{cases} 0 & 0 \le t < 2 \\ 0.75\phi(t-2) & t \ge 2 \end{cases}$$

Equivalentemente, podemos expresarlo como:

$$y(t) \;=\; \begin{cases} \phi(t) & 0 \le t < 1 \\ \phi(t) - 1.5\phi(t-1) & 1 \le t < 2 \\ \phi(t) - 1.5\phi(t-1) + 0.75\phi(t-2) & t \ge 2 \end{cases}$$

*Para graficarlo, recordemos que las funciones "a trozos" necesitan ser introducidas con el comando "`heaviside`" de la symbolic toolbox:

```
syms t real
resp_pulsos_sym= resp_escalon_sym ...
    - 1.5*heaviside(t-1)*subs(resp_escalon_sym, t, t-1) ...
    + 0.75*heaviside(t-2)*subs(resp_escalon_sym, t, t-2);
fplot(resp_pulsos_sym,[0 5], LineWidth=2), grid on,
legend("Posición ang.","Velocidad ang.",Location="best"),
title("Salida ante tren de escalones")
```


Salida ante tren de escalones
1.5
1
0.5
0
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Posición ang.
Velocidad ang.