Control **proporcional** de **nivel** de un **tanque** de líquido: modelado y análisis en dominio de Laplace

*****Este código funcionó sin errores en Matlab `R2022b` (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/tank1ModyBA.html

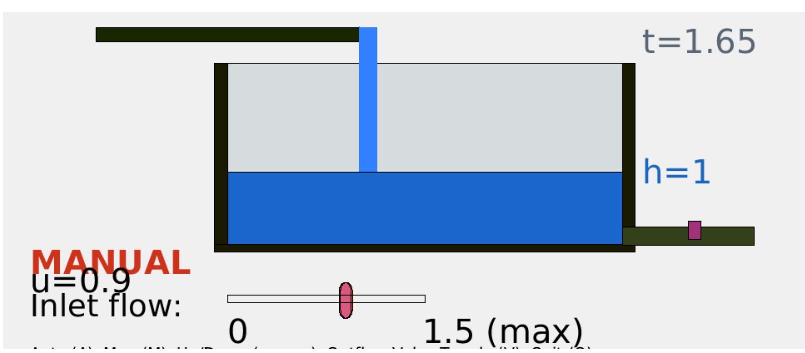
http://personales.upv.es/asala/YT/V/tankCLeq.html

http://personales.upv.es/asala/YT/V/tank1CLan.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:

$$\frac{d\Delta h}{dt} = \left(-\frac{\kappa_0}{6 \cdot 2\sqrt{h_0}}\right) \cdot \Delta h + \frac{1}{6} \cdot \Delta u - \sqrt{h_0} \cdot \Delta \kappa$$

Sustituyendo valores numéricos y eliminando Δ de la notación, llamando d a Δk, de "disturbance" (perturbación):

$$\dot{h} = -\frac{0.9}{2 \cdot 6} \cdot h + \frac{1}{6} \cdot u - \frac{1}{6} \cdot d$$

o 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)
```

```
A = -0.0750
```

```
B=[1/6 -sqrt(h0)/6]
```

```
B = 1×2
    0.1667   -0.1667
```

```
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='h'; tank.StateName='h';
```

Vamos a cambiar a "y" el nombre de la variable de salida, por usar la notación más frecuente en los libros de teoría:

```
tank.OutputName={'y'};
```

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)
```

```
MdT =

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

```
  s + 0.075

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

Continuous-time transfer function.
Model Properties
```

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**".

Propiedades en bucle abierto:

```
ganBA = dcgain(tank)
```

```
-4./pole(tank) %tiempo de establecimiento en bucle abierto
```

```
ans = 53.3333
```

Control "estático (proporcional)" en bucle abierto

El control más "trivial" del mundo en bucle abierto, LINEAL, si quisiéramos poner "r" metros de altura (de "incremento de altura sobre el punto de operación $h_0 = 1$") sería:

$$u_{BA} = \frac{1}{2.2222} \cdot r$$

```
1/ganBA(1)
```

```
ans = 0.4500
```

Obviamente, es la "linealización" de la ecuación en equilibrio $u_{eq} - \kappa_{eq}\sqrt{h_{eq}} = 0$, que resultaría en el control en bucle abierto NO LINEAL dado por $u_{BA} = \kappa_o\sqrt{r}$. Linealizando, en incrementales tendríamos $\Delta u_{BA} = \frac{\kappa_0}{2\sqrt{h_0}}\Delta r$.

En efecto

```
K_r_ba=kappa0/2/sqrt(h0)
```

```
K_r_ba = 0.4500
```

Pero ese control no podrá "acelerar" la respuesta, ni podría hacer frente a perturbaciones d no medibles.

Aunque pueden hacerse "trucos" para lo primero (por ejemplo un primer escalón "sobredimensionado", vídeo [linregla3]), sin sensor es imposible compensar el efecto de las aperturas inesperadas de válvula de salida.

Plantearemos, pues, control en bucle CERRADO suponiendo disponible existencia de un sensor de nivel de líquido.

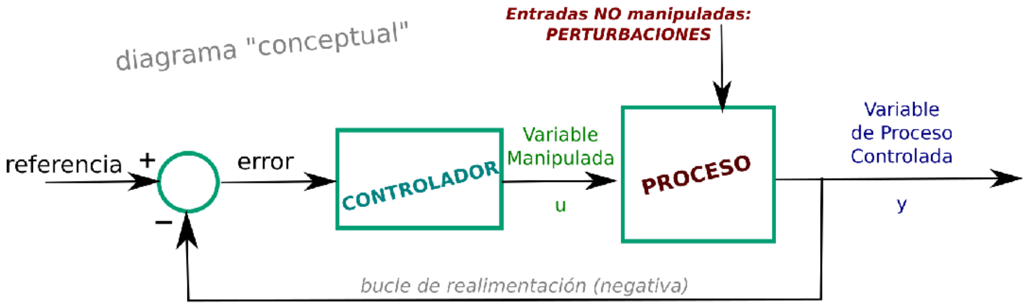
Modelo en bucle cerrado

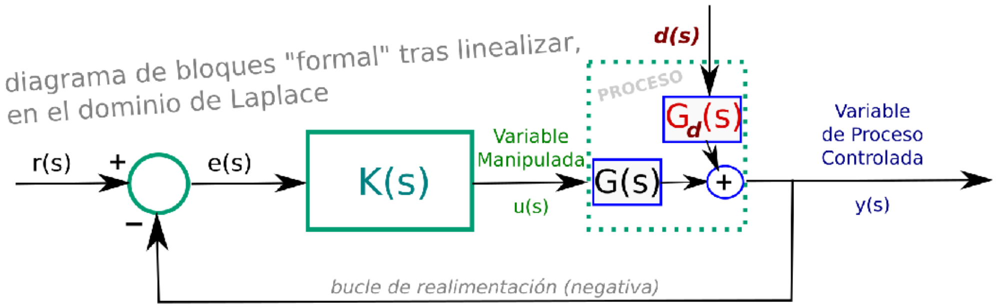
[tanque agua potable] El control de nivel tendrá variable manipulada (MV; manipulated variable en inglés) u (caudal de entrada) y variable perturbación d (incremento de válvula de salida), y variable controlada ("process variable PV", o "controlled variable" en la literatura en inglés) el nivel h.

[depuradora] En este caso (no objeto de este material), el caudal de entrada sería perturbación; la variable manipulada sería la válvula de salida.

Nos centraremos en el primer caso.

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:





El control proporcional será $u = K \cdot e$, siendo el error $e = r - y$ y K una constante.

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

Las ecuaciones de bucle cerrado son:

$$y = G(s) \cdot u + G_d(s) \cdot d\,, \quad u = K(s) \cdot e\,, \qquad e = r - y$$

que se suelen representar con el diagrama de bloques de la figura superior.

Con lo cual, de esas tres ecuaciones podremos despejar y, u, e en lo que constituirán las "ecuaciones en bucle **cerrado**".

Ecuaciones de bucle cerrado para seguimiento de referencia

Despejando, si suponemos $d = 0$:

$e = r - y = r - Gu = r - GKe\,,$

con lo que

$$e = \frac{1}{1 + GK} r\,.$$

Entonces,

$$u = Ke = \frac{K}{1 + GK} r\,,$$

$$y = Gu = \frac{GK}{1+GK} r \, .$$

Lo podemos resumir, en matriz de transferencia *[MUY IMPORTANTE, conviene memorizar]*:

$$\begin{pmatrix} e \\ u \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{1+GK} \\ \frac{K}{1+GK} \\ \frac{GK}{1+GK} \end{pmatrix} r$$

Ecuaciones de bucle cerrado (caso general con perturbación)

En el caso general con $d \neq 0$, entonces:

$e = r - y = r - (Gu + G_d d) = r - GKe - G_d d$, con lo que $e = \frac{1}{1+GK} r - \frac{G_d}{1+GK} d$.

$$u = Ke = \frac{K}{1+GK} r - \frac{KG_d}{1+GK} d \, ,$$

$$y = Gu + G_d d = \frac{GK}{1+GK} r - \frac{GKG_d}{1+GK} d + G_d d = \frac{GK}{1+GK} r + \left(1 - \frac{GK}{1+GK}\right) \cdot G_d d$$

que resulta en:

$$y = Gu + G_d d = \frac{GK}{1+GK} r + \frac{1}{1+GK} G_d d$$

Nótese que estas ecuaciones valdrían para cualquier $K(s)$ diferente a proporcional. Son las "ecuaciones en bucle cerrado" de (casi) cualquier sistema de control monovariable lineal (por ser precisos, de los sistemas de control monovariable de 1 grado de libertad, con 1 sensor). Los podemos resumir en matriz de transferencia como:

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

Análisis de propiedades del sistema en bucle cerrado

Probemos, por ejemplo, con $K = 3$.

```
s=tf('s');
K=tf(3); %proporcional
K.OutputName="u"; K.InputName="e";
```

Funciones de transferencia de bucle cerrado

Método 1: Sustituyendo en la matriz de transferencia teórica

```
MdT_BC=cierrabucle(K,G,Gd)
```

```
MdT_BC =

  From input "r" to output...
       s + 0.075
   e:  ---------
       s + 0.575

       3 s + 0.225
   u:  -----------
        s + 0.575

          0.5
   y:  ---------
       s + 0.575

  From input "d" to output...
        0.1667
   e:  ---------
       s + 0.575

          0.5
   u:  ---------
       s + 0.575

        -0.1667
   y:  ---------
       s + 0.575

Continuous-time transfer function.
Model Properties
```

Método 2: con "connect"

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");
FBB=connect(e_sum,K,tank,{'r','d'},{'e','u','y'}); %trabajar en "estados" es más eficiente, sobre todo en "multivariable"
tf(FBB)
```

```
ans =

  From input "r" to output...
       s + 0.075
   e:  ---------
       s + 0.575

       3 s + 0.225
   u:  -----------
        s + 0.575

          0.5
   y:  ---------
       s + 0.575
```

```
  From input "d" to output...
        0.1667
   e:  ---------
       s + 0.575

          0.5
   u:  ---------
       s + 0.575

        -0.1667
   y:  ---------
       s + 0.575

Continuous-time transfer function.
Model Properties
```

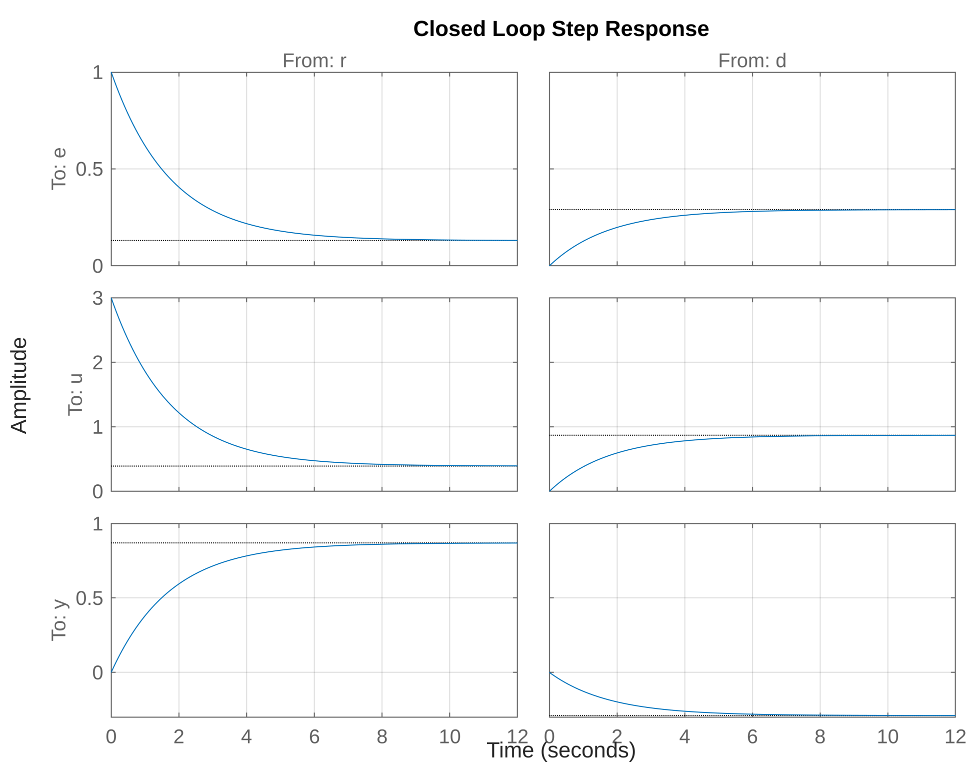
```
norm(FBB-MdT_BC) %coincidente, norm es cero si los sistemas son idénticos. Detalles no objetivo de este material.
```

```
ans = 2.1853e-16
```

Análisis de propiedades en bucle cerrado y simulación

Valor final, ganancia estática sustituyendo s por cero:

```
step(MdT_BC), grid on, title("Closed Loop Step Response")
```




```
ganBC=dcgain(MdT_BC)
```

```
ganBC = 3×2
```

```
    0.1304    0.2899
    0.3913    0.8696
    0.8696   -0.2899
```

El elemento (1,1) se denomina "error de posición", ante referencia:

```
e_pos_ref=ganBC(1,1) %adimensional
```

```
e_pos_ref = 0.1304
```

El efecto sobre la salida de la perturbación en bucle cerrado es

```
gan_y_d=ganBC(3,2) % metros por "unidad de cte válv."
```

```
gan_y_d = -0.2899
```

Por ejemplo, el "control" baja el efecto de "d" casi 10 veces:

```
gan_y_d/ganBA(2) %reducción de la variabilidad debido al control
(disturbance rejection)
```

```
ans = 0.1304
```

*Esa es la gran ventaja de la realimentación: gracias a medir que se está desviando, tomamos una acción correctora.

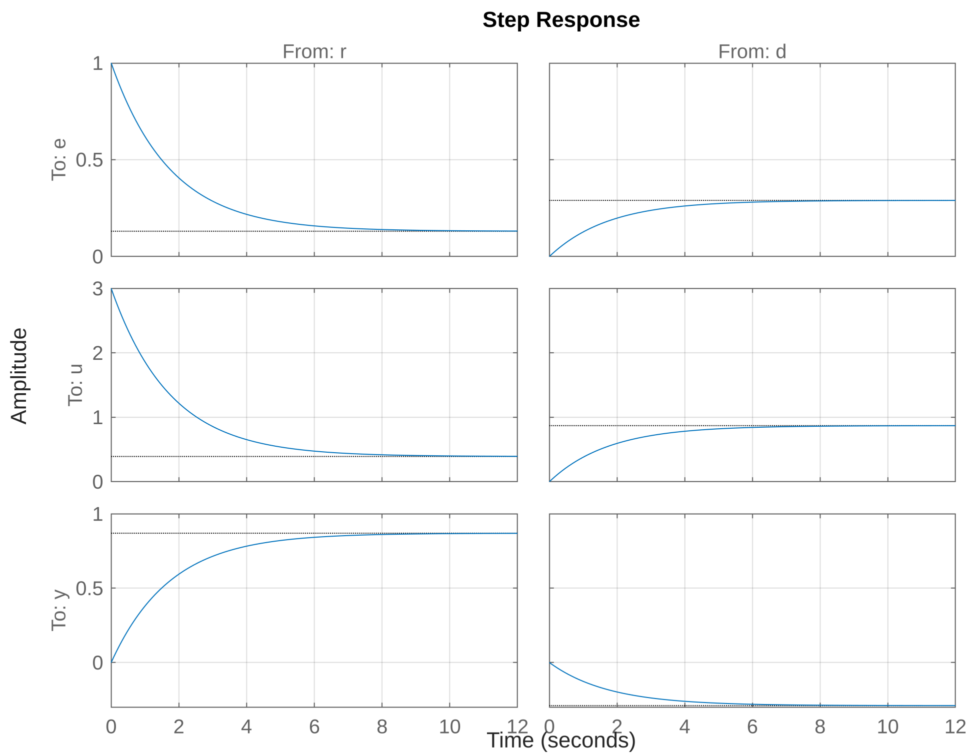
Tiempo de establecimiento de bucle cerrado:

```
-4./pole(MdT_BC)
```

```
ans = 2×1
    6.9565
    6.9565
```

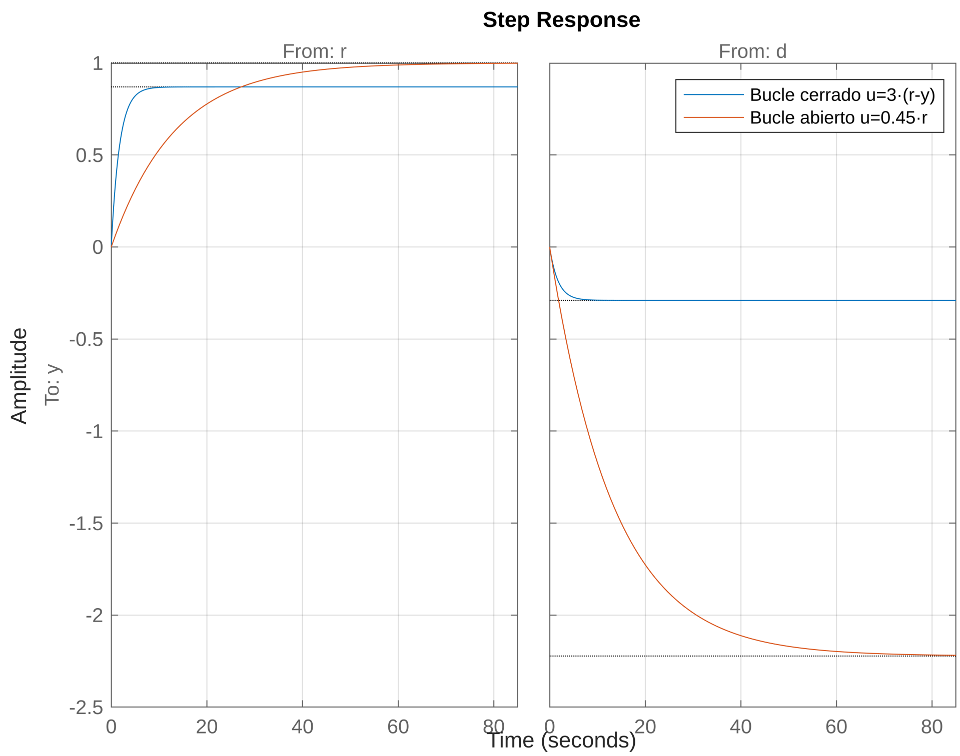
La simulación ante escalón (unitario, que no suele coincidir con amplitud "físicamente razonable", pero da igual porque todo será proporcional a lo que aquí veamos) es:

```
step(MdT_BC), grid on
```



Comparando bucle abierto con cerrado, tendríamos:

```
cosaBA=[tank(1)*K_r_ba tank(2)];
cosaBA.InputName={'r','d'};
step(MdT_BC(3,:),cosaBA,85)
grid on, legend("Bucle cerrado u=3·(r-y)","Bucle abierto u=0.45·r")
```



Funciones auxiliares: ecs. bucle cerrado

```
function MdT_BC=cierrabucle(K,G,Gd)
 S=1/(1+G*K);
 MdT_BC=minreal(S*[1 -Gd; K -K*Gd; G*K Gd]);
 MdT_BC.InputName={'r','d'};
 MdT_BC.OutputName={'e','u','y'};
end
```