

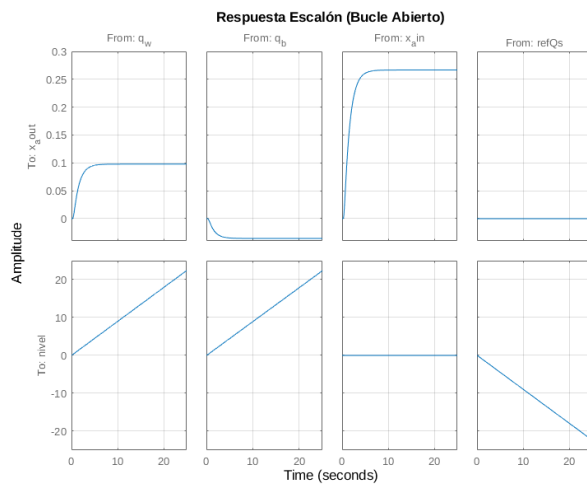
# Diseño de control para modelo linealizado de tanque de mezclado V2: desacoplamiento y prealimentación **dinámicas**

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

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

Este código funcionó correctamente con Matlab **R2019b**

```
load ModeloConCascada.mat
load Multibucle.mat %los PIs y el bucle cerrado, para justificar o no la necesidad
step(G2), grid on, title ("Respuesta Escalón (Bucle Abierto)")
```



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

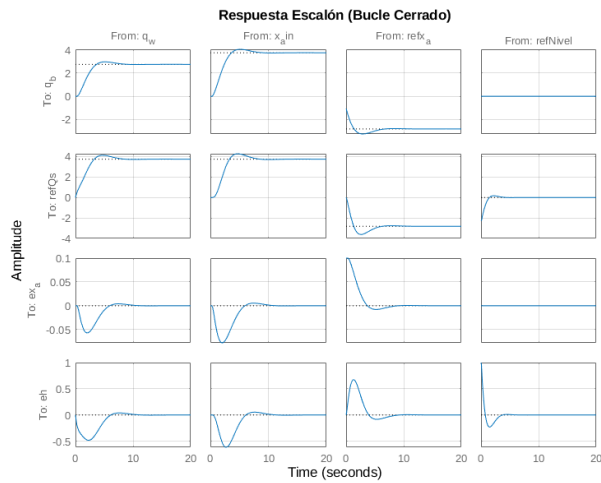
Referencias primarias tenemos para  $x_{aout}$  y para nivel.

Variables manipuladas serán  $refQs$  y  $q_{bin}$ .

La filosofía básica será multibucle nivel  $\rightarrow$   $refQs$ ,  $x_{aout} \rightarrow q_{bin}$ . Éste es el resultado de un diseño previo, que analizaremos en tiempo y frecuencia.

**Respuesta temporal multibucle+cascada:**

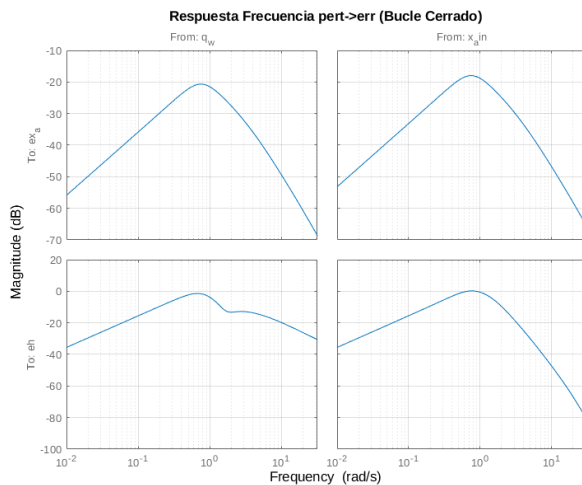
```
cosa=BC_multibucle*diag([1 .5 .1 1]); %amplitudes de perturb/referencias no unitarias
cosa.InputName=BC_multibucle.InputName;
step(cosa([3 4 6 5],:)), grid on, title ("Respuesta Escalón (Bucle Cerrado)")
```



### Respuesta en frecuencia multibucle+cascada:

El análisis de la respuesta en frecuencia me determinará el "ancho de banda" que un posible feedforward/desacoplamiento necesitaría. Como sólo quiero que el error (salidas 5 y 6) sea "pequeño", lo único interesante es el Bode de Magnitud.

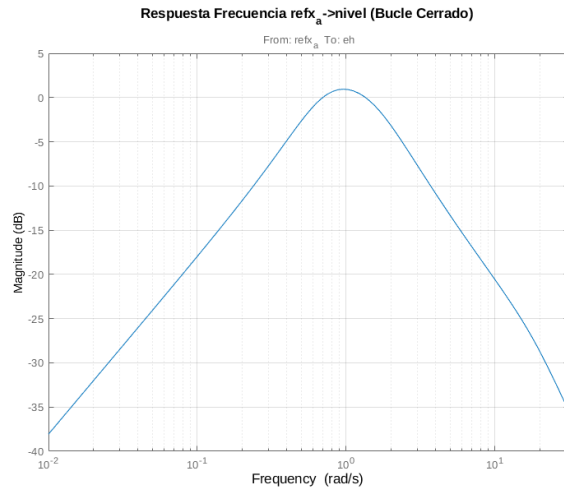
```
bodemag(cosa([6 5],[1 2]),logspace(-2,1.5,100)), grid on, title ("Respuesta Frecuencia
```



Se comprueba que sólo es necesario cancelar las perturbaciones hasta 10 rad/s, aproximadamente: luego el propio comportamiento del proceso de mezclado hace que se filtre su efecto; a frecuencias bajas el efecto se anula debido a las acciones integrales del control multibucle.

Si analizamos ahora el efecto "cruzado" (acoplamiento) entre el bucle de concentración y nivel, resulta:

```
bodemag(cosa(5,3),logspace(-2,1.5,100)), grid on, title ("Respuesta Frecuencia refx_a->
```



**Objetivo:** refinar el multibucle con prealimentación de  $q_{ain}$ ,  $x_{ain}$  para comprobar su efecto en simulación y decidir si merece o no la pena comprar dichos sensores adicionales, así como hacerlo estático o dinámico.

## Feedforward (prealimentación): opción MIMO --inversión de matrices--

**Nota:** el feedforward sobre el control de concentración podría considerarse con "ratio" y otras ecuaciones no lineales calculando fracciones de productos en los caudales entrantes y salientes, así se hará en el caso "no lineal". Pero el modelo linealizado sólo sabe de efectos "**aditivos**" y no es capaz de considerar efectos "**multiplicativos**" entre varias señales.

Supondremos que tenemos medida de las perturbaciones... En su momento se compararán simulaciones para decidir si compensa o no la modificación de la instrumentación para instalar esos sensores adicionales.

Las perturbaciones exógenas (no controlables) son las entradas 1 y 3:

```
Gdisturb=minreal(G2(:, [1 3]));
```

```
1 state removed.
```

```
Gdisturb.OutputDelay
```

```
ans = 2x1
    0.2489
         0
```

Las entradas manipuladas son las 2 y 4, pero hay un retraso:

```
Gcontrol=minreal(G2(:, [2 4]));
Gcontrol.OutputDelay
```

```
ans = 2x1
    0.2489
         0
```

## Caso 1: Prelimentación estática (sólo ganancia)

Como hay integradores, ganancia infinita cambiaremos por la respuesta a "muy baja frecuencia" ( $s = 10^{-3}$ ) :

```
GanPerturb=evalfr(Gdisturb,1e-3);
GanControl=evalfr(Gcontrol,1e-3);
```

Con ello, el FF multivariable, independientemente de si se desacopla o no, sería:

```
FF=-GanControl\GanPerturb
```

```
FF = 2x2
    2.7501    7.5004
    3.7502    7.5003
```

## Caso 2: Prealimentación dinámica (función de transf. completa)

Como el retardo coincide en Gdisturb y Gcontrol, se cancelarán en la inversa, por lo que podemos operar sin él (¡atención!: esto **NO** es el caso general):

```
GcontrolSinRetardo=Gcontrol;
GdisturbSinRetardo=Gdisturb;
GcontrolSinRetardo.OutputDelay=0;%en las inversas con retardo, Matlab se "queja".
GdisturbSinRetardo.OutputDelay=0;
```

Con lo que, aplicando la fórmula "como si no hubiese retardo", resulta:

```
FeedForwardDyn=-GcontrolSinRetardo\GdisturbSinRetardo;
minreal(tf(FeedForwardDyn))
```

```
ans =

From input "q_w" to output...
      8.956e09 s^3 + 8.506e21 s^2 + 2.087e23 s + 7.723e23
q_b:  -----
      s^3 + 8.707e10 s^2 + 6.186e22 s + 2.808e23

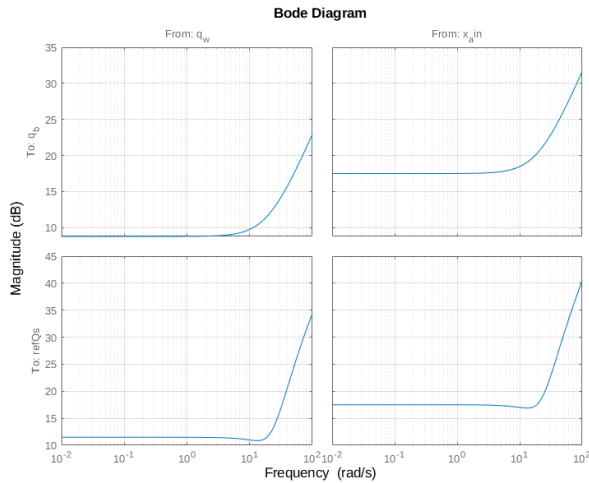
      5.141e10 s^5 + 4.48e22 s^4 + 2.272e24 s^3 + 7.216e25 s^2 + 1.099e27 s + 1.049e28
refQs: -----
      s^5 + 8.707e10 s^4 + 6.186e22 s^3 + 7.317e24 s^2 + 1.767e26 s + 2.796e27

From input "x_ain" to output...
q_b:  0.375 s + 7.5

      1.448 s^2 + 36.47 s + 695.6
refQs: -----
      s + 92.74
```

Continuous-time transfer function.

```
bodemag(FeedForwardDyn,logspace(-2,2,100)), grid on
```



No es realizable: como decisión de diseño (muy común), despreciaremos y haremos sólo la ganancia... cancelará correctamente hasta ancho de banda de aproximadamente 10 rad/s, lo que resultará satisfactorio (revisando bodes con el multibucle, no parece que sea necesario cancelar más allá de esa frecuencia).

```
FF2=evalfr(FeedForwardDyn,1e-3)
```

```
FF2 = 2x2
      2.7501    7.5004
      3.7502    7.5003
```

Obviamente, coincide con el cálculo a partir de las ganancias (frec. lenta).

```
FF
```

```
FF = 2x2
      2.7501    7.5004
      3.7502    7.5003
```

```
FF_tf=tf(FF);
%Esto lo hacemos simplemente para ponerlo como un bloque LTISystem de Simulink.
```

De hecho, en la solución final que se adoptará más adelante, se decidirá refinar el feedforward incorporando *no linealidad* ( $\approx$  *ratio control*) en el cálculo de punto de funcionamiento (constante), en vez de *dinámica*.

## Desacoplamiento

```
DesacQbToQs=-minreal(Gcontrol(2,2)\Gcontrol(2,1));
```

```
11 states removed.
```

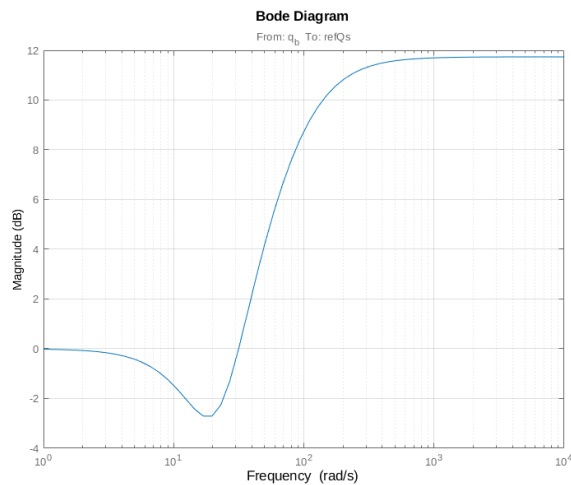
```
zpk(DesacQbToQs)
```

```
ans =
```

```
From input "q_b" to output "refQs":
3.8625 (s^2 + 25.18s + 480.2)
-----
      (s+20) (s+92.74)
```

Continuous-time zero/pole/gain model.

```
bodemag(DesacQbToQs), grid on
```



Es realizable, y no hay un pico demasiado grande a alta frecuencia, por lo que sí podríamos hacer desacoplamiento "dinámico", o usar sólo la ganancia y ser estático (descoplaría hasta ancho de banda de 10 rad/s, que sería suficiente según el análisis inicial).

```
DesacEstatico=dcgain(DesacQbToQs)
```

```
DesacEstatico = 1.0000
```

Esta vez, por variar y demostrar cómo simular el desacoplamiento dinámico, vamos a elegir dicho desacoplamiento dinámico en la solución final, aunque seguramente la solución simple "estática" hubiese funcionado suficientemente bien para cambios de referencia por debajo de 10 rad/s (semiperíodo de  $\pi/10 = 0.3$  unidades de tiempo).

```
save FFyDC.mat DesacQbToQs FF_tf %para futuras simulaciones
```

## Conclusiones

Se ha evaluado el ancho de banda necesario del feedforward una vez está conectado el control multibucle+cascada (que, debido a la acción integral, consigue las referencias deseadas a frecuencia cero). Se ha visto que 10 rad/s es suficiente y que dicho ancho de banda se consigue con la implementación constante (estática). Un análisis similar también sería válido para desacoplamiento, aunque se haya elegido, por motivos "ilustrativos" para otros vídeos la solución dinámica.