

DEVS: Un primer modelo

DEVS: un primer modelo

Introducción al modelo	3
Elementos del modelo	3
Construcción del modelo M/M/1	5
Parámetros finales y ejecución del modelo	16
Preguntas finales	16

Introducción al modelo

Este ejemplo muestra cómo modelar un sistema de una cola y un servidor con una sola fuente de tráfico y una capacidad de almacenamiento infinita. En la notación, la M significa markoviana; M/M/1 significa que el sistema tiene un proceso de llegada de Poisson, una distribución exponencial del tiempo de servicio y un servidor. La teoría de colas proporciona resultados teóricos exactos para algunas medidas de rendimiento de un sistema de colas M/M/1 y este modelo facilita la comparación de los resultados empíricos con los resultados teóricos correspondientes.

Este modelo es importante para el estudio ya que constituye la pieza fundamental para luego modelar sistemas más complejos, con múltiples servidores y múltiples colas de llegada. Comprender bien este sistema allanará el camino para construir modelos más complejos. Por ello, vamos a detallar su construcción paso a paso.

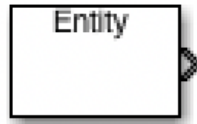
Elementos del modelo

A continuación se describen someramente los bloques con los que habitualmente se trabaja en la simulación de eventos discretos. Pueden encontrarse en la carpeta SimEvents de la librería de componentes de Simulink.

Todos los componentes estudiados anteriormente en los sistemas discretos pueden utilizarse también aquí si la algorítmica del problema lo requiere. No olvide que puede consultar la ayuda de cada componente en la ayuda general de Simulink. Asimismo, cada ventana asociada a un bloque tiene su propio botón de ayuda. Se recomienda encarecidamente su uso para familiarizarse con su estructura.

Generador de Entidades (Entity Generator)

Símbolo



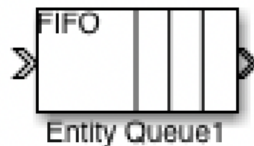
Descripción El bloque Generador de Entidades genera los elementos que entran en el sistema. Las entidades son elementos discretos de interés que se pueden definir en una simulación de eventos discretos. Una entidad puede llevar datos escalares, de bus o vectoriales. El significado de una entidad depende del modelo: una entidad puede representar clientes en un sistema de colas, paquetes de datos de un controlador remoto a un actuador, o cualquier elemento discreto que se defina.

Por defecto, el método de generación de entidades del bloque está basado en el tiempo. En este método, el bloque genera entidades utilizando tiempos de intergeneración especificados por el Periodo, a partir de una señal de entrada o una distribución estadística.

En la pestaña **Estadísticas** existen unos valores opcionales que pueden ser elegidos para visualizar los valores y el comportamiento de las entidades durante la simulación. Digamos que son, junto a un bloque **Scope**, la ventana que nos muestra lo que está sucediendo durante todo el proceso.

Cola (Entity Queue)

Símbolo



Descripción Este bloque almacena entidades o mensajes en una cola según el orden de llegada o la prioridad. Cada elemento en la cabeza de la cola sale cuando el bloque posterior está listo para aceptarlo. El bloque de cola y el bloque de cola de entidades son los mismos bloques con diferentes valores por defecto para la casilla sobrescribir el elemento más antiguo si la cola está llena.

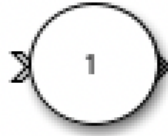
Se puede especificar la capacidad de la cola y la política cuando la cola está llena. El bloque admite tres políticas diferentes de ordenación de mensajes o colas: primero en entrar, primero en salir (FIFO), último en entrar, primero en salir (LIFO) y prioridad.

Lo mismo que ha dicho sobre la pestaña **Estadísticas** en el bloque **Entity Generator**, puede decirse de este bloque. Conviene estudiar con detenimiento las posibilidades que

ofrece.

Servidor (Entity server)

Símbolo

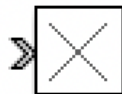


Descripción El bloque Servidor (Entity Server) procesa las entidades a medida que llegan. En una simulación de eventos discretos, un servidor almacena entidades durante un periodo de tiempo, llamado «tiempo de servicio», y luego intenta dar salida a la entidad. Durante el periodo de servicio, se dice que el bloque está sirviendo a la entidad que almacena. El bloque puede servir varias entidades simultáneamente y dar salida a cada entidad a través del puerto de salida a menos que el puerto esté bloqueado. Cuando el bloque permite la preferencia, una entidad en el servidor puede salir antes a través de un segundo puerto.

Normalmente, cuando el tiempo de servicio es variable, se usa una función de MATLAB para generar dicho tiempo asociándolo a una generación aleatoria con una función de distribución exponencial.

Terminador (Entity terminator)

Símbolo



Descripción El bloque Entity Terminator acepta y destruye entidades. Este bloque se utiliza para representar la salida de la entidad del modelo.

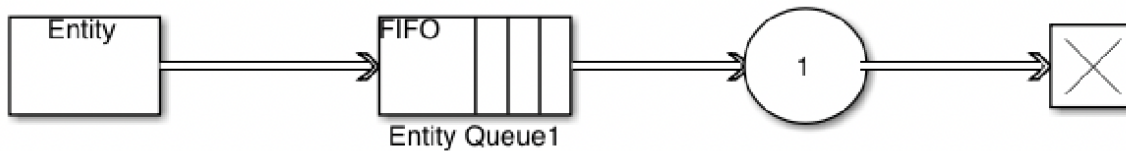
Para personalizar las acciones cuando las entidades entran, puede insertarse código de MATLAB en el campo **Entry Action** de la pestaña **Event Actions**.

Conviene tener presente la pestaña **Estadísticas** para saber el número de entidades que abandonan el sistema.

Construcción del modelo M/M/1

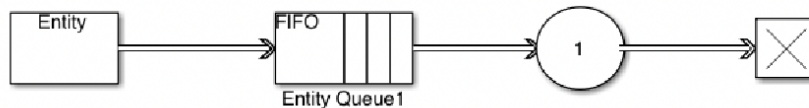
Paso 1: Componentes principales del modelo

En un proyecto en blanco, construimos el modelo general constituido por un Generador de entidades, una Cola, un Servidor y un Terminador y los unimos todos mediante sus correspondientes buses. Obsérvese que los buses, a diferencia de los modelos construidos hasta el momento, son más gruesos; esto es así para significar que a través de ellos pasan «entidades» (pasajeros, paquetes, clientes, ...) y no sólo valores de alguna variable:



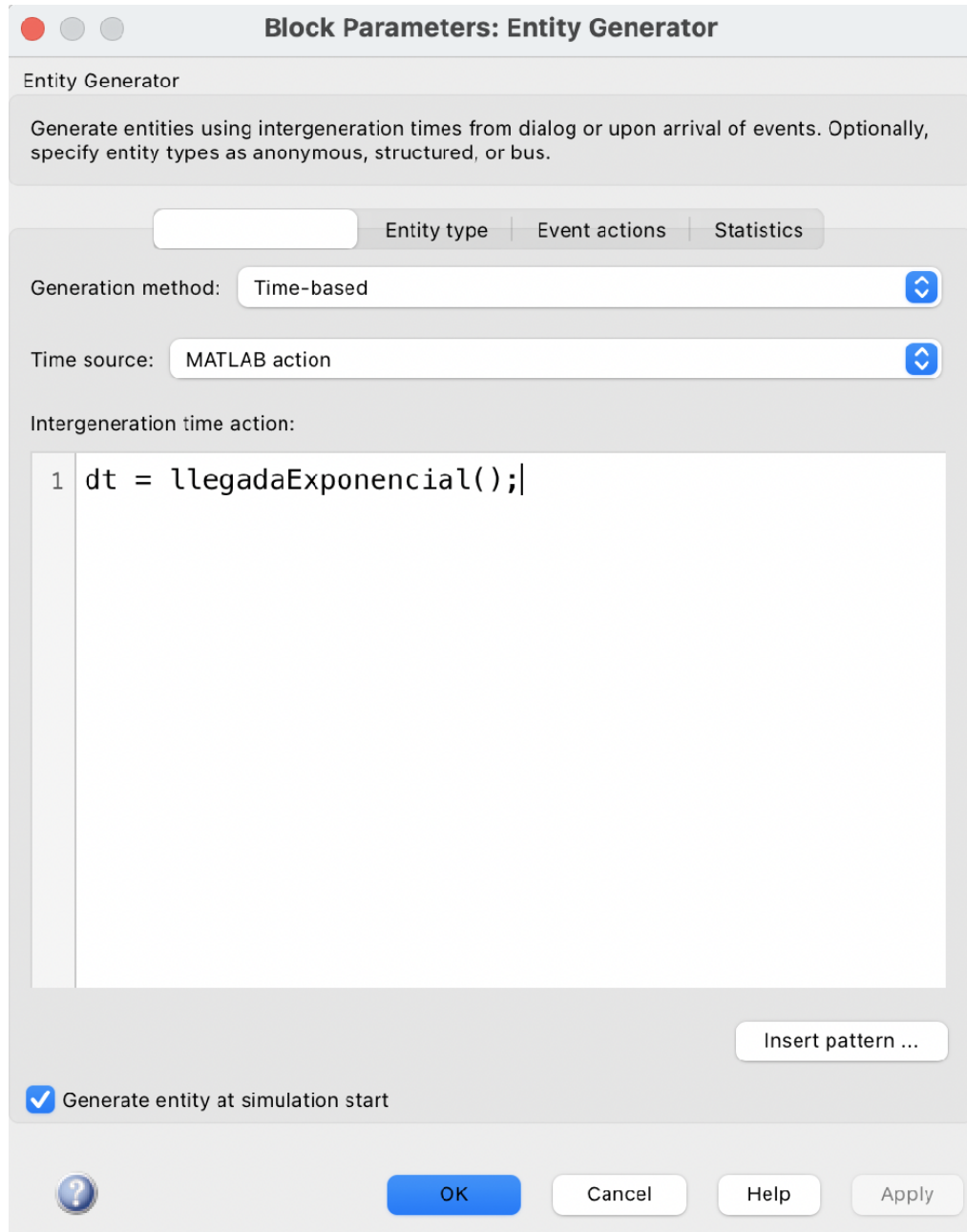
Paso 2: Botón de ajuste de tasa de llegada

En un modelo M/M/1, la primera «M» significa que la llegada de entidades al sistema es un proceso «Markoviano» es decir, estocástico, siguiendo una determinada distribución de probabilidad. En el caso de este sistema, la distribución utilizada será una distribución de Poisson. Como hemos estudiado, una distribución de este tipo se caracteriza por el parámetro «tasa de llegada», λ . Esa tasa, lógicamente, puede tomar diferentes valores. Para facilitar el uso final del modelo, y con objeto de no tener que introducir dicho parámetro a mano en un bloque «constante», vamos a introducir un bloque muy visual que obtendremos de la carpeta **Dashboard** de la biblioteca de bloques. El componente elegido será **Knob**. De momento no lo conectaremos con ningún otro componente.



Paso 3: Configuración del bloque Entidad

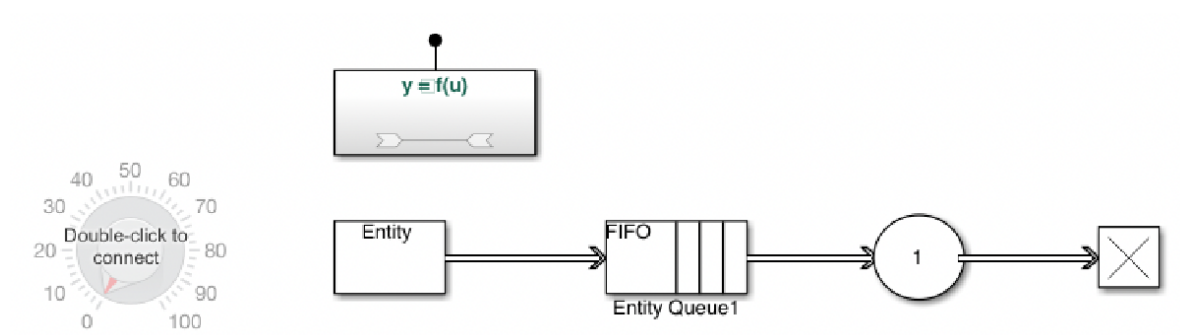
Nos vamos a centrar ahora en el bloque **Entity**. Si hacemos doble click sobre él, accedemos a la ventana siguiente:



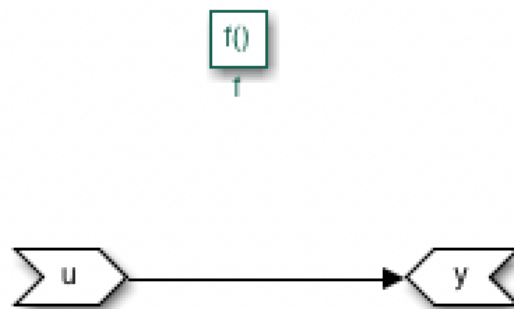
Vamos a definir una función que recoja la generación de entidades según la distribución de Poisson. Para ello, en **Time source** elegimos **MATLAB action** y en la ventana de **Integration time action** creamos una variable cualquiera (en nuestro ejemplo, `dt`) y la igualamos a la función que vamos a crear, en nuestro caso `llegadaExponencial()`. Ahora crearemos dicha función como una función de Simulink. La creación de dicha función se detalla en el paso 4.

Paso 4: Creación del bloque de funcionalidad de las entidades

Añadimos a continuación un nuevo bloque a nuestro modelo: **Simulink Function**. El modelo queda de la siguiente manera:

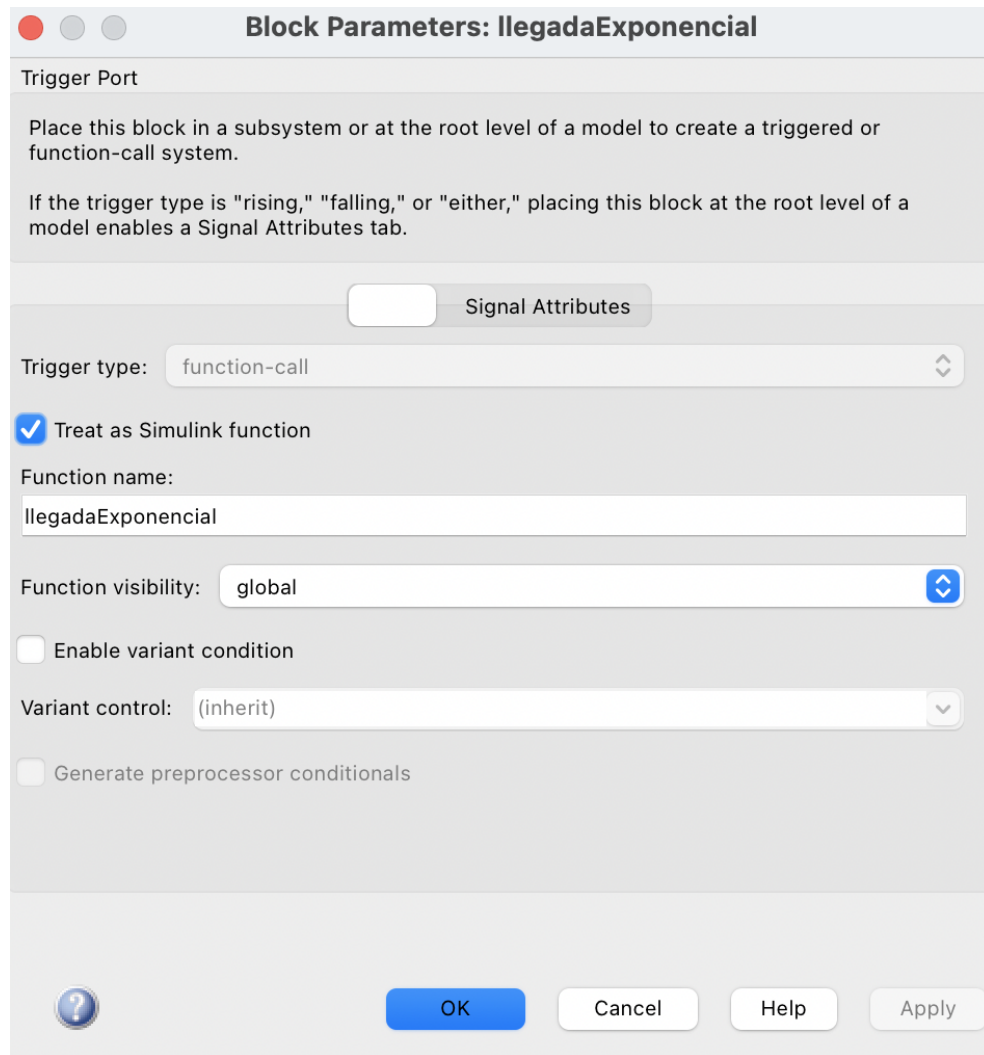


Ahora, y sobre el nuevo bloque introducido, hacemos doble click. Veremos algo parecido a lo siguiente:



Esta figura nos indica que podemos introducir nuestro modelo de función que tendrá una entrada «u» y una salida «y», además de los parámetros de la función que definimos a continuación.

Hacemos doble click sobre el cuadradito etiquetado como « $f()$ » y nos presentará la ventana siguiente:



Como se aprecia en la figura, en **Function Name** ponemos «llegadaExponencial» (el mismo nombre que dimos a la función en el paso 3) y en **Function visibility** elegimos «global». Damos al botón «OK». En estos momentos ya estamos en condiciones de definir nuestra función.

Hasta este momento, lo que hemos hecho ha sido asociar nuestro bloque de entidades a una función que todavía tenemos que elaborar.

Paso 5: Definición de la función «llegadaExponencial»

Ahora debemos construir un modelo en Simulink que reproduzca las funciones que se estudiaron en la Distribución de Poisson.

Como vimos en el capítulo anterior, para generar una muestra de una distribución exponencial negativa, utilizamos la siguiente expresión:

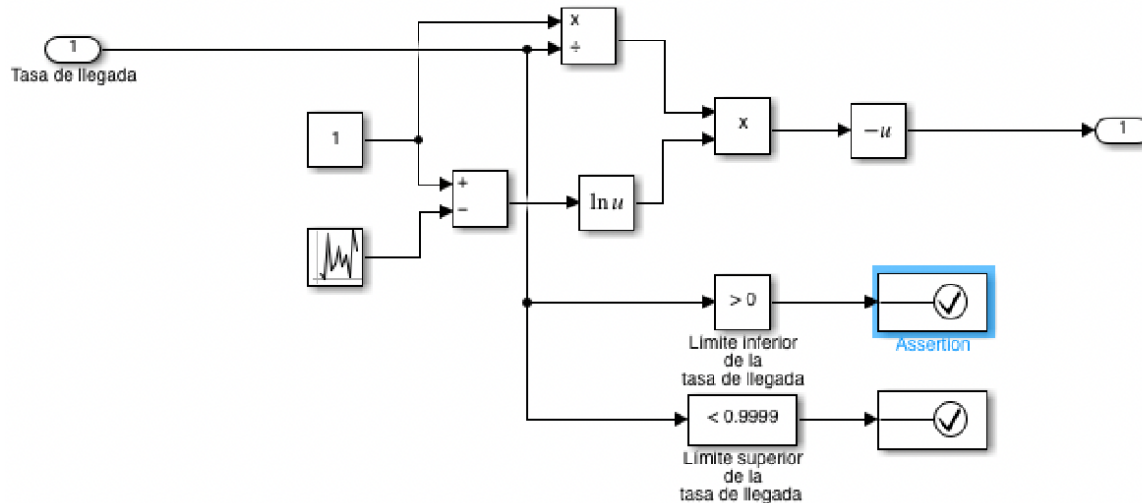
Si generamos un número aleatorio R donde $0 \leq R \leq 1$ entonces

$$C(t) = R = 1 - e^{-\lambda t}$$

$$e^{-\lambda t} = 1 - R$$

$$\lambda t = -\ln(1 - R)$$

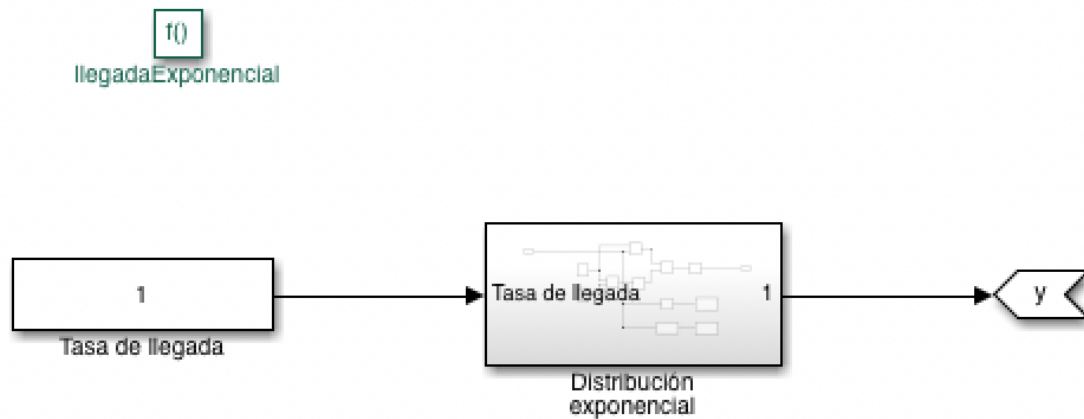
En esta expresión despejamos la t , intervalo de tiempo entre muestras, y obtenemos nuestra tasa de llegada. El modelo sería el siguiente (no borre los elementos de la figura de la función para realizar esta. Luego enlazaremos lo que teníamos con esto):



Merece un comentarios especial el bloque coloreado en la figura. Es el bloque **Assertion**. Este bloque controla que los límites de llegada estén entre cero y uno.

Ahora creamos un bloque **Constant** y lo llamaremos «Tasa de llegada». Este bloque es el que enlazaremos con el bloque **Knob** añadido en el paso 2.

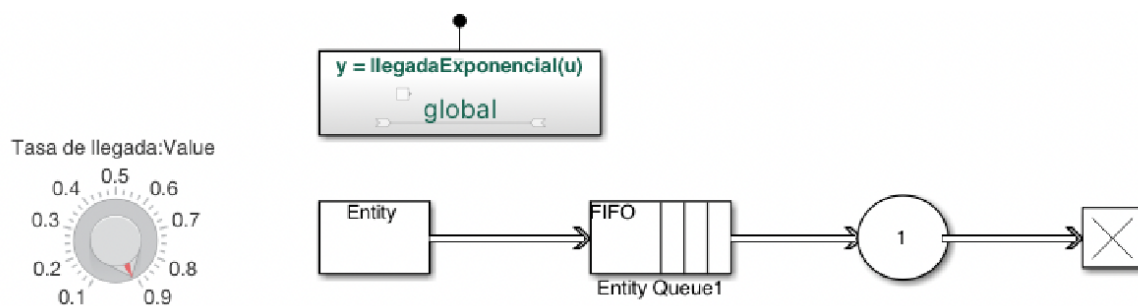
Por último, seleccionaremos toda nuestra función para crear un bloque que la enmascare y así hacer más legible el resultado. La entrada debe estar unida al boque «Tasa de llegada» y la salida será la «y» que ya teníamos de la figura anterior. Debe quedar algo así:



Por último, vamos a enlazar el bloque **Knob** con el bloque **Constant** «Tasa de llegada». Para ello, hacemos lo siguiente:

- 1.- Doble click en el bloque **Knob**.
- 2.- En el bloque de la función Simulink, hacemos doble click.
- 3.- Seleccionamos el bloque «Tasa de llegada»
- 4.- En la ventana del bloque **Knob**, pulsamos el «radio-button» para establecer la conexión (CONNECT).
- 5.- En la misma ventana, establecemos los límites mínimo (0.1) y máximo (0.9) de la escala que queremos que presente.
- 6.- Pulsamos OK.

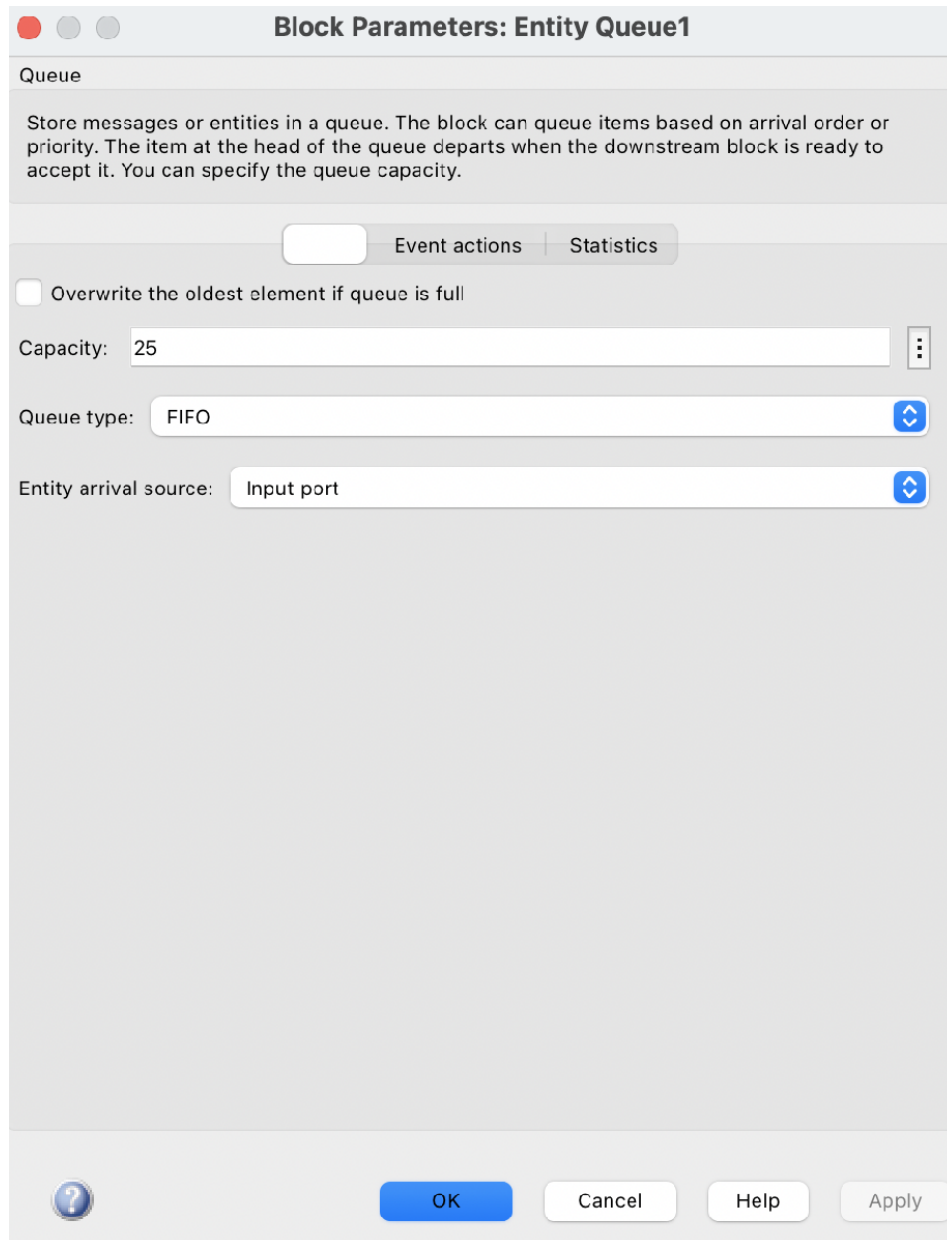
El modelo, en este momento, tiene que ser como el que se muestra en la figura:



Paso 6: Configuración de la cola FIFO

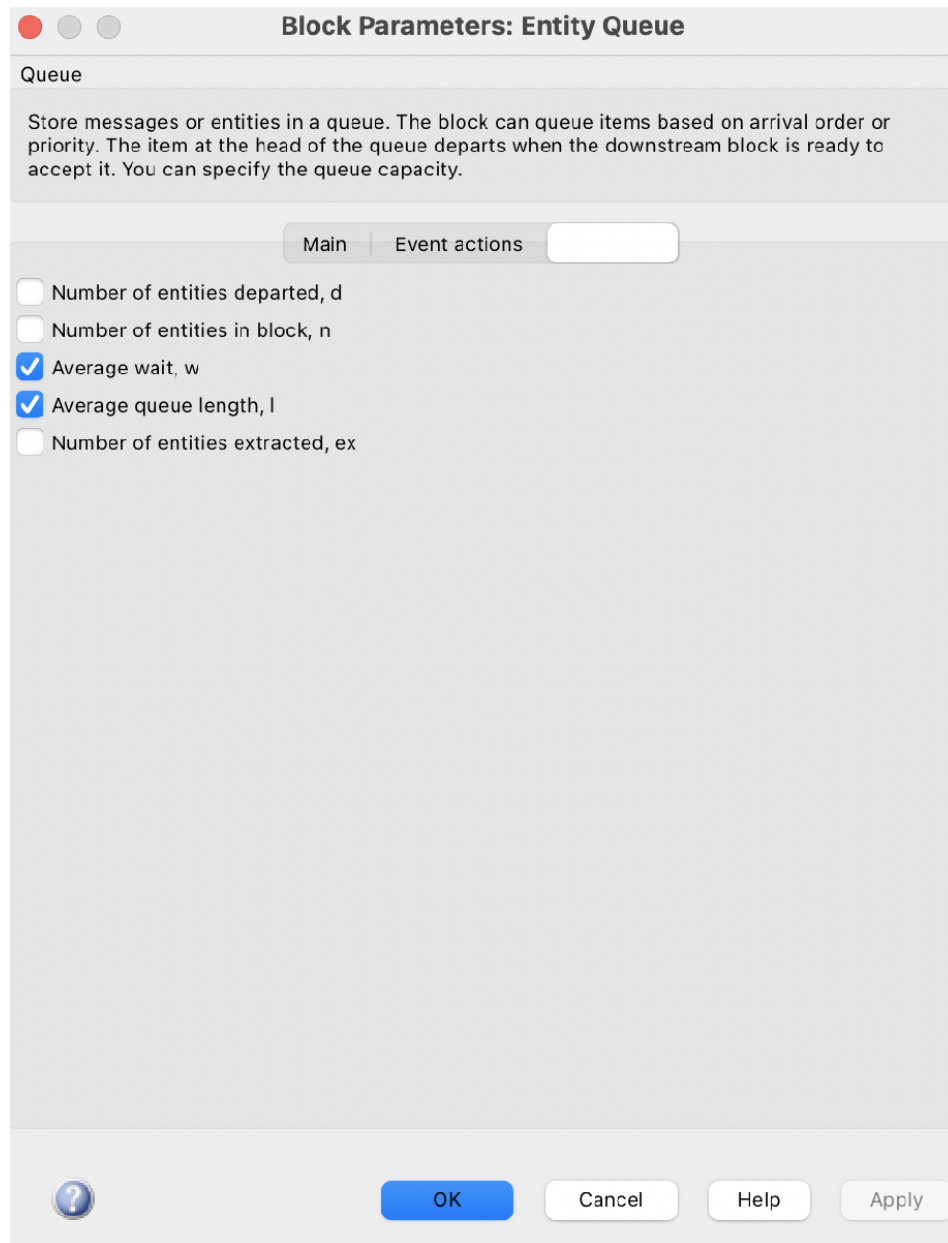
Después de haber configurado el generador de entidades de entrada, toca configurar la cola FIFO. En esta ocasión será más sencillo ya que, para este modelo, poco hay

que definir. Si hacemos doble click sobre el bloque Entity Queue, obtenemos la ventana siguiente:



En ella solo debemos modificar el parámetro **Capacity** a **Inf**, ya que no queremos limitar el número de elementos que pueden estar en la cola. Los demás parámetros los dejamos como están.

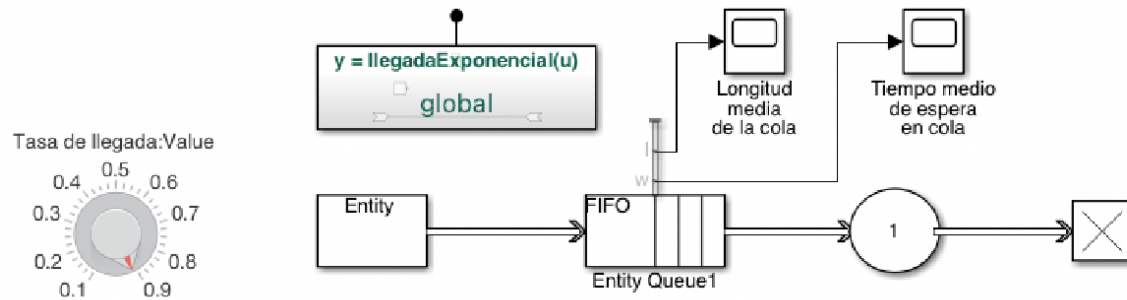
Ahora nos fijamos y hacemos click en la pestaña **Statistics**. Aquí podremos definir qué valores de la cola queremos visualizar para ver el comportamiento de la cola. Obtenemos la siguiente ventana:



En la cola FIFO nos interesan conocer dos parámetros: el tiempo medio de espera en cola (**Average wait, w**) y el número medio de entidades entrantes que están esperando en la cola (**Average queue length, l**). Simplemente seleccionamos estas casillas y pulsamos «OK».

Lo que provoca esta selección es que se habiliten dos puertos de salida en el bloque para que puedan ser conectados a un visor, por ejemplo, un bloque **Scope**.

Después de hechas estas operaciones el modelo deberá quedar como sigue:



Paso 7: Configuración del servidor

Nos queda por último configurar el servidor. Recordemos que el modelo que estamos diseñando es M/M/1, es decir, el tiempo de servicio del servidor también es «Markoviano», esto es, tiene un tiempo de servicio estocástico siguiendo una distribución de probabilidad exponencial negativa.

Una vez más deberemos recordar que la media de una distribución de Poisson es $\mu = \lambda t$. Por simplicidad vamos a considerar que $\lambda = 1$ y que la media $\mu = 1$. Por tanto, la ecuación que nos da el tiempo de servicio será:

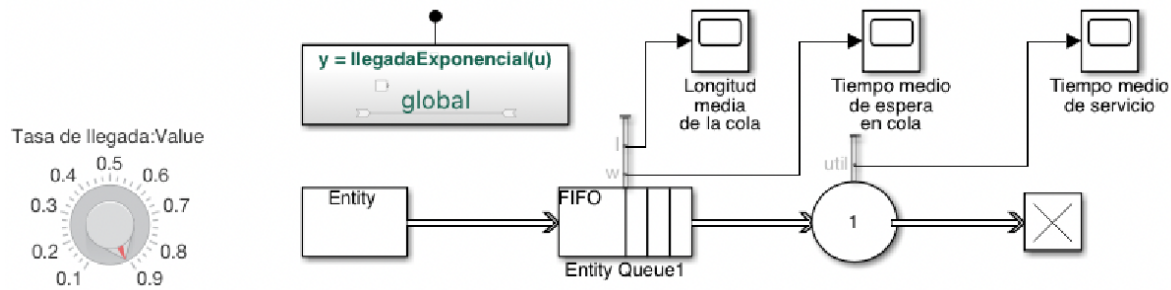
$$t = -\ln(1 - R)$$

Hacemos doble click sobre el bloque **Servidor** y obtenemos la siguiente ventana:

En el campo **Service time source** elegimos «MATLAB action» lo que nos permitirá añadir una porción de código en el campo **Service time value**. Introducimos el siguiente código:

```
% Tiempo de servicio exponencial con tasa de llegada igual a 1
media = 1;
dt = -media * log(1 - rand());
```

En la pestaña **Statistics**, seleccionamos el tiempo de utilización (**Utilization, util**) y añadimos un bloque **Scope** que etiquetamos como «Tiempo medio de servicio». Después de estas operaciones, el modelo quedará del siguiente modo:



Parámetros finales y ejecución del modelo

Antes de correr el modelo, deberemos ajustar los parámetros generales de la simulación como hicimos con los modelos discretos. En esta ocasión deberemos hacer:

- 1.- Seleccionar la pantalla de **Configuration parameters**
- 2.- En **Solver selection - Type** elegimos elegimos «Variable-step»
- 3.- En **Solver selection - Solver** elegimos elegimos «auto (Automatic solver selection)»
- 4.- Todos los demás parámetros se dejan por defecto.
- 5.- Podemos elegir un tiempo de parada de, por ejemplo, 1000.

Ahora es el momento de ejecutar el modelo y observar los resultados. Analícelos detenidamente y modifique la tasa de llegada para observar sus efectos.

Preguntas finales

Para comprobar que ha entendido bien el modelo, intente responder a las siguientes preguntas:

- a) En el momento de terminar la simulación, ¿cuántas entidades han salido del sistema?
- b) ¿Cuántas entidades se encuentran en cola?
- c) ¿Cuántas entidades están en el servidor pendientes de ser atendidas?
- d) ¿Cómo podríamos provocar un error para que entraran en juego los bloques **Assertion**?