

Caso Estudio: tratamiento térmico objeto 2 capas "en escalón" y "en rampa"

© 2023, Antonio Sala Piqueras, Universitat Politècnica de Valencia. Todos los derechos reservados.

Presentaciones en vídeo:

<http://personales.upv.es/asala/YT/V/term2mod1.html> , <http://personales.upv.es/asala/YT/V/term2mod2.html> , <http://personales.upv.es/asala/YT/V/term2step.html> , <http://personales.upv.es/asala/YT/V/term2ramp.html> .

Este código funcionó sin errores en Matlab **R2022b** (Linux)

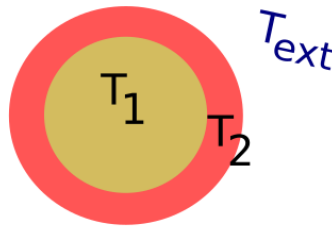
Objetivo: modelar el calentamiento de un sistema térmico de segundo orden. Obtener respuestas ante escalón y rampa, por Control Systems Toolbox o por Laplace. Analizar el perfil de calentamiento necesario para limitar la diferencia de temperaturas entre elementos del sistema.

Tabla de Contenidos

Modelado (repr. interna).....	1
Modelo de orden 2.....	2
Modelo de orden reducido, despreciando capacidad calorífica de capa exterior:.....	4
Modelos de orden superior (esbozo).....	4
Matriz de Transferencia y polos en Symbolic Toolbox.....	5
Introducción del modelo en Control Systems Toolbox.....	6
Respuesta temporal ante escalón (Control Systems Toolbox).....	7
Respuesta temporal ante rampa (Control Systems Toolbox).....	9
Respuesta temporal de las temperaturas ante rampa de entrada (Symbolic Toolbox)....	10
Limitación de velocidad de calentamiento para no superar diferencia de temperatura prefijada.....	12

Modelado (repr. interna)

Debemos modelar una pieza "1", rodeada de una pintura o capa aislante "2" (que debe calentarse primero para que se caliente la pieza "1") al introducirla en un horno cuya temperatura es T_{ext} :



Modelo de orden 2

A partir de:

$$M_1 C_{e,1} \frac{dT_1}{dt} = k_{12} \cdot (T_2 - T_1)$$

$$M_2 C_{e,2} \frac{dT_2}{dt} = k_{12} \cdot (T_1 - T_2) + k_{2e} \cdot (T_e - T_2)$$

Podemos expresar matricialmente la ecuación de estado normalizada como, sustituyendo parámetros constantes por valores numéricos:

```
Mce1=110; Mce2=15; k12=1; K2e=2.4; %parámetros constantes
A=[-k12/Mce1 k12/Mce1; k12/Mce2 -(k12+K2e)/Mce2]
```

```
A = 2x2
    -0.0091    0.0091
     0.0667   -0.2267
```

```
B=[0; K2e/Mce2]
```

```
B = 2x1
         0
     0.1600
```

```
C=eye(2); D=[0;0];
```

Con esto podemos construir $\dot{x} = Ax + Bu$, $y = Cx + Du$, siendo x el vector de estado $[T_1; T_2]$, y la salida $y = x$.

Si estudiamos el equilibrio (derivadas a cero), tendríamos $0 = Ax_{eq} + Bu_{eq}$, o sea $x_{eq} = -A^{-1}B \cdot u_{eq}$.

```
-inv(A)*B
```

```
ans = 2×1  
     1  
     1
```

Esto indica que el equilibrio se alcanza con todas las temperaturas iguales a la temperatura de entrada, como era de esperar intuitivamente.

Como las salidas son las mismas que los estados, obviamente la ecuación de salida en equilibrio $y_{eq} = Cx_{eq} + Du_{eq} = (-CA^{-1}B + D) \cdot u_{eq}$ acaba diciendo exactamente lo mismo:

```
-C*inv(A)*B+D
```

```
ans = 2×1  
     1  
     1
```

El punto de funcionamiento es cualquiera que elijamos con las tres temperaturas T_1 , T_2 , T_{ext} iguales (equilibrio térmico), usualmente todas iguales a la *temperatura ambiente antes de conectar el horno*. El sistema, en ese punto de operación, es lineal, si las temperaturas se consideran incrementos respecto al mencionado punto.

Las "diferencias de temperatura" producen estrés térmico: diferencias de temperatura excesivas producen tensión mecánica debido a la dilatación diferencial de los materiales, de modo que aparecen grietas (incluso pueden fracturarse piezas) o se despegan uniones en aplicaciones prácticas. Vamos, pues, a añadir esa variable en las salidas:

```
C=[eye(2);-1 1], D=[0;0;0]
```

```
C = 3×2  
     1     0  
     0     1  
    -1     1  
D = 3×1  
     0  
     0  
     0
```

Ahora en equilibrio, la diferencia térmica es cero, claro:

```
C*inv(-A)*B+D
```

```
ans = 3×1  
     1  
     1  
     0
```

Modelo de orden reducido, despreciando capacidad calorífica de capa exterior:

A partir de $M_2 C_{e,2} \rightarrow 0$, escribiríamos:

$$M_1 C_{e,1} \frac{dT_1}{dt} = k_{12} \cdot (T_2 - T_1)$$

$$0 = k_{12} \cdot (T_1 - T_2) + k_{2e} \cdot (T_e - T_2)$$

y podríamos tener:

$$T_2 = \frac{k_{12} T_1 + k_{2e} T_e}{k_{12} + k_{2e}}$$

y sustituyendo arriba:

$$M_1 C_{e,1} \frac{dT_1}{dt} = k_{12} \cdot \left(\frac{k_{12} T_1 + k_{2e} T_e}{k_{12} + k_{2e}} - T_1 \right) = \frac{k_{12} k_{2e}}{k_{12} + k_{2e}} (T_e - T_1) = \frac{1}{\frac{1}{k_{12}} + \frac{1}{k_{2e}}} (T_e - T_1)$$

que corresponde a la conexión "en serie" de dos "conductancias térmicas".

La salida "diferencia de temperaturas" sería aproximada por:

$$T_2 - T_1 = \frac{k_{2e}}{k_{12} + k_{2e}} (T_e - T_1)$$

El equilibrio sigue siendo $T_{1,eq} = T_{e,eq}$.

Aunque continuaremos en este material con el modelo con estados T_1 y T_2 , podría ser conveniente en aplicaciones a mayor escala (donde esto sea un "subsistema" de algo mucho mayor), el considerar si merece la pena o no tener un modelo "más complicado" del subsistema.

Modelos de orden superior (esbozo)

Realmente, puestos a complicar las cosas, en estos objetos esféricos la temperatura será una función del tiempo y de la posición... por simetría, de la distancia al centro r , o sea $T(t, r)$, si las condiciones iniciales/contorno tienen simetría radial. Un análisis, por tanto, más exacto, requeriría la ecuación en derivadas parciales (coordenadas esféricas) de transmisión de calor:

$$\frac{\delta T}{\delta t} = \frac{\kappa}{r^2} \frac{\delta}{\delta r} (r^2 \frac{\delta T}{\delta r})$$

o una aproximación por "elementos finitos" haciendo en vez de "dos" temperaturas, pues "dos capas en el cuerpo 2" y "siete capas en el cuerpo 1", para tener un modelo de orden 9, por ejemplo.

No entra en los objetivos de este material el plantear ese tipo de modelos, consultad vuestras notas de Transmisión de Calor o asignaturas de EDP.

Matriz de Transferencia y polos en Symbolic Toolbox

Para operar en transformada de Laplace y obtener las fórmulas explícitas de la respuesta, podemos operar con la Symbolic Math toolbox de Matlab.

```
syms s
MdT=simplify(C*inv(s*eye(2)-A)*B+D)
```

MdT =

$$\begin{pmatrix} \frac{12}{8250s^2 + 1945s + 12} \\ \frac{12(110s + 1)}{8250s^2 + 1945s + 12} \\ \frac{1320s}{8250s^2 + 1945s + 12} \end{pmatrix}$$

Como curiosidad, la temperatura T_1 verificaría la EDO:

$$8250 \frac{d^2 T_1}{dt^2} + 1945 \frac{dT_1}{dt} + 12T_1(t) = 12T_{ext}(t)$$

pero hemos, en cierto modo, perdido la "física", porque la condición inicial $\frac{dT_1}{dt}(0)$ que necesitamos para resolverla, realmente no es "conocida" al menos directamente, y lo que tiene significado físico es $T_1(0)$ y $T_2(0)$. Claro, $\frac{dT_1}{dt}(0) = \frac{k_{12}}{M_1 C_{e,1}} (T_2(0) - T_1(0))$, pero si tengo que escribir esa ecuación de estado para calcular la condición inicial, pues mejor trabajar con $\dot{x} = Ax + Bu$ que replica la "física" que sabemos comprender.

Polos del sistema (raíces del denominador de las FdTs):

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

Num =

$$\begin{pmatrix} 12 \\ 1320s + 12 \\ 1320s \end{pmatrix}$$

Den =

$$\begin{pmatrix} 8250s^2 + 1945s + 12 \\ 8250s^2 + 1945s + 12 \\ 8250s^2 + 1945s + 12 \end{pmatrix}$$

```
vpa (solve (Den (1) == 0) , 5)
```

ans =

$$\begin{pmatrix} -0.22942 \\ -0.0063402 \end{pmatrix}$$

Por tanto, la respuesta no será "oscilatoria", sino que será $Pe^{-0.00634t} + Qe^{-0.229t}$ (dos exponenciales estables) junto a alguna solución particular dependiente de la entrada.

Introducción del modelo en Control Systems Toolbox

```
termico=ss(A,B,C,D);  
termico.InputName="Text"; termico.OutputName={'T_1','T_2','T_2-T_1'};
```

La función (matriz) de transferencia es:

```
tf(termico)
```

ans =

From input "Text" to output...
0.001455

T_1: -----
s^2 + 0.2358 s + 0.001455

0.16 s + 0.001455
T_2: -----
s^2 + 0.2358 s + 0.001455

0.16 s - 3.925e-19
T_2-T_1: -----
s^2 + 0.2358 s + 0.001455

Continuous-time transfer function.

Los denominadores tienen raíces (polos del sistema) reales, que coinciden con las vistas arriba en "Symbolic":

```
zpk(termico)
```

```
ans =
```

```
From input "Text" to output...
```

```
0.0014545
```

```
T_1: -----  
      (s+0.00634) (s+0.2294)
```

```
      0.16 (s+0.009091)  
T_2: -----  
      (s+0.00634) (s+0.2294)
```

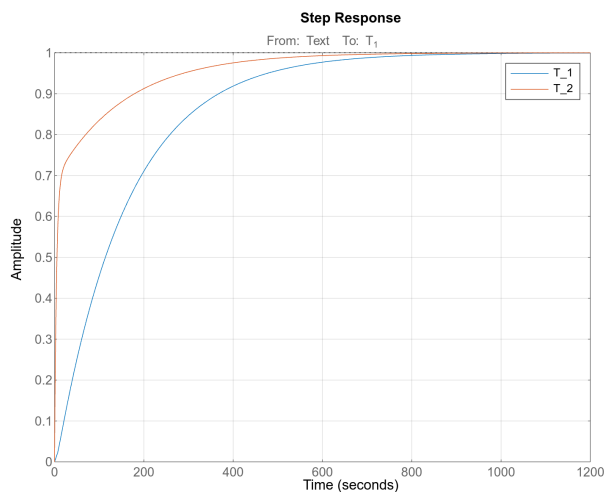
```
      0.16 s  
T_2-T_1: -----  
          (s+0.00634) (s+0.2294)
```

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

Respuesta temporal ante escalón (Control Systems Toolbox)

La respuesta ante un escalón de entrada $T_{ext} = 1$, será:

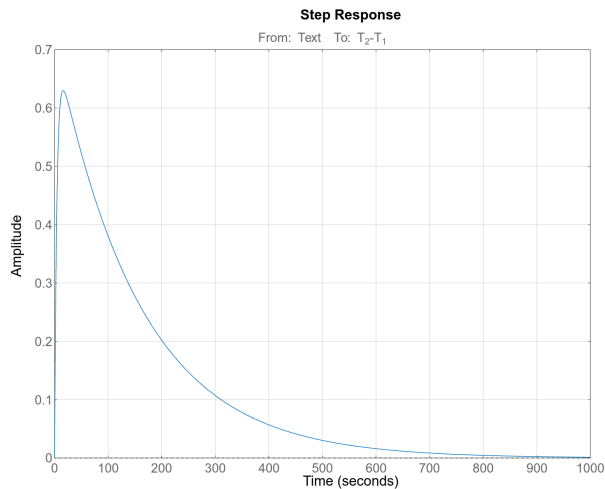
```
step(termico(1)), hold on, step(termico(2)), hold off, grid on, legend("T_1", "T_2")
```



Se observa un "calentamiento de T_2 " muy rápido, y un calentamiento "de ambas piezas" más lento.

La diferencia de temperaturas es:

```
step(termico(3),1000), grid on
```



De forma aproximada,

```
lospolos=pole(termico)
```

```
lospolos = 2x1
-0.0063
-0.2294
```

El tiempo de establecimiento (criterio 95%) asociado a las dos exponenciales (polos)

```
t_est=log(0.05)./lospolos
```

```
t_est = 2x1
-472.5003
-13.0580
```

También puede determinarse ese tiempo de establecimiento con la matriz A (de hecho, nos da información de la "forma" de esos "modos de respuesta libre" que la FdT no nos da):

```
[V, De] = eig(A)
```

```
V = 2x2
0.9571 -0.0412
0.2896 0.9991
De = 2x2
-0.0063 0
0 -0.2294
```

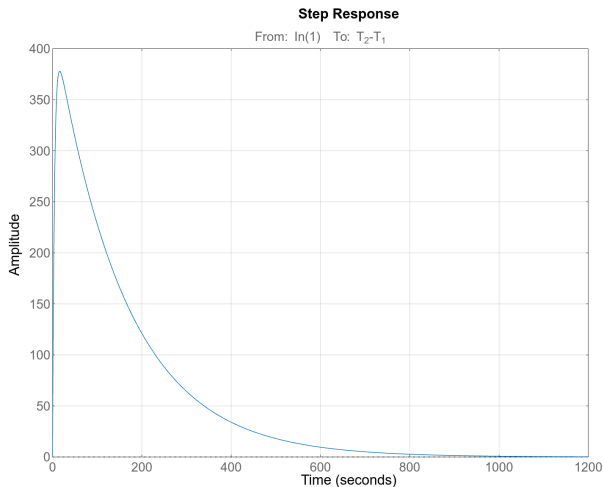
O sea, hay un fenómeno que dura 13 segundos asociado al "enfriamiento de la capa exterior" (porque $V(:,2)$ está fundamentalmente alineado con T_2), y otro fenómeno que dura 470 segundos, asociado al calentamiento simultáneo de ambos cuerpos, sobre todo el cuerpo 1 interior.

Tratamiento térmico a 600 °C (bueno, en rigor, 600 °C por encima del ambiente): la diferencia de temperatura máxima será, por linealidad de

```
600*0.63
```

```
ans = 378
```

```
step(termico(3)*600), grid on
```



Si no es aceptable, habrá que calentar "lentamente", por ejemplo, en forma de rampa.

De hecho, conseguir "tecnológicamente" que la temperatura exterior suba "instantáneamente" 600 grados es difícil si no imposible. Teniendo en cuenta que una de las dinámicas dura 13 segundos, seguramente tardemos más en calentar el horno, salvo que la tecnología fuera de "introducir el objeto en un horno ya precalentado por una abertura pequeña", o "sumergir en sales fundidas a 600°C".

Respuesta temporal ante rampa (Control Systems Toolbox)

La respuesta ante una rampa de entrada $T_{ext} = 1 \cdot t$, será calculada con:

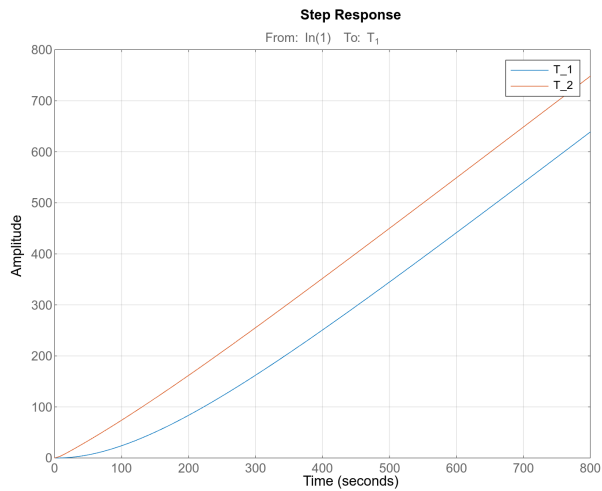
```
Integrador=tf(1,[1 0])
```

```
Integrador =
```

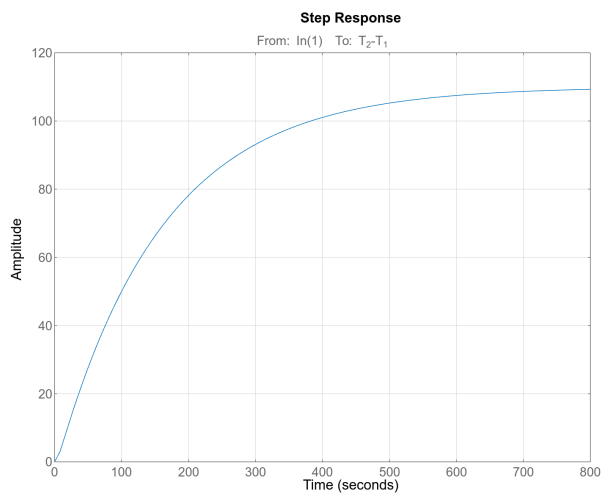
```
1
-
s
```

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

```
step(termico(1)*Integrador,800), hold on, step(termico(2)*Integrador,800), hold off, gr
```



```
step(termico(3)*Integrador,800), grid on
```



Respuesta temporal de las temperaturas ante rampa de entrada (Symbolic Toolbox)

Ante $T_{ext} = t$, tendríamos:

$$Y = MdT \cdot 1/s^2$$

Y =

$$\left(\begin{array}{c} \frac{12}{s^2 (8250 s^2 + 1945 s + 12)} \\ \frac{12 (110 s + 1)}{s^2 (8250 s^2 + 1945 s + 12)} \\ \frac{1320}{s (8250 s^2 + 1945 s + 12)} \end{array} \right)$$

Estas serían las fracciones simples:

```
Pepito=vpa(partfrac(Y,FactorMode="real"),5)
```

Pepito =

$$\left(\begin{array}{l} \frac{162.21}{s+0.0063402} - \frac{0.12389}{s+0.22942} - \frac{162.08}{s} + \frac{1.0}{s^2} \\ \frac{49.081}{s+0.0063402} + \frac{3.0025}{s+0.22942} - \frac{52.083}{s} + \frac{1.0}{s^2} \\ \frac{3.1264}{s+0.22942} - \frac{113.13}{s+0.0063402} + \frac{110.0}{s} \end{array} \right)$$

Y esta la transformada de Laplace inversa:

```
cosa=vpa(ilaplace(Pepito),5)
```

cosa =

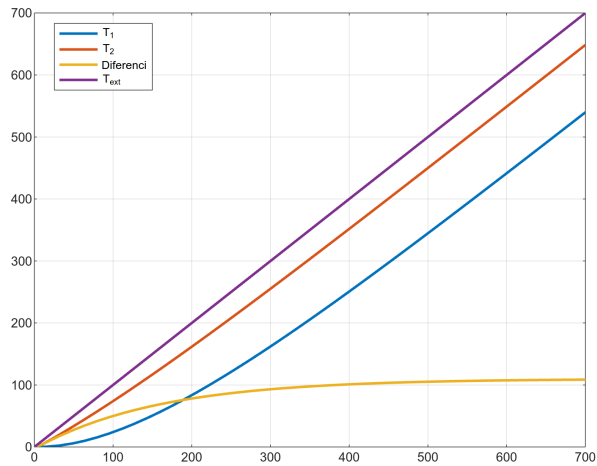
$$\left(\begin{array}{l} t + 162.21 e^{-0.0063402 t} - 0.12389 e^{-0.22942 t} - 162.08 \\ t + 49.081 e^{-0.0063402 t} + 3.0025 e^{-0.22942 t} - 52.083 \\ 3.1264 e^{-0.22942 t} - 113.13 e^{-0.0063402 t} + 110.0 \end{array} \right)$$

```
tempdif=cosa(3);
```

Las exponenciales son las mismas que en la respuesta ante escalón (están presentes en "casi" todos los transitorios con significado práctico, excepto casos "especiales" expresamente buscados para que no lo estén). El tiempo de establecimiento será, pues, idéntico a la respuesta escalón.

Si dibujamos la respuesta temporal, tenemos:

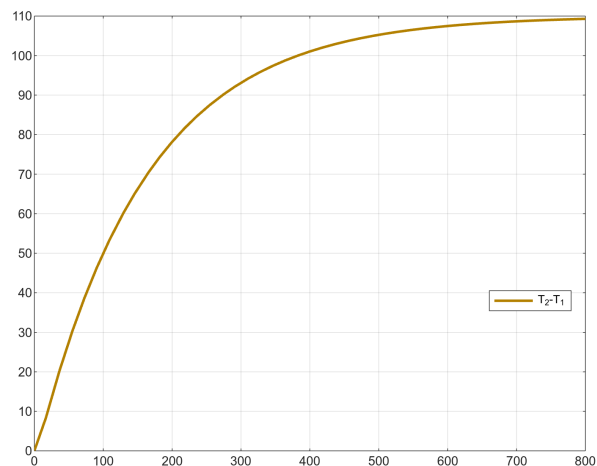
```
syms t real
fplot([cosa;t],[0 700],LineWidth=2), grid on, legend("T_1","T_2","Diferenci","T_{ext}"),
```



Limitación de velocidad de calentamiento para no superar diferencia de temperatura prefijada

La diferencia de temperatura entre capa exterior y cuerpo interior es:

```
fplot(tempdif,[0 800],LineWidth=2,Color=[.7 .5 0]), grid on, legend("T_2-T_1",Location=
```



Valor final:

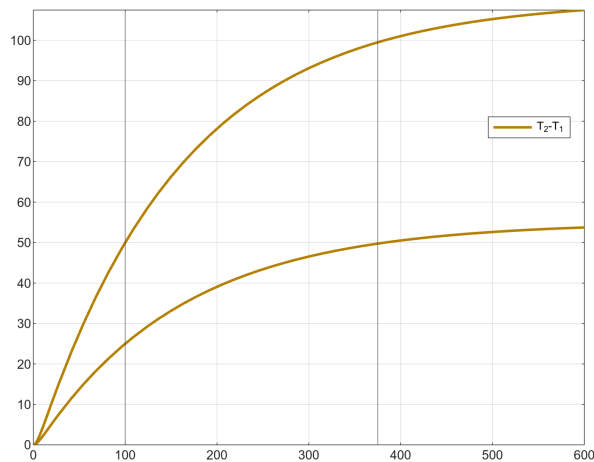
```
limit(tempdif,inf)
```

ans = 110.0

Diferencias de temperatura excesivas producen tensión mecánica debido a la dilatación diferencial de los materiales, de modo que aparecen grietas o se despegan uniones en aplicaciones prácticas.

Para que la diferencia de temperatura no supere los 50°, el tratamiento térmico con entrada rampa debería ser interrumpido a los 100 segundos, o bien, por linealidad, el tratamiento debería calentar las piezas con una tasa de calentamiento limitada, como mucho con $T_{ext} = \frac{50}{110}t$, para poder mantenerse todo el tiempo que se considerara oportuno. Obviamente, existirían opciones intermedias (por ejemplo, calentar $T_{ext} = 0.5t$ durante 375 segundos), como se observa en la figura inferior.

```
fplot([tempdif;tempdif/2],[0 600],LineWidth=2,Color=[.7 .5 0]), xline(100),xline(375),
```



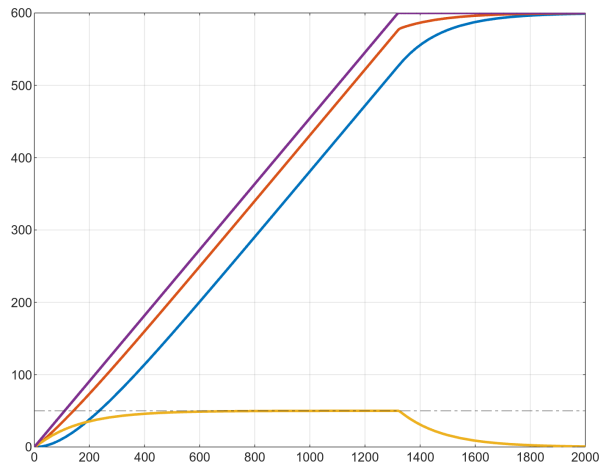
Pero si hay que llegar a los 600 grados sin superar la diferencia de temperatura límite, esas opciones "cortas" no serán válidas.

El perfil podría ser:

```
pendiente_rampa=50/110;
dur_rampa=600/pendiente_rampa
```

```
dur_rampa = 1320
```

```
larampa=pendiente_rampa*t-heaviside(t-dur_rampa)*pendiente_rampa*(t-dur_rampa);
Resp_rampa_truncada=pendiente_rampa*(cosa-subs(heaviside(t)*cosa,t,t-dur_rampa));
fplot([Resp_rampa_truncada;larampa],[0 2000],LineWidth=2, grid on
yline(50,'-.'))
```



Nota: podríamos calentar "de golpe" más rápido al principio y mantener "un poquito más" la rampa para que el cuerpo interior llegue a 600 grados más rápidamente, si ese es el objetivo del tratamiento. El perfil, obtenido por "ensayo y error", y simulando con ode45 las formas de onda "raras" (que son "complicadas de manejar en Laplace" y que "no me interesa la fórmula") podría ser:

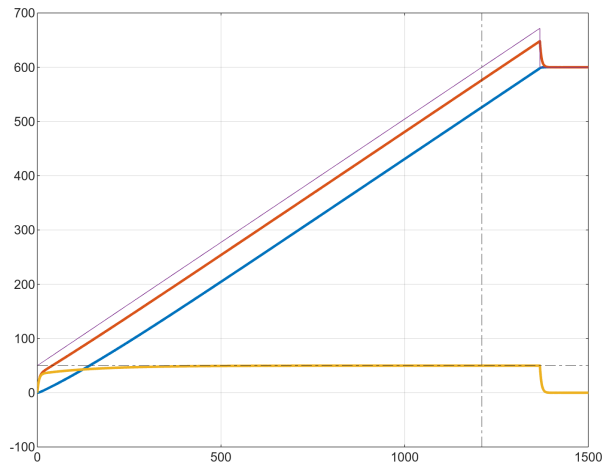
```
pendiente_rampa=50/110;
dur_rampa=550/pendiente_rampa+158
```

```
dur_rampa = 1368
```

```
larampa=50+pendiente_rampa*t*heaviside(dur_rampa-t)+heaviside(t-dur_rampa)*550;
Unum=matlabFunction(larampa);
tic
[tiempos,estados]=ode45(@(t,x) A*x+B*Unum(t),[0 1500],[0;0],odeset(RelTol=1e-5,AbsTol=1e-5));
toc
```

```
Elapsed time is 0.059449 seconds.
```

```
plot(tiempos,estados,LineWidth=2), grid on, hold on
plot(tiempos,estados*[-1;1],LineWidth=2)
plot(tiempos,Unum(tiempos))
yline(50,'-.'),xline(550/pendiente_rampa,'-.')
hold off
```



*Podría calcularse la fórmula superponiendo adecuadamente respuestas escalón (no calculada en simbólico, repetiríamos con $1/s$ en vez de $1/s^2$) y rampa (esa sí, ya calculada)... pero me da pereza y he "jugado un ratito" con `ode45`.