



Identificación de sistemas

- ✍ Definición del problema
- ✍ Metodología general
 - ✓ Sistemas no lineales
- ✍ Ejemplo
 - ✓ Utilización de Matlab
 - ✓ Sistema: Reactor biológico
 - ✓ Resultados



Definición del problema

- ✍ Definición del problema
 - ✓ Se propone un modelo
 - ✓ Ecuaciones diferenciales
 - Principios de conservación + Leyes
 - ✓ Parámetros
 - Particularización para un sistema real
 - ✓ Objetivos
 - ✓ Validar el modelo
 - ✓ Determinación de parámetros (identificación)



Definición del problema

Identificación

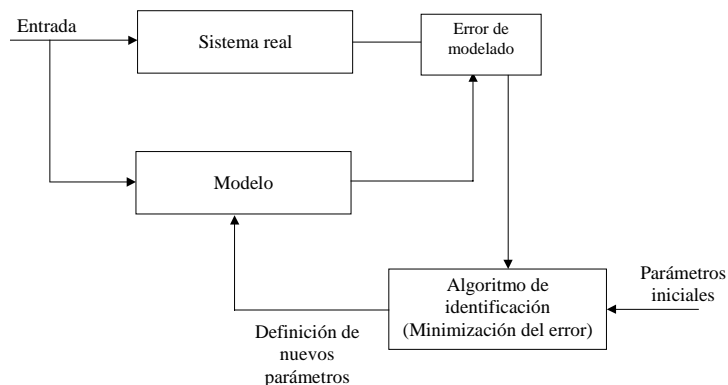
- ✓ Dado un sistema del que se conoce su funcionamiento de forma experimental determinar su modelo
- ✓ En general se supone que se conoce el tipo de modelo

Clasificación de técnicas

- ✓ Sistemas lineales
 - ✓ Métodos bien definidos
 - ✓ No aportan información acerca de la naturaleza del modelo
- ✓ Sistemas no lineales
 - ✓ Tienen en cuenta la naturaleza del modelo
 - ✓ Identifican el valor de los parámetros de las leyes aplicadas
 - Ley de Monod (m_{max} , K_m)



Metodología general





Metodología general

Elementos que intervienen

- ✓ Modelo
 - ✓ Ecuación diferencial, función de transferencia
- ✓ Función a minimizar
 - ✓ Entrada
 - Parámetros
 - ✓ Salida
 - Error sistema simulado/sistema real (eqm)
 - ✓ Realiza la simulación del sistema no-lineal
 - Ejemplo: *ode45*
- ✓ Función de minimización
 - ✓ `parametros=fmins("ferror",Par_iniciales);`



Ejemplo. Identificación en Matlab

Procedimiento en Matlab

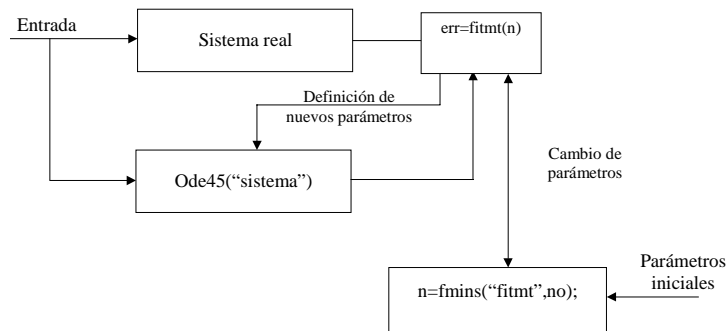
- ✓ Función *fmins*
 - ✓ Utiliza técnicas de optimización

Ejemplo

- ✓ Reactor biológico
 - ✓ Sistema con seis grados de libertad
 - ✓ Utilización de los datos experimentales



Ejemplo. Procedimiento en Matlab



Ejemplo. Modelo del sistema

Modelo del sistema

```
function qpunto=qtotal2(t,q)
global t_exp k1 ks k2 k3 kglic krimit kresp po Yatp
mu=(((2*kglic*q(2))+((4*po+1)*kresp*q(4)*(q(6)^3)))*Yatp);
Rrimit=k2*(q(3)/(krimit+q(3)));
Rresp=kresp*q(4)*(q(6)^3);
Rgluc=k1*(q(1)/(ks+q(1)));
our_t=(k3/k1)*q(5)*Rgluc;
qpunto(1)=-q(5)*Rgluc;
qpunto(2)=q(5)*(Rgluc-(kglic*q(2)+mu));
qpunto(3)=q(5)*(2*kglic*q(2)-(Rrimit+mu));
qpunto(4)=q(5)*(Rrimit-Rresp);
qpunto(5)=q(5)*mu;
qpunto(6)=-q(5)*Rresp+our_t;
```



Ejemplo

```
function err = fitmt(n2)
%
q0=[s(1) .3 0. .5 x(1) .5];
[t_t,q]=ode45('qttotal2',min(t_exp),max(t_exp),q0);
save datos t_t q
err1=0;
err2=0;
%Cálculos del error
```

Es necesario realizar una interpolación de datos

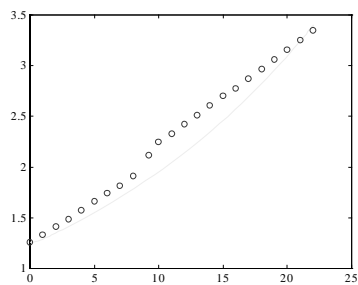
```
Fichero de ajuste
% Los valores iniciales de las constantes son
n2=[0.7 0.0118 1.1175 2.4e-4 0.0023 0.3568 1.1027 0.8517
0.0085];
opts(1)=0;
opts(2)=1e-4;
opts(3)=1;
opts(14)=100000;
ns = fminsearch('fitmt',n2,opts);
```

Función de optimización

Resultados



Ejemplo. Resultados del ajuste



El resultado son los parámetros

***K1, ks, k2, k3,
kglic, krimit,
kresp, po, Yatp***

