Control **Proporcional-Integral** de **nivel** de un tanque de líquido

Análisis y simulación en bucle cerrado (step)

© 2023, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

*****Este código funcionó sin errores en Matlab `R2023b` (Linux)

Objetivo: modelar y linealizar el tanque, introducir control proporcional, calcular y simular las funciones de transferencia de bucle cerrado, y comparar con la animación del depósito.

Presentaciones en vídeo (YouTube)**:**

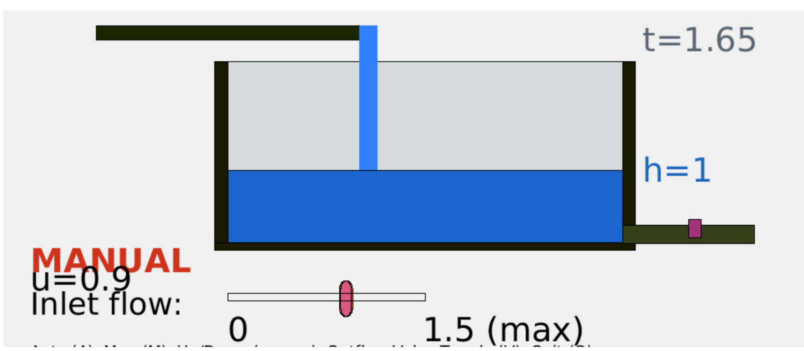
http://personales.upv.es/asala/YT/V/tank1PI1teo.html

http://personales.upv.es/asala/YT/V/tank1PI2step.html

Tabla de Contenidos

Modelado y linealización del tanque de líquido (sin control)

Consideremos el siguiente sistema:



Dos entradas: caudal u y grado de apertura de válvula de salida κ.

El modelo no lineal del depósito de líquido es:

$$\frac{dh}{dt} = \frac{1}{6} \cdot \left(u - \kappa \cdot \sqrt{h}\right)$$

Ecuaciones en equilibrio: $u_{eq} - \kappa_{eq}\sqrt{h_{eq}} = 0$.

Punto operación nominal $\kappa_0 = 0.9$, $u_0 = 0.9$, $h_0 = 1$.

```
kappa0=0.9; u0=0.9; h0=1;
```

Linealización, en variables incrementales, sustituyendo valores numéricos y eliminando Δ de la notación, llamando d a Δk, de "disturbance" (perturbación),en forma normalizada $\frac{dh}{dt} = A \cdot h + B_{1\times 2} \cdot \begin{pmatrix} u \\ d \end{pmatrix}_{2\times 1}$,

```
A=-kappa0/6/2/sqrt(h0);
B=[1/6 -sqrt(h0)/6];
tank=ss(A,B,1,[0 0]); %la salida es el nivel del tanque directamente C=1, D=0
tank.InputName={'u','d'}; tank.OutputName='y'; tank.StateName='h'
```

```
tank =

  A =
          h
   h  -0.075

  B =
           u        d
   h   0.1667  -0.1667

  C =
       h
   y   1

  D =
       u  d
   y   0  0

Continuous-time state-space model.
Model Properties
```

```
dcgain(tank)
```

```
ans = 1×2
    2.2222   -2.2222
```

El modelo en función de transferencia (formalmente, en "matriz" de transferencia, que hay dos entradas) será $MdT(s) = C(sI - A)^{-1}B + D$, con dimensiones 1×2, en concreto con elementos:

```
MdT=tf(tank);
```

Las FdT asociadas a cada entrada son:

```
G=MdT(1) %de variable manipulada a controlada
```

```
G =

  From input "u" to output "y":
   0.1667
  ---------
```

```
  s + 0.075

Continuous-time transfer function.
Model Properties
```

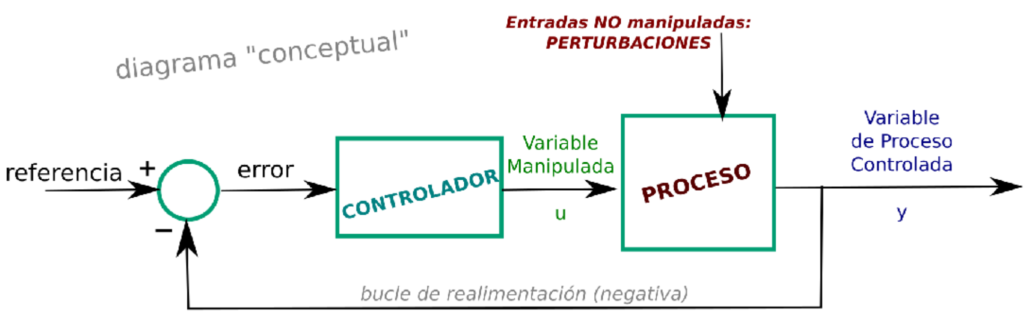
```
Gd=MdT(2) %de perturbación a variable controlada
```

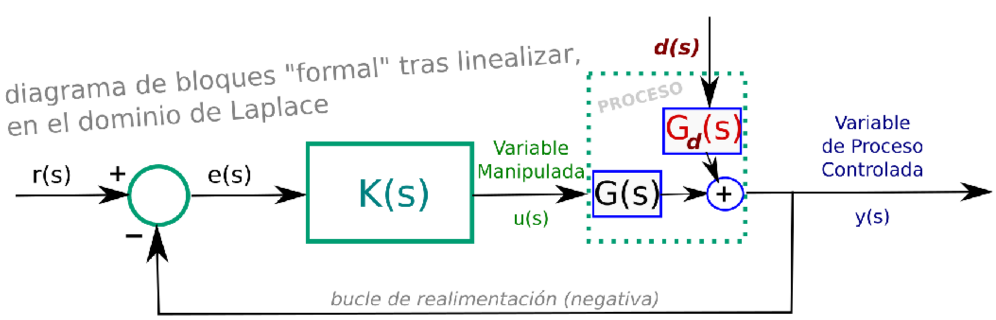
```
Gd =

  From input "d" to output "y":
   -0.1667
  ---------
  s + 0.075

Continuous-time transfer function.
Model Properties
```

Con lo que el modelo en el dominio de Laplace será $y = G(s) \cdot u + G_d(s) \cdot d$; esta se denomina "ecuación del modelo en bucle **abierto**".

Modelo en bucle cerrado con control PI

Este es el diagrama de bloques de un sistema de control realimentado (bucle cerrado), suponiendo que la dinámica del sensor y su ruido de medida son despreciables:





En general, un controlador dinámico será $u = K(s) \cdot e$.

Las ecuaciones de bucle cerrado son:

$$\begin{pmatrix} e \\ u \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{1+GK} & -\frac{G_d}{1+GK} \\ \frac{K}{1+GK} & -\frac{KG_d}{1+GK} \\ \frac{GK}{1+GK} & \frac{G_d}{1+GK} \end{pmatrix} \begin{pmatrix} r \\ d \end{pmatrix}$$

Análisis de propiedades del sistema en bucle cerrado (comparamos varios reguladores)

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

Vamos a comparar un regulador proporcional:

```
Kp=tf(3); %proporcional
Kp.OutputName="u_control"; Kp.InputName="e";
```

Y tres reguladores proporcional-integral (PI), de ganancia integral creciente:

```
K_PI_1=3*(1+0.03/s); %PI corrigiendo ligeramente el "P".
K_PI_1.OutputName="u_control"; K_PI_1.InputName="e"
```

```
K_PI_1 =

  From input "e" to output "u_control":
  3 s + 0.09
  ----------
      s

Continuous-time transfer function.
Model Properties
```

```
K_PI_2=3*(1+0.075/s); %PI de "cancelación"
K_PI_2.OutputName="u_control"; K_PI_2.InputName="e"
```

```
K_PI_2 =

  From input "e" to output "u_control":
  3 s + 0.225
  -----------
       s

Continuous-time transfer function.
Model Properties
```

```
K_PI_3=3*(1+0.4/s); %PI que modifica bastante la dinámica respecto al P
(esperemos que para bien).
K_PI_3.OutputName="u_control"; K_PI_3.InputName="e"
```

```
K_PI_3 =

  From input "e" to output "u_control":
  3 s + 1.2
  ---------
      s
```

```
Continuous-time transfer function.
Model Properties
```

Añadiremos un polo en el actuador, que sí está en la animación, la valvula tiene una dinamica interna que hace que se tarde un poquito en dar el caudal que se manda:

```
Actuador=1/(0.25*s+1);
Actuador.InputName='u_control';Actuador.OutputName='u';
```

Funciones de transferencia de bucle cerrado, propiedades y simulación

El comando "connect" simplifica interconexiones arbitrarias de bloques, donde la conexión viene prefijada por las propiedades "InputName" y "OuputName" de cada uno de ellos.

```
e_sum=sumblk("e=r-y");
BucleCerradoP=connect(e_sum,Kp,tank,Actuador,{'r','d'},{'y','u_control'});
%podemos sacar "e" y "u" pero no lo hacemos por brevedad
dcgain(BucleCerradoP)
```

```
ans = 2×2
    0.8696   -0.2899
    0.3913    0.8696
```

```
zpk(BucleCerradoP(1,1))
```

```
ans =

  From input "r" to output "y":
           2
  --------------------
  (s+0.6768) (s+3.398)

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

```
BucleCerradoPI_1=connect(e_sum,K_PI_1,tank,Actuador,{'r','d'},
{'y','u_control'});
dcgain(BucleCerradoPI_1)
```

```
ans = 2×2
    1.0000         0
    0.4500    1.0000
```

```
zpk(BucleCerradoPI_1(1,1))
```

```
ans =

  From input "r" to output "y":
            2 (s+0.03)
  -------------------------------
  (s+3.405) (s+0.643) (s+0.02741)

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

```
BucleCerradoPI_2=connect(e_sum,K_PI_2,tank,Actuador,{'r','d'},
{'y','u_control'});
```

```
dcgain(BucleCerradoPI_2)
```

```
ans = 2×2
    1.0000         0
    0.4500    1.0000
```

```
zpk(BucleCerradoPI_2(1,1))
```

```
ans =
 
  From input "r" to output "y":
           2 (s+0.075)
  ------------------------------
  (s+3.414) (s+0.5858) (s+0.075)
 
Continuous-time zero/pole/gain model.
Model Properties
```

```
BucleCerradoPI_3=connect(e_sum,K_PI_3,tank,Actuador,{'r','d'},
{'y','u_control'});
dcgain(BucleCerradoPI_3)
```

```
ans = 2×2
    1.0000         0
    0.4500    1.0000
```

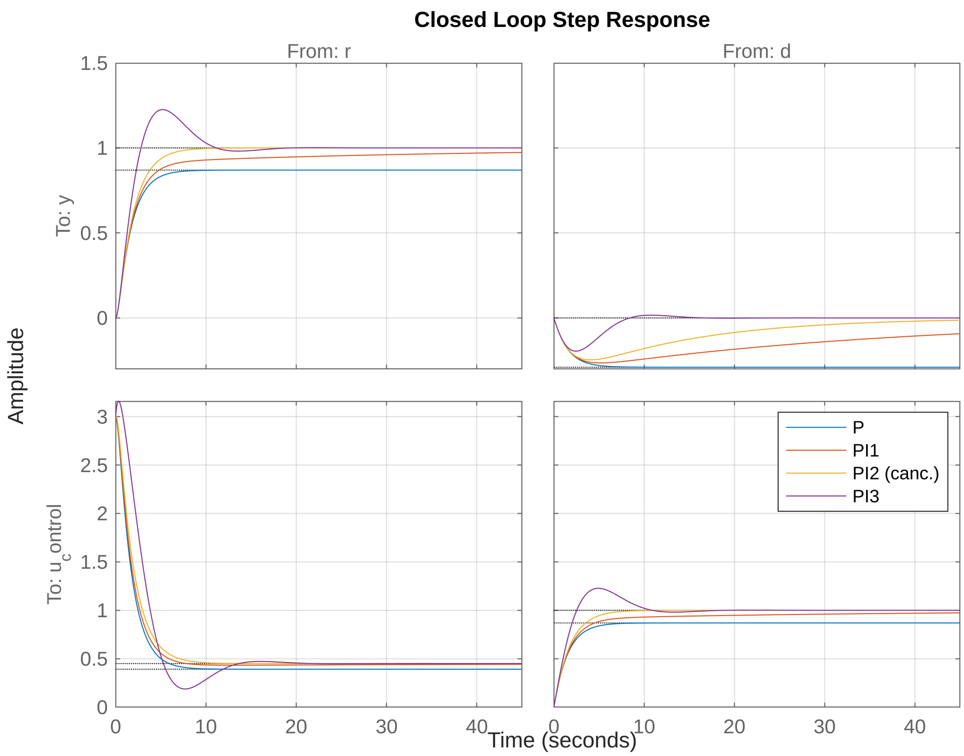
```
zpk(BucleCerradoPI_3(1,1))
```

```
ans =
 
  From input "r" to output "y":
              2 (s+0.4)
  ---------------------------------
  (s+3.48) (s^2 + 0.5948s + 0.2299)
 
Continuous-time zero/pole/gain model.
Model Properties
```

```
pole(BucleCerradoPI_3)
```

```
ans = 3×1 complex
  -3.4802 + 0.0000i
  -0.2974 + 0.3761i
  -0.2974 - 0.3761i
```

```
step(BucleCerradoP,BucleCerradoPI_1,BucleCerradoPI_2,BucleCerradoPI_3,45),
grid on,
title("Closed Loop Step Response"), legend("P","PI1","PI2 (canc.)","PI3")
```



Closed Loop Step Response
From: r
From: d
Amplitude
To: y
To: u_control
Time (seconds)
P
PI1
PI2 (canc.)
PI3