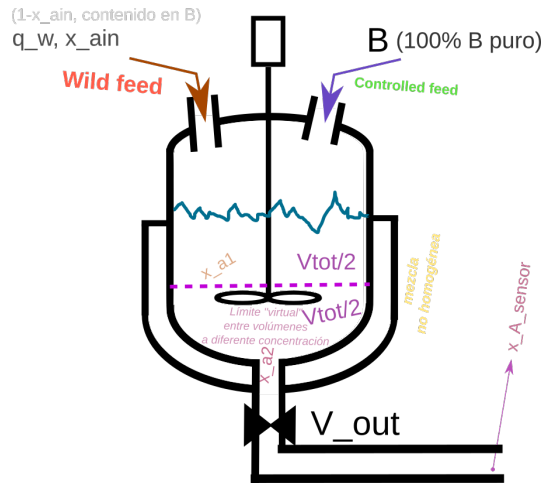


Linealización mediante Simulink en un conjunto de varios puntos de funcionamiento de un sistema

© 2020, Antonio Sala Piqueras.

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

Este código funcionó correctamente con Matlab R2019b.



Punto de funcionamiento nominal para control posterior

Calculemos un punto de funcionamiento de caudales. Sistema debe diseñarse nominalmente para un caudal "wild feed" entrante dado por:

```
PFqw=1.5; %1.5 nominalmente, rango esperado [1 2]
```

Y una concentración nominal de:

```
PFxain=.75; %75 por cien de pureza en A, rango esperado [.6 .9]
```

Nuestro objetivo es producir una concentración de salida de A dada por:

```
PFxasalida=0.2;
```

Con la válvula de salida posicionada nominalmente en:

```
PFVout=0.6;
```

Tendremos:

```
caudalAentrante=PFqw*PFxain
```

```
caudalAentrante = 1.1250
```

```
caudalBentranteizquierda=PFqw-caudalAentrante
```

```
caudalBentranteizquierda = 0.3750
```

```
caudalTotal=caudalAentrante/PFxasalida
```

```
caudalTotal = 5.6250
```

Con lo que el caudal de B entrante debería ser:

```
PFqb=caudalTotal-PFqw
```

```
PFqb = 4.1250
```

```
%return
```

Linealización en varios puntos de funcionamiento

Realmente, para comprobar el grado de no-linealidad, probaremos un conjunto de caudales y concentraciones de entrada. La conc. de salida deseada no cambiará:

```
PFqa_rango= [1 1 1 1.5 2 2 2];  
PFxa_rango= [.6 .75 .88 0.75 .62 .75 .9];  
PFVout_rango=[.2 .3 .4 0.6 0.6 .95 .95];  
GuardarNominal=4;  
%PFqa_rango=PFqa;PFxa_rango=PFxa;PFVout_rango=PFVout
```

Por tanto, para cada punto haremos:

```
N_PF=length(PFqa_rango);  
for i=1:N_PF  
    PFqw=PFqa_rango(i);  
    PFxain=PFxa_rango(i);  
    PFVout=PFVout_rango(i);  
    caudalAentrante=PFqw*PFxain;  
    caudalBentranteizquierda=PFqw-caudalAentrante;  
    caudalTotal=caudalAentrante/PFxasalida  
    PFqb=caudalTotal-PFqw
```

Linealizamos ahora con Simulink: simulamos hasta 40 segundos para obtener equilibrio y usamos "linearize":

```
IO = getlinio('tanquemezcladoSimularLinealizar');  
options = linearizeOptions('SampleTime',0,'UseExactDelayModel','on');  
[lin]=linearize('tanquemezcladoSimularLinealizar',40,IO,options);  
simout=sim("tanquemezcladoSimularLinealizar");  
  
SistemasLinealiz{i}=lin;  
nivel{i}=simout.Hfinal;  
end
```

```
caudalTotal = 3.0000  
PFqb = 2.0000  
caudalTotal = 3.7500  
PFqb = 2.7500  
caudalTotal = 4.4000  
PFqb = 3.4000  
caudalTotal = 5.6250
```

```

PFqb = 4.1250
caudalTotal = 6.2000
PFqb = 4.2000
caudalTotal = 7.5000
PFqb = 5.5000
caudalTotal = 9
PFqb = 7

```

```

G=SistemasLinealiz{GuardarNominal};
NivelNominal=nivel(GuardarNominal)

```

```

NivelNominal = 1x1 cell array

    {[7.1747]}

```

```

zpk(G)

```

```

ans =

```

```

From input "q_w" to output...

```

```

                                -2.7797e-12 (s-1.35e11) (s+0.3564)
x_aout: exp(-0.25*s) * -----
                                (s+4.54) (s+0.8455) (s+0.3564)

```

```

                                0.90909
nivel: -----
                                (s+0.3564)

```

```

                                0.35636
Qs: -----
                                (s+0.3564)

```

```

From input "q_b" to output...

```

```

                                -6.7336e-11 (s+4.053e10) (s+0.3564)
x_aout: exp(-0.25*s) * -----
                                (s+4.54) (s+20) (s+0.8455) (s+0.3564)

```

```

                                18.182
nivel: -----
                                (s+0.3564) (s+20)

```

```

                                7.1273
Qs: -----
                                (s+0.3564) (s+20)

```

```

From input "x_ain" to output...

```

```

                                1.0235
x_aout: exp(-0.25*s) * -----
                                (s+4.54) (s+0.8455)

```

```

nivel: 0

```

```

Qs: 0

```

```

From input "Valv_out" to output...

```

```

                                5.9526e-10 (s+0.503) (s+5.203)
x_aout: exp(-0.25*s) * -----
                                (s+4.54) (s+20) (s+0.8455) (s+0.3564)

```

```

                                -170.45
nivel: -----
                                (s+0.3564) (s+20)

```

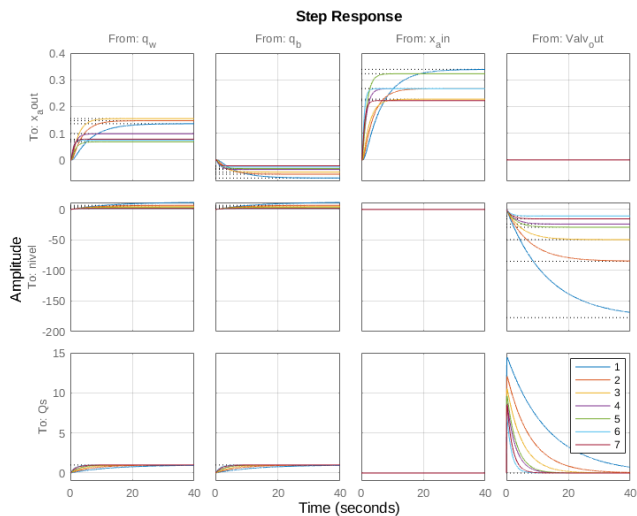
$$Qs: \frac{187.5}{(s+0.3564)(s+20)}$$

Continuous-time zero/pole/gain model.

```
save ModeloNominal.mat G NivelNominal
```

Comprobemos la diferente respuesta escalón (1 unidad de tiempo de simulación) en diferentes puntos de funcionamiento:

```
for i=1:N_PF
    step(SistemasLinealiz{i},40)
    hold on
end
hold off
grid on
legend('1','2','3','4','5','6','7')
```



Y la respuesta en frecuencia:

```
for i=1:N_PF
    bodemag(SistemasLinealiz{i},logspace(-2,2))
    hold on
end
hold off
grid on
legend('1','2','3','4','5','6','7')
```

