

Estimación de la fuerza que actúa sobre un sistema mecánico: filtro H-infinito y comparación con Kalman (H2)

© 2020, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Este código ejecutó correctamente en Matlab R2020a

Vídeo presentación disponible en <http://personales.upv.es/asala/YT/V/hinfobs1.html> y <http://personales.upv.es/asala/YT/V/hinfobs2.html>

Objetivo:

El filtro de Kalman estacionario puede ser obtenido como un problema de optimización $\mathcal{H}_2$.

Con la misma planta generalizada, podrá ser obtenida la solución $\mathcal{H}_\infty$. Variando los pesos haremos modificaciones de diseño y compararemos con el filtro de Kalman ($\mathcal{H}_2$).

Modelo

```
polopert=.4; %fuerzas no consignadas... son un ancho de banda limitado...
Gs=ss([0 1 0;-5 -1 2;0 0 -polopert],[0 0;2 0;0 polopert],[1 0 0],0);
zpk(Gs)
```

ans =

```
From input 1 to output:
      2
```

```
-----
(s^2 + s + 5)
```

```
From input 2 to output:
      0.8
```

```
-----
(s+0.4) (s^2 + s + 5)
```

Continuous-time zero/pole/gain model.

```
Ts=0.04;
Gdiscr=c2d(Gs,Ts,'zoh'); %hacemos el discreto...
A=Gdiscr.A;B=Gdiscr.B(:,1);G=Gdiscr.B(:,2);%matrices ec. estado
C=Gdiscr.C; D=Gdiscr.D(1); %matrices ec. salida
%Desviación típica ruido de proceso (Q) y medida (R):
desvtip_ruidomed=0.05;
desvtip_ruidoproc=5;
%Valrianzas ruido de proceso (Q) y medida (R):
R=desvtip_ruidomed^2;
Q=desvtip_ruidoproc^2;
```

1.- Filtro H2 (equiv. filtro de Kalman estacionario)

Calculamos el filtro de Kalman discreto:

```
[elestimadorKalestac,~,~,~,Z]=kalman(Gdiscr,Q,R,0,1,1);
```

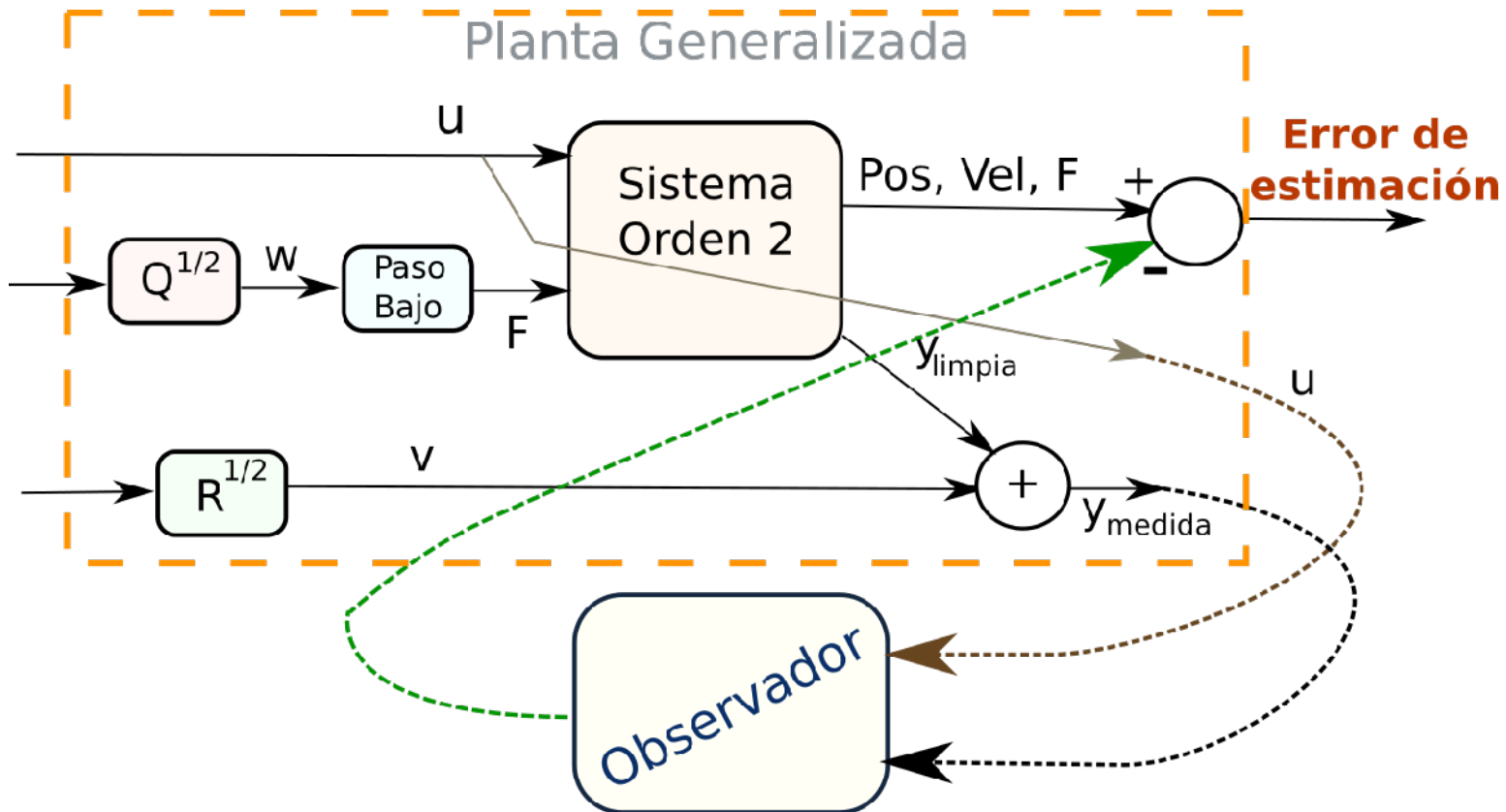
Las salidas del estimador son la medida filtrada y el estado interno estimado:

```
elestimadorKalestac.OutputName'
```

```
ans = 1x4 cell  
'y1_e'      'x1_e'      'x2_e'      'x3_e'
```

```
estimadorEstadoKalman=elestimadorKalestac(2:4,:);
```

Ahora lo calculamos mediante optimización $\mathcal{H}_2$ sobre un problema de planta generalizada:



```
Bgen=[B G zeros(3,1) zeros(3,3)];  
Cgen=[eye(3);zeros(1,3); C];  
Dgen=[zeros(3,3) -eye(3);1 0 0 0 0 0;0 0 1 0 0 0];  
PlantaGen=ss(A,Bgen,Cgen,Dgen,Ts);  
PlantaGen.InputName={'u','w','v','x1estim','x2estim','x3estim'};  
PlantaGen.OutputName={'observerr1','observerr2','observerr3','u','ymedida'};  
  
Win1=blkdiag(1,desvtip_ruidoproc,desvtip_ruidomed);  
Wout1=eye(3);  
Wout=blkdiag(Wout1,1,1);  
Win=blkdiag(Win1,eye(3));  
PlantaGenPonderada=Wout*PlantaGen*Win;  
[OO2,CL,GAM2,INFO]=h2syn(PlantaGenPonderada,2,3);  
GAM2 %desv típica err. pred de estado si entradas ruido blanco vza 1.
```

```
GAM2 = 0.2658
```

```
OO2.InputName={'u','ymedida'};
```

Comparemos con el resultado del filtro de Kalman:

```
norm(OO2-estimadorEstadoKalman,inf) %idénticos, pico resp. frec. diferencia =0
```

```
Warning: Ignoring all input names because of name conflicts.  
ans = 7.5728e-14
```

```
sqrt(trace(Z)) %raíz cuadrada de "varianza total"
```

```
ans = 0.2658
```

```
sqrt(diag(Z))' %desviación típica
```

```
ans = 1×3  
0.0240 0.1090 0.2413
```

```
PlantaYObs2=lft(PlantaGen*Win,OO2);  
norm(PlantaYObs2(1,:)) %en bucle cerrado la desv. típica de x1 estimada es:
```

```
ans = 0.0240
```

```
norm(PlantaYObs2(2,:)) %en bucle cerrado la desv. típica de x2 estimada es:
```

```
ans = 0.1090
```

```
norm(PlantaYObs2(3,:)) %en bucle cerrado la desv. típica de x3 estimada es:
```

```
ans = 0.2413
```

Para ver la utilidad del sensor, comparamos con la predicción en bucle abierto de la vza de x(3) si entradas ruido blanco vza 1:

```
norm(PlantaGenPonderada(1,2:3))
```

```
ans = 0.2007
```

```
norm(PlantaGenPonderada(2,2:3))
```

```
ans = 0.2398
```

```
EfecPertX3=PlantaGenPonderada(3,2:3);  
stdx3=norm(EfecPertX3)
```

```
stdx3 = 0.4472
```

2.- Filtro H-infinito

Aunque "ruido blanco" no es lo mismo que "límite de amplitud en cada frecuencia", cogemos mismos pesos de ENTRADA que H2.

Vamos a ver distintas opciones de pesos de salida

```

prueba=1;s=tf('s');
switch prueba
    case 1
        Wout1=eye(3); %observador "tal cual" minimizar norma de x estimado.
    case 2
        Wout1=diag([1e-4,1e-4,1/.5631]); %minimizar norma del estimado de x3 en concreto
    case 3
        Plantilla1=(1+0.015*s)/s*6; %limitar amplificación de alta frecuencia
        Plantilla2=tf(0.689); %limitar error constante
        Filt1=1/c2d(Plantilla1,Ts,'tustin');
        Filt2=1/c2d(Plantilla2,Ts,'tustin');
        Wout1=blkdiag(1e-4,1e-4,[Filt1;Filt2]);
    case 4
        Plantilla1=(1+0.02*s)/s*3.3; %limitar amplificación alta frecuencia
        Plantilla2=tf(.201*(1+s/0.3)/(1+s/8)); %limitar baja frecuencia
        Filt1=1/c2d(Plantilla1,Ts,'tustin');
        Filt2=1/c2d(Plantilla2,Ts,'tustin');
        Wout1=blkdiag(1e-4,1e-4,[Filt1;Filt2]);
    case 5
        Plantilla1=(1+0.0365*s)^2/s^2*35.5; %limitar amplificación alta frecuencia
        Plantilla2=tf(.165*(1+s/0.3)/(1+s/2.5)); %limitar baja frecuencia
        Filt1=1/c2d(Plantilla1,Ts,'tustin');
        Filt2=1/c2d(Plantilla2,Ts,'tustin');
        Wout1=blkdiag(1e-4,1e-4,[Filt1;Filt2]);

end

Wout=blkdiag(Wout1,1,1);
PlantaGenPonderada=Wout*PlantaGen*Win;

[OOi,CL,GAMi,INFO]=hinfsyn(PlantaGenPonderada,2,3,'Tol',1e-4);
GAMi

```

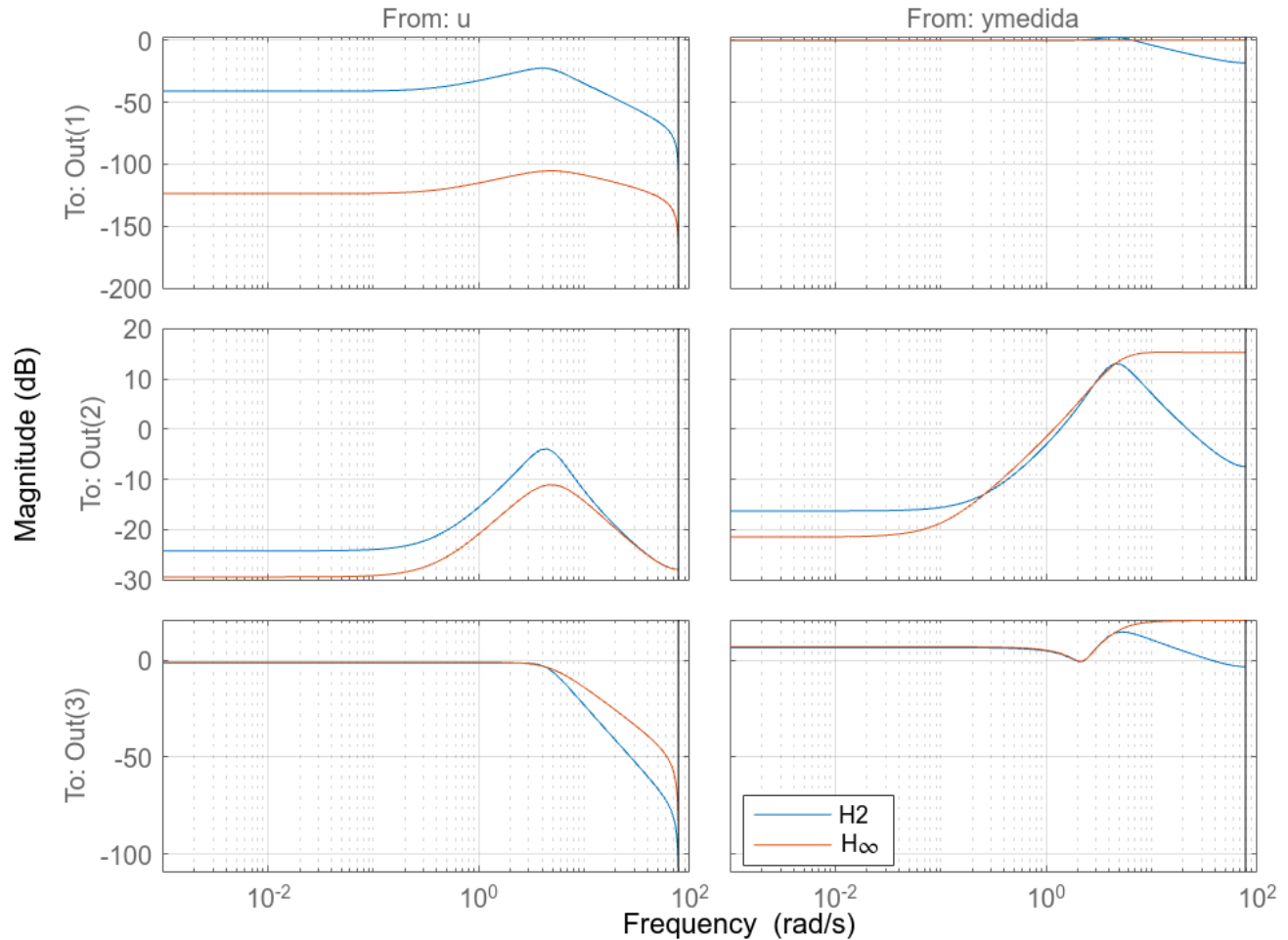
```
GAMi = 0.6046
```

```
OOi.InputName={'u','ymedida'};
```

Comparemos el diagrama de Bode de Magnitud de observador H2 y Hinfinito

```
bodemag(OO2,OOi), legend('H2','H\infty','Location','best'),grid on
```

Bode Diagram

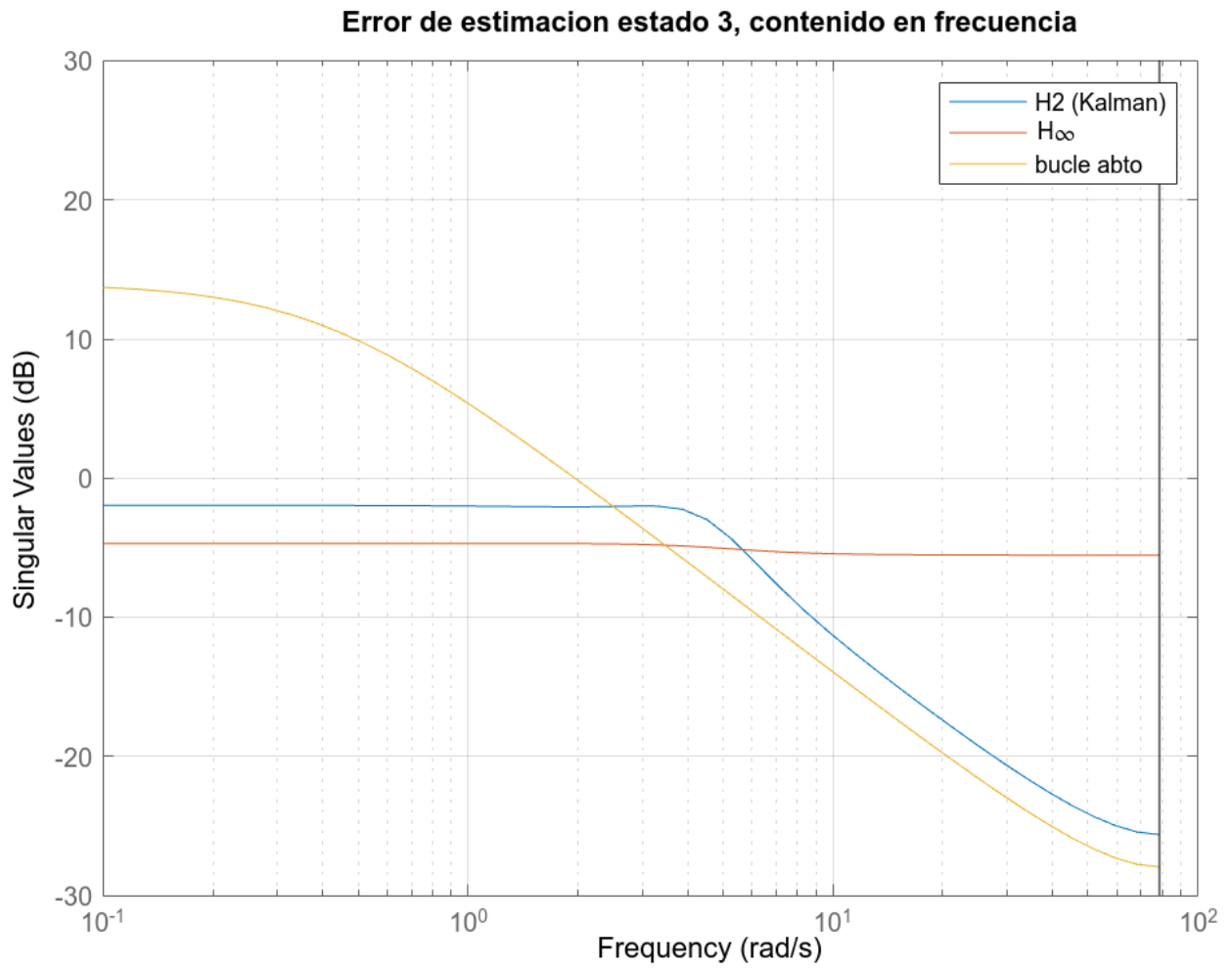


Comparemos error de observacion Kalman y Hinf con el contenido en frecuencia de x(3) en bucle abierto debido a las perturbaciones:

```
PWi=PlantaGen*Win;
PlantaYObsiWi=lft(PWi,OOi);
norm(PlantaYObsiWi(3,2:3),inf)
```

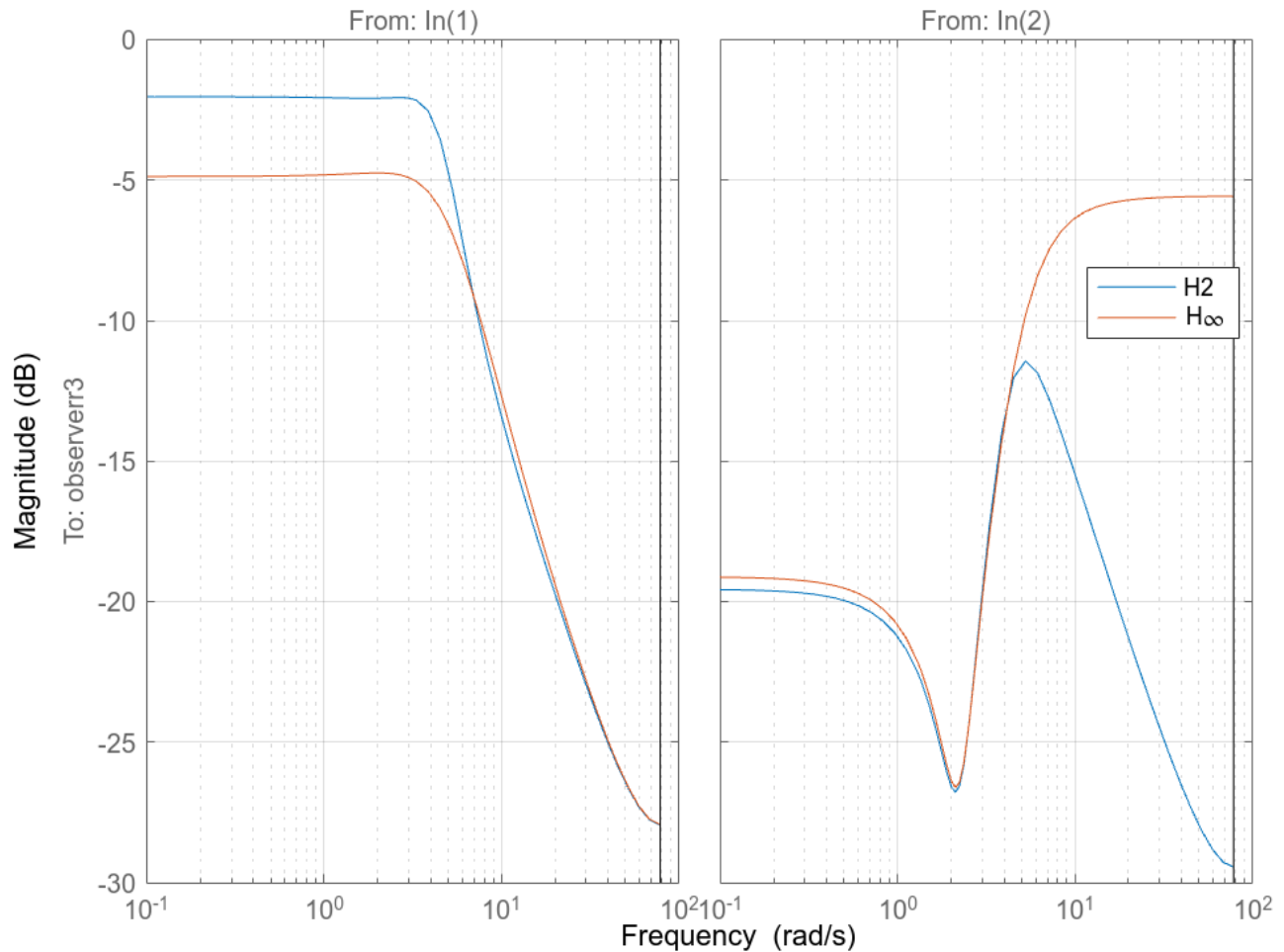
```
ans = 0.5817
```

```
sigma(PlantaYObs2(3,2:3),PlantaYObsiWi(3,2:3),PWi(3,2:3)), grid on, legend('H2 (Kalman)')
if(prueba>=3)
    hold on
    sigma(1/Filt1,'-.',1/Filt2,'-.')
    hold off, legend('H2 (Kalman)','H\infty','bucle abto','Plantilla')
end
ylim([-30 30]), title('Error de estimacion estado 3, contenido en frecuencia')
```



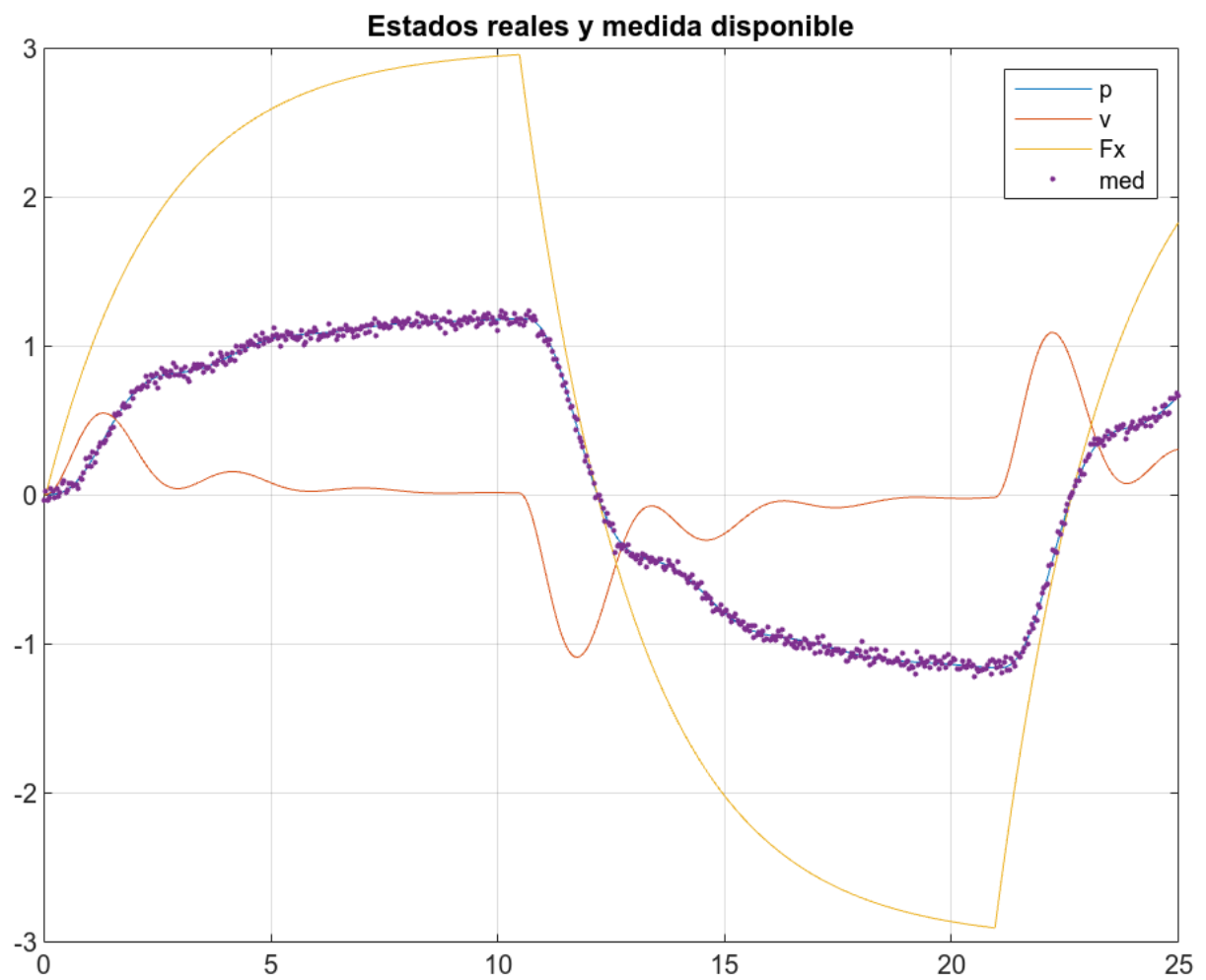
```
bodemag(PlantaYObs2(3,2:3),PlantaYObsiWi(3,2:3)), legend('H2','H\infty','Location','best')
title('Efecto ruido de proceso In(1) y medida In(2) sobre err. x3')
```

Efecto ruido de proceso $\ln(1)$ y medida $\ln(2)$ sobre err. x3

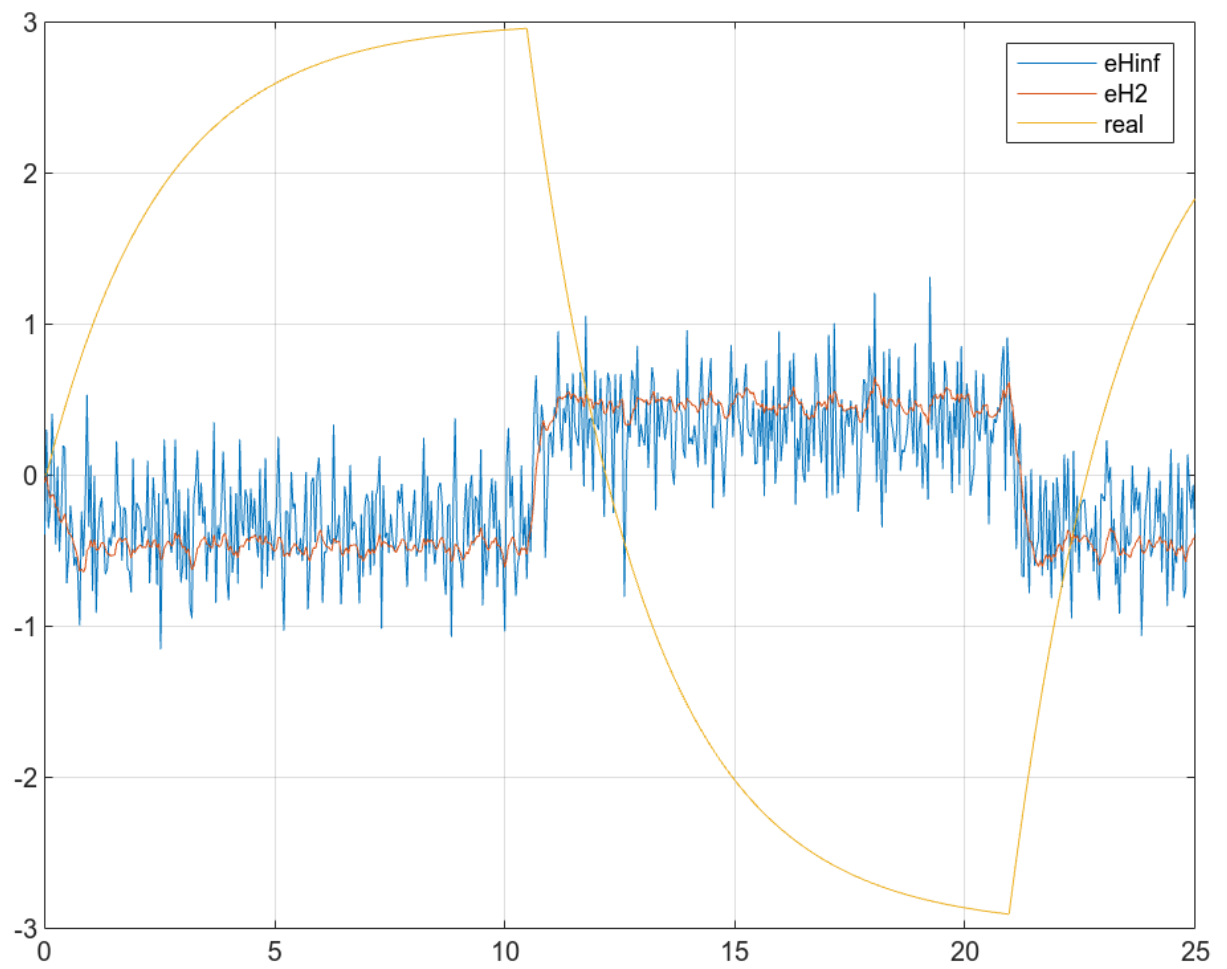


Simulación temporal comparativa H_2 , H_∞

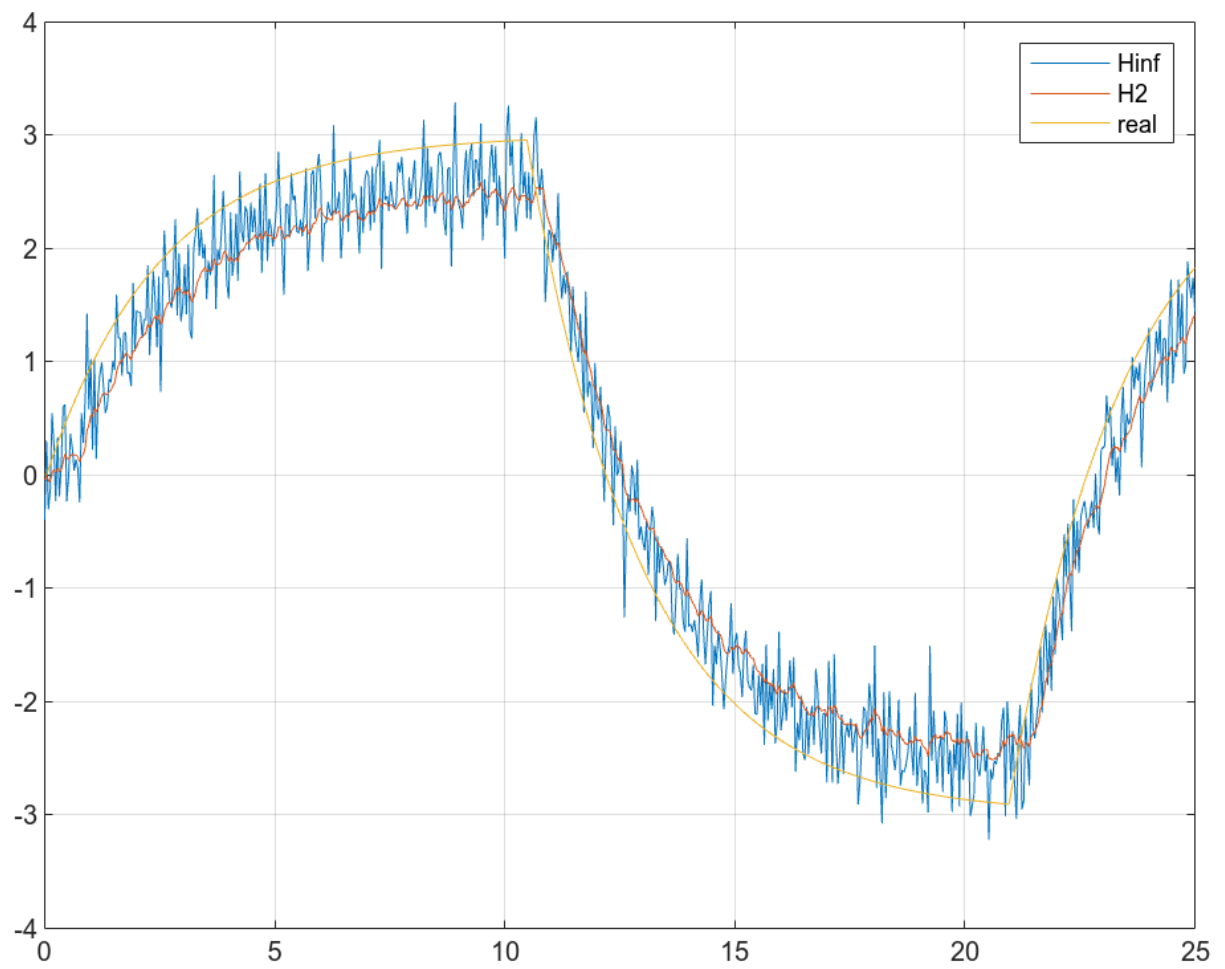
```
Tiempos=0:Ts:25;
u=zeros(size(Tiempos));
w=sign(sin(0.3*Tiempos))*3;
v=randn(size(Tiempos))*0.03;
GenInput=[u; w; v]';
ttt=lsim(PlantaGen([1 2 3 5],1:3),GenInput);
ymed=ttt(:,4);
plot(Tiempos,ttt(:,1:3)),
hold on
plot(Tiempos,ymed,'.');
hold off
legend('p','v','Fx','med'), grid on, title('Estados reales y medida disponible')
```



```
Yi=lsim(OOi,[u' ymed]);
Y2=lsim(OO2,[u' ymed]);
plot(Tiempos,[Yi(:,3)-ttt(:,3) Y2(:,3)-ttt(:,3) ttt(:,3) ]), grid on, legend('eHinf','e
```



```
plot(Tiempos,[Yi(:,3) Y2(:,3) ttt(:,3)]), grid on, legend('Hinf','H2','real')
```



Conclusiones
