Práctica 3. Sistemas dinámicos continuos

Contenidos

Problema 2

La población en el continente asiático en el año 2000 era de aproximadamente 4830 millones de personas y, en aquel momento, crecía a un ritmo de un 1.73% por año. Suponiendo que el crecimiento de la población se rigiera por el modelo exponencial, calcular el valor estimado de la población en dicho continente en el año 2025.

a) resolver el problema analíticamente.

b) construir el modelo en Simulink y verificar el resultado.

Solución

a) Resolver el problema analíticamente.

Modelo de crecimiento exponencial de la población del continente asiátocp.

$$\frac{\mathrm{dP}}{\mathrm{dt}} = r \cdot P$$

$P(t) =$ población en un tiempo t

$r =$ tasa de crecimiento

con $P(t_o) = P_o$

Limpieza del entorno de trabajo

```
clear     % limpieza del espacio de trabajo
clc       % limpieza de línea de comando
close all % cerrar todas las figuras
```

Sustitución de datos del enunciado

```
t0 = 2000; % Año del tiempo inicial
P0 = 4830; % Población en el tiempo inicial (en millones)
r = 0.0172;% Tasa de crecimiento 1.72%
```

Resolución de la EDO

Cálculo de la solución de la ecuación diferencial ordinaria con MATLAB:

```
syms P(t)
edo = diff(P, t, 1) == r*P;
cond = [P(t0)==P0];

P(t) = dsolve(edo, cond)
```

```
P(t) =
```

$$4830\,e^{\frac{43\,t}{2500}}\,e^{-\frac{172}{5}}$$

Cálculo de la población en el 2025

Sustitución del tiempo en la solución:

```
Pa_2025 = double(P(2025))
```

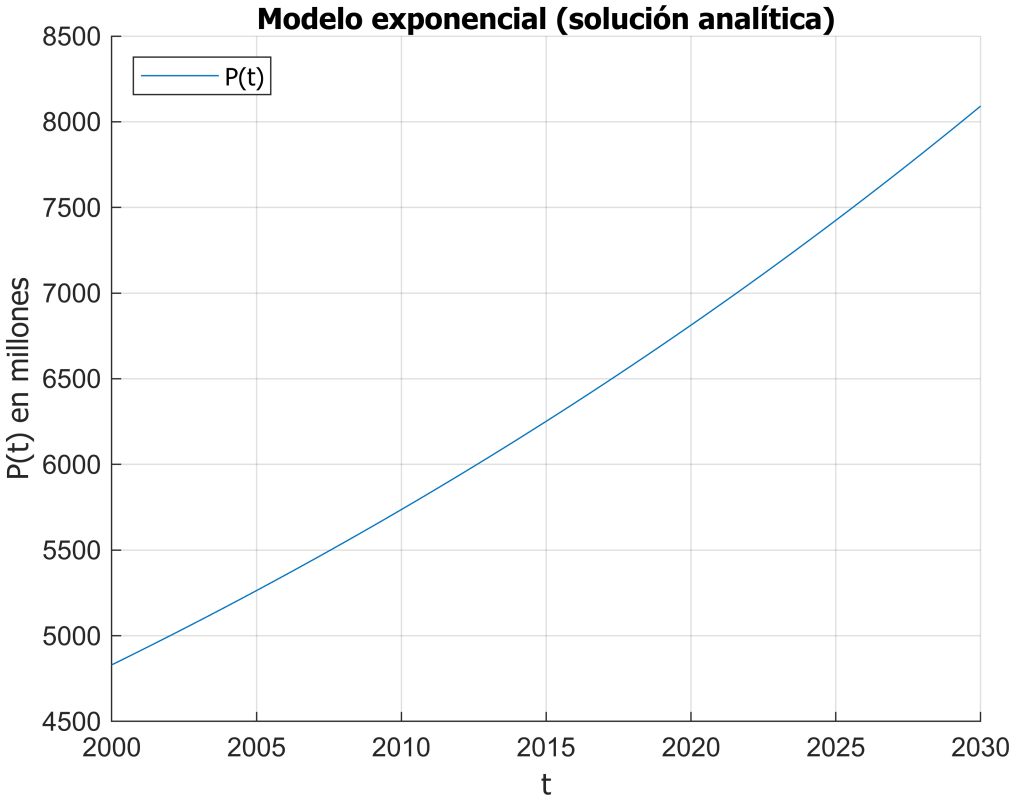
```
Pa_2025 = 7.4250e+03
```

Gráfica de la solución

```
t_values = 2000:0.5:2030;
figure;
hold on;
% Evaluar la solución en los valores de tiempo dados
P_values = subs(P, t, t_values);

% Graficar la solución
plot(t_values, P_values,'DisplayName','P(t)');
hold on

% Detalles de la gráfica
xlabel('t','FontName','latex');
ylabel('P(t) en millones','FontName','latex');
title('Modelo exponencial (solución analítica)','FontName','latex');
grid on;
legend('show','Location','northwest','FontName','latex')
hold off;
```



b) Construir el modelo en Simulink y verificar el resultado.

Configuración usuario

Se muestran los ajustes elejidos por el usuario aplicables a cualquier modelo de simulación discreta.

Configuración del solver

```
solver = struct();
solver.type = "Variable-Step";
solver.name = "ode45";
solver.start = 2000;
solver.stop = 2025;
solver.step = 1;
```

Nombre del modelo

```
model_name = "modelo5";
```

Condiciones iniciales y constantes del modelo:

```
r_values = 0.0173;
ic_values = [2000 4830];
```

sim_config, función que simula el modelo de nombre model_name para una configuración de solver almacenada en el struct solver. Recibe como argumentos el nombre del modelo y un struct con la configuración del solver.

```
sim_data = sim_config(model_name, solver); % End of file definition
```

De sim_data se extrae la información de la simulación gracias a bloques output.

Obtenemos un data set con las señales del campo yout

```
signals = sim_data.yout;
P1 = signals{1}.Values.Data;
P = signals{2}.Values.Data;
signal_time = signals{1}.Values.Time;
```

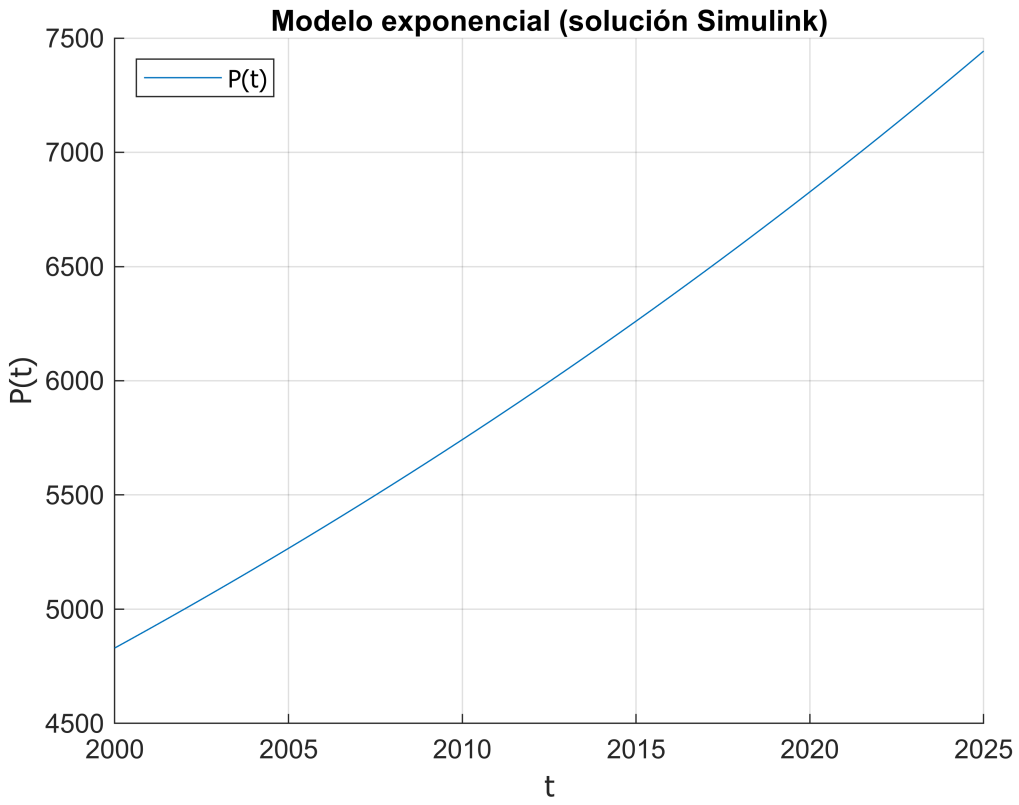
Cálculo de la población en el 2025

```
[m, ~] = size(P);
Pb_2025 = double(P(m)) % el último elemento de la simulación que finaliza en 2025
```

```
Pb_2025 = 7.4435e+03
```

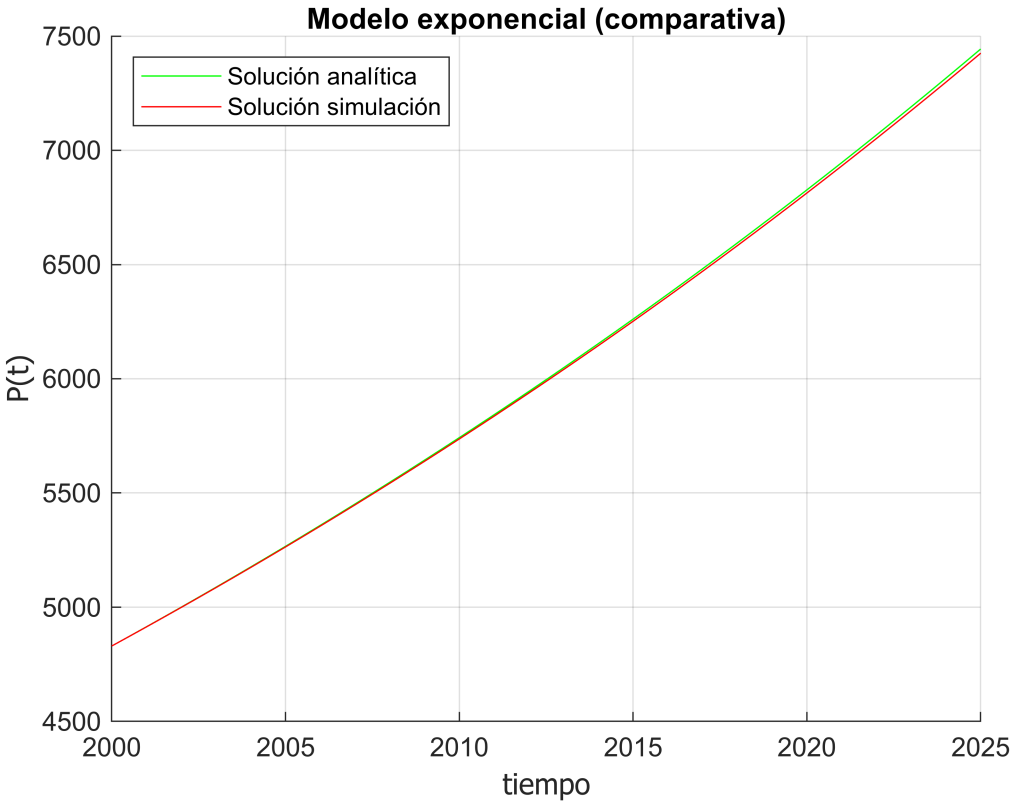
Gráfica Simulink

```
figure
hold on
% Gráfica de las señales
plot(signal_time, P,'DisplayName','P(t)')
grid on
% Detalles de la gráfica
title('Modelo exponencial (solución Simulink)');
xlabel('t','FontName','latex');
ylabel('P(t)','FontName','latex');
grid on;
legend('show','Location','northwest','FontName','latex')
hold off;
```



Juntamos las dos gráficas para ver si el resultado es el mismo (que a primera vista lo parece)

```
[m, n] = size(P1);
figure
xlim([2000, 2025])
hold on
plot(signal_time, P,'Color','g')
hold on
plot(t_values, P_values,'Color','r')
grid on
xlabel("tiempo","FontName","latex");
ylabel("P(t)","FontName","latex");
title('Modelo exponencial (comparativa)');
legend('Solución analítica','Solución simulación','Location','northwest');
hold off
```



Observamos que se superponen. Además la población para ambos en el 2025 es:

```
Pa_2025
```

```
Pa_2025 = 7.4250e+03
```

```
Pb_2025
```

```
Pb_2025 = 7.4435e+03
```

Problema 5

En el problema del estudio del sistema de amortiguación de un coche, suponga que debemos añadir el efecto de la fricción de la rueda con el asfalto. Modifique las ecuaciones, construya el modelo con Simulink y pruebe el resultado con la entrada en escalón unitario.

Solución

Procedimiento

Definimos las variables del modelo de simulación de amortiguación de un coche.

$x =$ posición del coche
$y =$ posición del resorte en el extremo del muelle de las ruedas del coche
$r =$ elevación del terreno

El objetivo es modelar un sistema en donde el coche amortigue los baches del terreno a través de la suspensión del muelle. Para que el conductor, apenas note la irregularidad del terreno.

Ecuaciones del modelo

Las fuerzas que aplican al modelo son la fuerza elástica, la fuerza del cinética y la fuerza del resorte.

(Recordamos que la derivada de la posición es la velocidad, y la derivada de la velocidad es la aceleración).

$$F_k = K_c(x - y)$$
$$F_c = C_c(x' - y')$$
$$F_r = K_r(y - r)$$

La fuerza elástica es proporcional a la posición de *x* e *y*. Con *Kc* constante elástica.

La fuerza del coche es proporcional a la velocidad relativa entre *x* e *y*.

La fuerza del resorte que es proporcional al terreno y al resorte (ya que está en contacto con el terreno).

Para modelar la fricción, vamos a agregar un término proporcional a la velocidad relativa de resorte, ya que a mayor velocidad, mayor es la fricción. Modificamos la ecuación de la fuerza:

$$F_c = C_c(x' - y') - \mu \cdot y'$$

De manera que la fricción es proporcional a la velocidad.

Ley de Newton:

$$F = m\,a$$

Como por la Ley de Newton, la suma de fuerzas es igual a cero, tenemos:

La suma de fuerzas del coche:

$$m_c\,x'' = -F_k - F_c$$

La suma de fuerzas del resorte:

$$m_r y'' = F_k + F_c - F_r$$

$$\text{con } m_c = \text{masa del coche } y\, m_r = \text{masa del resorte}$$

La aceleración del resorte y del coche, se ve afectada negativamente (en sentido opuesto al movimiento) por la fricción.

Sustituyendo las fuerzas por sus respectivas fórmulas, obtenemos:

$$x'' = -\frac{C_c}{m_c}x' + \frac{C_c}{m_c}y' - \frac{K_c}{m_c}x + \frac{K_c}{m_c}y - \frac{\mu y'}{m_c}$$
$$y'' = \frac{C_c}{m_r}x' - \frac{C_c}{m_r}y' + \frac{K_c}{m_r}x - \frac{K_c}{m_r}y - \frac{K_r}{m_r}y + \frac{K_r}{m_r}r - \frac{\mu y'}{m_r}$$

A continuación modelamos las ecuaciones en Simulink, y graficamos los resultados.

Limpieza del entorno de trabajo

```
clear     % limpieza del espacio de trabajo
clc       % limpieza de línea de comando
close all % cerrar todas las figuras
```

Configuración usuario

Se muestran los ajustes elejidos por el usuario aplicables a cualquier modelo de simulación discreta.

Configuración del solver

```
solver = struct();
solver.type = "Variable-Step";
solver.name = "ode45";
solver.start = 0;
solver.stop = 20;
solver.step = 1;
```

Nombre del modelo

```
model_name = "amortiguacion";
```

Constantes de ejemplo para el modelo de un sistema de amortiguación de un coche:

```
mu = 0.03;
cc = 30;
mc = 60;
mr = 12;
kc = 20;
kr = 700;
```

sim_config, función que simula el modelo de nombre model_name para una configuración de solver almacenada en el struct solver. Recibe como argumentos el nombre del modelo y un struct con la configuración del solver.

```
sim_data = sim_config(model_name, solver); % End of file definition
```

De sim_data se extrae la información de la simulación gracias a bloques output.

Obtenemos un data set con las señales del campo yout

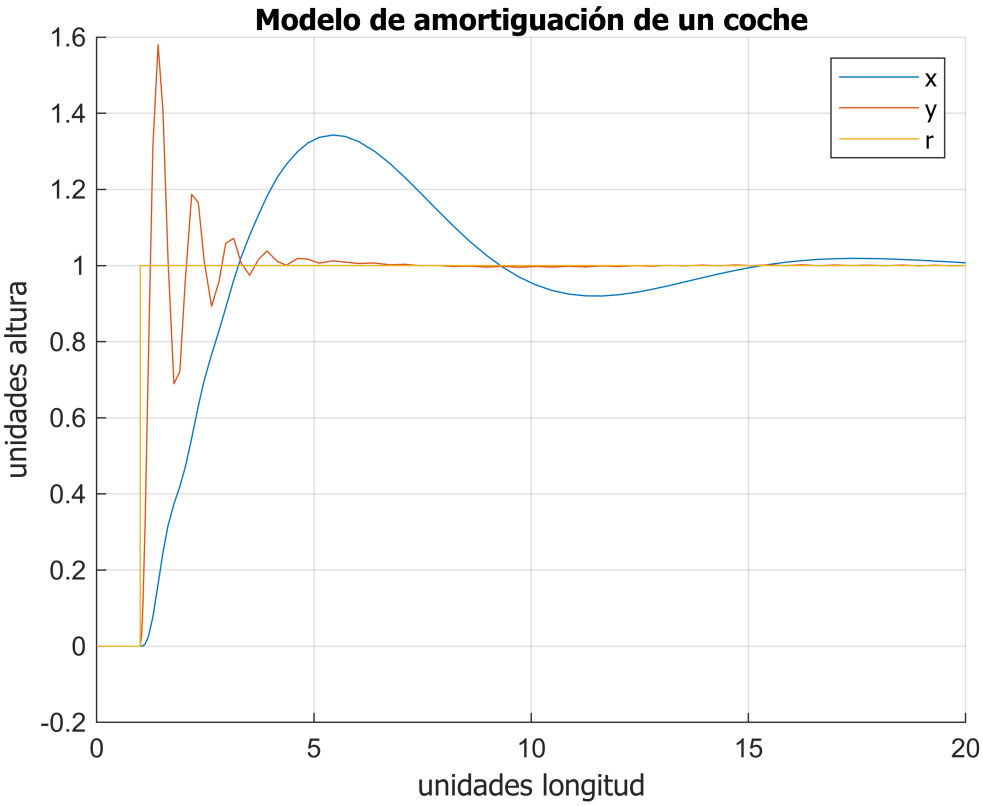
```
signals = sim_data.yout;
x = signals{1}.Values.Data;
y = signals{2}.Values.Data;
r_terreno = signals{3}.Values.Data;
signal_time = signals{1}.Values.Time;
```

Gráfica 1

```
[m, n] = size(x);

for i =1:n
    figure
```

```
    hold on
    plot(signal_time, x(:,i))
    hold on
    plot(signal_time, y(:,i))
    grid on
    plot(signal_time, r_terreno(:,i))
    grid on
    xlabel("unidades longitud","FontName","latex");
    ylabel("unidades altura","FontName","latex");
    title(['Modelo de amortiguación de un coche'],'FontName','latex');
    legend('x','y','r','FontName','latex');
    hold off
end
```



Observamos que el movimiento del coche está amortiguado, la curva *x* traza un movimiento más uniforme que la de *y*, ya que el resorte avanza por el terreno y el muelle amortigua al coche.

Funciones disponibles para todas las secciones

MATLAB permite utilizar la última sección de un livescript para definir funciones globales.

Decorador de la función sim de MATLAB

Esta función ejecuta la función sim, tras establecer los parámetros del solver. Recibe como argumentos el nombre del modelo y un struct con la configuración del solver.

```
function sim_data = sim_config(model_name, solver)
```

Carga del modelo con load_system

```
load_system(model_name)
% open_system(model_name) will also open it
```

Configuración con set_param

```
% Set solver parameters
set_param(model_name, ...
    SolverType  = solver.type, ...
    SolverName  = solver.name, ...
    StartTime   = string(solver.start),...
    StopTime    = string(solver.stop), ...
    FixedStep   = string(solver.step) ...
    )
% More model parameters can be retrieved with:
% model_data = get_param(model_name, "ObjectParameters")
% Access blocks params with "model_name/block_name"
```

Simulación del modelo con sim

```
sim_data = sim(model_name);
% SimulationMode="normal" by default
end
```

Se utiliza el bloque Out1 para cada señal.