Comparación de observadores (orden completo, orden reducido): ejemplo masa (doble integrador)

Presentaciones en vídeo:

http://personales.upv.es/asala/YT/V/roobsml1.html , http://personales.upv.es/asala/YT/V/roobsml2.html .

Este código funcionó con Matlab **R2021a**

Objetivos: Diseñar observadores de orden completo y orden reducido por asignación de polos. Comparar sus respuestas en tiempo y frecuencia.

Tabla de Contenidos

Modelo

```
A=[0 1;0 0]; B=[0;1]; C=[1 0];D=0;
sys=ss(A,B,eye(2),D);
sys.InputName='u (Force)';
sys.OutputName={'pos','vel'};
tf(sys)
```

```
ans =

  From input "u (Force)" to output...
          1
   pos:  ---
         s^2

         1
   vel:  -
         s

Continuous-time transfer function.
```

Diseño de observadores (asign. polos)

```
polodeseado=-2;
t_est_deseado=-4/polodeseado  % 98%
```

```
t_est_deseado = 2
```

Observador de orden completo

```
Lcomplet=place(A',C',polodeseado*[1 2])'
```

```
Lcomplet = 2×1
    6.0000
    8.0000
```

```
%pondremos un segundo polo el doble de rápido que el dominante.
eig(A-Lcomplet*C)
```

```
ans = 2×1
   -4.0000
   -2.0000
```

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}) = (A - LC)\hat{x} + Bu + Ly$$

```
observador_ordencompleto = ss(A-Lcomplet*C,[B Lcomplet],eye(2),0);
observador_ordencompleto.InputName={'u (Force)','pos medida'};
observador_ordencompleto.OutputName={'pos estim','vel estim'};
```

Observador de orden reducido

Repitiendo la teoría (sin matriz de transformación T porque $y = x_1$, pues podemos decir que el modelo dirá $\dot{v} = u$, pero podemos poner un "observador" $\frac{d\hat{v}}{dt} = u + L(\frac{dy}{dt} - \hat{v})$.

Para que sea realizable, $\frac{d(\hat{v} - Ly)}{dt} = u - L\hat{v}$, con lo que definiremos $w := \hat{v} - Ly$, esto es, una ecuación de estado:

$$\frac{dw}{dt} = u - L(w + Ly) = -Lw + u - L^2 y,$$

y ecuación de salida $\hat{v} = w + Ly$.

Fórmulas "de teoría" genéricas

Sea el proceso:

$$\begin{bmatrix}\dot{\overline{x}}_1\\ \dot{\overline{x}}_2\end{bmatrix}=\begin{bmatrix}\overline{A}_{11} & \overline{A}_{12}\\ \overline{A}_{21} & \overline{A}_{22}\end{bmatrix}\begin{bmatrix}\overline{x}_1\\ \overline{x}_2\end{bmatrix}+\begin{bmatrix}\overline{B}_1\\ \overline{B}_2\end{bmatrix}u;\qquad y=\overline{x}_1$$

Entonces, el observador de orden reducido tiene las ecuaciones de estado y de salida dadas por:

$$\frac{dw}{dt}=(\bar{A}_{22}-L\bar{A}_{12})w+\left((\bar{A}_{22}-L\bar{A}_{12})L+\bar{A}_{21}-L\bar{A}_{11}\;\;\bar{B}_2-L\bar{B}_1\right)\begin{pmatrix}y\\ u\end{pmatrix}$$
$$\hat{x}_2=w+Ly$$

```
L_red=place(A(2,2)',A(1,2)',polodeseado)'
```

```
L_red = 2
```

```
Aobs=A(2,2)-L_red*A(1,2)
```

```
Aobs = -2
```

```
Bobs_u=B(2)-L_red*B(1)
```

```
Bobs_u = 1
```

```
Bobs_y=Aobs*L_red+A(2,1)-L_red*A(1,1)
```

```
Bobs_y = -4
```

```
Cobs=1
```

```
Cobs = 1
```

```
Dobs_u=0;
Dobs_y=L_red
```

```
Dobs_y = 2
```

```
obs_ord_red=ss(Aobs,[Bobs_u Bobs_y],Cobs, [Dobs_u Dobs_y]);
obs_ord_red.InputName={'u (Force)','pos medida'};
obs_ord_red.OutputName='vel_estim';
```

Análisis en función de transferencia y resp. en frecuencia

*Nota: este análisis no es necesario para la implementación, dado que están pensados para un entorno "state space" con condiciones iniciales no nulas. El objetivo es, simplemente, ver qué ecuaciones tienen una vez el efecto de las condiciones iniciales ha desaparecido, para "aprender"... pero podría omitirse esta sección sin ningún problema.

```
s=tf('s');
zpk(observador_ordencompleto)
```

```
ans =

  From input "u (Force)" to output...
                   1
   pos estim:  -----------
```

```
              (s+4) (s+2)

                 (s+6)
   vel estim:  -----------
              (s+4) (s+2)

  From input "pos medida" to output...
              6 (s+1.333)
   pos estim:  -----------
              (s+4) (s+2)

                  8 s
   vel estim:  -----------
              (s+4) (s+2)

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

```
zpk(obs_ord_red)
```

```
ans =

  From input "u (Force)" to output "vel_estim":
    1
  -----
  (s+2)

  From input "pos medida" to output "vel_estim":
   2 s
  -----
  (s+2)

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

```
minreal(tf(obs_ord_red)*[1;1/s^2])
```

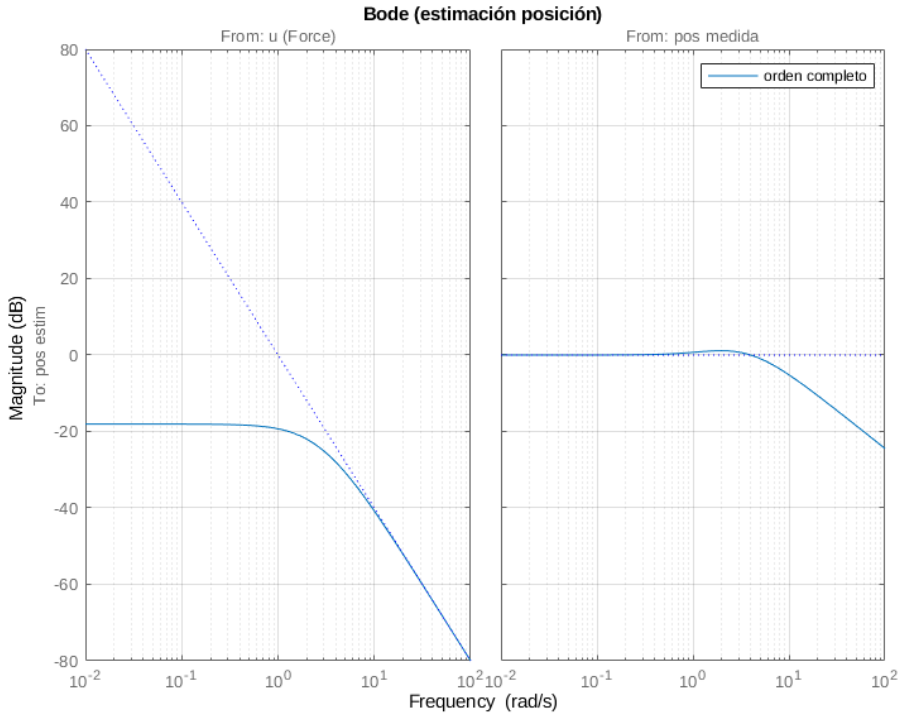
```
ans =

  From input to output "vel_estim":
  1
  -
  s

Continuous-time transfer function.
```

```
minreal(zpk(observador_ordencompleto)*[1;1/s^2])
```

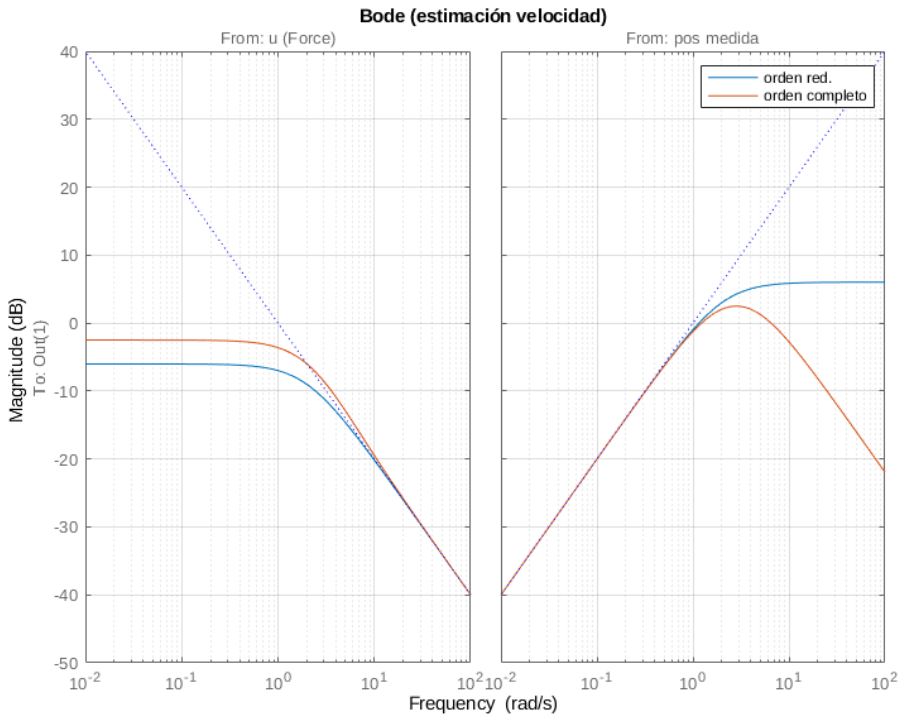
```
ans =

  From input to output...
               1
   pos estim:  ---
               s^2

               1
   vel estim:  -
               s

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

Resp. en frecuencia (estacionaria)

```
bodemag(observador_ordencompleto(1,:),[1/s^2 1],':'), grid on, title('Bode (estimación
legend("orden completo")
```




```
bodemag(obs_ord_red,observador_ordencompleto(2,:),[1/s s],':'), grid on, title('Bode (e
legend("orden red.","orden completo")
```



Simulación temporal

Generamos forma de onda de entrada, y simulamos para obtener una serie de datos.

La simulación de los ruidos "en tiempo continuo" no es nada "rigurosa" desde el punto de vista formal, pero nos vale "cualitativamente" con `lsim` discretizando a 0.02 segundos.

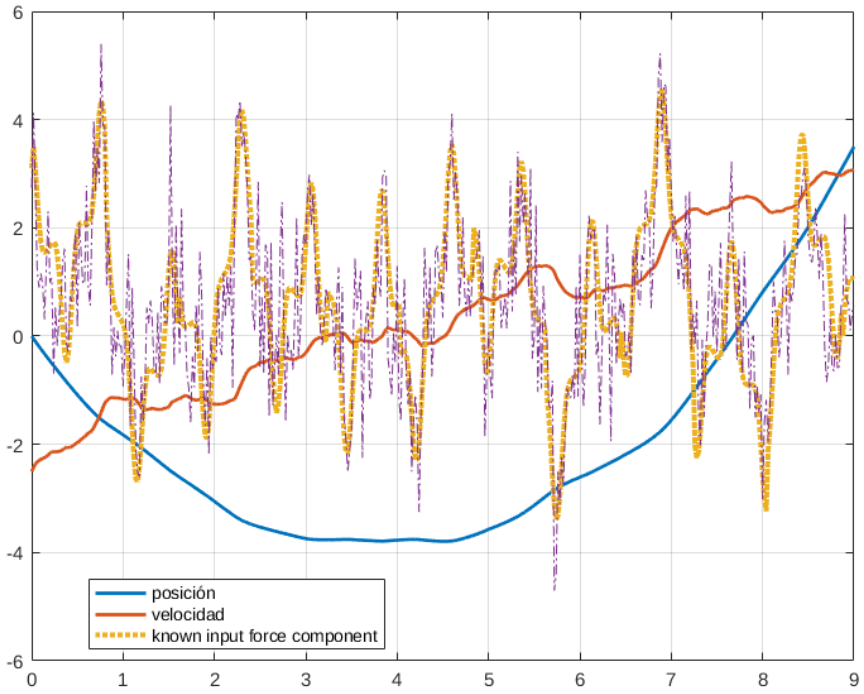
```
T=0:0.02:9;
u=sin(T*3)+0.5*sign(sin(T*3.9))+2.5*cos(T*8.2).^5+.6;

simularconruido=1;


process_noise=simularconruido*(randn(size(u))*0.9-sign(u).*min(0.45,abs(u)));%fuerzas c
size(u)
```

```
ans = 1×2
     1   451
```

```
y=lsim(sys,u+process_noise,T,[0;-2.5]);
size(y)
```

```
ans = 1×2
   451     2
```

```
plot(T,y,LineWidth=2), grid on, hold on,
plot(T,u,':',LineWidth=3)
plot(T,u+process_noise,'-.')
hold off, legend("posición","velocidad","known input force component",Location="best")
```



Contaminaremos también con ruido de medida:

```
posmed = y(:,1) + simularconruido*0.08*randn(size(y(:,1)));
entradaobs=[u' posmed];
size(entradaobs)
```

```
ans = 1×2
   451     2
```

La salida del observador es el estimado de la velocidad:

El de orden completo es:

```
x_obs_completo=lsim(observador_ordencompleto,entradaobs,T,[0;0]);
```

El de orden reducido es:

```
vest_obs_red=lsim(obs_ord_red,entradaobs,T,0);
```

Las ventajas de los observadores se verán claras al comparar con un "derivador * filtro con el polo deseado":

```
s=tf('s');
Filtro_a_puro_ojo = s * 1/(s/(-polodeseado)+1)
```

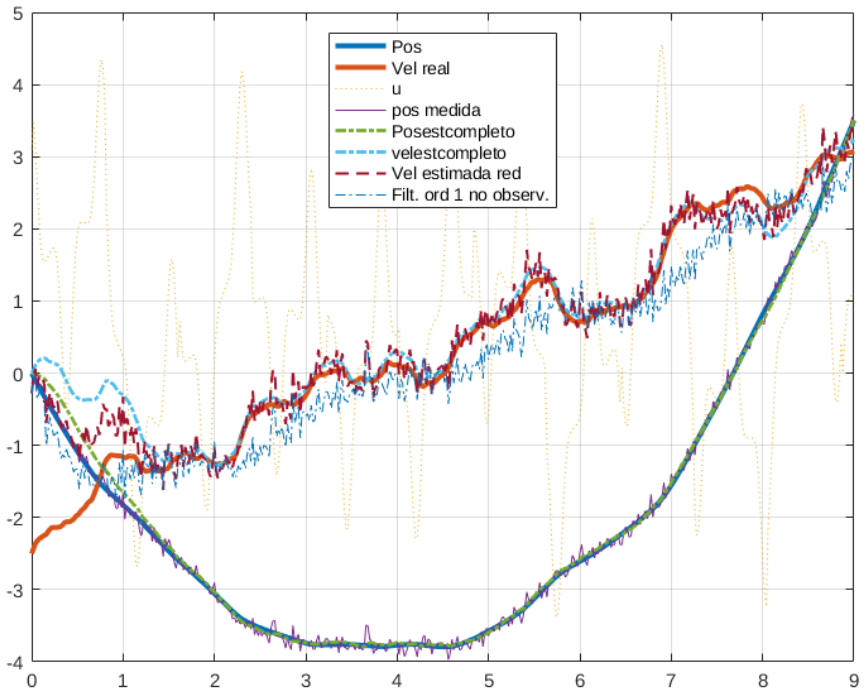
```
Filtro_a_puro_ojo =

   2 s
```

```
  -----
  s + 2

Continuous-time transfer function.
```

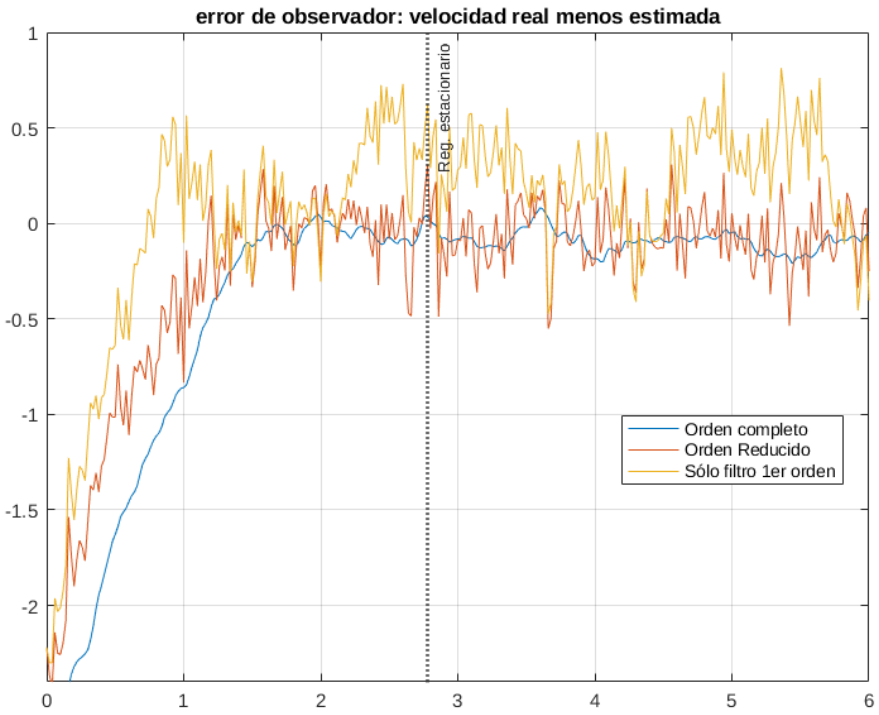
```
vest_solofiltrado=lsim(Filtro_a_puro_ojo,posmed,T,0);
```

Representemos todo gráficamente:

```
plot(T,y,LineWidth=3), hold on, grid on
plot(T,u,':')
hold on
plot(T,posmed)
plot(T,x_obs_completo,'-.',LineWidth=2)
plot(T,vest_obs_red,'--',LineWidth=1.5)
plot(T,vest_solofiltrado,'-.')
legend('Pos','Vel real','u','pos medida','Posestcompleto','velestcompleto','Vel estimac
hold off
```




```
err_complet=y(:,2)-x_obs_completo(:,2);
err_red=y(:,2)-vest_obs_red;
err_filtr=y(:,2)-vest_solofiltrado;
plot(T,[err_complet err_red err_filtr]), grid on,
title("error de observador: velocidad real menos estimada"),xlim([0 6]), ylim([-2.4 1])
legend("Orden completo","Orden Reducido","Sólo filtro 1er orden",Location="best")
```



● Filtro 1er orden vs. orden reducido tienen más o menos la misma varianza a alta frecuencia debida a ruidos, pero el filtro tiene además un error "sistemático" al no incorporar el hecho de que conocer la fuerza de entrada te permite calcular mejor la velocidad.

● El obs. de orden completo es de orden 2, tiene un transitorio más lento, pero una variabilidad a alta frecuencia mucho menor.

● ambos observadores no tienen error "sistemático" si los ruidos de proceso son ruido blanco de media cero (sí lo tienen si hay no-linealidades, derivas, etc.)

La integral del error cuadrático (una vez ha pasado el transitorio) es (informalmente, lo consideraremos una aproximación a "varianza por ruido" + "error sistemático^2"):

```
sum(err_complet(140:end).^2)
```

```
ans = 6.6463
```

```
sum(err_red(140:end).^2)
```

```
ans = 11.2354
```

```
sum(err_filtr(140:end).^2)
```

```
ans = 49.9758
```

Conclusiones

Hemos aplicado la teoría de observador de orden completo y orden reducido a un sistema (doble integrador) que representa una masa de 1 Kg sujeta a una fuerza de entrada conocida, para estimar únicamente la velocidad en el caso de orden reducido. Hemos comparado resultados con/sin ruidos.