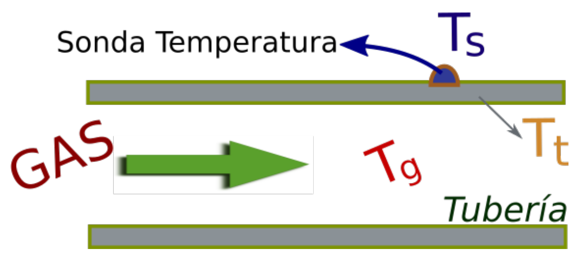
Ejemplo filtro de Kalman para estimar temperatura del gas que circula por una tubería

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

Este código ejecutó sin errores en Matlab **R2019a**

Modelado



Suponiendo T_g (temperatura gas) y T_a (temperatura ambiente) como entradas, tendríamos unas constantes de transmisión de calor y capacidades caloríficas que darían lugar a la ecuación de estado:

$$\frac{d}{dt}\begin{pmatrix} T_s \\ T_t \end{pmatrix} = \begin{pmatrix} (-k_{st}-k_{sa})/C_s & k_{st}/C_s \\ k_{st}/C_t & (-k_{st}-k_{ta}-k_{tg})/C_t \end{pmatrix}\begin{pmatrix} T_s \\ T_t \end{pmatrix} + \begin{pmatrix} 0 & k_{sa}/C_s \\ k_{tg}/C_t & k_{ta}/C_t \end{pmatrix}\begin{pmatrix} T_g \\ T_a \end{pmatrix}$$

Nota: este es un modelo simplificado de parámetros concentrados; la transmisión de calor en una tubería se rige por ecuaciones en derivadas parciales fuera de los objetivos de este modelado. Supondremos, por tanto, que la tubería es "corta" y que las pérdidas de entalpía del gas (calor transmitido a la tubería) son despreciables.

Vamos a dar valores numéricos

```
kst=.3; ktg=.25; kta=.08; ksa=0.01; Cs=.4; Ct=1.5;
A=diag([1/Cs 1/Ct])*[-kst-ksa kst; kst -ktg-kta-kst];
B=diag([1/Cs 1/Ct])*[0 ksa; ktg kta];
sys=ss(A,B,eye(2),0);
sys.InputName={'Tgas','Tambiente'};
sys.OutputName={'Tsonda','Ttuberia'};
```

Incorporación de modelo de perturbaciones

```
anchobandagas=0.16; s=tf('s');
GeneradorTgas=1/(s+anchobandagas)^2; %multipl. por ruido blanco...
```

Por simplicidad, descartemos variaciones de Tamb...

```
sys2=ss([sys(:,1);1]*GeneradorTgas);
sys2.OutputName{3}='Tgas';
```

Discretizamos:

```
Ts=0.2; sysd=c2d(sys2,Ts,'zoh');
%existe una forma teóricamente mejor de discretizar ruidos continuos,
%omitida por brevedad
```

Análisis de propiedades

```
Eba=eig(sysd.a)' %polos bucle abierto
```

```
Eba = 1×4
    0.8149    0.9663    0.9685    0.9685
```

```
Q=300; %varianza ruido de proceso
Zba=gram(sysd,'c')*Q %vzas-covarianzas estado en bucle abierto:
```

Zba = 4×4

10^3 ×

```
    1.4134    1.4605   -0.4614    0.9439
    1.4605    1.5306   -0.4158    1.0523
   -0.4614   -0.4158    1.4994    0.0002
    0.9439    1.0523    0.0002    0.9154
```

```
VzaSalidas=sysd.c*Zba*sysd.c'
```

VzaSalidas = 3×3

10^3 ×

```
    1.4134    1.4605    1.8878
    1.4605    1.5306    2.1045
    1.8878    2.1045    3.6618
```

```
dt_salidas=sqrt(diag(VzaSalidas))'
```

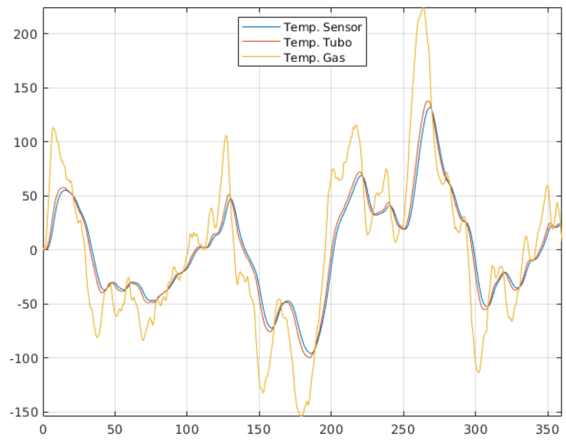
```
dt_salidas = 1×3
   37.5947   39.1229   60.5128
```

```
dt_Tgas=dt_salidas(3) %desviación. tip. temperatura gas a priori (sin sensores)
```

```
dt_Tgas = 60.5128
```

Simulación (bucle abierto)

```
N=1800;Tiempo=(0:(N-1))*Ts;
perturbacionRuidoBlanco=randn(1,N)*sqrt(Q);
Y=lsim(sysd,perturbacionRuidoBlanco);
plot(Tiempo,Y), grid on, legend('Temp. Sensor','Temp. Tubo','Temp. Gas','Location','bes
```



Filtro de Kalman estacionario

Con un sensor "horrible (vza 1e9)" coincide con la estimación del estado en bucle abierto

```
Csensor=sysd.c(1,:);  %medimos la salida 1 (sonda)
[Mhorrible,~,Zhorrible,Ehorrible]=dlqe(sysd.a,sysd.b,Csensor,Q,1e9);
Mhorrible'
```

```
ans = 1×4
10^-5 ×
    0.1413    0.1460   -0.0461    0.0944
```

```
Zhorrible
```

```
Zhorrible = 4×4
10^3 ×
    1.4132    1.4604   -0.4613    0.9439
    1.4604    1.5305   -0.4158    1.0522
   -0.4613   -0.4158    1.4993    0.0002
    0.9439    1.0522    0.0002    0.9154
```

```
Zba %esto era lo que obtenía el gramiano
```

```
Zba = 4×4
10^3 ×
    1.4134    1.4605   -0.4614    0.9439
    1.4605    1.5306   -0.4158    1.0523
   -0.4614   -0.4158    1.4994    0.0002
    0.9439    1.0523    0.0002    0.9154
```

```
Ehorrible'
```

```
ans = 1×4 complex
   0.9686 - 0.0006i   0.9686 + 0.0006i   0.9661 + 0.0000i   0.8149 + 0.0000i
```

```
fliplr(Eba)
```

```
ans = 1×4
    0.9685    0.9685    0.9663    0.8149
```

con un sensor "perfecto" tendería a la estimación tiempo mínimo (polos muy rápidos)

```
[Mperfecto,~,Zperfecto,Eperfecto]=dlqe(sysd.a,sysd.b,Csensor,Q,4e-11);
Mperfecto %ganancia "enorme"
```

```
Mperfecto = 4×1
10^3  ×
    0.0010
    0.0152
    9.8336
    0.3373
```

La varianza del estado estimado con ese sensor perfecto:

```
Zperfecto %pero al ser de orden>1, el ruido de proceso no puede eliminarse
```

```
Zperfecto = 4×4
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0207    0.0002
    0.0000    0.0207  148.8296    1.5397
    0.0000    0.0002    1.5397    0.0161
```

La varianza del las tres salidas (sonda,tubo,gas) estimadas (medidas sólo es la primera) con ese sensor ideal:

```
VzaErrorEstimSalidasPerfecto=sysd.c*Zperfecto*sysd.c'
```

```
VzaErrorEstimSalidasPerfecto = 3×3
    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0004
    0.0000    0.0004    0.0646
```

Esta sería la desviación típica de las salidas estimadas (sonda,tubo,gas) en grados:

```
dt_perfecto=sqrt(diag(VzaErrorEstimSalidasPerfecto))'
```

```
dt_perfecto = 1×3
    0.0000    0.0017    0.2541
```

```
Eperfecto' %tres polos rápidos, 1 más lento
```

```
ans = 1×4
   -0.8982   -0.1252   -0.0769   -0.0020
```

Diseñemos un observador en bucle cerrado con un sensor "normal", ni muy bueno ni muy malo:

```
desvtipsensor=0.6;
R=desvtipsensor^2;%desv. típica ruido medida sensor al cuadrado
[M,~,Z,E]=dlqe(sysd.a,sysd.b,Csensor,Q,R);
M %ganancia observador
```

```
M = 4×1
    0.3736
    0.9657
    9.2138
    3.2496
```

```
Z %vza estado estimado
```

```
Z = 4×4
10^3 ×
    0.0001    0.0003    0.0033    0.0012
    0.0003    0.0011    0.0173    0.0046
    0.0033    0.0173    1.0155    0.1130
    0.0012    0.0046    0.1130    0.0227
```

```
sysd.c*Z*sysd.c' %matriz vzas-cov salidas estimadas (sonda,tubo,gas)
```

```
ans = 3×3
    0.1345    0.3477    2.3397
    0.3477    1.1239    9.2431
    2.3397    9.2431   90.6515
```

```
E
```

```
E = 4×1 complex
   0.8701 + 0.2101i
   0.8701 - 0.2101i
   0.7565 + 0.0718i
   0.7565 - 0.0718i
```

el comando kalman da el mismo resultado, y forma el observador:

```
[Observador,~,~,Mx,Z,~]=kalman(sysd(1,:),Q,R); %elegimos la primera salida para kalman
eig(Observador) %polos del observador
```

```
ans = 4×1 complex
   0.8701 + 0.2101i
   0.8701 - 0.2101i
   0.7565 + 0.0718i
   0.7565 - 0.0718i
```

```
Mx %ganancia
```

```
Mx = 4×1
    0.3736
    0.9657
    9.2138
    3.2496
```

```
Z %vza estado interno
```

```
Z = 4×4
10^3 ×
    0.0001    0.0003    0.0033    0.0012
    0.0003    0.0011    0.0173    0.0046
    0.0033    0.0173    1.0155    0.1130
    0.0012    0.0046    0.1130    0.0227
```

```
VzaSalidasKalm=sysd.c*Z*sysd.c' %vza salidas
```

```
VzaSalidasKalm = 3×3
    0.1345    0.3477    2.3397
    0.3477    1.1239    9.2431
    2.3397    9.2431   90.6515
```

```
dt_Kalm=sqrt(diag(VzaSalidasKalm))' %de sonda, tubo, gas
```

```
dt_Kalm = 1×3
    0.3667    1.0601    9.5211
```

```
dt_estimgas_consensor=dt_Kalm(3) %con ese sensor, esta es la d.t. del estimado
```

```
dt_estimgas_consensor = 9.5211
```

```
dt_estimgas_consensor/dt_Tgas %mejoramos 6 veces la precisión respecto a b.a.
```

```
ans = 0.1573
```

Simulación del filtro de Kalman

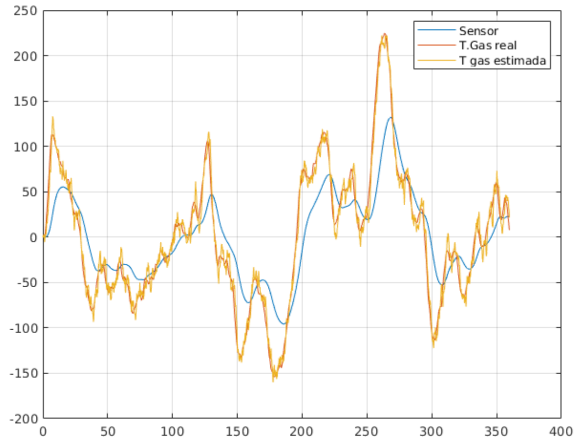
```
size(Observador)
```

```
State-space model with 5 outputs, 1 inputs, and 4 states.
```

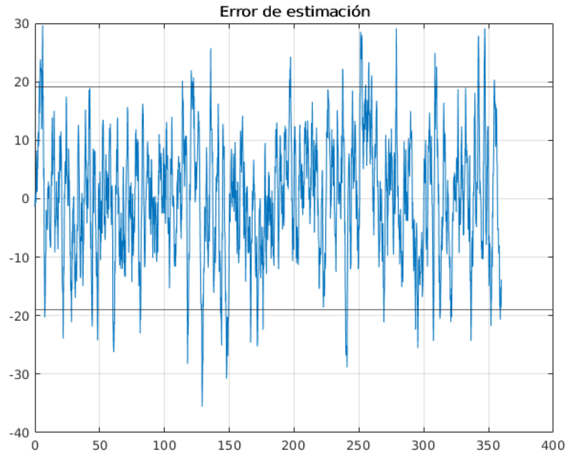
```
Observador.OutputName'
```

```
ans = 1×5 cell array
    {'Tsonda_e'}    {'x1_e'}    {'x2_e'}    {'x3_e'}    {'x4_e'}
```

```
ObservadorEstado=Observador(2:end,:);
ObservadorTemperaturas=sysd.c*ObservadorEstado;
ObservadorTemperaturas.OutputName={'Tsonda_e','Ttubo_e','Tgas_e'};
%añadimos a la simulación de Tsonda el  ruido de medida
TgasEstim=lsim(ObservadorTemperaturas(3),Y(:,1)+randn(size(Tiempo'))*desvtipsensor);
plot(Tiempo,[Y(:,1) Y(:,3) TgasEstim]), grid on, legend('Sensor','T.Gas real','T gas es
```




```
plot(Tiempo,Y(:,3)-TgasEstim), grid on, title('Error de estimación')
yline(-dt_estimgas_consensor*2);yline(dt_estimgas_consensor*2);
```


Error de estimación
30
20
10
0
-10
-20
-30
-40
0
50
100
150
200
250
300
350
400