Identificación de modelo de primer/segundo orden + retardo con procest (system ID toolbox)

Presentación en vídeo: http://personales.upv.es/asala/YT/V/procest.html

*Este código ejecutó sin errores en Matlab **R2019b**

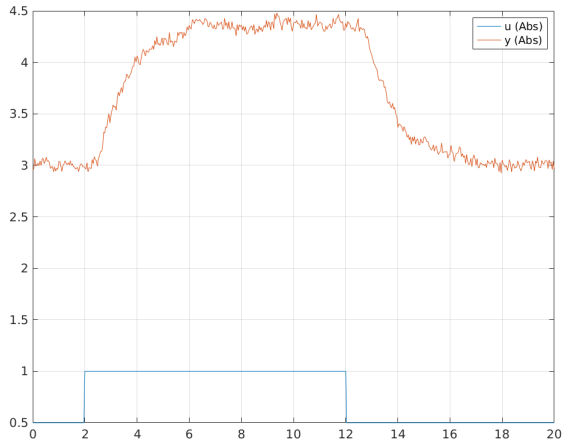
Motivación: Los modelos de primer orden + retardo, $G(s)=\frac{k}{\tau s+1}e^{-ds}$, son una aproximación de dinámicas no oscilatorias muy usada en procesos químicos (donde existen retardos de transporte, transferencias de calor, reacciones químicas, etc... sin oscilaciones). También son frecuentesmodelos de segundo orden + retardo en la forma $G(s)=\frac{k}{(\tau_1 s+1)(\tau_2 s+1)}e^{-ds}$. Existen reglas/tablas para sintonizar controladores (PID) en esos tipos de sistemas.

Objetivos: el objetivo de este material es usar la System Identification Toolbox (`procest`) para ajustar esos modelos a unos datos experimentales.

Tabla de Contenidos

Visualización de los datos

```
load Experimento.mat
plot(T,[ureal yreal]), grid on, legend('u (Abs)','y (Abs)')
```



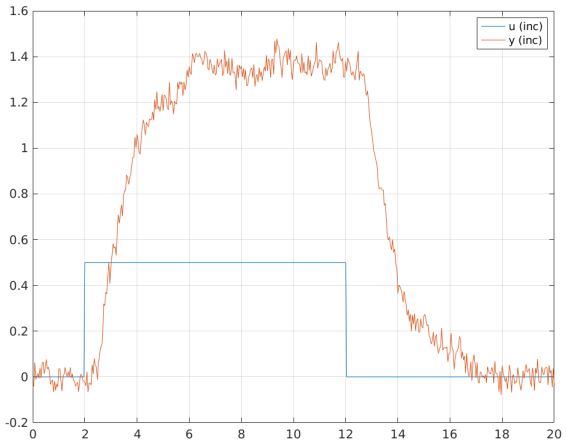
Variables incrementales: para identificar un modelo ***lineal*** alrededor de un punto de funcionamiento, debemos restar el valor anterior al escalón de entrada, y el valor medio de la salida antes de aplicar el escalón.

```
muestras_antes_escalon=1:round(2/Ts);
y_pf=mean(yreal(muestras_antes_escalon));
u_pf=mean(ureal(muestras_antes_escalon));
y=yreal-y_pf; u=ureal-u_pf;
plot(T,[u y]), grid on, legend('u (inc)','y (inc)')
```

Identificación con process model estimation (system ID toolbox)

Primer orden + retardo

```
Data1=iddata(y,u,Ts);
tic,sys=procest(Data1,'P1D');toc
```



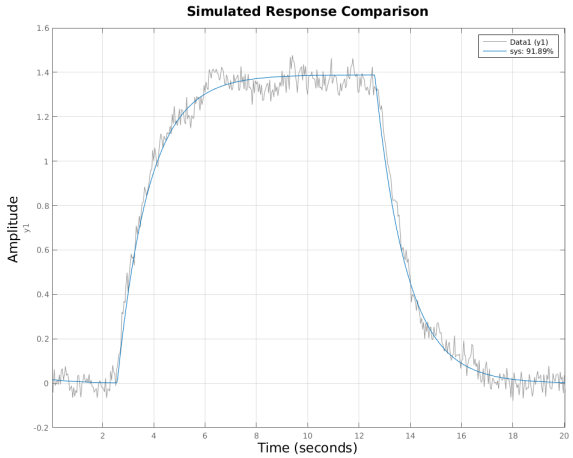

```
Elapsed time is 0.485527 seconds.
```

```
present(sys)
```

```
sys =
Process model with transfer function:
             Kp
  G(s) = ---------- * exp(-Td*s)
          1+Tp1*s

         Kp = 2.7773 +/- 0.0070757
        Tp1 = 1.2333 +/- 0.016837
         Td = 0.53752 +/- 0.011491

Parameterization:
    'P1D'
   Number of free coefficients: 3
   Use "getpvec", "getcov" for parameters and their uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 4, Number of function evaluations: 9

Estimated using PROCEST on time domain data "Data1".
```

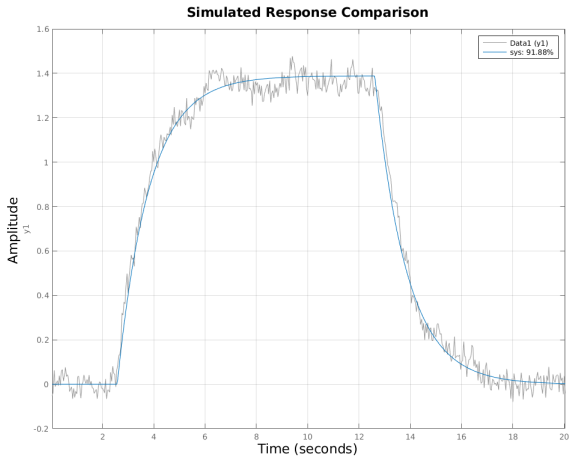
```
Fit to estimation data: 91.88%
FPE: 0.0024, MSE: 0.002372
More information in model's "Report" property.
```

```
compare(Data1,sys), grid on
```



*Por defecto, ajusta las condiciones iniciales por si acaso los datos experimentales no estaban inicialmente en equilibrio. Para desactivarlo:

```
opt = compareOptions('InitialCondition','zero');
compare(Data1,sys,opt), grid on
```



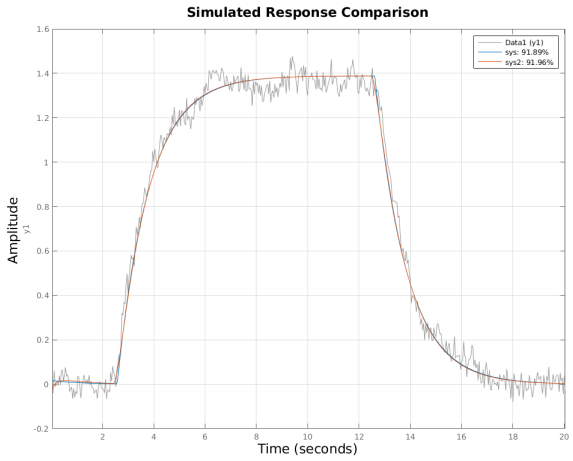
Segundo orden + retardo

```
sys2=procest(Data1,'P2D');
present(sys2)
```

```
sys2 =
Process model with transfer function:
                Kp
  G(s) = ----------------- * exp(-Td*s)
         (1+Tp1*s)(1+Tp2*s)
```

```
          Kp = 2.7763 +/- 0.0071204
         Tp1 = 1.2065 +/- 0.023838
         Tp2 = 0.16454 +/- 0.046711
          Td = 0.3908 +/- 0.037375

Parameterization:
    'P2D'
   Number of free coefficients: 4
   Use "getpvec", "getcov" for parameters and their uncertainties.

Status:
Termination condition: Near (local) minimum, (norm(g) < tol)..
Number of iterations: 6, Number of function evaluations: 16

Estimated using PROCEST on time domain data "Data1".
Fit to estimation data: 91.94%
FPE: 0.002375, MSE: 0.002338
More information in model's "Report" property.
```

```
compare(Data1,sys,sys2), grid on
```



No parece que la diferencia de ajuste justifique el incremento de complejidad del modelo. Habría que confirmar con datos de validación (experimento repetido en otro momento, con otra amplitud, o/y con otra forma de onda) que los modelos son aceptables.

Conclusiones

Los modelos de primer orden + retardo son una simplificación de la dinámica de sistemas muy usual en industria de procesos y en literatura sobre sintonizado de PIDs. Su identificación "manual" resulta sencilla basándose en las fórmulas bien conocidas de sistemas de primer orden. Su identificación automática también resulta sencilla en unas pocas líneas de código de la System Identification Toolbox de Matlab; la toolbox también permite identificar sistemas de segundo orden + retardo rápidamente, para comparar la diferencia de ajuste.