Modelado, simulación y linealización de un elemento calefactor de líquido de primer orden

Presentación en vídeo: http://personales.upv.es/asala/YT/V/term1lin.html

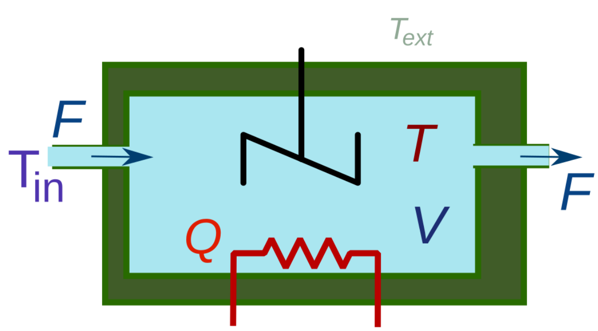
Este código funcionó correctamente con Matlab **R2020b**

Objetivo: modelar un tanque (volumen constante) por el que circula agua que es calentada por una resistencia calefactora. Simular el modelo no lineal. Linealizar alrededor de un punto de funcionamiento y simular dicho modelo linealizado.

Tabla de Contenidos

Modelo de primeros principios

Modelo de un tanque con una resistencia (proporcionando potencia calorífica Q, supuesta conocida, variable de entrada) que calienta el líquido (luego sustituiremos numéricamente los valores para el agua) que circula por dicho tanque (el agitador hace que la temperatura se iguale "rápidamente" en todo el volumen):



- Entradas:

```
syms F real %caudal de entrada
syms Tin real %temp. entrada
syms Q real %potencia calorífica de la resistencia
```

- Parámetros constantes:

```
syms V real%volumen tanque
syms rho real %densidad
syms kappa real %fugas hacia exterior (suponemos pto. Func. Text = 0)
syms Ce real %calor específico másico
```

- Estado y su derivada:

```
syms T  real %temperatura del líquido (dentro, igual a la de salida)
syms dTdt real % la derivada temporal de la vble estado
```

El modelo físico ($T_{ext} \equiv 0$, κ no dependiente de F) es:

$$\underbrace{V\rho}\, C_e \frac{dT}{dt} = \underline{F\rho} C_e T_{in} - \underline{F\rho} C_e T - \kappa T + Q$$

Lo introducimos en la Symbolic toolbox:

```
Modelo=  V*rho*Ce*dTdt == F*rho*Ce*Tin - F*rho*Ce*T - kappa*T+ Q;
```

Repr. interna:

```
dTdt_sym=simplify(solve(Modelo,dTdt),50)
```

dTdt_sym =

$$\frac{Q - T\,\kappa}{\mathrm{Ce}\,V\,\rho} - \frac{F\,(T - \mathrm{Tin})}{V}$$

Es no lineal por el producto FT y FT_{in}.

No podemos, pues, por el momento expresar el modelo como $\frac{dx}{dt} = A \cdot x + Bu$ que es la forma normalizada **lineal**.

Datos numéricos

Sustituyamos datos numéricos para un ejemplo:

```
V=0.01; %m^3
Ce=4.18e3; %J/Kg/K
rho=1000; %Kg/m^3
kappa=15; %W/K, sin considerar dependencia con el caudal (cambio coef. convección)
```

Esta es la expresión simbólica de la ecuación de estado sustituyendo los valores numéricos constantes conocidos:

```
dTdt_ev=eval(dTdt_sym); vpa(dTdt_ev,4)
```

ans = $2.392\text{e-}5\, Q - 0.0003589\, T - 100.0\, F\, (T - 1.0\, \mathrm{Tin})$

```
dTdt_num=matlabFunction(dTdt_ev) %compilada como función Matlab
```

```
dTdt_num = function_handle with value:
    @(F,Q,T,Tin)Q./4.18e+4-T.*3.588516746411483e-4-F.*(T-Tin).*1.0e+2
```

Simulación no lineal

Simulemos con ode45 ante funciones del tiempo para calor, caudal y temp. entrada.

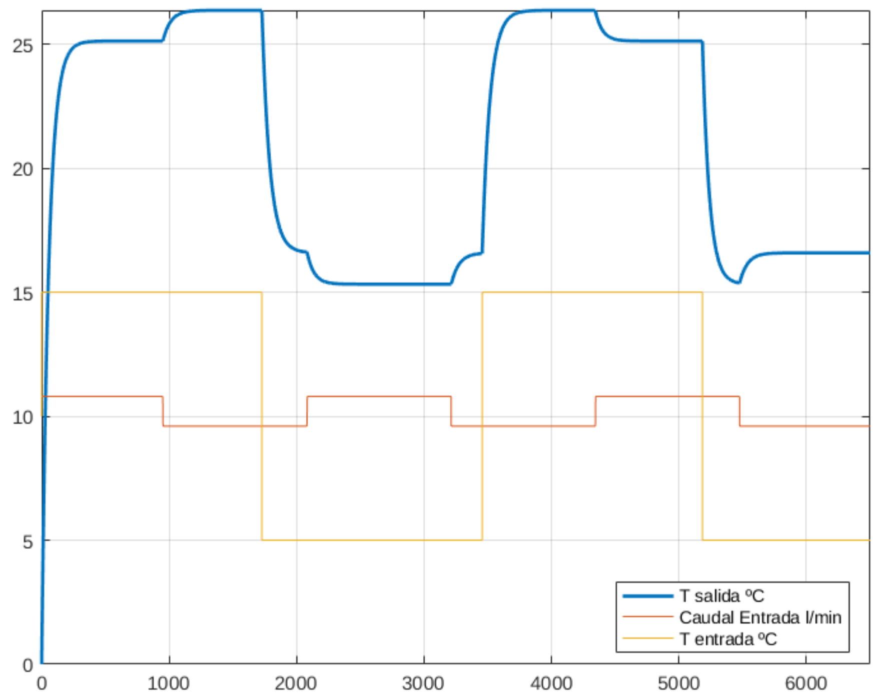
Asumiremos que las entradas están "cerca" de cierto punto de funcionamiento:

```
Q_pf=8000; %8 Kw
F_pf=0.00017; %0.17 l/s
Tin_pf=10; %incremento sobre temp. exterior, que asumimos "cero"

Q_t=@(t) Q_pf;
F_t=@(t) F_pf+0.0001*sign(sin(t/360+0.5))*0.1;%para linealización
Tin_t=@(t) Tin_pf+5*sign(sin(t/550));


options = odeset('RelTol',1e-5,'AbsTol',1e-4) ;
ModeloNum=@(t,T) dTdt_num(F_t(t),Q_t(t),T,Tin_t(t));
[tiempo,Tsol]=ode45(ModeloNum,[0 6500],0,options);


plot(tiempo,Tsol,"LineWidth",2), hold on
plot(tiempo,[F_t(tiempo)*60000 Tin_t(tiempo)]), hold off,
grid on, axis tight
legend("T salida °C","Caudal Entrada l/min","T entrada °C","location","best")
```



Observamos que:

- si disminuye caudal sube lentamente temperatura de salida, si sube caudal baja rápidamente temperatura de salida,
- si sube/baja temperatura de entrada, sube/baja temperatura de salida.

Cálculo de puntos de equilibrio

Cuando $\frac{dT}{dt} = 0$ estaremos en condiciones de equilibrio constante:

```
Tequilibrio=simplify(solve(0==dTdt_sym,T))
```

```
Tequilibrio =
```

$$\frac{Q + \mathrm{Ce}\, F\, \mathrm{Tin}\, \rho}{\kappa + \mathrm{Ce}\, F\, \rho}$$

Si $F \to \infty$ entonces $T_{equilibrio} \to T_{in}$.

Si $F \to 0$, entonces $T_{equilibrio} \to \frac{1}{15}Q$.

Sustituyendo los valores numéricos concretos (constantes) de un punto de funcionamiento fijando valores de Q, F y T_{in}:

Resulta:

```
Teqnum=eval(subs(Tequilibrio,{Q,F,Tin},{Q_pf,F_pf,Tin_pf}))
```

```
Teqnum = 20.8186
```

```
T_pf=Teqnum;
```

Linealización

Linealicemos los productos caudal*temp en el punto nominal de funcionamiento calculado arriba.

Buscamos un modelo con ecuación de estado $\frac{dx}{dt}_{1\times1} = A_{1\times1} \cdot x_{1\times1} + B_{1\times3} \cdot u_{3\times1}$, y una salida de temperatura $y_{1\times1} = C_{1\times1} \cdot x_{1\times1} + D_{1\times3} \cdot u_{3\times1}$.

Primero hacemos derivadas parciales

```
Estados=T; %primer orden, un sólo estado.
Asym=jacobian(dTdt_sym,Estados)
```

```
Asym =
```

$$-\frac{F}{V} - \frac{\kappa}{\mathrm{Ce}\, V\, \rho}$$

```
Entradas=[Q F Tin];
Bsym=jacobian(dTdt_sym,Entradas)
```

```
Bsym =
```

$$\begin{pmatrix} \frac{1}{\mathrm{Ce}\, V\, \rho} & -\frac{T - \mathrm{Tin}}{V} & \frac{F}{V} \end{pmatrix}$$

*ahora faltaría sustituir F, T y T_{in} por los valores en el ***punto de funcionamiento***.

Análisis de polos de modelo linealizado

Con $F_{p.f.} > 0$ el sistema linealizado es estable. La dinámica de la temperatura (linealizada, pequeños incrementos) es más rápida cuanto mayor sea el caudal. Se corresponde con la simulación.

Sin disipación κ, la constante de tiempo del modelo linealizado es igual al tiempo de residencia nominal $\frac{V}{F_{p.f.}}$.

Con disipación, como κ crece con la superficie, el factor $\frac{\kappa}{V}$ en el polo disminuye con el tamaño de la instalación (tamaño^2/tamaño^3).

Valores numéricos del modelo linealizado

Sustituimos el punto de funcionamiento y evaluamos:

Si sustituimos los parámetros constantes, queda:

```
Anum=eval(subs(Asym,{Q,F,Tin,T},{Q_pf,F_pf,Tin_pf,Teqnum}))
```

```
Anum = -0.0174
```

```
Bnum=eval(subs(Bsym,{Q,F,Tin,T},{Q_pf,F_pf,Tin_pf,Teqnum}))
```

```
Bnum = 1×3
10^3 ×
    0.0000   -1.0819    0.0000
```

Haremos un cambio de unidades, con Q en kW, F en l/m:

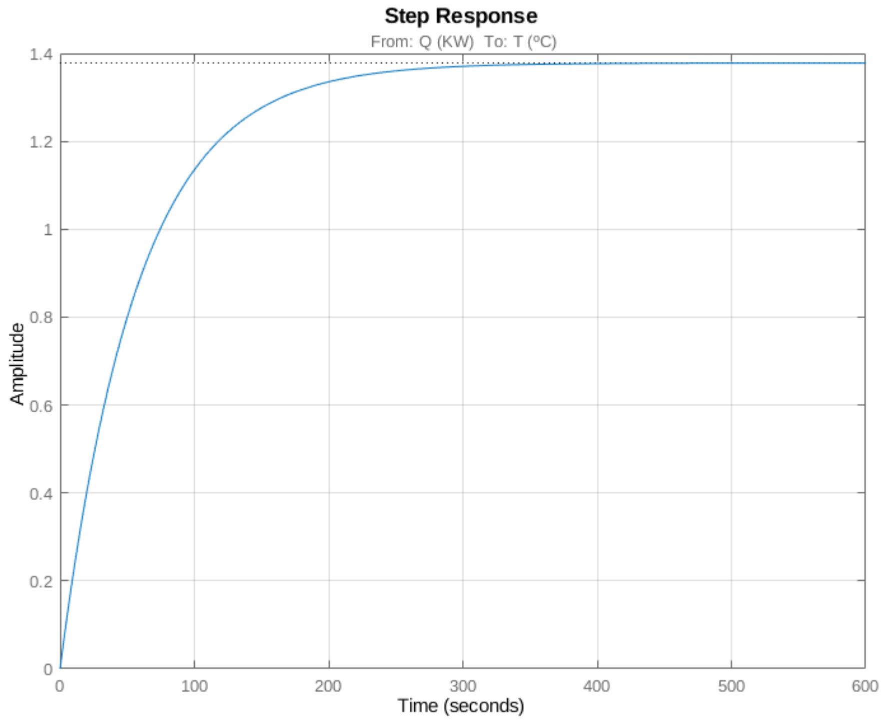
```
Bnum2=Bnum*diag([1000 0.001/60 1])
```

```
Bnum2 = 1×3
    0.0239   -0.0180    0.0170
```

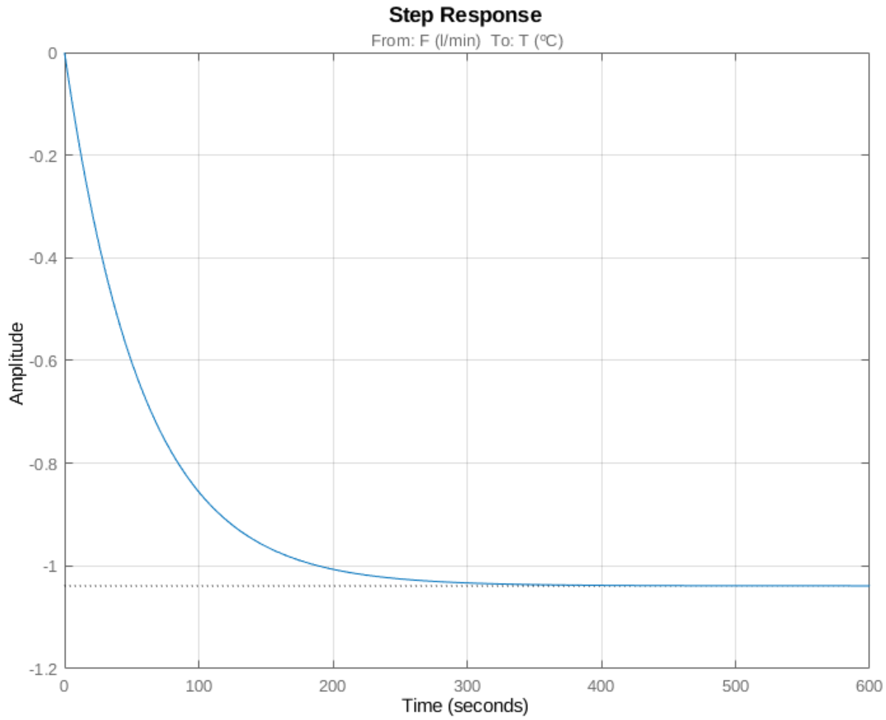
```
sysLin=ss(Anum,Bnum2,1,0); %y=Cx+Du  ---> y=1*T+0*u
```

Nota sobre ec. salida: la matriz D debería ser 3×1, esto es $[0\ \ 0\ \ 0]$, porque hay 3 entradas. Al poner un cero como cuarto argumento en el comando `ss`, Matlab crea una matriz de ceros de las dimensiones adecuadas.

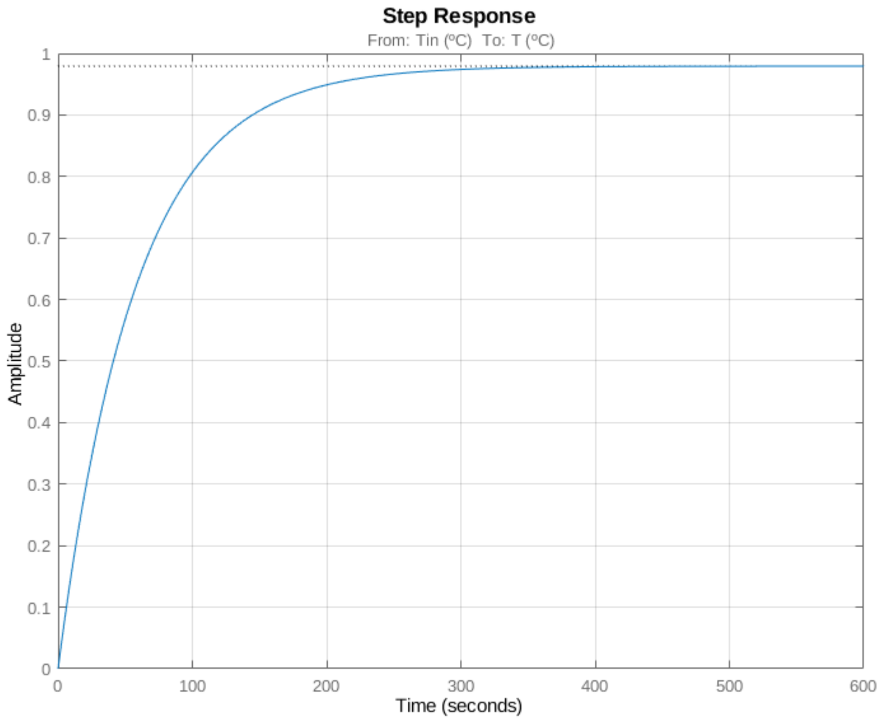
```
%sysLin.InputName={'Q (W)','F (m^3/s)','Tin (°C)'};
sysLin.InputName={'Q (KW)','F (l/min)','Tin (°C)'};
sysLin.OutputName='T (°C)';
step(sysLin(1)), grid on
```




```
step(sysLin(2)), grid on
```




```
step(sysLin(3)), grid on
```



Los valores finales (incrementales) de temperatura los podemos obtener de $0 = Ax_{eq} + Bu_{eq}$, $x_{eq} = -A^{-1}Bu_{eq}$.

La matriz $-A^{-1}B$ se denomina matriz de ganancia estática, y es lo que multiplica a cada incremento de potencia, caudal o temperatura de entrada:

```
Gan=-inv(Anum)*Bnum2
```

```
Gan = 1×3
    1.3782   -1.0387    0.9793
```

```
dcgain(sysLin) %este es el comando Matlab para obtener la ganancia estática
```

```
ans = 1×3
    1.3782   -1.0387    0.9793
```

Los elementos de la matriz de ganancia estática tendrán unidades de $ºC/kW$, $ºC/(l/m)^{-1}$, $ºC/ºC$

Comparación linealizado/no lineal

```
incrQ_t=@(t) Q_t(t)-Q_pf;
incrF_t=@(t) F_t(t)-F_pf;
incrTin_t=@(t) Tin_t(t)-Tin_pf;
%B lo pongo en las unidades originales, como se hizo la no lineal
dincT_dt_linealizado=@(incF,incQ,incT,incTin) ...
    Anum*incT+Bnum*[incQ;incF;incTin];
```

```
ModeloNumL=@(t, incrT) ...
    dincT_dt_linealizado(incrF_t(t),incrQ_t(t),incrT,incrTin_t(t));
[tiempo2,incrTsol]=ode45(ModeloNumL,[0 6500],0,options);

plot(tiempo,Tsol,"LineWidth",2), hold on
plot(tiempo,[F_t(tiempo)*60000 Tin_t(tiempo)])

plot(tiempo2,[incrTsol+T_pf],'LineWidth',2), grid on, axis tight
legend("T salida °C","Caudal Entrada l/min","T entrada °C","Tlinealizado °C","location'
hold off
```