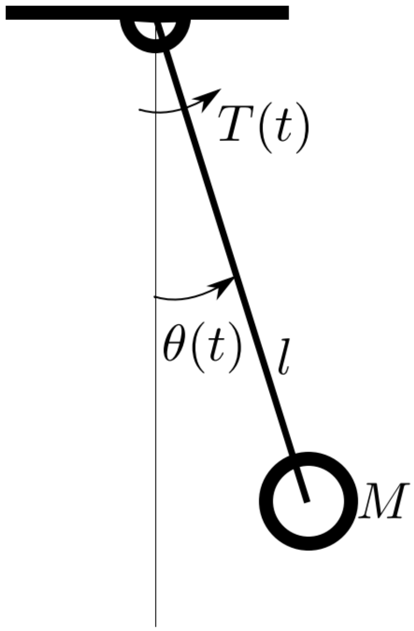
Práctica 1.3: Péndulo: comparación linealizado vs no lineal

Diagrama de fuerzas y sus ecuaciones

Objetivo: en este Live Script construiremos el péndulo simple en dos versiones: la versión no lineal (que es la verdadera) y la linealizada (que es una aproximación de la anterior).



Dado el péndulo que se muestra en la figura y sabiendo que la dinámica del mismo viene dada por las siguientes ecuaciones:

$$\sum T_i(t) = I\ddot{\theta}(t)$$

$$I = mL^2$$

donde I es el momento de inercia del péndulo, m su masa, L la longitud de la varilla y $\theta(t)$ el ángulo respecto a la vertical.

Se considera un par de entrada externo, $T(t)$, además del par debido al peso, $T_p(t)$, y el par debido al rozamiento, $T_r(t)$:

$$T_p(t) = -mgL\sin\theta(t)$$

$$T_r(t) = -bL^2\dot{\theta}(t)$$

donde g es la aceleración de la gravedad y b el coeficiente de rozamiento de la masa (se considera despreciable el rozamiento en la varilla).

Parámetros

```
m = 2;
b = 0.8;
L = 3;
```

```
g = 9.81;
theta_0 = 0.1;
```

El signo negativo en T_p se debe a que el par gravitatorio tiende a devolver el péndulo a la posición de equilibrio. En el caso de T_r, el signo se debe a que la fuerza siempre se opone al sentido del movimiento.

La ecuación dinámica completa queda entonces:

$$T(t) + T_p(t) + T_r(t) = I\,\ddot{\theta}(t)$$

$$T(t) - mgL\sin\theta(t) - bL^2\dot{\theta}(t) = mL^2\,\ddot{\theta}(t)$$

Procedemos a poner en forma de espacio de estados. En este caso, al haber una derivada segunda, se añade una ecuación de la forma: $\dot{\theta} = \omega$ para poder expresar todo en función de los estados y sus derivadas primeras (es decir, para "deshacernos" de la segunda derivada).

$$T(t) - mgL\sin\theta(t) - bL^2\omega(t) = mL^2\,\dot{\omega}(t)$$

Vamos a comparar el modelo no lineal con su aproximación lineal. **IMPORTANTE**:el funcionamiento real es el no lineal, el otro es una aproximación de este. Para ello, expresaremos las ecuaciones en forma de espacio de estados (aislando las derivadas de primer orden y sin tener derivadas de mayor orden). Linealizando la ecuación no lineal ya en EE obtenemos la versión linealizada.

1) **No lineal** usando la ecuación exacta

$$\dot{\omega} = -\frac{b}{m}\omega - \frac{g}{L}\sin\theta + \frac{T}{mL^2}$$

$$\dot{\theta} = \omega$$

2) **Linealizado** alrededor de $\theta = 0$. Linealizando en torno a $\theta_0 = 0$ se obtiene:

$$\dot{\omega} = -\frac{b}{m}\omega - \frac{g}{L}\theta + \frac{T}{mL^2}$$

$$\dot{\theta} = \omega$$

NOTA: Al introducir un escalón de par, el punto de equilibrio cambia. Ese transitorio se modelaría mejor si se recalcula la linealización para ese punto, pero lo omitimos por simplicidad (para no tener que considerar distintos valores de punto de equilibrio y, por tanto, distintas versiones linealizadas de A).

1) Espacio de estados (EE)

Construimos las matrices de espacio de estados, considerando los estados $x = \begin{bmatrix}\omega \\ \theta\end{bmatrix}$

```
A = [ -b/m    -g/L;
       1       0   ];
```

```
B = [ 1/(m*L^2);
      0       ];

C = [0 1];  % Salida = theta
D = 0;
```

Con la siguiente instrucción, se construye el sistema en espacio de estados

```
sys_ss = ss(A,B,C,D);
```

2) Función de Transferencia

A partir de la matriz en espacio de estados, se puede obtener la función de transferencia usando la función tf.

```
sys_tf = tf(sys_ss)
```

```
sys_tf =

      0.05556
  ------------------
  s^2 + 0.4 s + 3.27

Continuous-time transfer function.
Model Properties
```

Podemos comprobar que coincide con la calculada "manualmente" (simplificando diagrama de bloques). A continuación, dos maneras de definir una FdT sabiendo los coeficientes de numerador y denominador.

Primero, escribiendo directamente en función de s. Para ello, primero hay que definir s como una FdT, ya que s no es una variable por defecto. Haciendo eso, el programa ya interpreta que cualquier función donde aparezca s es una FdT.

```
s   = tf('s');
sys_tf_calc = 1/(m*L^2*s^2 + b*L^2*s + m*g*L)
```

```
sys_tf_calc =

            1
  ----------------------
  18 s^2 + 7.2 s + 58.86

Continuous-time transfer function.
Model Properties
```

La segunda se puede hacer expresando los coeficientes del numerador y denominador en el polinomio en s. La última posición siempre indica el término independiente. Por ejemplo, si el vector fuera [2 1 0] el polinomio asociado sería $2s^2 + s$. Importante darse cuenta que, si no hay término independiente (o, en general, si se "salta" cualquier término) tiene que haber un 0 en dicha posición.

```
num = 1;
den = [m*L^2 b*L^2  m*g*L];
tf(num,den)
```

```
ans =
```

```
           1
  ----------------------
  18 s^2 + 7.2 s + 58.86

Continuous-time transfer function.
Model Properties
```

Vemos que obtenemos lo mismo en ambas, pero no parece exactamente la misma que obtuvimos a partir del espacio de estados. Esto es porque una misma FdT puede expresarse de distintas formas equivalentes..

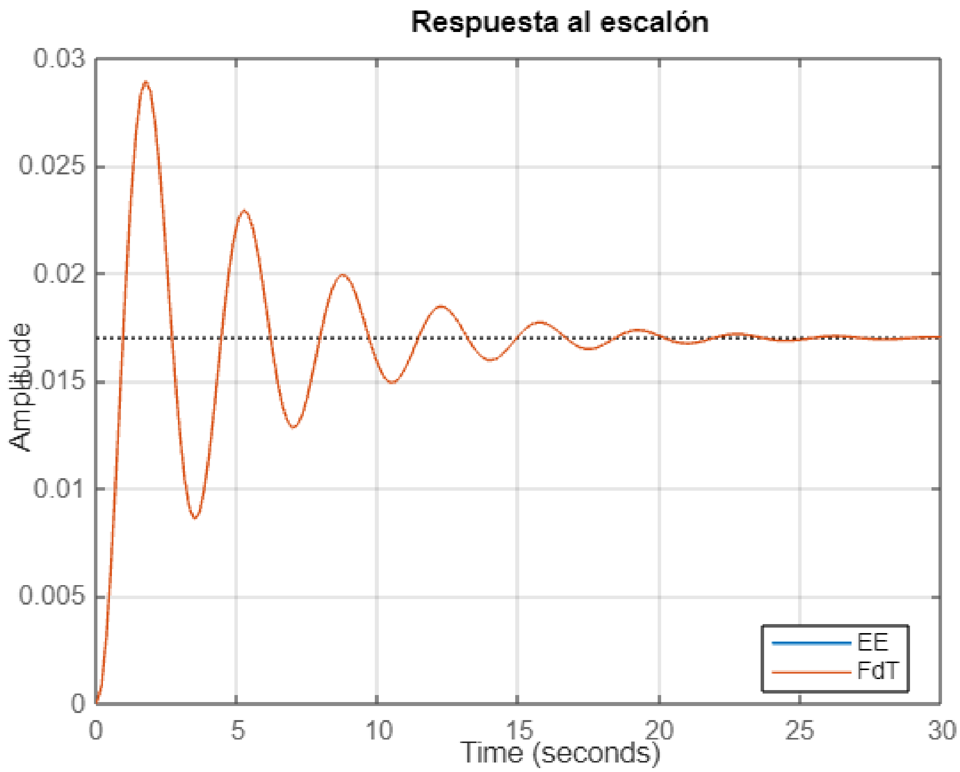
Dividiendo numerador y denominador de la segunda entre 18, se obtiene la primera.

```
num = num/den(1);
den = den/den(1);
tf(num,den)
```

```
ans =

       0.05556
  ------------------
  s^2 + 0.4 s + 3.27

Continuous-time transfer function.
Model Properties
```

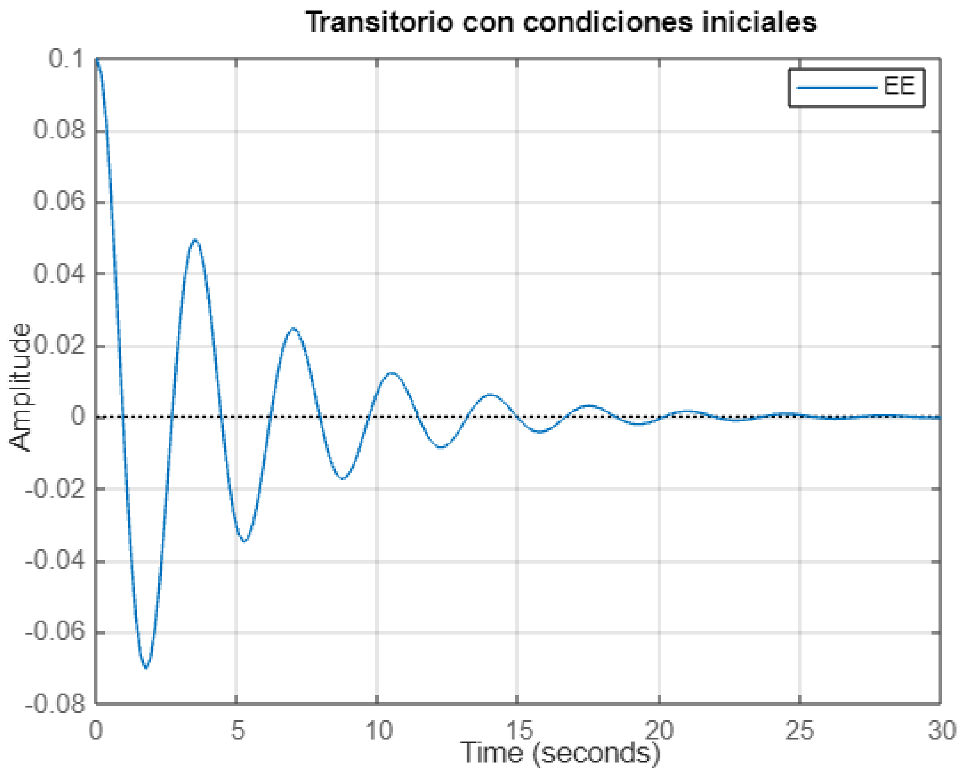
Finalmente, pintamos la respuesta a escalón. Esto se puede hacer tanto con el modelo en espacio de estados (EE) como con la función de transferencia (FdT).

```
step(sys_ss,sys_tf);
grid on;
title('Respuesta al escalón');
legend('EE','FdT','Location','Best');
```



En este caso particular, más que un escalón en un par de entrada, suele interesarnos lo que ocurre al soltar desde un determinado ángulo inicial (ya definido antes como theta_0). Eso se hace de manera muy similar con la función initialplot, a la que se le añade la información de los valores iniciales de los estados. En este caso, consideramos omega_0=0 (velocidad inicial nula, se suelta el péndulo desde un ángulo theta_0).

```
omega_0 = 0;
initialplot(sys_ss,[omega_0;theta_0]);
grid on;
title('Transitorio con condiciones iniciales');
legend('EE','Location','Best');
```



Esta función no se puede usar sobre la FdT, sino que necesita el sistema en EE.

3) Simulación paso a paso: lineal vs no lineal

En esta última parte, se hará la simulación calculando la propia evolución usando espacio de estados. Se comparará el modelo real, el no lineal, con su versión aproximada linealizada.

Primero, vamos a crear el vector de tiempo de la simulación, que iremos recorriendo paso a paso:

```
dt   = 1e-3;
Tend = 20;
t = 0:dt:Tend;
N    = length(t);
```

A continuación, se crean los vectores. Para un escalón de entrada unitario, se haría:

```
x_lin = zeros(2,N);  % [omega; theta]
x_nl  = zeros(2,N);
y_lin = zeros(1,N);
y_nl  = zeros(1,N);
u     = ones(1,N);   % escalón unitario
```

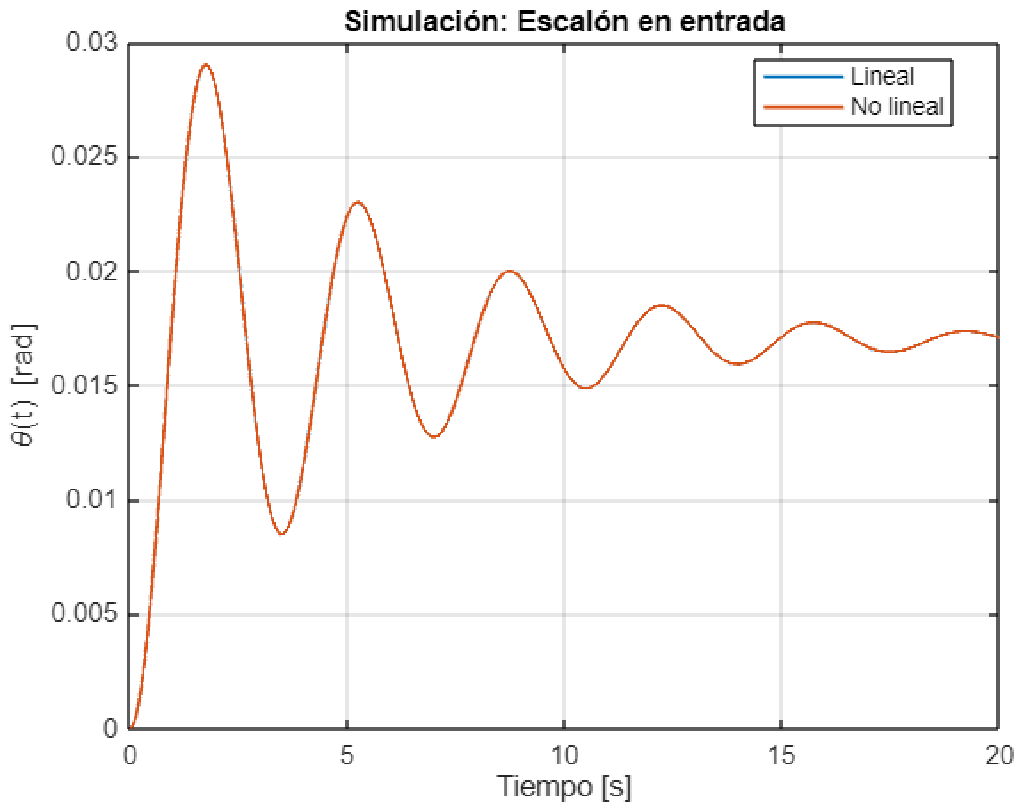
Y el bucle de simulación:

```
for k = 1:N-1
    % ------ Modelo lineal: xdot = A x + B u
```

```
    dx_lin = A*x_lin(:,k) + B*u(k);
    x_lin(:,k+1) = x_lin(:,k) + dt*dx_lin;

    % ------ Modelo no lineal
    omega  = x_nl(1,k);
    theta  = x_nl(2,k);

    domega = -b/m*omega - g/L*sin(theta) + 1/(m*L^2)*u(k);
    dtheta = omega;

    x_nl(1,k+1) = omega + dt*domega;
    x_nl(2,k+1) = theta + dt*dtheta;

    y_lin(k) = x_lin(2,k);
    y_nl(k)  = x_nl(2,k);
end
y_lin(end) = x_lin(2,end);
y_nl(end)  = x_nl(2,end);
```

Y comparamos ambos resultados poniéndolos en la misma gráfica:

```
figure;
plot(t,y_lin,'LineWidth',1.2); hold on; grid on;
plot(t,y_nl,'LineWidth',1.2);
xlabel('Tiempo [s]');
ylabel('\theta(t) [rad]');
title('Simulación: Escalón en entrada');
legend('Lineal','No lineal','Location','Best');
```



Mientras que si queremos considerar el caso de condiciones iniciales distintas de 0, basta con darle ese valor al primer elemento de dicho vector. Es decir:

```
x_lin = zeros(2,N);  x_lin(:,1) = [omega_0;theta_0];
x_nl  = zeros(2,N);  x_nl(:,1) = [omega_0;theta_0];
y_lin = zeros(1,N);
y_nl  = zeros(1,N);
u     = zeros(1,N);   % En este caso, no consideramos entrada
```

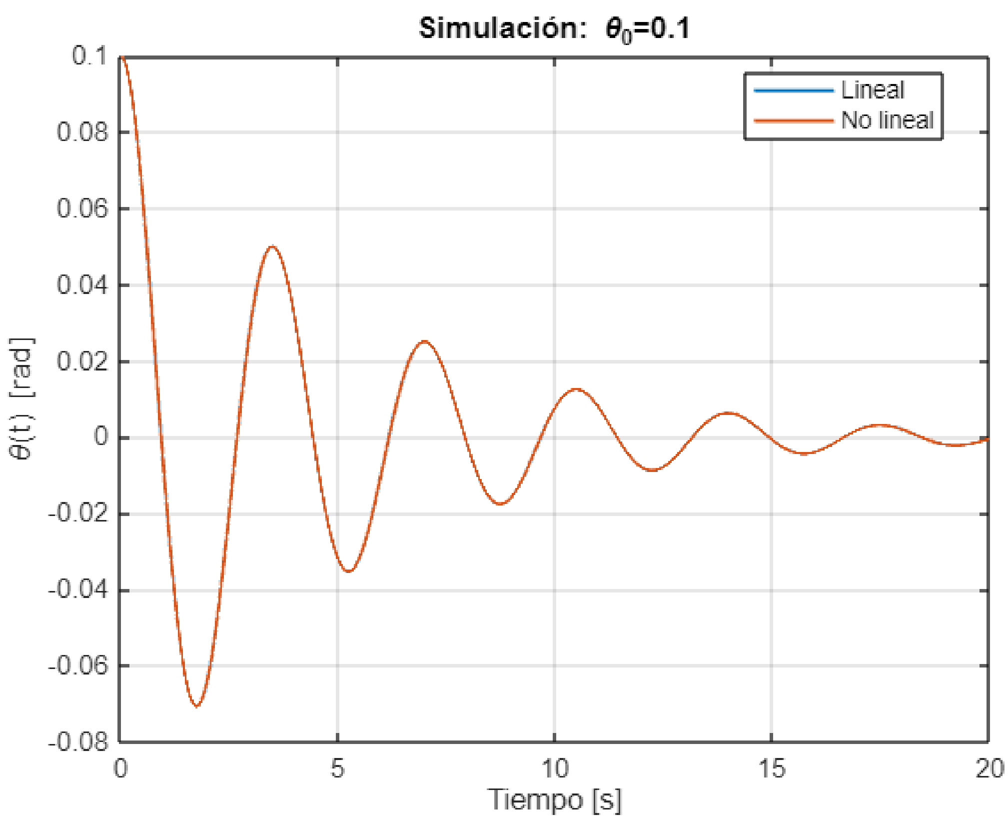
Y el bucle de simulación (idéntico al anterior).

```
for k = 1:N-1
    % ------ Modelo lineal: xdot = A x + B u
    dx_lin = A*x_lin(:,k) + B*u(k);
    x_lin(:,k+1) = x_lin(:,k) + dt*dx_lin;

    % ------ Modelo no lineal
    omega  = x_nl(1,k);
    theta  = x_nl(2,k);

    domega = -b/m*omega - g/L*sin(theta) + 1/(m*L^2)*u(k);
    dtheta = omega;

    x_nl(1,k+1) = omega + dt*domega;
    x_nl(2,k+1) = theta + dt*dtheta;
```

```
    y_lin(k) = x_lin(2,k);
    y_nl(k)  = x_nl(2,k);
end
y_lin(end) = x_lin(2,end);
y_nl(end)  = x_nl(2,end);
```

Y comparamos ambos resultados poniéndolos en la misma gráfica:

```
figure;
plot(t,y_lin,'LineWidth',1.2); hold on; grid on;
plot(t,y_nl,'LineWidth',1.2);
xlabel('Tiempo [s]');
ylabel('\theta(t) [rad]');
title('Simulación: \theta_0=0.1');
legend('Lineal','No lineal','Location','Best');
```



En ambos casos, vemos que la aproximación lineal es idéntica a la evolución real del péndulo. Esto es porque las desviaciones son suficientemente pequeñas. Si consideramos, por ejemplo, un ángulo inicial 10 veces el considerado anteriormente:

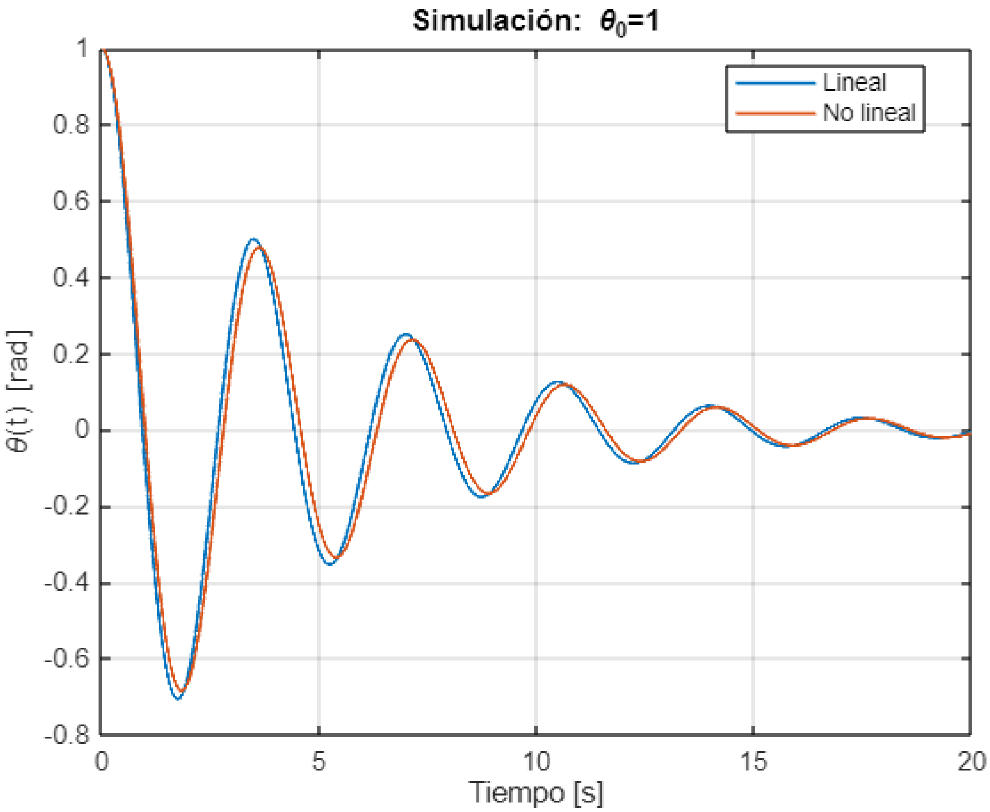
```
x_lin = zeros(2,N);  x_lin(:,1) = [omega_0;10*theta_0];
x_nl  = zeros(2,N);  x_nl(:,1) = [omega_0;10*theta_0];
y_lin = zeros(1,N);
y_nl  = zeros(1,N);
u     = zeros(1,N);   % En este caso, no consideramos entrada
```

Y el bucle de simulación (idéntico al anterior).

```
for k = 1:N-1
    % ------ Modelo lineal: xdot = A x + B u
    dx_lin = A*x_lin(:,k) + B*u(k);
    x_lin(:,k+1) = x_lin(:,k) + dt*dx_lin;

    % ------ Modelo no lineal
    omega  = x_nl(1,k);
    theta  = x_nl(2,k);

    domega = -b/m*omega - g/L*sin(theta) + 1/(m*L^2)*u(k);
    dtheta = omega;

    x_nl(1,k+1) = omega + dt*domega;
    x_nl(2,k+1) = theta + dt*dtheta;

    y_lin(k) = x_lin(2,k);
    y_nl(k)  = x_nl(2,k);
end
y_lin(end) = x_lin(2,end);
y_nl(end)  = x_nl(2,end);
```

Y comparamos ambos resultados poniéndolos en la misma gráfica:

```
figure;
plot(t,y_lin,'LineWidth',1.2); hold on; grid on;
plot(t,y_nl,'LineWidth',1.2);
xlabel('Tiempo [s]');
ylabel('\theta(t) [rad]');
title('Simulación: \theta_0=1');
legend('Lineal','No lineal','Location','Best');
```



En este caso, se aprecia una diferencia notable.