Modelado y Simulación de Sistemas Dinámicos
Práctica 1 - Sistemas de Tiempo Discreto

Ernesto Kofman

Problema 1. Crecimiento de una Población En ausencia de limitaciones ambientales, muchas especies se reproducen de manera tal que el nacimiento de individuos es proporcional al tamaño de la población. Llamaremos $P(t)$ a dicho tamaño, b a la *tasa de natalidad* (número de nacimientos por unidad de tiempo) y d a la *tasa de mortalidad* (número de fallecimientos por unidad de tiempo). Tanto b como d estarán expresados como fracción del tamaño actual de la población (es decir, $b = 0.1$ implica que el número de nacimientos por unidad de tiempo es un $10\,\%$ del tamaño de la población).

Se pide lo siguiente:

1. Obtener un modelo de *Ecuaciones en Diferencias* que exprese la dinámica de $P(t)$
2. Escribir una función en Octave que permita simular el modelo a partir de una condición inicial arbitraria. Observar el resultado que se obtiene con $b = 0.1$ y $d = 0.02$ para $P(0) = 1$.
3. Modificar el modelo para contemplar limitaciones ambientales, suponiendo ahora que la tasa de mortalidad es proporcional al número de individuos, es decir, $d(t) = a \cdot P(t)$ para cierta constante $a > 0$.
4. Simular este nuevo modelo con $a = 0.01$, $b = 0.1$, $P(0) = 1$.
5. Para este último caso, calcular analíticamente los puntos de equilibrio.

Problema 2. Modelo Epidemiológico SIR discreto El siguiente modelo [1] es una versión de tiempo discreto de un modelo clásico de epidemiología conocido como SIR (Susceptible, Infected, Recovered).

$$\begin{aligned} S(t+1) &= S(t) - \frac{\alpha \cdot S(t) \cdot I(t)}{N} \\ I(t+1) &= I(t) + \frac{\alpha \cdot S(t) \cdot I(t)}{N} - \gamma \cdot I(t) \\ R(t+1) &= R(t) + \gamma \cdot I(t) \end{aligned} \tag{2a}$$

Las variables $S(t)$, $I(t)$ y $R(t)$ representan el número de individuos susceptibles, infectados y recuperados, respectivamente. Los parámetros, en tanto, son N (población total), α (tasa de contacto) y γ (tasa de recuperación). α expresa el número de individuos en promedio por día con los que cada individuo infectado está en suficiente contacto como para infectarlo.

Se pide entonces

1. Programar en Octave una función que permita simular este modelo desde condiciones iniciales arbitrarias. Simular el modelo con parámetros $N = 10^6$, $\alpha = 1$, $\gamma = 0.5$ para la condición inicial $S(0) = N$, $I(0) = 10$.
2. Un problema con la función anterior es que el modelo y la rutina de simulación están mezclados en un único algoritmo, lo que por muchas razones es una mala práctica. La manera más correcta de hacerlo es tener una función que implemente el algoritmo de simulación y otra que implemente cada modelo. Para la primera, un posible código es el siguiente:

```
function [t,x] = dtsim(f,x0,ti,tf)
  t=[ti:tf];
  x(:,1)=x0;
  for k=1:length(t)-1
    x(:,k+1)=f(x(:,k),t(k));
```

```
  end
end
```

El modelo, en tanto, puede representarse como

```
function x=discreteSIR(pre_x,t)
  al=1;
  gam=0.5;
  N=1e6;
  pre_S=pre_x(1);
  pre_I=pre_x(2);
  pre_R=pre_x(3);
  S = pre_S - al * pre_S * pre_I / N;
  I = pre_I + al * pre_S * pre_I / N - gam * pre_I;
  R = pre_R + gam * pre_I;
  x=[S;I;R];
endfunction
```

Repetir el punto anterior utilizando estas funciones.

Problema 3. Modelo SEIR Discreto Una modificación del modelo anterior surge al considerar que en ciertas enfermedades las personas no comienzan a contagiar apenas se exponen al virus, sino que hay un tiempo de incubación. Esto agrega una variable más al modelo, $E(t)$, que representa el número de individuos expuestos que aún no contagian:

$$\begin{aligned} S(t+1) &= S(t) - \frac{\alpha \cdot S(t) \cdot I(t)}{N} \\ E(t+1) &= E(t) + \frac{\alpha \cdot S(t) \cdot I(t)}{N} - \mu \cdot E(t) \\ I(t+1) &= I(t) + \mu \cdot E(t) - \gamma \cdot I(t) \\ R(t+1) &= R(t) + \gamma \cdot I(t) \end{aligned} \tag{3a}$$

Se propone entonces modificar el ejemplo anterior para considerar este nuevo modelo y simularlo con el parámetro $\mu = 0.5$.

Problema 4. Modelo SEIR Discreto con Retardos Explícitos Una mejora al modelo anterior surge al considerar explícitamente el tiempo que transcurre desde que cada individuo se expone hasta que se torna infeccioso y hasta que se recupera [2], lo que lleva al siguiente sistema de ecuaciones:

$$\begin{aligned} N^E(t) =& \frac{R_0}{T_R - T_I} \cdot \frac{I(t) \cdot S(t)}{N} \\ S(t+1) =& S(t) - N^E(t) \\ E(t+1) =& E(t) + N^E(t) - N^E(t-T_I) \\ I(t+1) =& I(t) + N^E(t-T_I) - N^E(t-T_R) \\ R(t+1) =& R(t) + N^E(t-T_R) \end{aligned} \tag{4a}$$

donde T_I y T_R son los tiempos (enteros) hasta la infección y recuperación, respectivamente, y R_0 es el número reproductivo básico (cuantos individuos son expuestos al virus por cada infectado). Notar que este modelo no está escrito en la forma típica: $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), t)$, ya que aparecen dependencias con valores anteriores de la variable $N^E(t)$.

Se pide entonces:

1. Re-escribir las Ecs.(4a) como $\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), t)$. Para esto, pueden tomarse como variables de estado $N_i^E(t) = N^E(t-i)$ para $i = 1, 2, \cdots, T_R$.

2. Implementar en Octave el modelo correspondiente y simularlo con parámetros $T_I = 3$, $T_R = 12$ y $R_0 = 1.5$.

Problema 5. Implementación en Modelica Una alternativa para implementar los modelos anteriores es usar el lenguaje de modelado Modelica. Para el caso del modelo del Problema 2, una posible representación es la siguiente

```
model discreteSIR
  parameter Real N = 1e6;
  discrete Real S(start = N), I(start = 10), R;
  parameter Real alpha = 1, gamma = 0.5;
algorithm
  when sample(0, 1) then
    S := pre(S) - alpha * pre(S) * pre(I) / N;
    I := pre(I) + alpha * pre(S) * pre(I) / N - gamma * pre(I);
    R := pre(R) + gamma * pre(I);
  end when;
end discreteSIR;
```

1. Implementar este modelo en la herramienta OpenModelica y simularlo.
2. Implementar los modelos de los Problemas 3-4

Problema 6. Modelo de Múltiples Poblaciones El siguiente modelo considera que hay individuos infectados que pueden moverse entre poblaciones:

```
 model discreteSIRimp
  parameter Real N = 1e6;
  discrete Real S(start = N), I(start = 10), R,imp,exp;
  parameter Real alpha = 1, gamma = 0.5, m=0.01;
algorithm
  when sample(0, 1) then
    S := pre(S) - alpha * pre(S) * pre(I) / N;
    I := pre(I) + alpha * pre(S) * pre(I) / N - gamma * pre(I) + imp - exp;
    R := pre(R) + gamma * pre(I);
    exp:= m* pre(I);
  end when;
end discreteSIRimp;
```

Aquí, la variable `exp` es el número de individuos infectados que abandonan la población, mientras que `imp` son los que llegan. El parámetro `m` representa la movilidad.

Suponiendo ahora que hay dos poblaciones que intercambian individuos según estas reglas, se puede construir el siguiente modelo:

```
 model multiSIR
  discreteSIRimp M1, M2(I.start = 0);
algorithm
  when sample(0, 1) then
    M1.imp := M2.exp;
    M2.imp := M1.exp;
  end when;
end multiSIR;
```

1. Implementar en OpenModelica el modelo de dos poblaciones, donde una inicialmente no tiene infectados.
2. Implementar ahora un modelo de 3 poblaciones de tal manera que la primera intercambie individuos con la segunda, y la segunda lo haga además con la tercera. Suponer que los individuos que salen de la segunda población van la mitad a la 1 y la mitad a la 3.

3. Analizar los incovenientes que encontraría para implementar estos modelos como funciones de Octave.

Problema 7. Diagramas de Bloques de un Modelo SIR Implementar el modelo del Problema 2 como Diagrama de Bloques de Modelica. La Figura 1 muestra una posible solución.

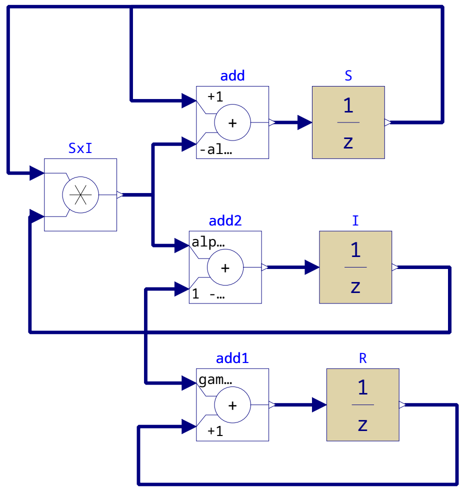

Figura 1: Diagrama de Bloques en Modelica

Definir como parámetros del modelo α, γ, N y la condición inicial $I(t=0)$

Problema 8. Diagramas de Bloques de un Modelo de dos Poblaciones Modificar el Diagrama de Bloques del Problema 7 para considerar ingreso y egreso de individuos infectados, de la misma forma que en el Problema 6. Se puede usar como guía la Figura 2.

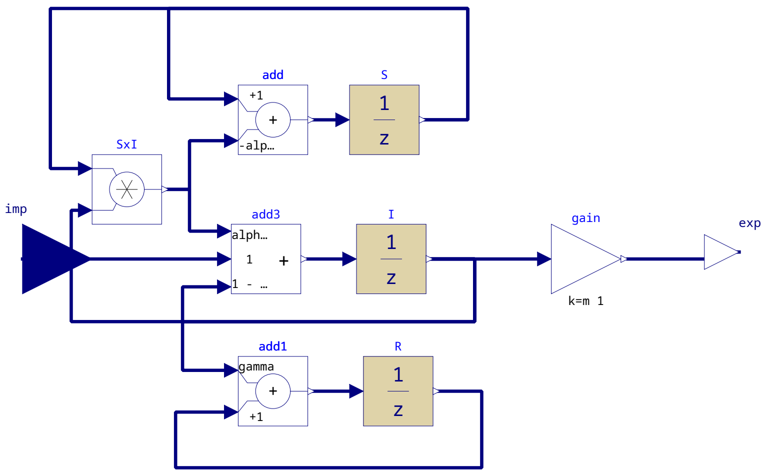

Figura 2: Diagrama de Bloques con Entradas y Salidas en Modelica

Conectar dos modelos como el de dicha figura para reproducir lo hecho en el Problema 6.

Problema 9. Diagramas de Bloque en PowerDEVS Repetir los Problemas 7-8 utilizando PowerDEVS. Proponer además un modelo con más de dos poblaciones.

Problema 10. Modelo en Tiempo Discreto y Control de un Robot Móvil La discretización del modelo de un robot móvil tipo auto que se desplaza en un plano [3] produce el siguiente sistema de ecuaciones en diferencias:

$$
\begin{aligned}
x(t+h) &= x(t) + h \cdot (\cos(\theta) \cdot \cos(\psi) \cdot v(t)) \\
y(t+h) &= y(t) + h \cdot (\sin(\theta) \cdot \cos(\psi) \cdot v(t)) \\
\theta(t+h) &= \theta(t) + h \cdot \frac{(\sin(\psi) \cdot v(t))}{L} \\
v(t+h) &= v(t) + h \cdot a(t) \\
\psi(t+h) &= \psi(t) + h \cdot \omega_\psi(t)
\end{aligned}
\tag{10a}
$$

Aquí, $x(t)$ e $y(t)$ representan la posición en el plano del centro del robot, y $\theta(t)$ representa el ángulo que forma el vehículo con el eje horizontal. Por otro lado, $v(t)$ es la velocidad del robot y $\psi(t)$ es el ángulo que forman las ruedas delanteras (dirección) con el robot. El modelo tiene un parámetro L que representa la longitud entre el eje delantero y el trasero y un parámetro h correspondiente al período de la discretización. Además hay dos entradas: la aceleración del vehículo ($a(t)$) y la velocidad de rotación de la dirección $\omega_\psi(t)$.

Se pide entonces

1. Implementar el modelo en PowerDEVS y simularlo suponiendo condiciones iniciales nulas, parámetros $L = 1$, $h = 0.1$, y entradas

$$
a(t) = \begin{cases} 1 & \text{si } t < 2 \\ 0 & \text{si } t \geq 2 \end{cases} \qquad \omega_\psi(t) \begin{cases} 0.1 & \text{si } t < 1 \\ -0.1 & \text{si } 1 < t < 2 \\ 0 & \text{si } t \geq 2 \end{cases}
$$

   Interpretar que trayectoria realiza el robot.

2. Suponer ahora que las entradas se calculan según la siguiente *ley de control*

$$
a(t) = K_v \cdot (v_{\text{ref}} - v(t)); \quad \omega_\psi(t) = K_\psi \cdot (\psi_{\text{ref}}(t) - \psi(t));
$$

   donde K_v y K_ψ son parámetros de control de velocidad y dirección, v_{ref} es la velocidad deseada y la dirección de referencia se calcula según:

$$
\psi_{\text{ref}}(t) = K_y \cdot (\arctan(y_{\text{ref}} - y(t))/L_0 - \theta(t))
$$

   siendo K_y y L_0 parámetros de control de posición, cuya referencia es y_{ref}.

   Tomando parámetros $K_v = 1$, $K_\psi = 2$, $K_y = 0.5$, $L_0 = 4$ y las referencias $v_{\text{ref}} = 2$ e $y_{\text{ref}} = 1$, implementar la ley de control y simular el sistema, interpretando el resultado.

Referencias

[1] Linda JS Allen. Some discrete-time SI, SIR, and SIS epidemic models. *Mathematical biosciences*, 124(1):83–105, 1994.

[2] Mariana Bergonzi, Ezequiel Pecker-Marcosig, Ernesto Kofman, and Rodrigo Daniel Castro. Discrete-time modeling of covid-19 propagation in argentina with explicit delays. *Computing in Science & Engineering*, 2020.

[3] Alessandro De Luca, Giuseppe Oriolo, and Claude Samson. Feedback control of a nonholonomic car-like robot. In *Robot motion planning and control*, pages 171–253. Springer, 1998.