

Diseño de control para modelo linealizado de tanque de mezclado (V2): controlador "esclavo" de caudal en cascada

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

Presentación vídeo: <http://personales.upv.es/asala/YT/V/tmcas.html>

Este código funcionó correctamente con Matlab R2019b

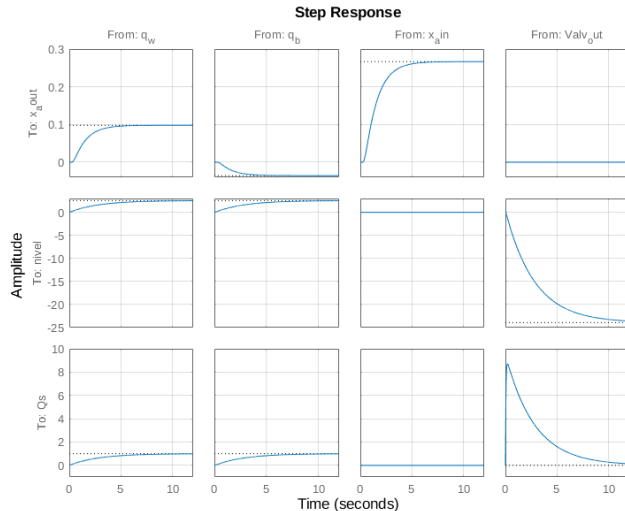
Objetivos: Calcular un controlador esclavo para manejar válvula de salida ante referencia de caudal y generar el modelo con dicho bucle ya cerrado para posteriores controles de concentración y nivel.

Tabla de Contenidos

1.- Modelado y planteamiento del problema.....	1
Análisis del modelo del subsistema para control en cascada (sensor extra) de caudal de salida.....	2
2.- Sintonizado del regulador con pident.....	3
Bucle cerrado del control de caudal (slave).....	4
Incorporación en modelo conjunto.....	5

1.- Modelado y planteamiento del problema

```
load ModeloNominal.mat
step(G,12), grid on
```

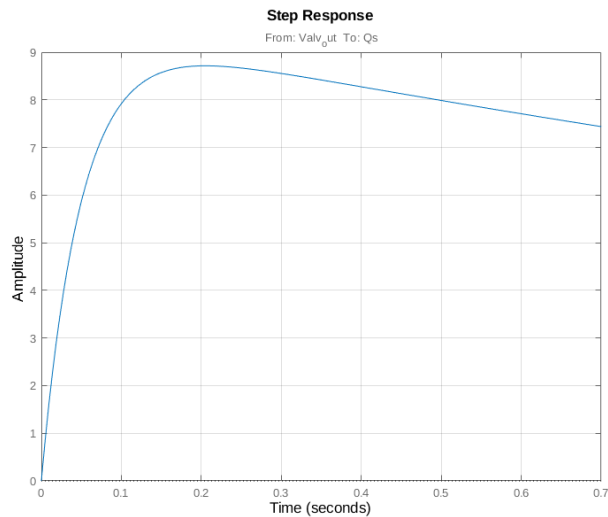


El sistema tiene dos **perturbaciones** (caudal q_{ain} , composición x_{ain}), del "**wild feed**".

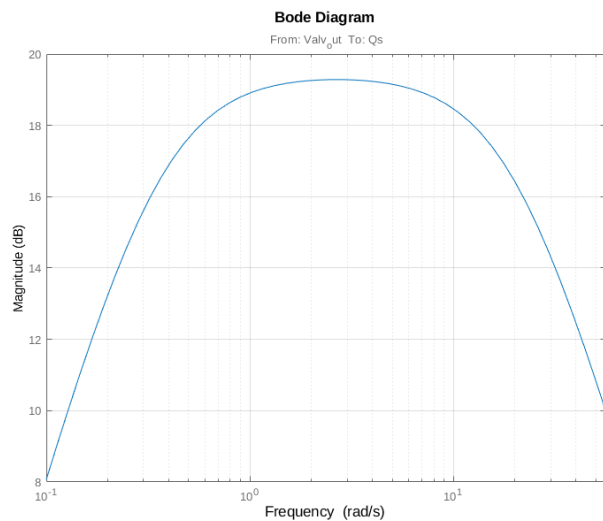
Referencias primarias: tenemos para x_{aout} y para nivel.

Variables manipuladas: en el modelo original, son $Valv_{out}$ y q_{bin} ... como tenemos un sensor extra, haremos "cascada" Qs - $valv_{out}$, y luego multibucle.


```
step(G_caudal,0.7), grid on
```



```
bodemag(G_caudal,logspace(-1,1.75)), grid on
```



```
norm(G_caudal,'inf')
```

```
ans = 9.2109
```

2.- Sintonizado del regulador con pidtune

```
s=tf('s');  
K_caudal=pidtune(G_caudal,'PI') %ORIGINAL de Matlab por defecto
```

```
K_caudal =
```

$$K_p + K_i * \frac{1}{s}$$

```
with Kp = 0.0276, Ki = 2.56
```

Continuous-time PI controller in parallel form.

Bucle cerrado del control de caudal (slave)

```
CL=feedback(G_caudal*K_caudal,1);zpk(CL), dcgain(CL)
```

ans =

```
From input to output "Qs":  
5.178 s (s+92.74)  
-----  
s (s^2 + 25.53s + 487.3)
```

Continuous-time zero/pole/gain model.
ans = Inf

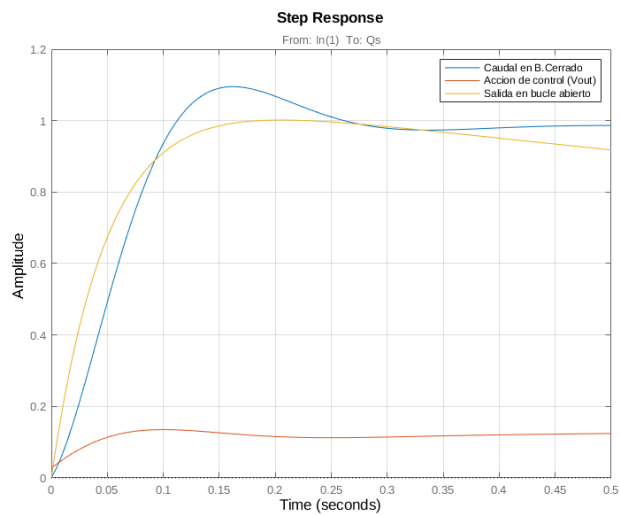
```
CL_acccontrol=minreal(feedback(K_caudal,G_caudal)); zpk(CL_acccontrol)
```

ans =

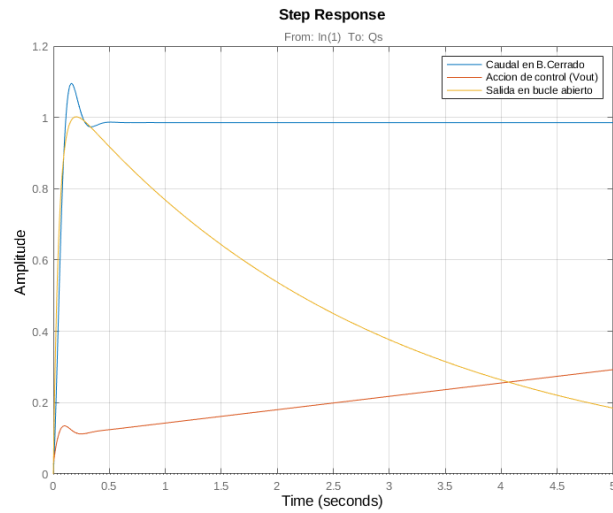
```
0.027616 (s+92.74) (s+20) (s+0.3564)  
-----  
s (s^2 + 25.53s + 487.3)
```

Continuous-time zero/pole/gain model.

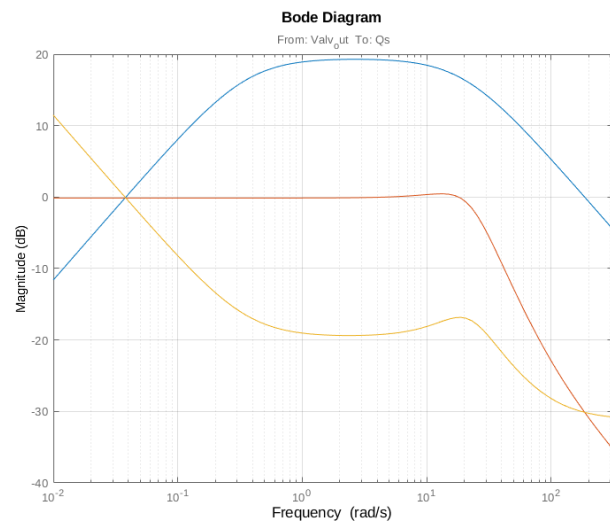
```
step(CL,CL_acccontrol,G_caudal/8.7,.5), grid on  
legend('Caudal en B.Cerrado','Accion de control (Vout)','Salida en bucle abierto')
```



```
step(CL,CL_acccontrol,G_caudal/8.7,5), grid on  
legend('Caudal en B.Cerrado','Accion de control (Vout)','Salida en bucle abierto')
```



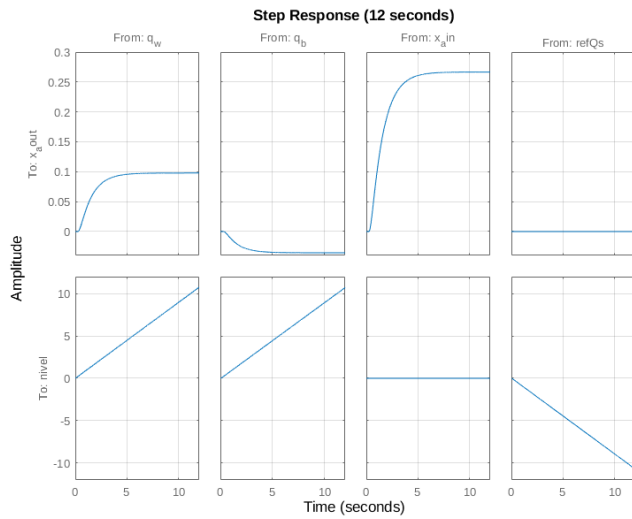
```
bodemag(G_caudal,CL,CL_acccontrol,logspace(-2,2.5,100)), grid on
```



Incorporación en modelo conjunto

Una vez decidido el regulador de caudal, procedemos a cerrar el bucle, con nueva entrada refQ:

```
K_caudal.InputName={'errqs'}; K_caudal.OutputName={'Valv_out'};
Sum=sumblk('errqs=refQs-Qs');
G2=connect(G,K_caudal,Sum',{'q_w','q_b','x_ain','refQs'},{'x_aout','nivel'});
step(G2,12), grid on, title("Step Response (12 seconds)")
```



Vemos que básicamente, la respuesta de nivel es un integrador, dado que el cascada nos ha creado una "referencia caudal de salida" como entrada virtual...

```
save ModeloConCascada.mat G2 K_caudal
```