



MODELADO Y SIMULACIÓN DE SISTEMAS DINÁMICOS

TRABAJO PRÁCTICO 1 SISTEMAS DE TIEMPO DISCRETO

Tomás Fernández de Luco, F-3443/6

Comentarios generales

Los archivos adjuntos en la entrega contienen las definiciones de funciones y modelos pedidos en cada ejercicio. En cada uno se agregan comentarios que explican los valores esperados en cada argumento donde se consideró útil. Además, los archivos tienen definidos los parámetros necesarios para realizar las simulaciones requeridas en cada apartado.

En el caso de los problemas que involucren Octave, se agregan también archivos de script que realizan las gráficas que se analizan más adelante en el informe. En las carpetas de los problemas resueltos en PowerDEVS, se agrega un archivo de notas donde se especifican los parámetros que deben inicializarse en Scilab para poder simular cada modelo.

Problema 1

En el primer modelo que se define, en cada instante de tiempo la población aumenta de manera proporcional a su tasa de natalidad y disminuye de la misma forma con la tasa de mortalidad. Como resultado, la ecuación de la población termina siendo de la forma $P(t) = (1 + b - a)^t P(0)$ para una población inicial dada $P(0)$. Esta fórmula nos da como resultado un comportamiento exponencial.

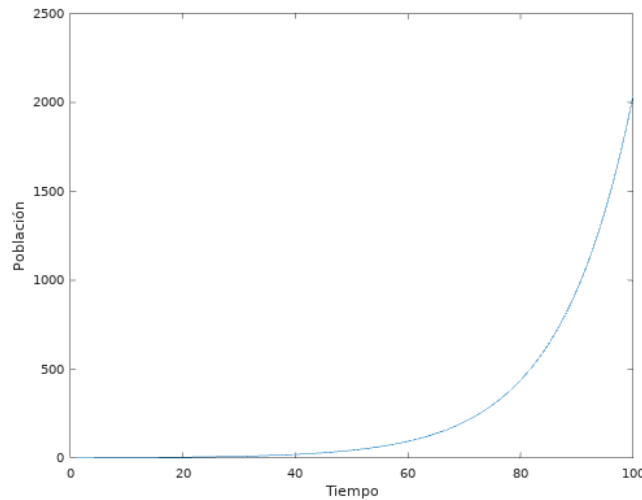


Figura 1: Simulación sin restricciones ambientales.

Por otro lado, al considerar limitaciones ambientales, la población se modifica rápidamente al principio, pero su pendiente se va volviendo progresivamente más horizontal hasta tener un comportamiento tangencial cerca de un valor de estabilización.

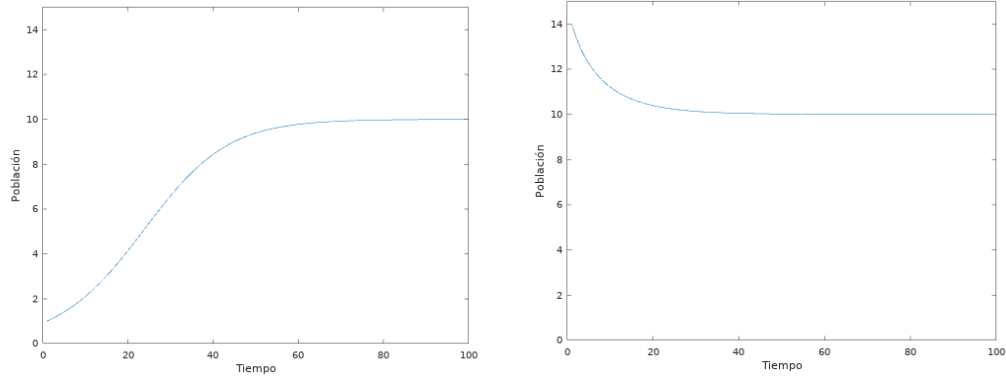


Figura 2: Simulaciones con restricciones ambientales y diferentes poblaciones iniciales.

Para determinar los puntos de equilibrio del sistema, partimos de la definición para determinar el estado tal que:

$$\begin{aligned}
 P(t) &= P(t+1) \\
 P(t) &= (1 + b - aP(t))P(t) \\
 0 &= (1 + b - aP(t))P(t) - P(t) \\
 0 &= (b - aP(t))P(t)
 \end{aligned}$$

Luego, para que el producto sea nulo debemos tener que alguno de los dos términos lo sea. Entonces, tenemos dos alternativas:

$$\begin{aligned}
 P(t) &= 0 \\
 b - aP(t) &= 0 \implies b = aP(t) \implies P(t) = b/a
 \end{aligned}$$

Por lo que la población se estabiliza cuando sea 0 o b/a . Podemos ver gráficamente en los ejemplos anteriores que el primer punto de equilibrio es inestable, mientras que el segundo es asintóticamente estable.

Problema 2

Luego de definir los modelos de las dos formas expresadas en el ejercicio, se realizaron simulaciones con ambos en iguales condiciones iniciales. El hecho de que los resultados sean iguales da confianza de que la primera implementación es correcta.

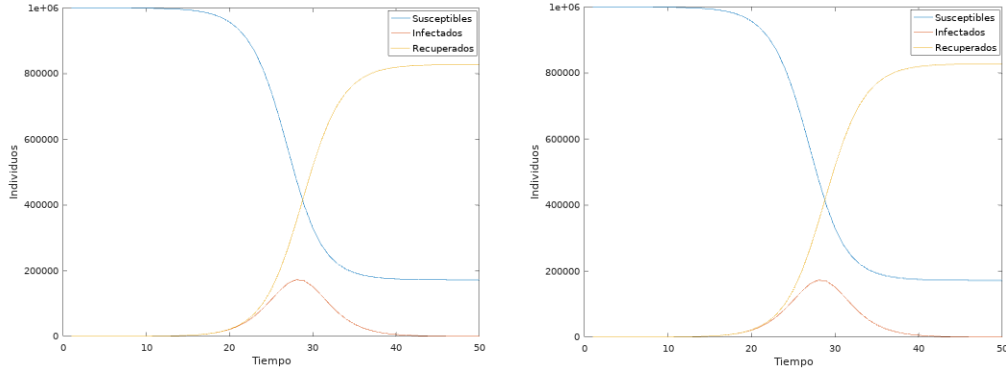


Figura 3: Simulaciones del modelo SIR con ambas implementaciones del algoritmo de simulación.

Puede verse que incluso con un número bajo de infectados iniciales, se llega a infectar a alrededor del 85% de la población. Simulaciones posteriores con un número de infectados inicial de hasta 50.000 muestran valores finales de susceptibles y recuperados similares, con la única diferencia siendo cuán rápido se llega al pico de infectados. Un detalle interesante es que la curva de contagios comienza a disminuir recién cuando el número individuos recuperados alcanza al de susceptibles, lo cual parecería tener sentido dado que la mitad de la población ya no sería infectable, lo cual vuelve cada vez más difícil contagiar nuevos individuos.

Problema 3

El resultado de la simulación con este modelo es muy parecido al anterior en la forma general de las curvas y los valores finales de recuperados y susceptibles. Sin embargo, el tiempo de simulación necesario para analizar la forma general de las curvas es el doble que antes, pues la presencia de los individuos expuestos demora la propagación de la infección en la población. Una consecuencia de esto es la curva de infectados considerablemente aplanada, llegando a un pico en un valor cercano a la mitad que la del modelo anterior.

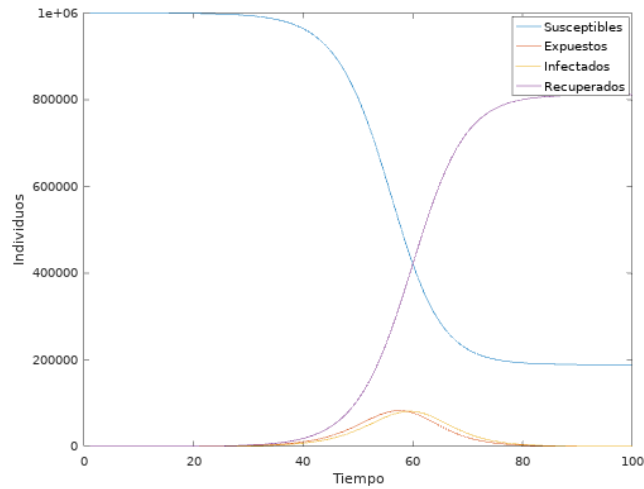


Figura 4: Simulación del modelo SEIR.

Problema 4

Para reescribir las ecuaciones del modelo, se puede partir de la definición sugerida $N_i^E(t) = N^E(t - i)$, pudiendo así despejar las siguientes igualdades:

$$\begin{aligned} N_1^E(t+1) &= N^E(t+1-1) = N^E(t) \\ N_2^E(t+1) &= N^E(t+1-2) = N^E(t-1) = N_1^E(t) \end{aligned}$$

Análogamente, se pueden despejar las ecuaciones de las demás N_i^E para ver que cada una corresponde a la del i anterior, desfasada una unidad de tiempo. Como resultado final, llegamos al nuevo conjunto de ecuaciones del modelo:

$$\begin{aligned} N^E(t) &= \frac{R_0}{T_R - T_I} \cdot \frac{I(t) \cdot S(t)}{N} \\ S(t+1) &= S(t) - N^E(t) \\ E(t+1) &= E(t) + N^E(t) - N_{T_I}^E(t) \\ I(t+1) &= I(t) + N_{T_I}^E(t) - N_{T_R}^E(t) \\ R(t+1) &= R(t) + N_{T_R}^E(t) \\ N_1^E(t+1) &= N^E(t) \\ N_i^E(t+1) &= N_{i-1}^E(t) \quad i = 2, \dots, T_R \end{aligned}$$

La simulación de este nuevo modelo muestra un incremento incluso más lento de la curva de infectados debido a que la demora que implican los períodos de incubación retrasa el ritmo de aparición de infecciones en la población.

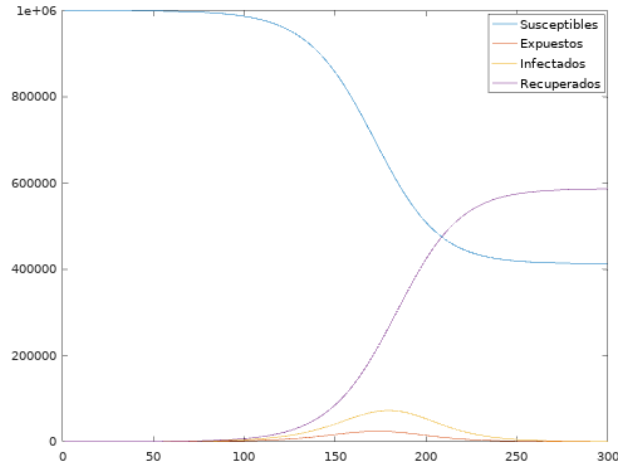


Figura 5: Simulación del modelo SEIR con Retardos Explícitos.

Problema 5

La traducción de los modelos a OpenModelica fue lo suficientemente directa, tomando como base el ejemplo dado en el enunciado. Las siguientes gráficas muestran que las simulaciones realizadas bajo las mismas condiciones iniciales que en los problemas anteriores dan los mismos resultados:

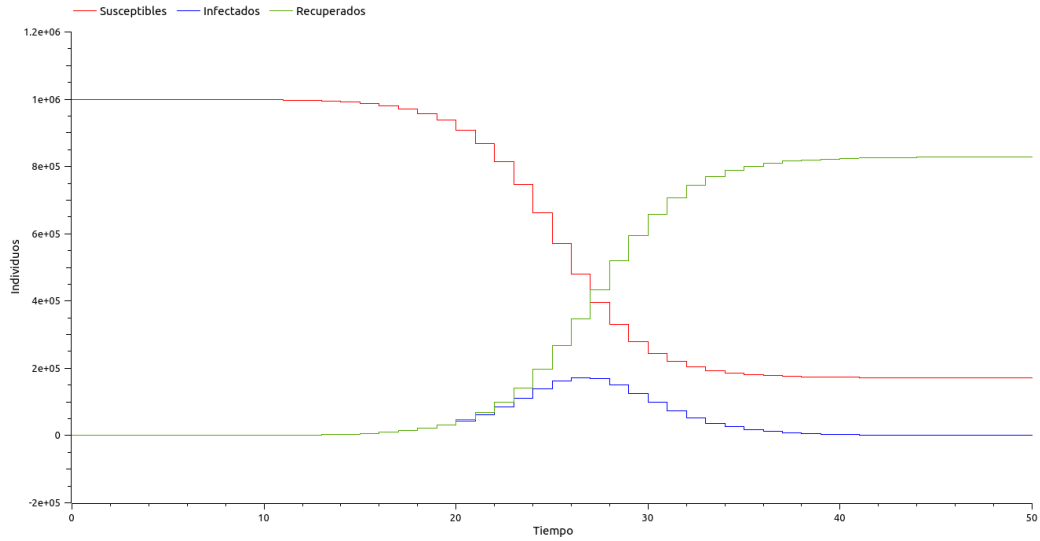


Figura 6: Simulación en OpenModelica del modelo SIR.

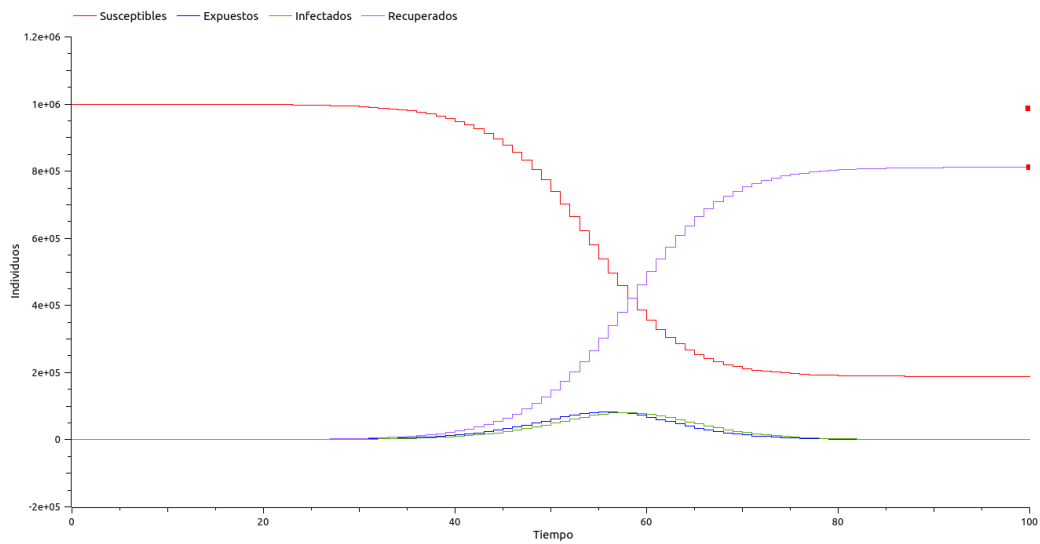


Figura 7: Simulación en OpenModelica del modelo SEIR.

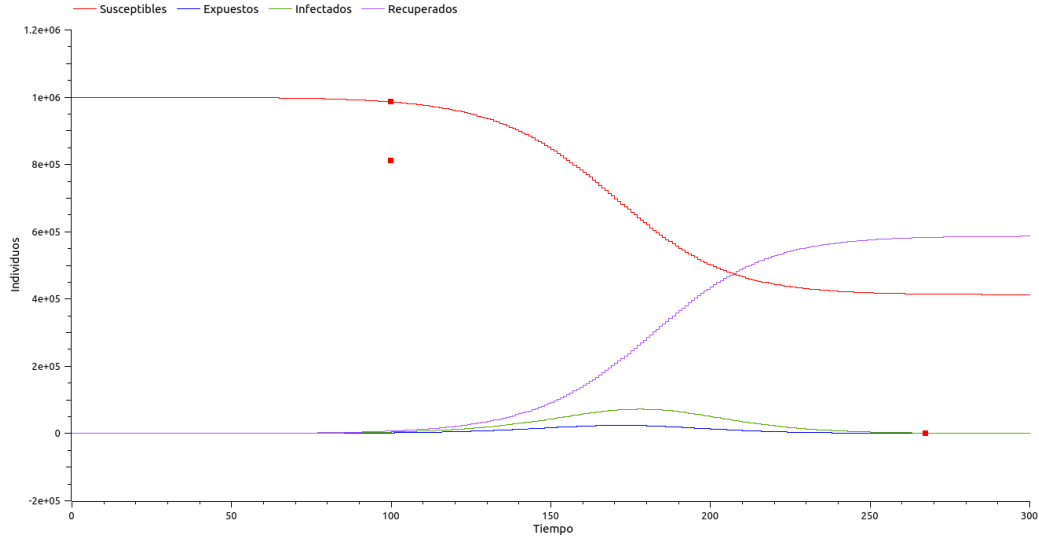


Figura 8: Simulación en OpenModelica del modelo SEIR con Retardos Explícitos.

Problema 6

En la simulación con dos poblaciones, resulta llamativo que la que tuvo el pico de infectados más alto fue la que comenzó con ninguno. Puede verse en la figura que para el momento en el que la curva azul comienza a ascender, la curva roja es considerablemente mayor. Esto significa que los infectados en la segunda población estarán incrementados considerablemente en cada instante de tiempo por los casos importados de la primera, lo cual hará que tengan un crecimiento de casos mayor.

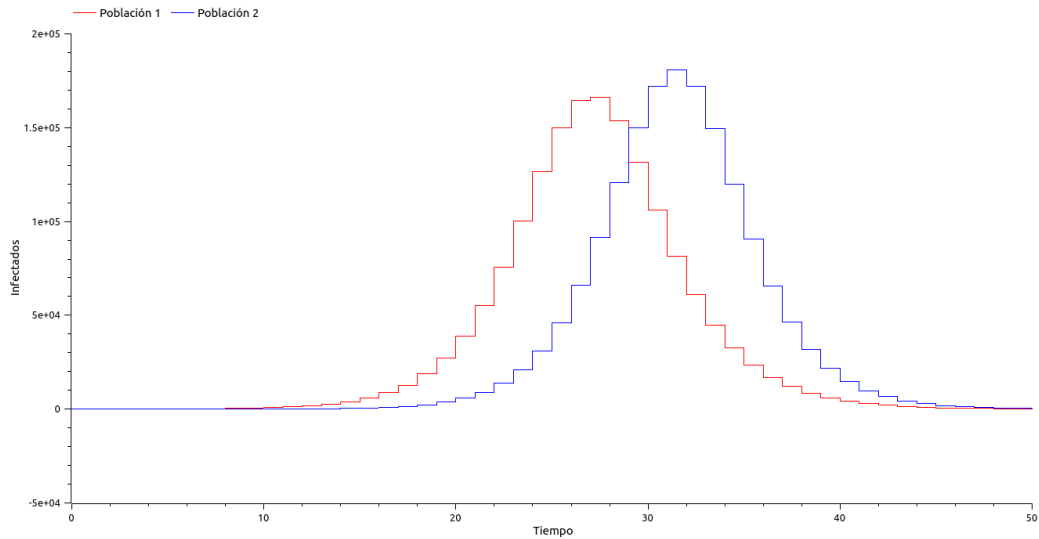


Figura 9: Simulación del modelo SIR de dos poblaciones.

Sin embargo, la situación del ejercicio anterior no ocurre entre las poblaciones 2 y 3 de la siguiente simulación. Si bien los casos importados son suficientes para que la curva de la tercera población sea mayor que de la primera, el hecho de que sólo sean la mitad de las exportaciones de la segunda los reduce lo suficiente para que el pico no sea el mayor de todas las curvas.

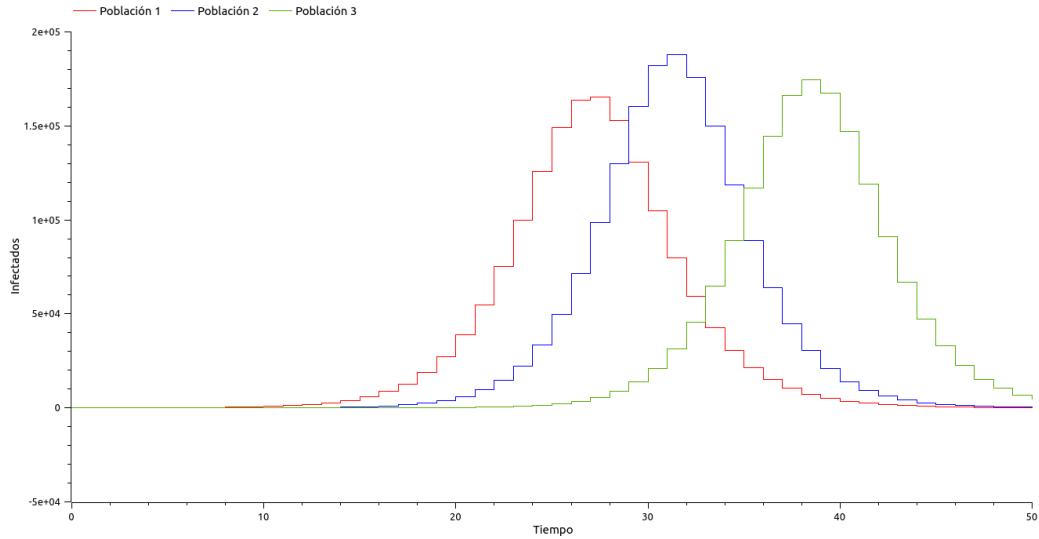


Figura 10: Simulación del modelo SIR de tres poblaciones.

La principal ventaja que encuentro en utilizar OpenModelica en comparación a las funciones de Octave es la sencilla composicionalidad que se obtiene entre los modelos. Cada unidad puede definirse en su módulo propio, y luego es muy sencillo instanciarlos con diferentes parámetros, lo cual evita el tener que estar lidiando con grandes cantidades de argumentos en funciones que implementen dichos modelos. A su vez, se puede desligar mucho mejor la implementación del modelo de la simulación, lo cual proporciona una abstracción de conceptos más sencilla. Estas ventajas ayudan a obtener modelos más legibles y compactos, disminuyen el error humano involucrado en reimplementar aspectos de la simulación y facilita el proceso de debugging al permitir evaluar rápidamente la evolución de distintas variables.

Problemas 7 y 8

Las simulaciones de los modelos hechos con diagramas de bloques obtuvieron resultados similares a los de las implementaciones anteriores, lo cual ayuda a corroborar su correcto funcionamiento.

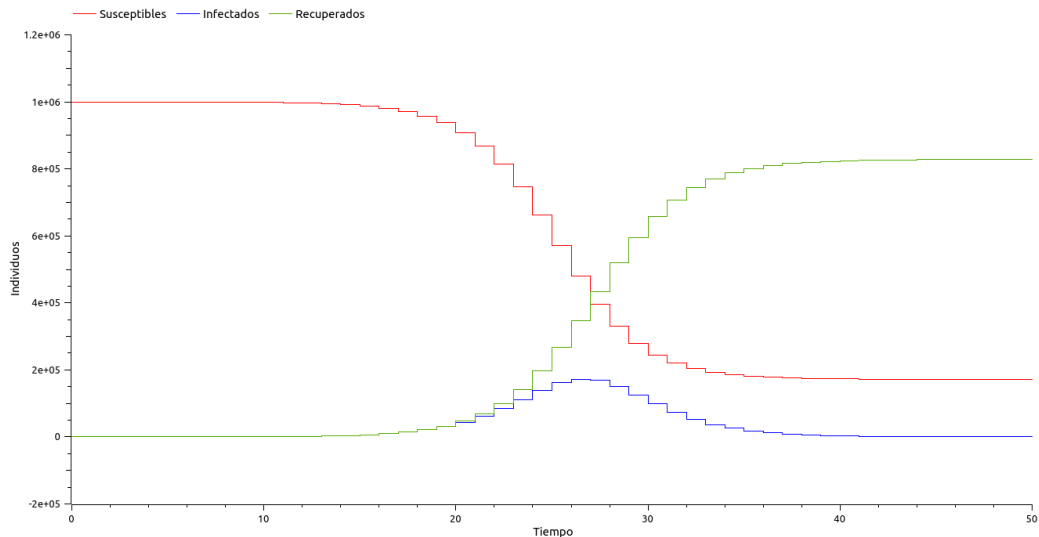


Figura 11: Simulación con diagrama de bloques de OpenModelica del modelo SIR.

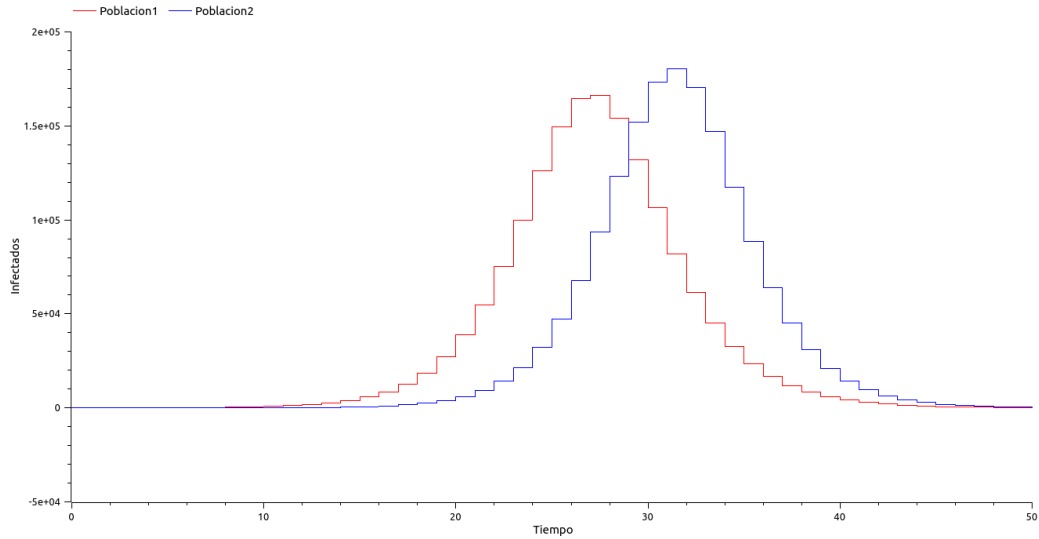


Figura 12: Simulación con diagrama de bloques de OpenModelica del modelo SIR de dos poblaciones.

Problema 9

Las curvas resultantes en las simulaciones con la nueva implementación son similares a las que ya se habían visto anteriormente:

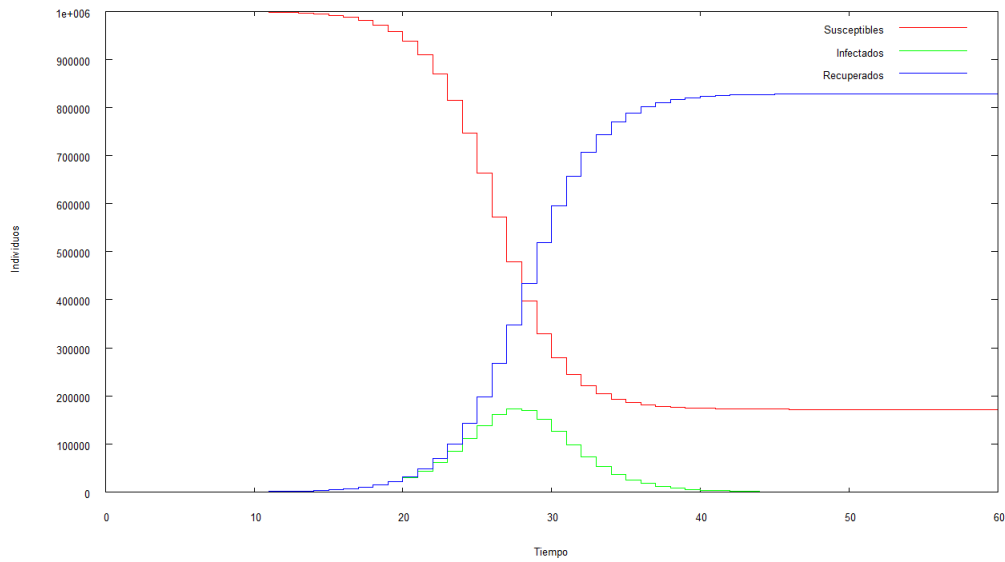


Figura 13: Simulación con PowerDEVS del modelo SIR.

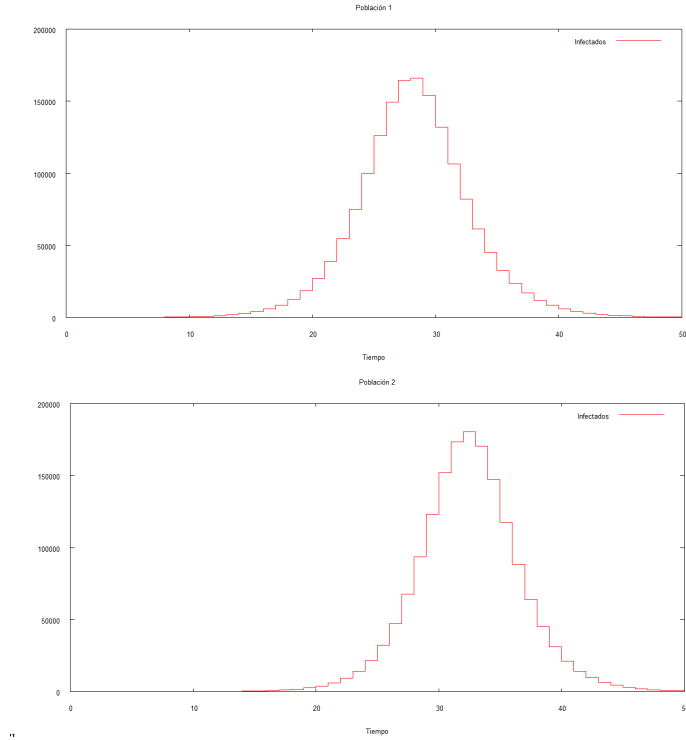


Figura 14: Simulación con PowerDEVS del modelo SIR de dos poblaciones.

Luego, se considera el caso de cuatro poblaciones que comienzan sin infectados. Además, tienen las siguientes características:

- Población 1: Recibe 100 infectados en el primer instante de tiempo.
- Población 2: Recibe 10 infectados cada instante de tiempo durante los primeros diez.
- Población 3: Recibe los casos exportados de las poblaciones 1 y 2.
- Población 4: Recibe los casos exportados de la población 3, y no exporta casos.

Las características que se quieren analizar son las diferencias de las curvas de infectados entre las poblaciones 1 y 2, y entre la 3 y 4. Notar que en la figura se grafican los casos exportados por cada población en vez del número de infectados por una cuestión de simplicidad en la estructuración de los modelos en PowerDEVS. Ya que el interés radica en comparar las curvas a grandes rasgos, estos valores son suficientes al ser proporcionales a los infectados totales.

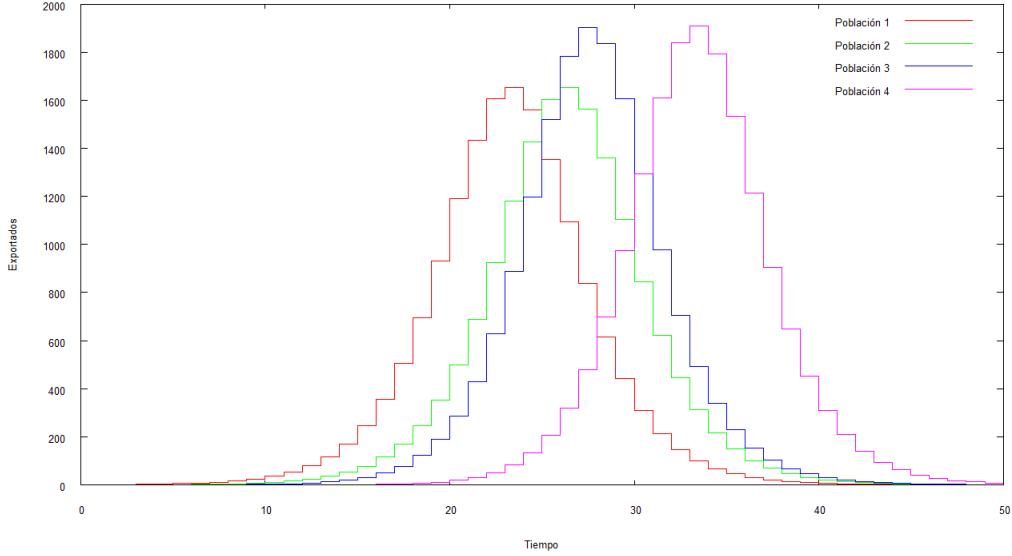


Figura 15: Simulación con PowerDEVS del modelo SIR de múltiples poblaciones.

Si bien hay un ingreso mayor de casos en la primera población, no resulta lo suficientemente relevante frente a la cantidad total de individuos para alterar el tamaño general de la curva de infectados. Como consecuencia, las curvas de las poblaciones 1 y 2 son casi idénticas, con la diferencia fundamental siendo que la primera llega a su pico tres unidades de tiempo antes que la otra.

La misma situación ocurre entre las poblaciones 3 y 4, con lo que parece ser que exportar casos no reduce drásticamente la evolución de los contagios dentro de una población bajo este modelo.

Problema 10

En la primer simulación, el robot acelera durante los primeros dos segundos mientras gira un poco sus ruedas y luego las endereza. Luego, parecería que se traslada a velocidad constante en línea recta. Los resultados de la simulación validan esto, mostrando que ambas coordenadas crecen linealmente en el tiempo a partir del segundo 2, con lo cual se interpreta que el robot se está moviendo a velocidad constante con un ángulo pequeño con respecto a la horizontal.

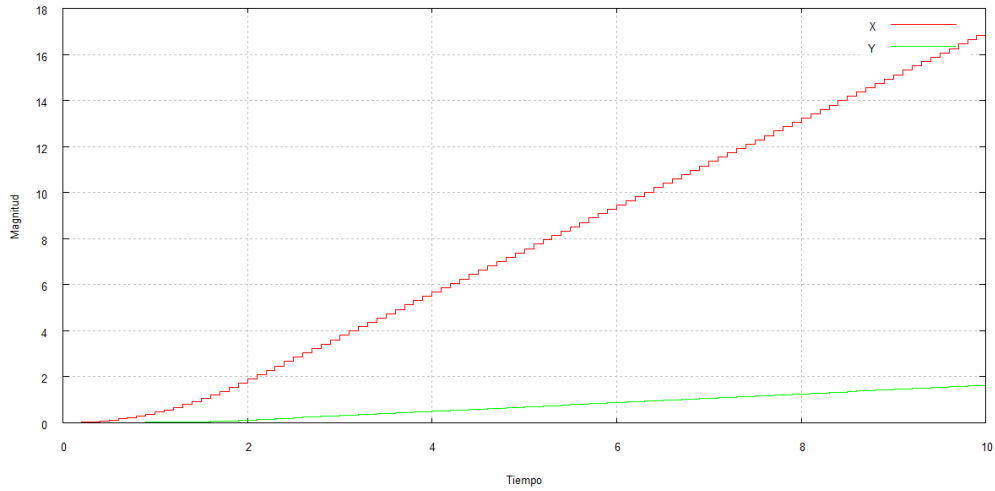


Figura 16: Simulación con PowerDEVS del modelo de desplazamiento de un robot.

Por otro lado, para el caso que utiliza una ley de control, se ve que la coordenada X evoluciona de una forma similar, mientras que la coordenada Y permanece constante y varía lentamente antes de estabilizarse otra vez. Es decir, el robot se movió por la horizontal, luego se desplazó a una línea paralela desplazada en el eje Y y continuó en la misma dirección que antes.

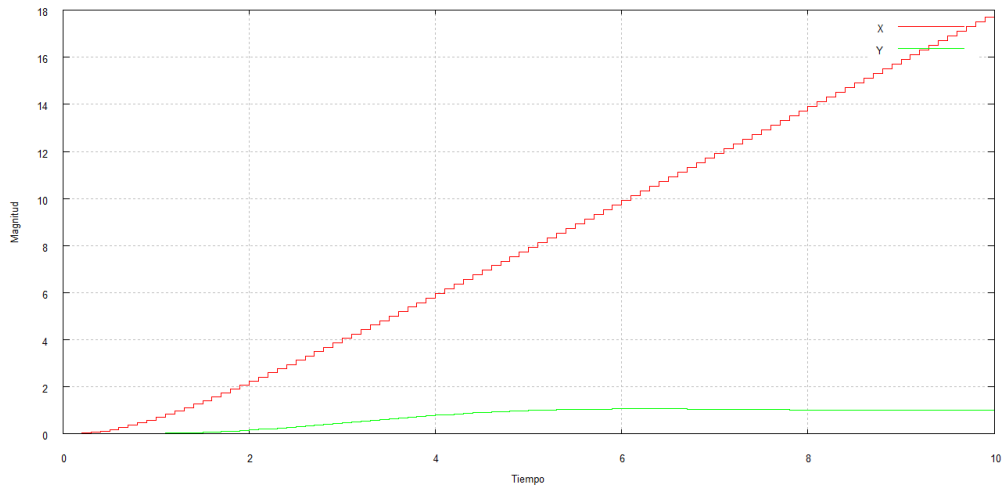


Figura 17: Simulación con PowerDEVS del modelo de desplazamiento de un robot con leyes de control.