

Respuesta temporal: fracciones simples y transformada inversa de Laplace con retardos,

a) sistema 1er orden con retardo

Presentaciones vídeo: <http://personales.upv.es/asala/YTV/dly1er1.html> , <http://personales.upv.es/asala/YTV/dly1er2.html>

b) sistema 1er orden inestable ante pulso de entrada

Presentación en vídeo: <http://personales.upv.es/asala/YTV/dly1erpul.html>

© 2022, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Este código ejecutó sin errores en Matlab R2022b (Linux)

Objetivo: revisar diferentes ejemplos de cálculo de respuesta temporal por transformada de Laplace inversa usando la descomposición en fracciones simples (partial fraction expansion) de la *Symbolic Math Toolbox* de Matlab. En este caso, **con presencia de retardos** en función de transferencia, entrada o ambas.

Tabla de Contenidos

Respuesta temporal: fracciones simples y transformada inversa de Laplace con retardos,	1
a) sistema 1er orden con retardo.....	1
b) sistema 1er orden inestable ante pulso de entrada.....	1
Ejemplo 1: Sistema de 1er orden con retardo ante escalón, c.i. = 0.....	1
Método 1: solucionar sub-problema sin retardo y luego retrasar la solución.....	2
Método 2: por Laplace manejando exponenciales en "s":	4
Descomposición en fracciones simples "manual".....	5
Descomposición en fracciones simples "automática".....	6
Transformada Inversa de Laplace.....	6
Ejemplo 2: Sistema inestable 1er orden ante pulso rectangular, c.i. no nulas.....	7
Método 1 (intuitivo y posiblemente lo que yo recomiendo): solucionar "fragmentos" sin retardo y luego ya retrasar y superponer.....	9
Método 2: Laplace con operador retardo (exp(-ds)).....	12
Descomposición en fracciones simples "manual".....	13
Descomposición en fracciones simples "automática".....	13

```
syms y(t) %función del tiempo simbólica
syms s %variable de Laplace
syms A B real %incógnitas varias en descomposiciones "a mano"
```

Ejemplo 1: Sistema de 1er orden con retardo ante escalón, c.i. = 0

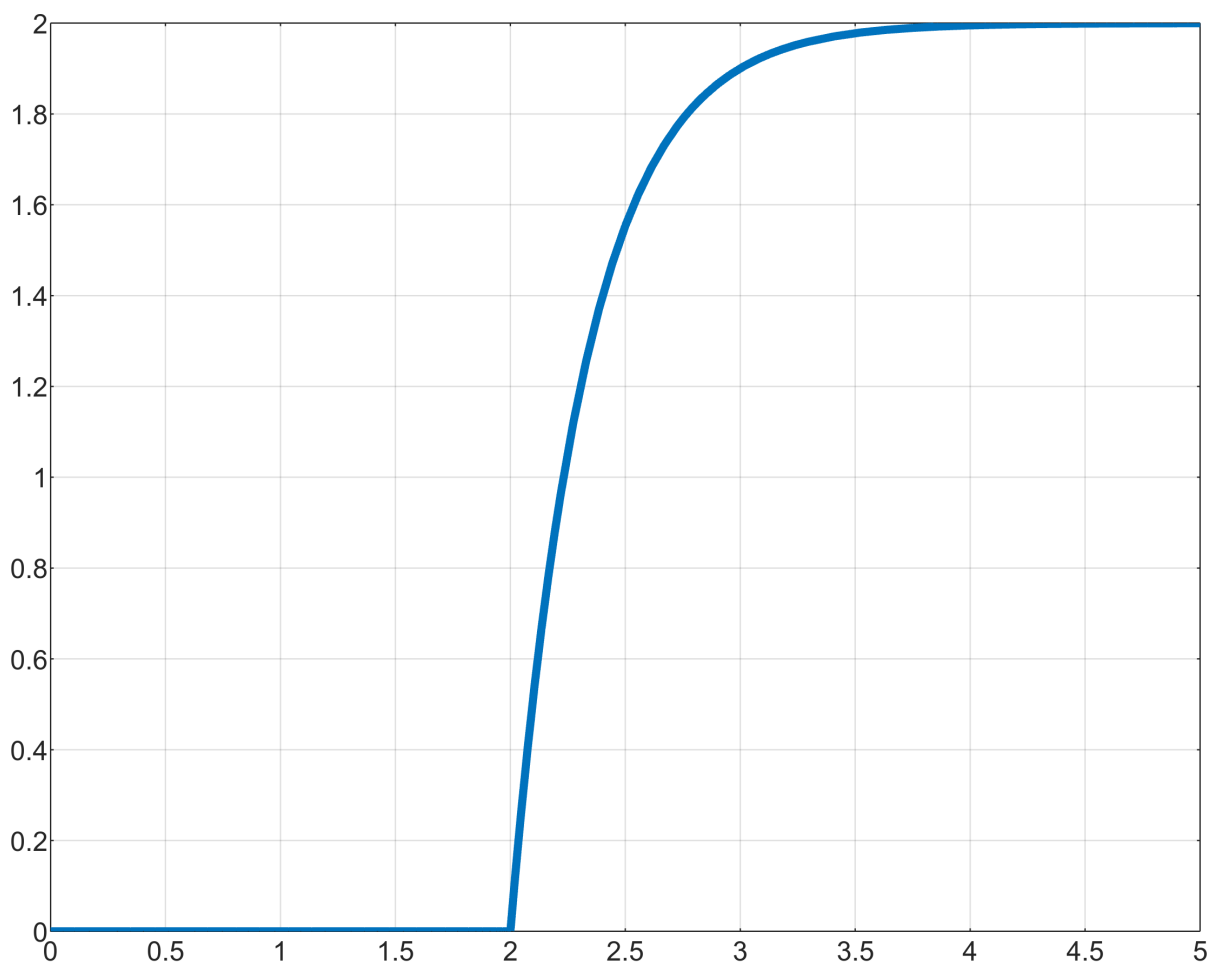
Consideremos: $\frac{dy}{dt} + 3y(t) = u(t - 2)$ ante entrada escalón de 6 unidades, suponiendo $y(0) = 0$.

Usemos "dsolve" para ver qué tiene que dar al final. El escalón en symbolic toolbox es "heaviside"; es necesario usarlo cuando no está aplicado en $t = 0$.

```
sol=dsolve(diff(y)+3*y==6*heaviside(t-2), y(0)==0); %esto debe dar al final.  
simplify(sol,100)
```

```
ans = -(sign(t - 2) + 1) (e6-3t - 1)
```

```
fplot(sol,[0 5],LineWidth=3), grid on
```



Método 1: solucionar sub-problema sin retardo y luego retrasar la solución

Básicamente es la solución sin retardo, recorrida 2 unidades a la derecha... esto se entiende un poco mejor, quizás:

```
sol_sinretardo=dsolve(diff(y)+3*y==6, y(0)==0)
```

```
sol_sinretardo = 2 - 2e-3t
```

Será, al retrasar

$$y(t) = \begin{cases} 0 & t < 2 \\ 2 - 2e^{-3(t-2)} & t \geq 2 \end{cases}$$

Podríamos resolver $y' + 3y = 6$ por Laplace (o lo que sea) y luego "retrasarlo". Posiblemente, ¡hasta lo recomiendo!.

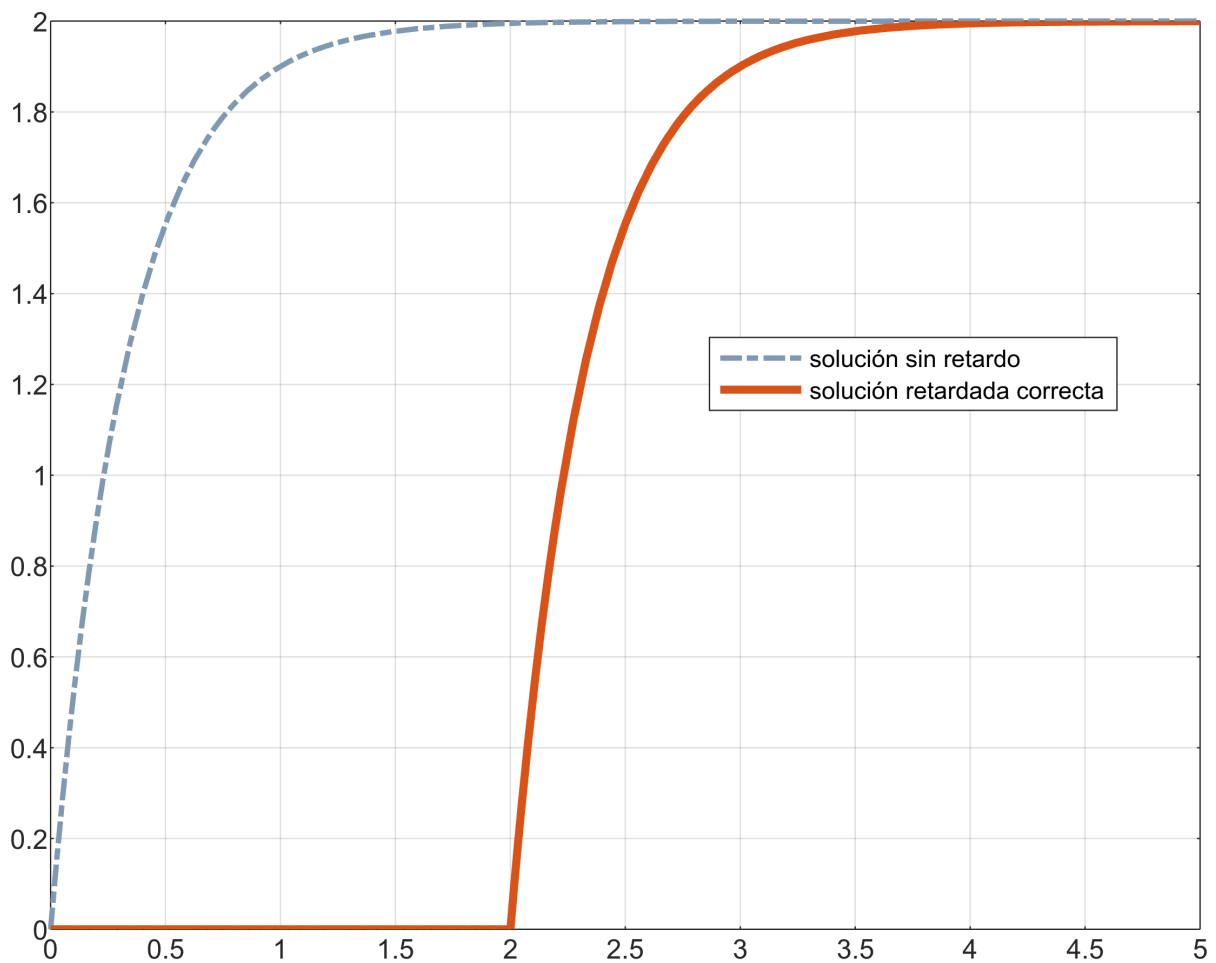
No lo detallo, me creo "dsolve" porque problemas similares se han resuelto en otros materiales. Con Matlab retrasaríamos como:

```
yretrasada=subs(heaviside(t)*sol_sinretardo,t,t-2)
```

```
yretrasada = -heaviside(t - 2) (2e6-3t - 2)
```

Gráficamente:

```
fplot(sol_sinretardo,[0 5],LineWidth=2,LineStyle="-.",Color=[.5 .6 .7]), grid on, hold on  
fplot(yretrasada,[0 5],LineWidth=3), hold off %coincide con dsolve, obviamente  
legend("solución sin retardo","solución retardada correcta",Location="best")
```



Método 2: por Laplace manejando exponenciales en "s": e^{-ds}

Por ser "puristas" siguiendo las metodologías en muchos libros, vamos a hacerlo por Laplace con "la transformada de Laplace del operador retardo", que es e^{-ds} siendo $d \geq 0$ el tiempo de retardo. En efecto, una función $f(t)$ "retrasada" se escribiría

$$f_{retra}(t) = \begin{cases} 0 & t < d \\ f(t-d) & t \geq d \end{cases}$$

de modo que, usando el cambio $[t-d = \tau, t = \tau + d]$:

$$\mathcal{L}(f_{retra}) = \int_d^\infty f(t-d)e^{-st} dt = \int_0^\infty f(\tau)e^{-s(\tau+d)} d\tau = e^{-d \cdot s} \int_0^\infty f(\tau)e^{-s\tau} d\tau = e^{-d \cdot s} \cdot \mathcal{L}(f)$$

Esto de llenar de "exponenciales en s " las expresiones (ojo, no en TIEMPO!) ya no tiene tanta interpretación "intuitiva" como el primer método, aunque es "común" entre el profesorado que explica este tema (yo mismo, je)... Bueno, al final, si entiendes que las dos cosas son esencialmente lo mismo, no es tan difícil, pero la primera vez asusta un poco y puedes confundirte entre las exponenciales "en s " y "en t ".

La transformada de Laplace de $\frac{dy}{dt} + 3y(t) = u(t-2)$ es:

$$sY(s) - y(0) + 3Y(s) = e^{-2s} \cdot U(s),$$

esto es:

$$Y(s) = \frac{e^{-2s}}{s+3} U(s) + \frac{y(0)}{s+3}$$

```
G=exp(-2*s)/(s+3); %Funcion de transferencia
U=6/s; %Transformada de Laplace de la entrada u(t)=6, constante (escalón).
%suponemos y(0)=0, por lo que no consideramos término de cond. iniciales
Y=G*U %Transformada de Laplace de la salida ante c.i.=0
```

Y =

$$\frac{6e^{-2s}}{s(s+3)}$$

Descomposición en fracciones simples "manual"

Para hacerlo "manualmente"; deberíamos igualar $Y(s) = \frac{A}{s} + \frac{B}{s+3}$

```
Desco=A/s+B/(s+3)
```

Desco =

$$\frac{A}{s} + \frac{B}{s+3}$$

Pasando a común denominador:

$$\frac{6e^{-2s}}{s(s+3)} = Y(s) = \frac{A}{s} + \frac{B}{s+3} = \frac{A(s+3) + Bs}{s(s+3)}$$

```
Desco2=collect(simplifyFraction(Desco),s)
```

Desco2 =

$$\frac{(A+B)s+3A}{s^2+3s}$$

Deberíamos igualar numeradores:

```
sol=solve([6*exp(-2*s) 0]==[3*A A+B],{A,B})
```

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:

A: 2*exp(-2*s)

B: -2*exp(-2*s)

Descomposición en fracciones simples "automática"

```
partfrac(Y) %igual que arriba, claro.
```

ans =

$$\frac{2e^{-2s}}{s} - \frac{2e^{-2s}}{s+3}$$

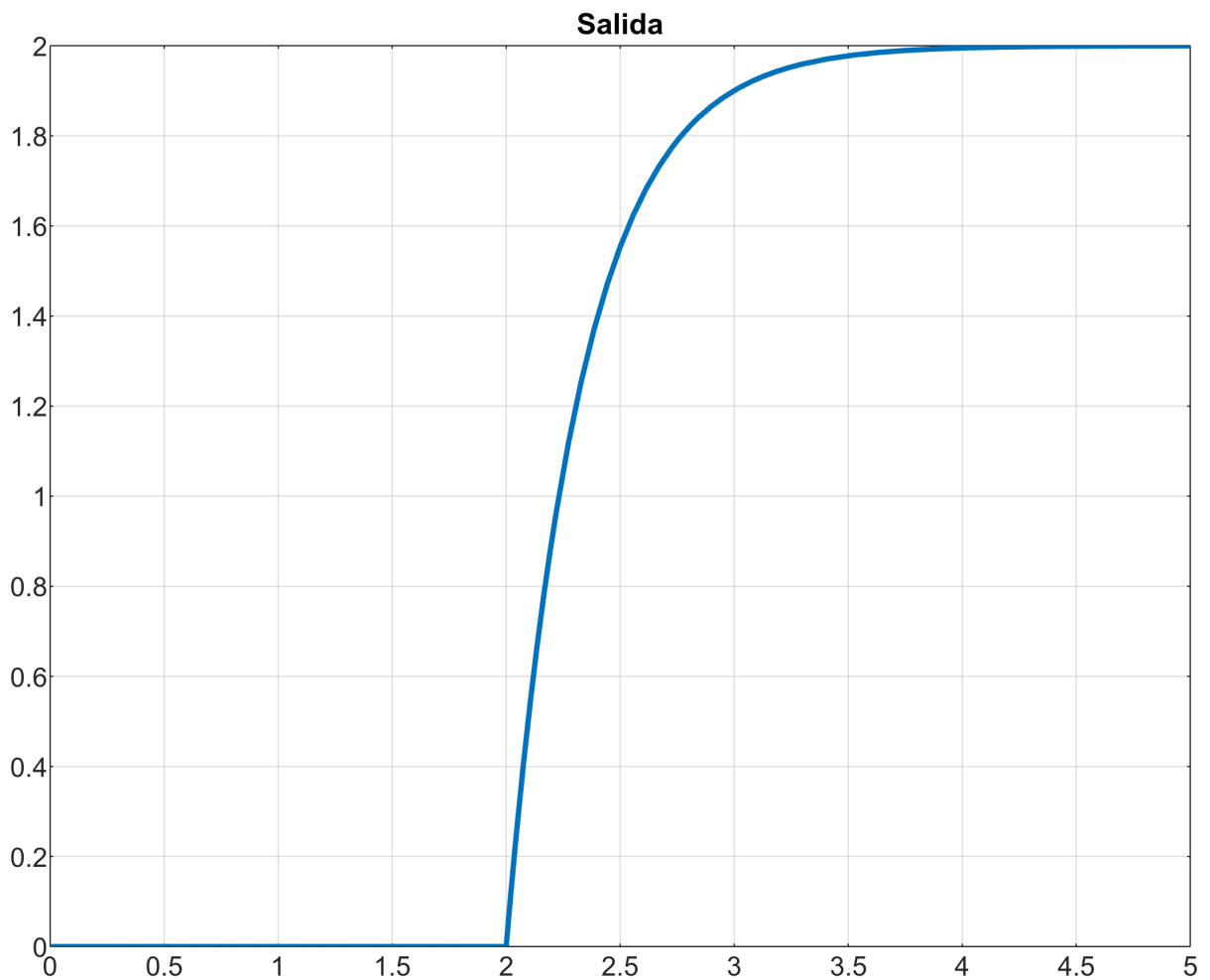
Transformada Inversa de Laplace

De hecho, la Symbolic toolbox hace la transformada inversa de todo (como él quiera, usando o no `partfrac` internamente) sin necesidad de lo de arriba:

```
Y_t=simplify(ilaplace(Y),30)
```

$$y_t = -2 \operatorname{heaviside}(t-2) (e^{6-3t} - 1)$$

```
fplot(Y_t,[0 5],LineWidth=2), grid on,title("Salida")
```



Ejemplo 2: Sistema inestable 1er orden ante pulso rectangular, c.i. no nulas

El modelo EDO del sistema es: $\frac{dy}{dt} - \frac{1}{3}y(t) = u(t)$

Su transformada es:

$$sY(s) - y(0) - Y(s)/3 = U(s),$$

esto es:

$$Y(s) = \frac{1}{s - 1/3}U(s) + \frac{y(0)}{s - 1/3}$$

```
G=1/(s-1/3); %Función de transferencia
```

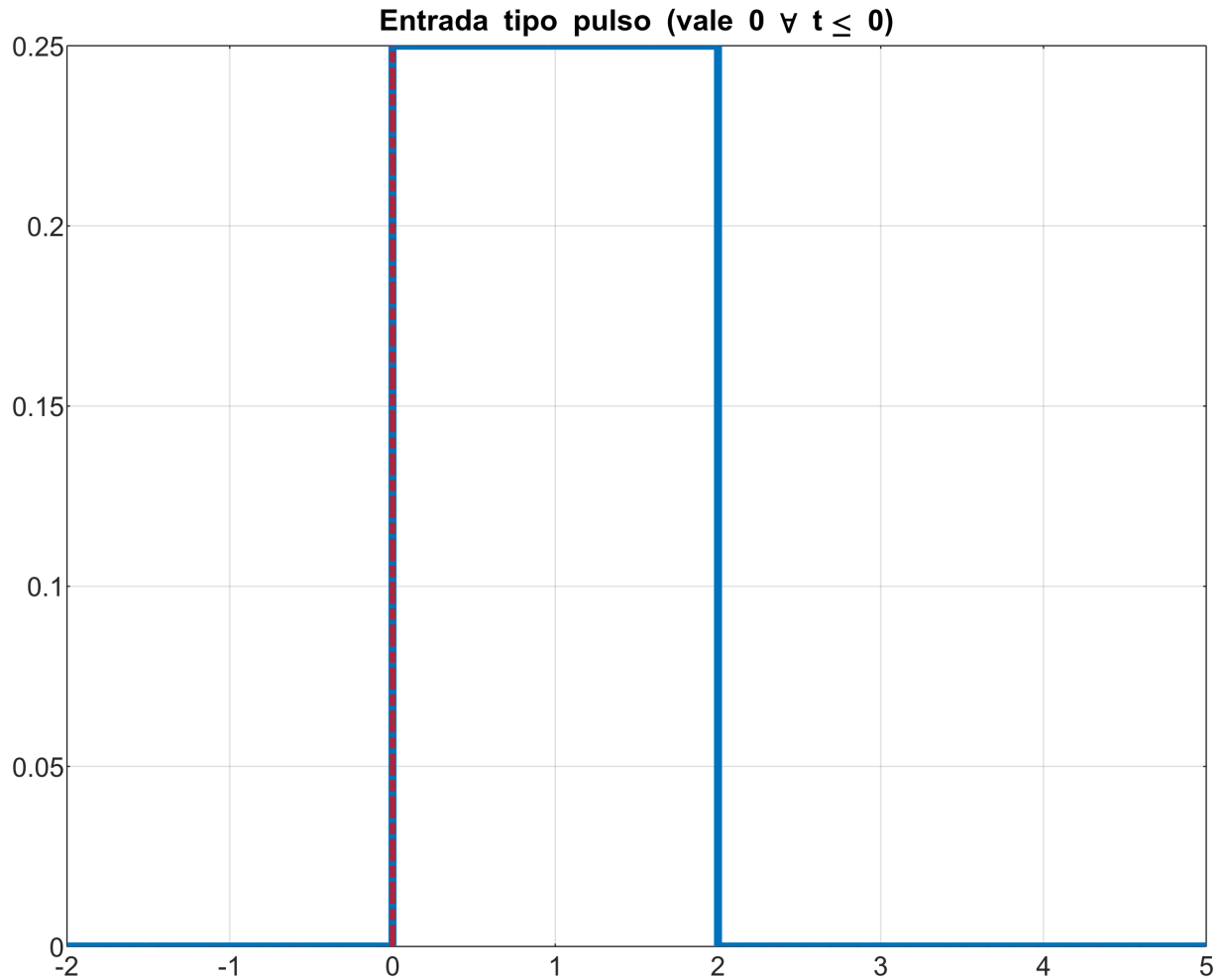
La entrada va a ser un pulso rectangular de amplitud 0.25 y duración 2 segundos:

```
U_time=0.25*(heaviside(t)-heaviside(t-2))
```

```
U_time =
```

$$\frac{\text{heaviside}(t)}{4} - \frac{\text{heaviside}(t-2)}{4}$$

```
fplot(U_time,[-2 5],LineWidth=3), grid on, xline(0,'-.r',LineWidth=2)
title("Entrada tipo pulso (vale 0 \forall t \leq 0)")
```



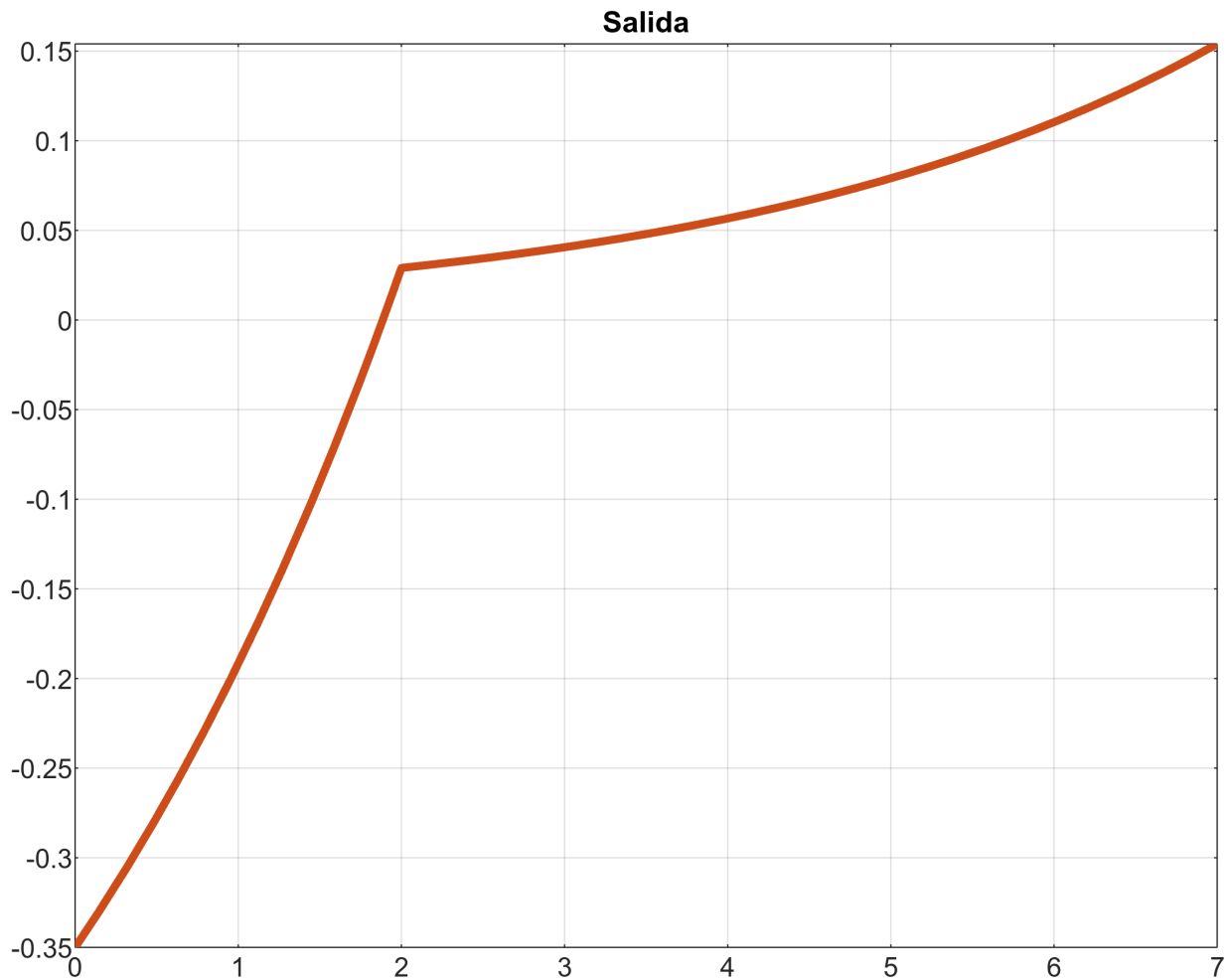
```
y0=-0.35; %condición inicial
sol=simplify( dsolve(diff(y)-y/3==U_time, y(0)==y0) , 50) %esto debe dar al final.
```

```
sol =
```

$$\frac{3 \text{ heaviside}(t-2)}{4} - \frac{7 e^{t/3}}{20} - \frac{3 \text{ heaviside}(t)}{4} + \frac{3 e^{t/3} \text{ heaviside}(t)}{4} - \frac{3 \text{ heaviside}(t-2) e^{t/3} e^{-\frac{2}{3}}}{4}$$

No lo entiende "ni Dios", pero vamos a dibujarlo:

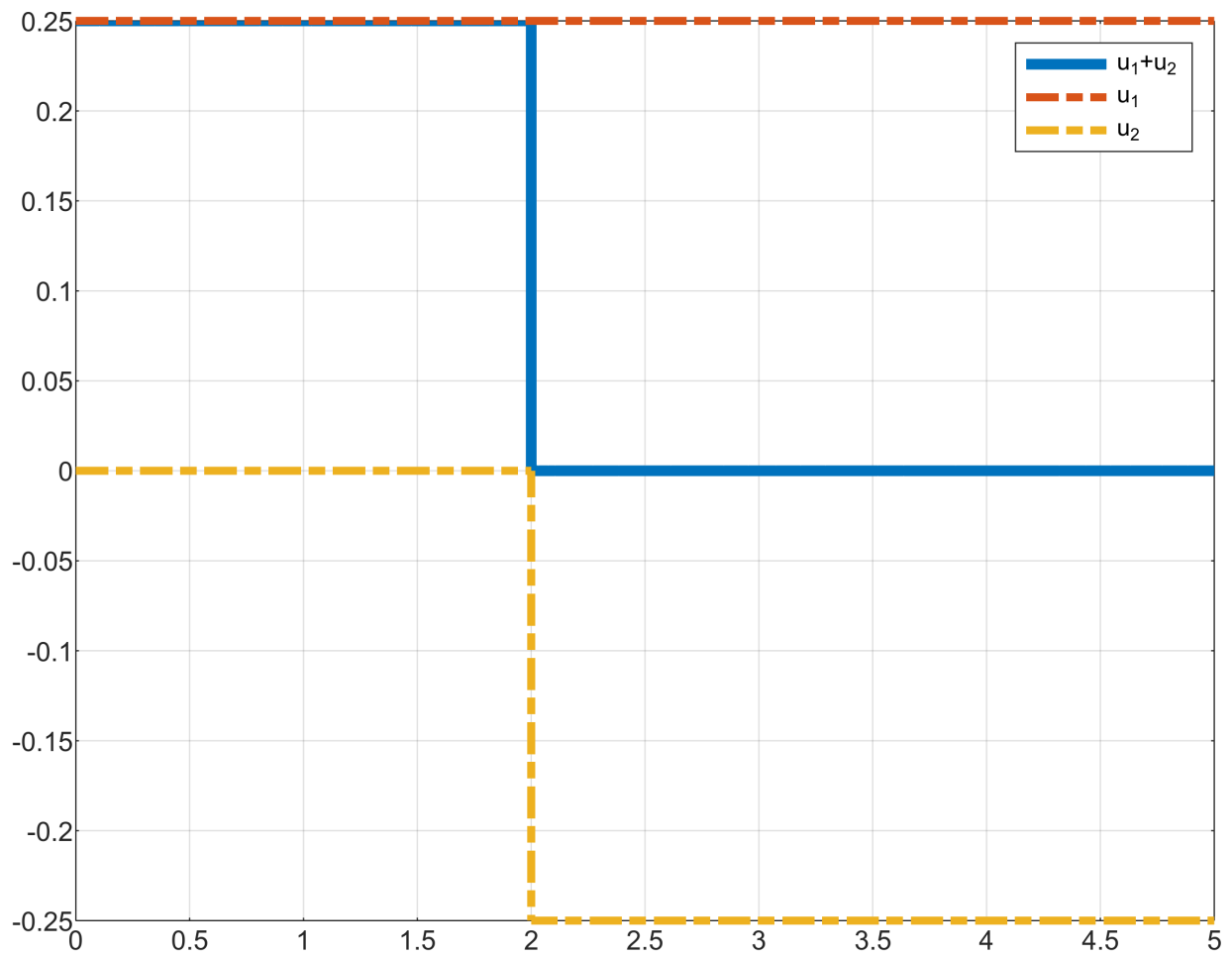
```
fplot(sol,[0 7],LineWidth=3,Color=[.8 .3 .1]), grid on, title("Salida")
```



Método 1 (intuitivo y posiblemente lo que yo recomiende): solucionar "fragmentos" sin retardo y luego ya retrasar y superponer.

La entrada es la superposición de 2 escalones:

```
U1=0.25/s;  
U2=-0.25/s*exp(-2*s); %el mismo retrasado 2 segundos, deshaciendo la subida.  
fplot(ilaplace([U1+U2]),[0 5], LineWidth=4),  
hold on  
fplot(ilaplace([U1;U2]),[0 5], LineWidth=3, LineStyle='-.'),  
hold off  
grid on, legend("u_1+u_2","u_1","u_2")
```



La solución ante $u_1(t) = 0.25$, $U_1(s) = 0.25/s$ será:

$$Y1_s = G * U1$$

$$Y1_s =$$

$$\frac{1}{4s \left(s - \frac{1}{3} \right)}$$

Con las adecuadas fracciones simples y tablas, resultará en:

$$Y1_t = \text{ilaplace}(Y1_s)$$

$$Y1_t =$$

$$\frac{3e^{t/3}}{4} - \frac{3}{4}$$

La salida ante $u_2(t) = \begin{cases} 0 & t < 2 \\ -0.25 & t \geq 2 \end{cases}$, $U_2(s) = -\frac{0.25}{s} \cdot e^{-2s}$ será la misma que ante u_1 pero retrasada, o sea

$$y_2(t) = \begin{cases} 0 & t < 2 \\ -y_1(t-2) & t \geq 2 \end{cases}$$

```
Y2_t=subs(-heaviside(t)*Y1_t,t,t-2)
```

Y2_t =

$$-\text{heaviside}(t-2) \left(\frac{3e^{\frac{t}{3}-\frac{2}{3}}}{4} - \frac{3}{4} \right)$$

y, por último la respuesta libre (salida ante entrada cero y condiciones iniciales no nulas) será:

```
Yfree=ilaplace(y0/(s-1/3))
```

Yfree =

$$-\frac{7e^{t/3}}{20}$$

con lo cual, la salida será $y(t) = y_1(t) + y_2(t) + y_{free}(t)$, esto es:

$$y(t) = -\frac{7}{20}e^{t/3} + \frac{3}{4}(e^{t/3} - 1) + \begin{cases} 0 & t < 2 \\ -\frac{3}{4}(e^{(t-2)/3} - 1) & t \geq 2 \end{cases}$$

que, reorganizando queda:

$$y(t) = \begin{cases} -\frac{7}{20}e^{t/3} + \frac{3}{4}(e^{t/3} - 1) & t < 2 \\ -\frac{7}{20}e^{t/3} + \frac{3}{4}(e^{t/3} - 1) - \frac{3}{4}(e^{(t-2)/3} - 1) & t \geq 2 \end{cases}$$

*Usando Matlab,

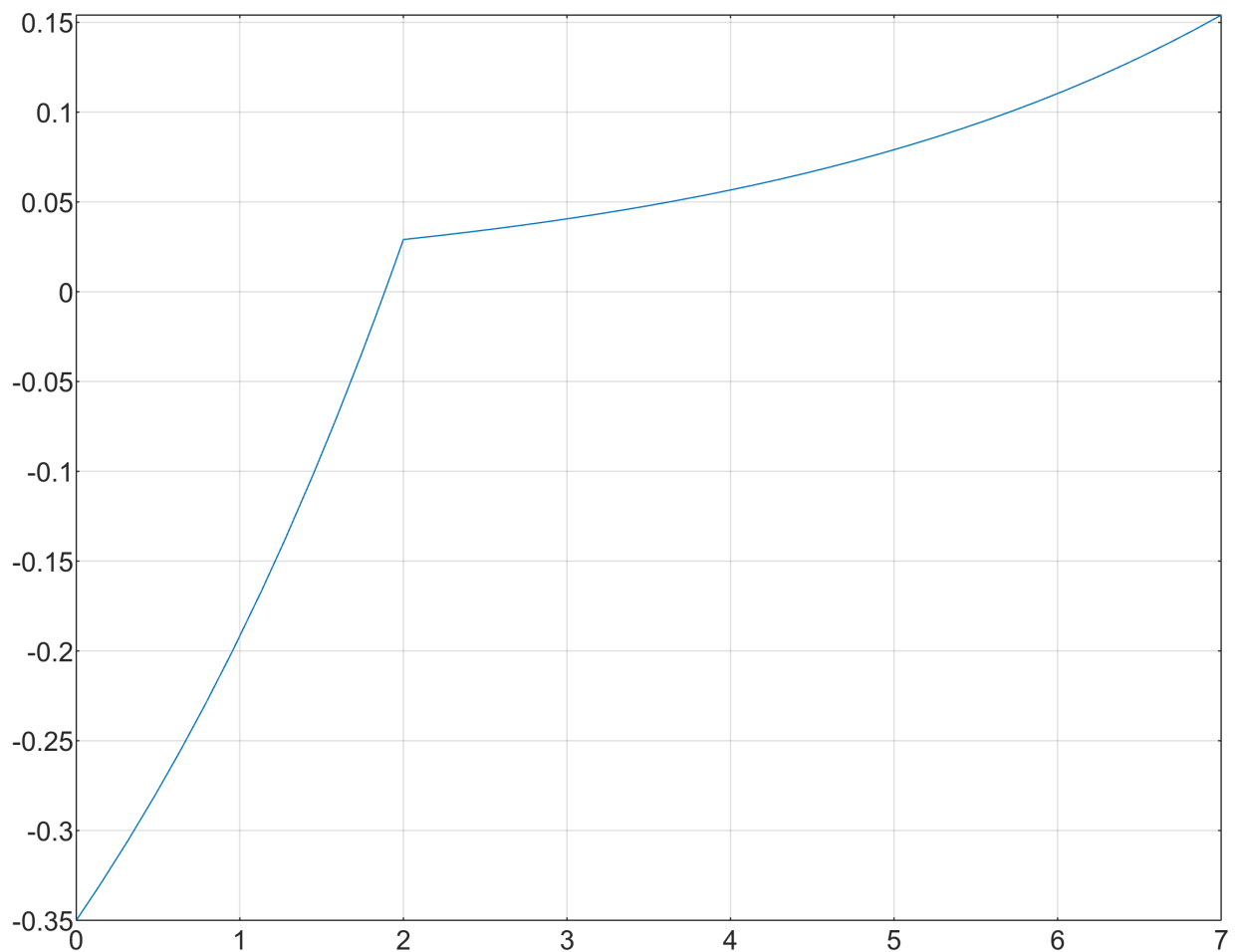
```
Y_metodo1=simplify(Yfree+Y1_t+Y2_t)
```

Y_metodo1 =

$$\frac{2e^{t/3}}{5} - \text{heaviside}(t-2) \left(\frac{3e^{\frac{t}{3}-\frac{2}{3}}}{4} - \frac{3}{4} \right) - \frac{3}{4}$$

coincide con `dsolve`, como era de esperar.

```
fplot(Y_metodo1,[0 7]), grid on
```



Método 2: Laplace con operador retardo ($\exp(-ds)$).

Ahora por Laplace, poco a poco a mano:

```
U=laplace(U_time)
```

U =

$$\frac{1}{4s} - \frac{e^{-2s}}{4s}$$

```
Y=simplify(G*U+y0/(s-1/3),70)
```

Y =

$$-\frac{3(7s + 5e^{-2s} - 5)}{20s(3s - 1)}$$

Descomposición en fracciones simples "manual"

Para hacerlo "manualmente"; deberíamos igualar $Y_{forz}(s) = \frac{A}{s} + \frac{B}{s - \frac{1}{3}}$

$$\text{Desco} = A/s + B/(s - 1/3)$$

Desco =

$$\frac{A}{s} + \frac{B}{s - \frac{1}{3}}$$

Pasando a común denominador:

$$Y_{forz}(s) = \frac{A}{s} + \frac{B}{s - \frac{1}{3}} = \frac{A(s - \frac{1}{3}) + Bs}{s(s - \frac{1}{3})}$$

$$\text{Desco2} = \text{collect}(\text{simplifyFraction}(\text{Desco}), s)$$

Desco2 =

$$\frac{(3A + 3B)s - A}{3s^2 - s}$$

Como los denominadores coinciden el de Yforz y el de Desco2, deberíamos igualar numeradores:

$$\text{sol} = \text{solve}([15/20 * (1 - \exp(-2*s)) - 21/20] == [-A \quad 3 * (A+B)], \{A, B\})$$

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:

$$A: (3 * \exp(-2*s))/4 - 3/4$$

$$B: 2/5 - (3 * \exp(-2*s))/4$$

Descomposición en fracciones simples "automática"

La Symbolic Toolbox lo hace en una línea, lo de arriba "sobra" si tenemos Matlab a mano...

$$\text{partfrac}(Y)$$

ans =

$$\frac{\frac{3e^{-2s}}{4} - \frac{3}{4}}{s} - \frac{\frac{9e^{-2s}}{4} - \frac{6}{5}}{3s - 1}$$

```
Y_t=ilaplace(Y)
```

```
Y_t =
```

$$\frac{2e^{t/3}}{5} - \frac{3\operatorname{heaviside}(t-2)\left(e^{\frac{t}{3}-\frac{2}{3}} - 1\right)}{4} - \frac{3}{4}$$

```
fplot(Y_t,[0 7]), grid on, yline(0)
```

