

Modelado Parte II

Enrique E. Tarifa, Facultad de Ingeniería, UNJu

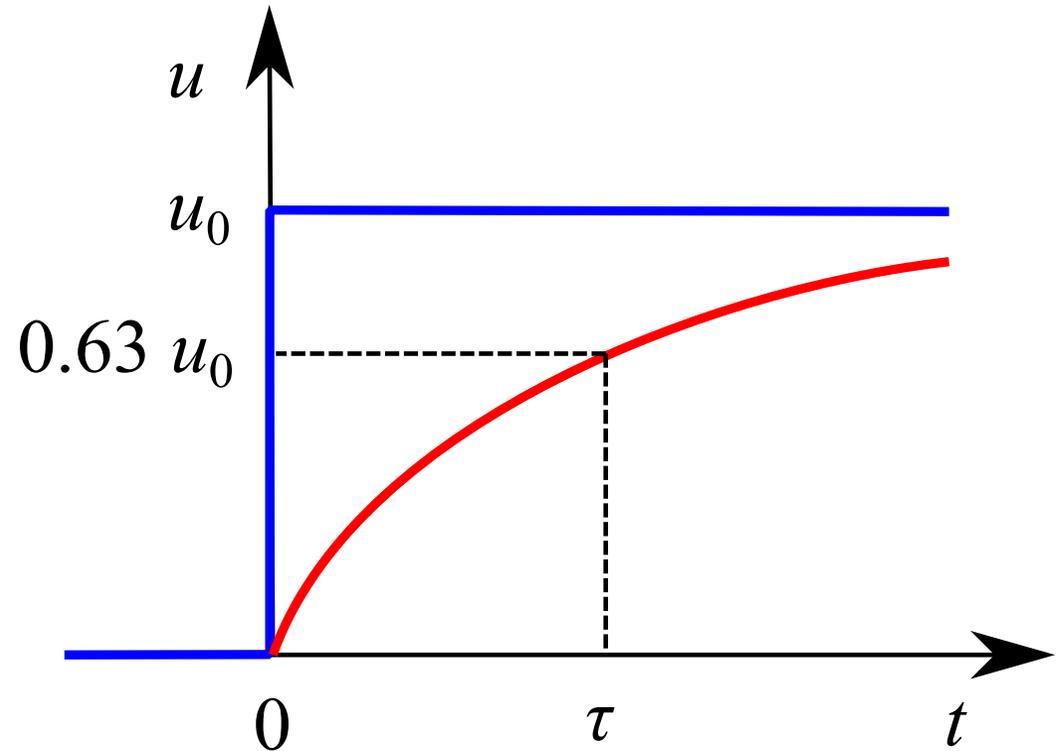
Sistema de primer orden

Sistema de primer orden

$$\begin{cases} \frac{dy}{dt} = \frac{u-y}{\tau} \\ y(0) = 0 \end{cases}$$

$$u = \begin{cases} 0 & t < 0 \\ u_0 & t \geq 0 \end{cases}$$

$$y = \begin{cases} 0 & t < 0 \\ u_0 (1 - e^{-t/\tau}) & t \geq 0 \end{cases}$$



Estado pseudoestacionario

- Estado dinámico: $\tau \frac{dy}{dt} = u - y$
- Estado estacionario: $\frac{dy}{dt} = 0 \Rightarrow u - y = 0$
- Estado pseudoestacionario: $\tau \rightarrow 0 \Rightarrow u - y \rightarrow 0$

Alternativas para la resolución

Enfoques para la resolución de modelos



Orientado a ecuaciones

- E-Z Solve
- JSim
- EMSO
- Modelica
- gPROMS
- Berkeley Madonna

$$\frac{dy}{dt} = \frac{u_0 - y}{\tau}$$

[Video de Simulación con Berkeley Madonna](#)

```
{Sistema de primer orden.  
u0: valor del escalón en la entrada.  
tau: es la constante de tiempo.  
y: valor de salida}
```

```
METHOD RK4
```

```
STARTTIME = 0
```

```
STOPTIME = 10
```

```
DT = 0.01
```

```
; Inicialización
```

```
INIT y = 0
```

```
; Sistema ODEs
```

```
y' = (u0-y)/tau
```

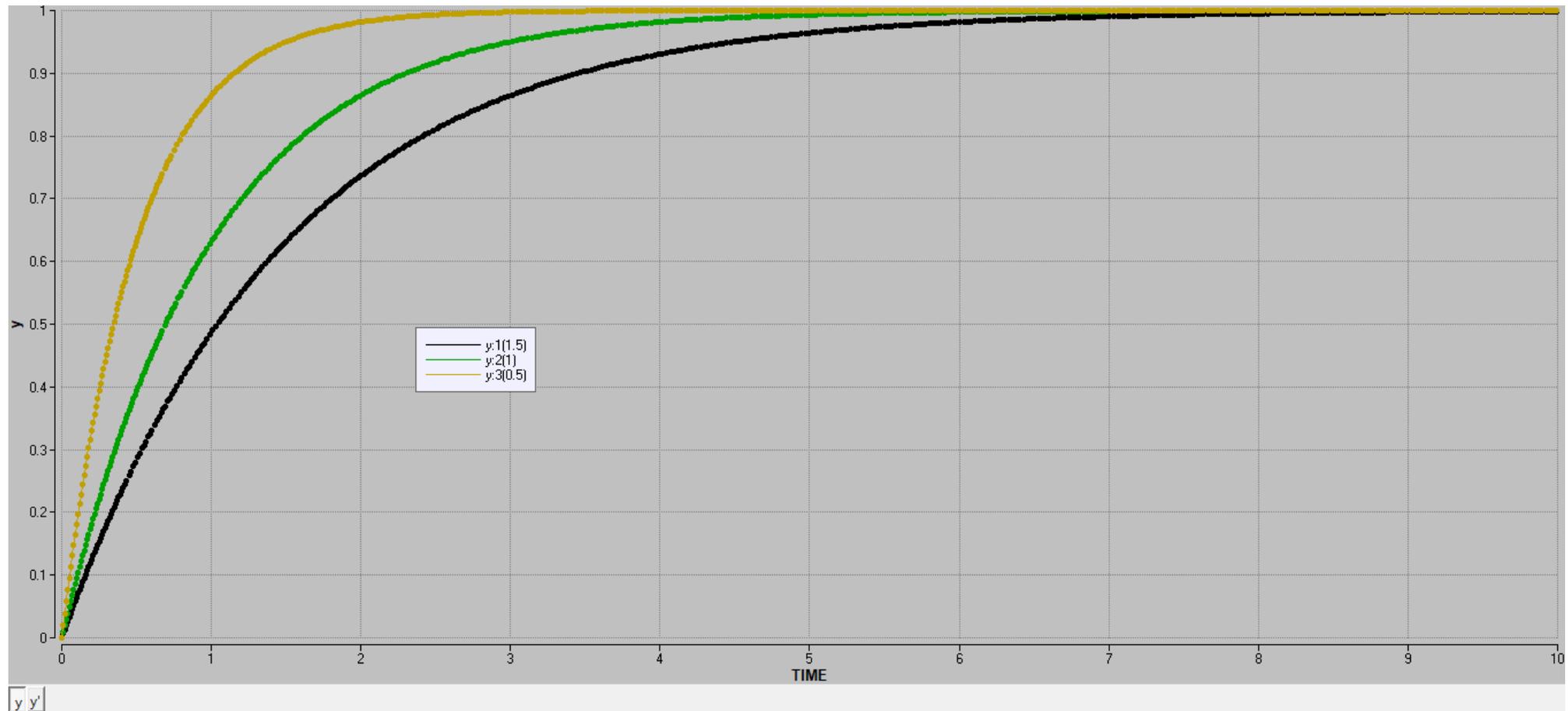
```
; Sistema AEs
```

```
; Datos
```

```
u0 = 1
```

```
tau = 1
```

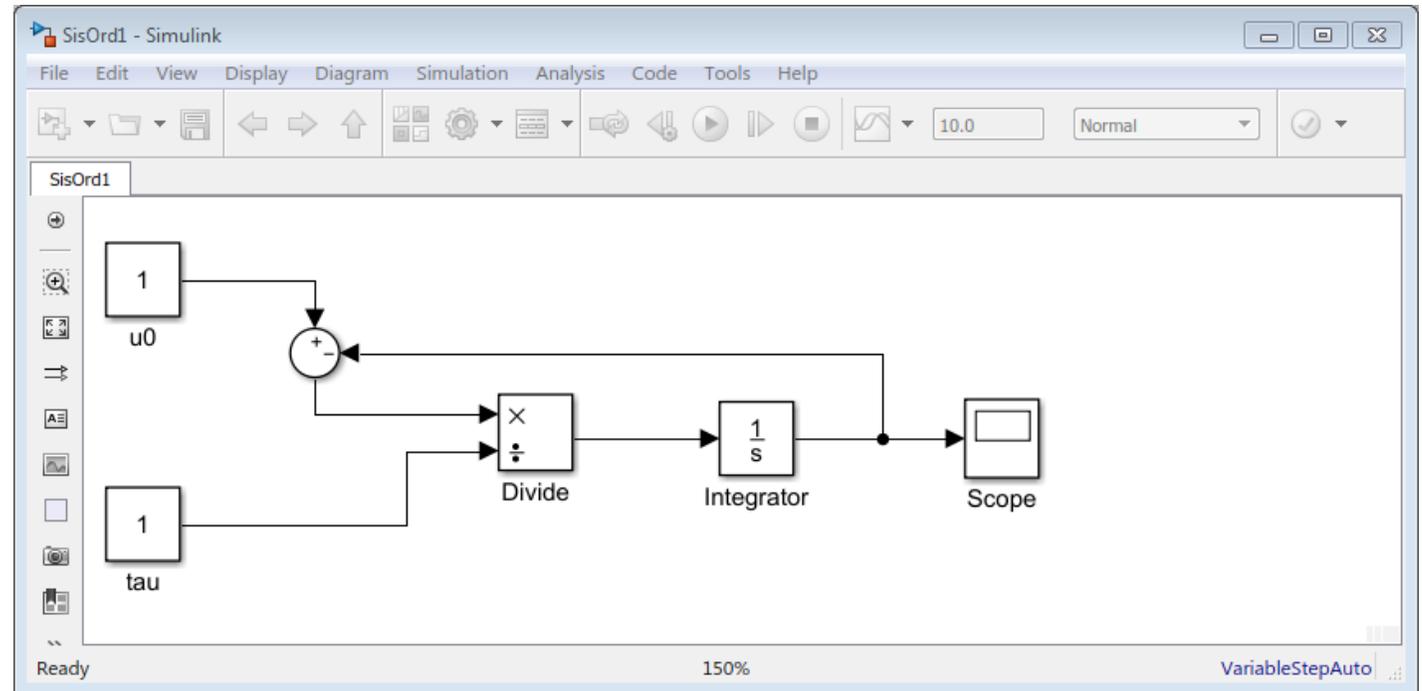
Constante de tiempo



Orientado a diagramas de bloques

- Simulink
- ViSim
- Xcos

$$\frac{dy}{dt} = \frac{U_0 - y}{\tau}$$



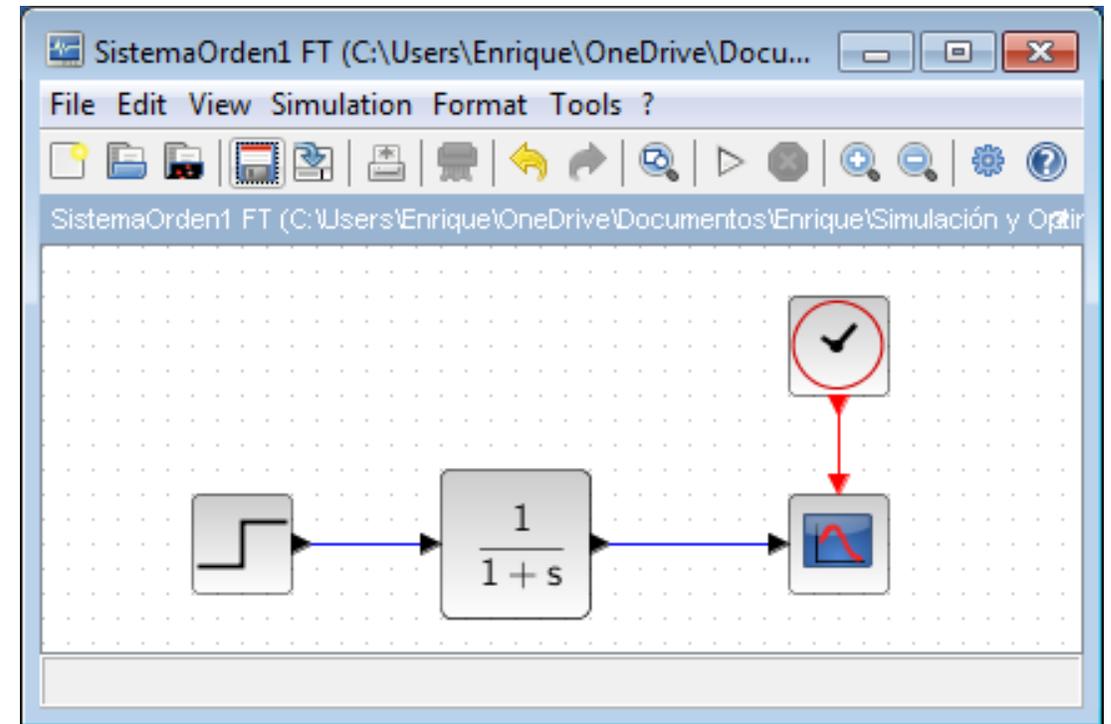
[Video Simulación con Simulink](#)

Orientado a diagramas de bloques

- Simulink
- ViSim
- Xcos

$$\frac{dy}{dt} = \frac{U_0 - y}{\tau}$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{1 + \tau s}$$



Simulación con Xcos

Condiciones iniciales iguales a cero.

Orientados a programación

- Lenguajes:
 - Fortran, C, Pascal, Python, Julia
- Entornos:
 - Matlab, Octave, Scilab
 - Spyder, Visual Studio Code
 - Mathcad, SMath

$$\frac{dy}{dt} = \frac{u_0 - y}{\tau}$$

[Video Simulación con Matlab](#)

[Manual de GNU Octave](#)

```
% Sistema de primer orden

% Datos
global tau u0
tau = 1;
u0 = 1;

% ODEs
function dy = ODEs(y,t)
    global tau u0
    dy = (u0-y)/tau;
endfunction

% Parámetros de simulación
tfin = 10;
nts = 20;

% Inicialización
tpts = linspace(0, tfin, nts)';
y0 = 0;

% Resolución
y = lsode('ODEs',y0,tpts);

% Gráfica
figure(1);
plot(tpts,y)
```