

# Control bi-frecuencia (dual-rate) por realimentación de la salida

© 2018, Antonio Sala Piqueras, Universitat Politècnica de València. *Todos los derechos reservados.*

Este código ejecutó sin errores en Matlab R2018b.

Presentación en vídeo en: <http://personales.upv.es/asala/YTV/drml.html>

## Tabla de Contenidos

|                                                          |   |
|----------------------------------------------------------|---|
| Modelado en tiempo continuo.....                         | 1 |
| Diseño de observador y realimentación del estado.....    | 1 |
| Diseño de observador.....                                | 1 |
| Diseño de controlador por realimentación del estado..... | 2 |
| Simulación.....                                          | 2 |
| Bucle de simulación.....                                 | 2 |
| Gráficas de resultados.....                              | 3 |

## Modelado en tiempo continuo

```
A=[0 1;-4 -0.75]; B=[0;1]; C=[1 0];  
syscont=ss(A,B,C,0);  
nsalidas=size(C,1)
```

```
nsalidas = 1
```

```
nestados=size(A,1)
```

```
nestados = 2
```

```
nentradas=size(B,2)
```

```
nentradas = 1
```

## Diseño de observador y realimentación del estado

Especificación de tiempo de establecimiento (aprox) y polos continuos y luego distretizado...

### Diseño de observador

```
t_est_deseado=0.4;  
polocontin=-4/t_est_deseado
```

```
polocontin = -10
```

```
Tsensor=0.085;  
sys_discret_obs=c2d(syscont,Tsensor,'zoh');  
polodiscretoobservador=exp(polocontin*Tsensor)
```

```
polodiscretoobservador = 0.4274
```

```
L=place(sys_discret_obs.A',sys_discret_obs.A'*sys_discret_obs.C',polodiscretoobservador)
```

## Diseño de controlador por realimentación del estado

```
t_est_deseado=0.6;  
polocontin=-4/t_est_deseado
```

```
polocontin = -6.6667
```

```
Tcontrol=0.05;  
sys_discret_contr=c2d(syscont,Tcontrol,'zoh');  
polodiscretocontrol=exp(polocontin*Tcontrol)
```

```
polodiscretocontrol = 0.7165
```

```
K=place(sys_discret_contr.A,sys_discret_contr.B,polodiscretocontrol*[1 0.98]);
```

## Simulación

```
Tsimulacion=0.005;%infinitamente pequeño... o casi.
```

Modelo discreto al período de simulación.

```
sys_simulacion=c2d(syscont,Tsimulacion,'zoh');
```

### Condiciones de la simulación

```
Tfinal=1.4;  
Tiempos=0:Tsimulacion:Tfinal;  
xreal=[8;6]; %condiciones iniciales planta real  
relojUltimoEvento=0;%última vez que se midió  
xestimado=[0;0]; %condiciones iniciales observador  
u=0;%control inicial... antes de la primera actualización  
LL=length(Tiempos);  
  
grafyreal=zeros(nsalidas , LL);  
grafxreal=zeros(nestados , LL);  
grafu=zeros(size(B,2) , LL);  
tiemposmed=[];tiempos_c=[];medidas=[];controles=[];  
tiempostodo=[];estadosestim=[];estimacionesalmedir=[];
```

### Bucle de simulación

```
for i=1:LL  
    reloj=Tiempos(i);  
    if(DetectaEvento(reloj,Tsensor))  
        instante_medir=true;  
    else  
        instante_medir=false;
```

```

end
if(DetectaEvento(reloj,Tcontrol))
    instante_actuar=true;
else
    instante_actuar=false;
end

if(instante_medir || instante_actuar)
    xestimado=simulaZOH(syscont,reloj-relojUltimoEvento,xestimado,u);
    relojUltimoEvento=reloj;
    if(instante_medir)
        y=C*xreal;%leer tarj. acdq. datos
        xestimado=xestimado+L*(y-C*xestimado);%observador current
        tiemposmed=[tiemposmed reloj];
        medidas=[medidas y];
        estimacionesalmedir=[estimacionesalmedir xestimado];

    end
    if(instante_actuar)
        u=-K*xestimado;%escribir tar. adq. datos...
        tiempos_c=[tiempos_c reloj];
        controles=[controles u];
    end
    tiempostodo=[tiempostodo reloj];
    estadosestim=[estadosestim xestimado];
end

%simulamos proceso real, a período de simulación...

%gráficas de salida real:
yreal=C*xreal;
grafyreal(:,i)= yreal;
grafxreal(:,i)= xreal;

grafu(:,i)=u;
xreal=sys_simulacion.A*xreal+sys_simulacion.B*u;%esperamos Tsimulacion y repetimos
end

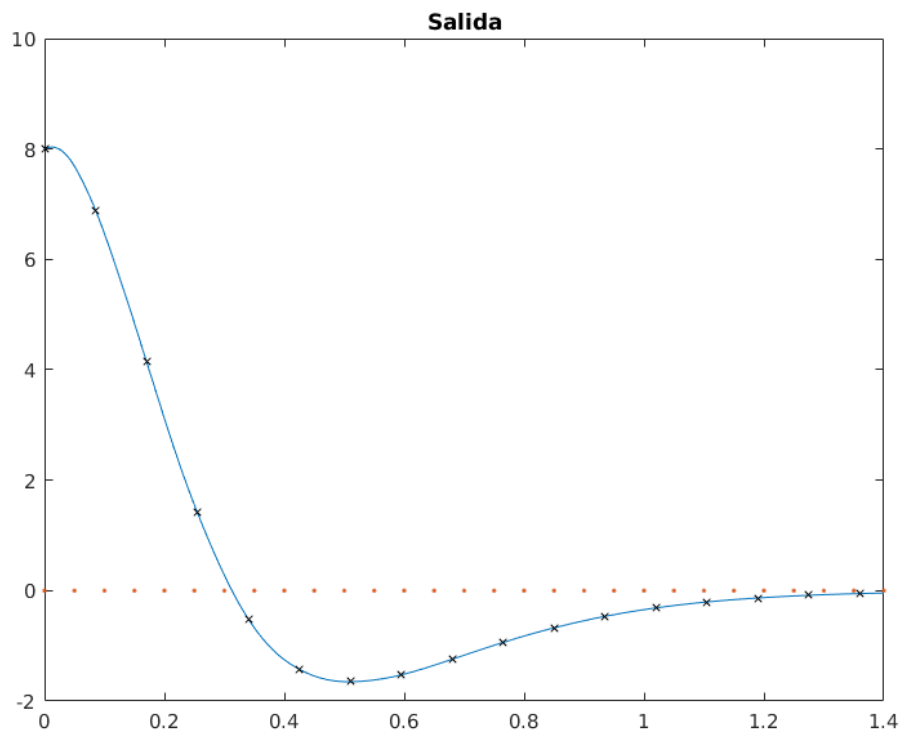
```

## Gráficas de resultados

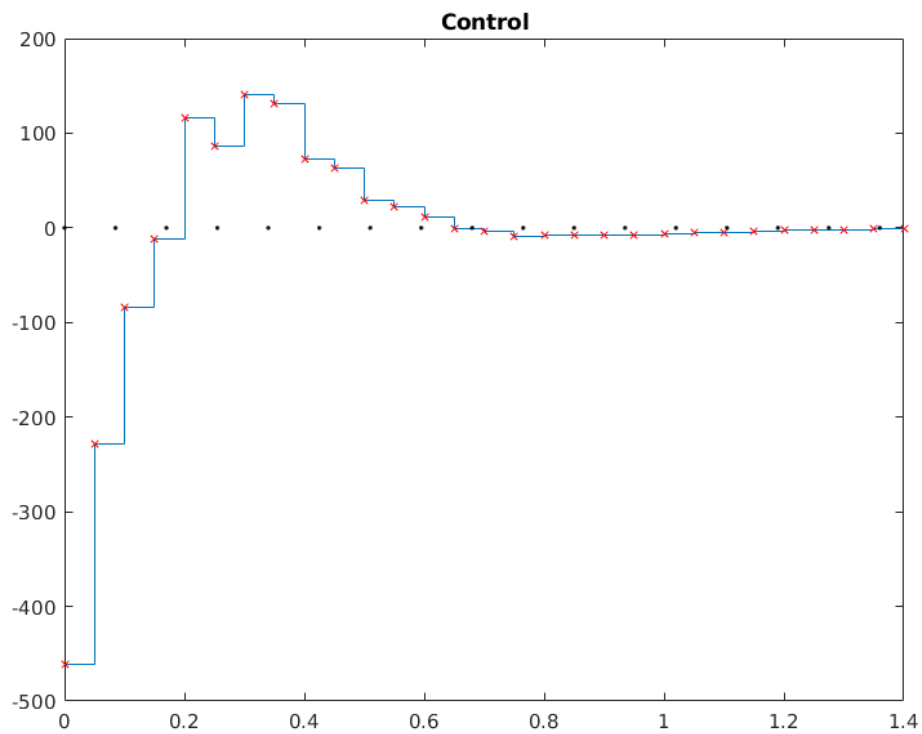
```

plot(Tiempos,grafyreal)
hold on
plot(tiemposmed,medidas,'xk')
plot(tiempos_c,0*tiempos_c,'.')
hold off
title('Salida')

```



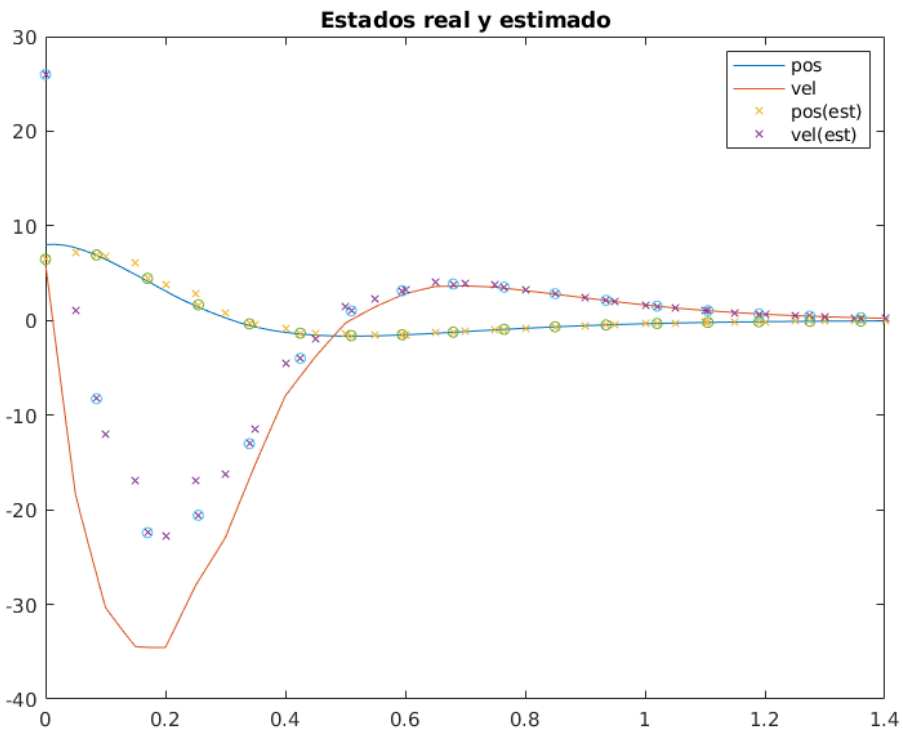
```
stairs(Tiempos,grafu)
hold on
plot(tiempos_c,controles,'xr')
plot(tiemposmed,0*tiemposmed,'.k')
hold off
title('Control')
```



```

plot(Tiempos, grafxreal)
hold on
plot(tiempostodo, estadoestim, 'x')
plot(tiemposmed, estimacionesalmedir, 'o')
hold off
legend('pos', 'vel', 'pos(est)', 'vel(est)')
title('Estados real y estimado')

```



```
function xnext=simulaZOH(sys,incr_tiempo,x,u)
    n_estados=size(sys.A,1);
    n_entradas=size(sys.B,2);
    Agorda=[sys.A sys.B; zeros(n_entradas,n_estados+n_entradas)];%dot u=0, estado ampliado
    xu_next=expm(Agorda*incr_tiempo)*[x;u]; %solución ecuación estado con u cte.
    xnext=xu_next(1:n_estados); %sólo devolvemos el estado.
end

function detectado=DetectaEvento(reloj,periodo)
    detectado=0;
    if(mod(reloj,periodo)<0.001*periodo)
        detectado=1;
    elseif(mod(reloj,periodo)>0.999*periodo)
        detectado=1;
    end
end
```