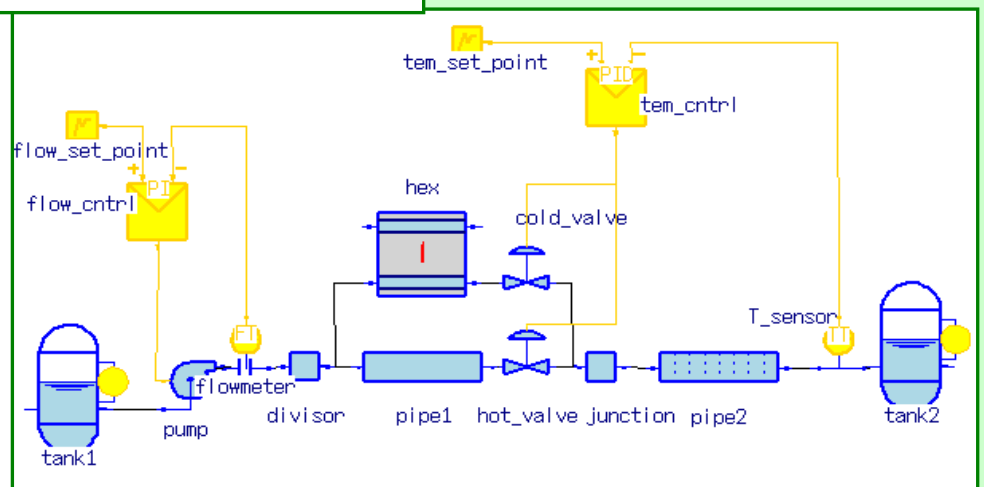
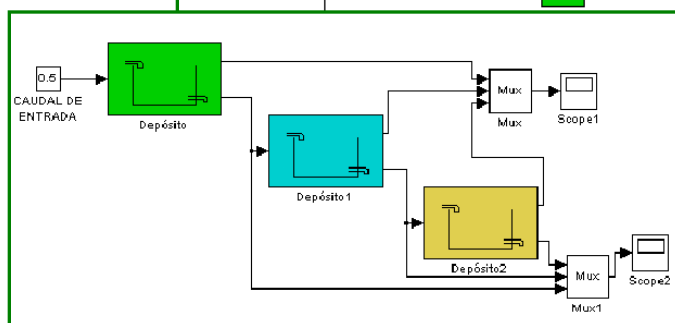
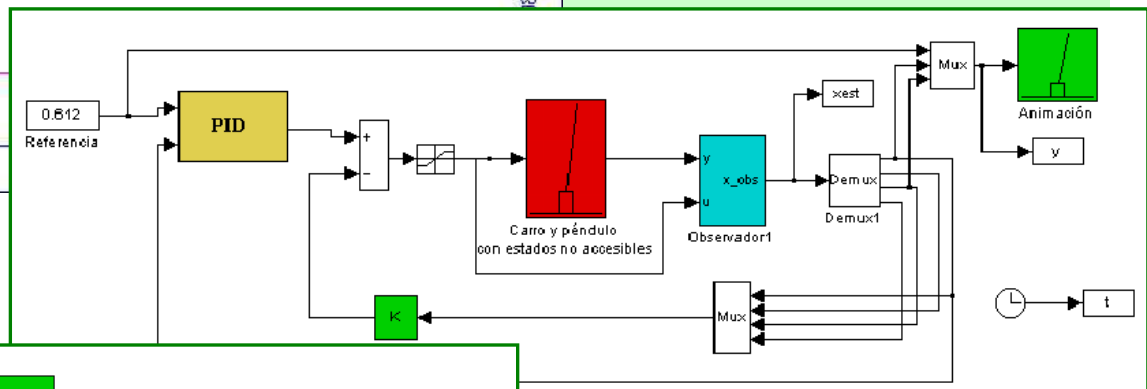
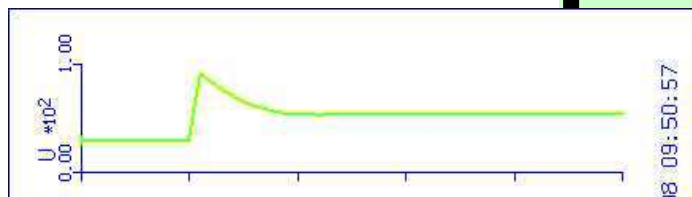


GUIONES DE PRÁCTICAS DE MODELADO Y SIMULACIÓN DE SISTEMAS DINÁMICOS

Jorge E. Jiménez Hornero



PRÁCTICA 0

INTRODUCCIÓN A MATLAB

OBJETIVOS: Introducción a MATLAB: familiarización con el entorno, utilización de vectores y matrices, uso de las funciones preimplementadas, creación de ficheros de comandos (ficheros .m) y definición y llamada de funciones creadas por el usuario.

RESOLUCIÓN: Se seguirán los siguientes puntos:

1. Generar un vector fila con los números del 7 al 12
2. Crear una matriz de 5x5
3. Especificar una submatriz de A con los dos primeros elementos de la tercera columna de A y llamarla B.
4. Crear dos matrices de 2x2, y hacer $A*B$, $A.*B$, A/B , $A./B$, $A\B$ y $A.\B$
5. Calcular la inversa de A y su determinante
6. Escribir un fichero m que cree una matriz cuyos elementos sean el inverso de la suma de los índices de su fila y columna
7. Escribir una función que devuelva el factorial de un número.
8. Escribir un fichero .m que calcule los 16 primeros de la sucesión de Fibonacci, cuyos dos primeros elementos son unos, y los siguientes la suma de los dos elementos precedentes.
9. Calcular la función seno para 1000 números entre 0 y 360°.
10. Si la presión de un depósito en función de la temperatura viene dada por la expresión

$$P = 0.0015T^2 + 0.0042T + 1.352 \text{ atm.}$$

Calcular la presión para temperaturas desde 0 °C hasta que la presión alcance 3.2 atm. con incrementos de T de 0.01 grado. Obtener una gráfica de P frente a T. Crear una función que devuelva las raíces de una ecuación de segundo grado.

PRACTICA 1

MODELADO Y SIMULACIÓN CON MATLAB

OBJETIVO: Familiarizarse con el uso del motor de resoluciones de ecuaciones diferenciales de MATLAB. Para ello se programarán tres ejemplos, que mostrarán las ventajas y limitaciones de esta herramienta.

PROBLEMA 1.1: CAÑÓN

La dinámica de los proyectiles de un famoso cañón de la Guerra de la Independencia americana tenía las siguientes características:

- El rozamiento, tanto en la componente horizontal como vertical del movimiento, era una fuerza proporcional al producto del módulo de la velocidad del proyectil por el componente de ésta en cada uno de los ejes del movimiento.
- La constante de proporcionalidad en este rozamiento es $R=2.3E-5$
- La masa del proyectil es $m=1$ Kg
- La velocidad de salida era de 600 m/s

Se pide:

1. Plantear las ecuaciones de movimiento para cada uno de los ejes (x e y).
2. Codificar las ecuaciones en MATLAB sin tener en cuenta eventos.
3. Representar la salida para diferentes ángulos de disparo, en la misma gráfica.
4. Repetir los apartados a b y c deteniendo la integración en el momento de que el proyectil alcance el suelo.

PROBLEMA 1.2: PELOTA

Representar la dinámica de una pelota lanzada hacia arriba con una velocidad inicial dada, y desde una altura dada. Utilizar eventos para detectar el momento en que vuelva al suelo y detener la integración.

Una vez hecho esto, utilizar la función para representar la altura en función de tiempo, de una pelota que al caer rebotara en el suelo, con un coeficiente de restitución de $r=0.8$.

PROBLEMA 1.3: FRENO DEL AVION

El objeto del sistema de frenado que se muestra a continuación es el de parar una aeronave en movimiento. La simulación está orientada a investigar las masas del sistema de frenado y avión, así como su velocidad tal que no se excedan los límites de las tensiones de los cables, el desplazamiento del pistón de frenado y en último caso la longitud efectiva de la pista.

La geometría del problema viene dada en la figura inferior. Las ecuaciones que gobiernan el comportamiento del sistema vienen dadas por:

$$\begin{aligned} m_3 y_3'' &= f k_2 - f_d & y_3(0) &= y_3'(0) = 0 \\ m_2 y_2'' &= 2 f k_1 - f k_2 & y_2(0) &= y_2'(0) = 0 \\ m_1 x_3'' &= -2 f k_1 \sin \theta & x(0) &= 0; \quad x'(0) = 290 \text{ pies/seg} \end{aligned}$$

con las fuerzas expresadas por:

$$f k_2 = \begin{cases} k_2 (y_2 - y_3) & \text{si } y_2 > y_3 \\ 0 & \text{si } y_2 \leq y_3 \end{cases} \quad f k_1 = \begin{cases} k_1 (y_1 - 2y_2) & \text{si } y_1 > 2y_2 \\ 0 & \text{si } y_1 \leq 2y_2 \end{cases}$$

$$f_d = f(y_3)(y_3')^2$$

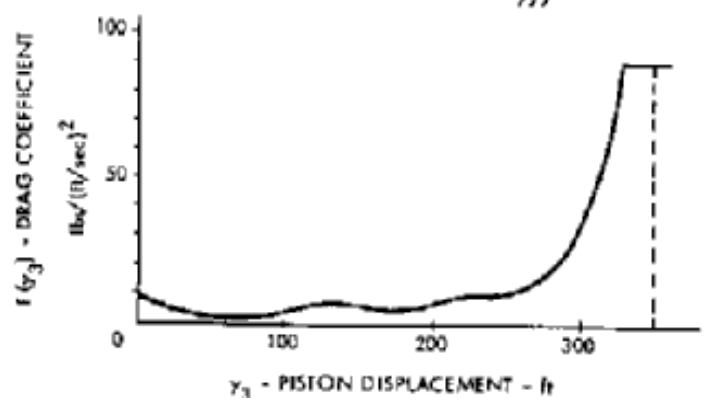
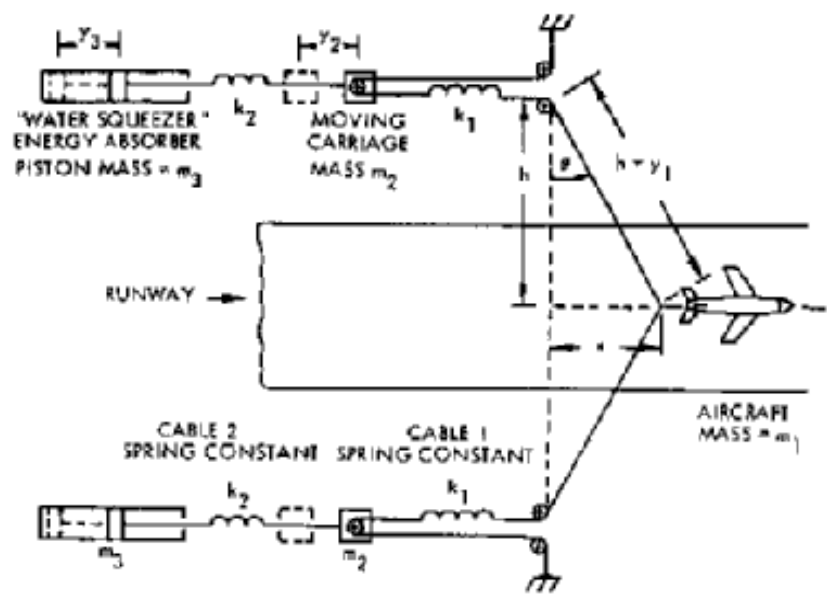
con las siguientes constantes del sistema

$$\begin{aligned} m_1 &= 140 \text{ slugs,} & k_1 &= 4550 \text{ libras/pie} \\ m_2 &= 45.28 \text{ slugs,} & k_2 &= 25300 \text{ libras/pie} \\ m_3 &= 20 \text{ slugs,} & h &= 125 \text{ pies} \end{aligned}$$

Los valores de la no linealidad $f(y_3)$ (función de amortiguamiento) vienen dados según la tabla:

y_3	-10	0	30	60	120	150	180	210	240	270	282	294	306	312	324
$f(y_3)$	8.33	8.33	4	1.6	5.2	5.2	6.6	8.3	10.7	16	21	28	41	50	90

Bajo estos supuestos, construir un programa MATLAB que simule el sistema y muestre las gráficas de posición y velocidad del avión.



PRACTICA 2

MODELADO Y SIMULACIÓN CON SIMULINK

OBJETIVOS: Familiarizarse con el uso del entorno basado en bloques de Matlab Simulink. Para ello se realizarán los dos problemas siguientes.

PROBLEMA 2.1: EL PÉNDULO SIMPLE

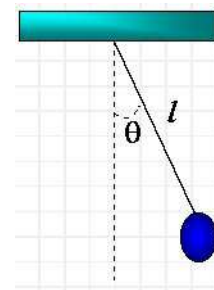
Sea un péndulo simple con las siguientes características.

l : longitud de la varilla es (1 m).

g : aceleración de la gravedad (9.8 m/s^2).

m : masa de la bola (0.1 kg).

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l} \text{sen}\theta$$



1. Crear un modelo SIMULINK. Probar qué ocurre al cambiar los distintos parámetros l y m , así como las condiciones iniciales de posición y velocidad. Utilizar el bloque de animación suministrado.
2. Volver a realizar el modelo, pero ahora teniendo en cuenta el coeficiente de rozamiento B de 0.02.

PROBLEMA 2.2: DEPÓSITO

Un depósito para líquidos presenta una dinámica que puede ajustarse a la siguiente ecuación diferencial

$$A \frac{dh}{dt} = q_e - q_s = q_e - K\sqrt{gh}$$

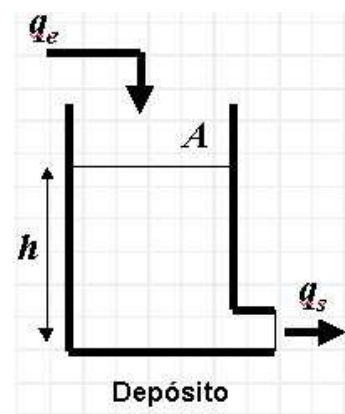
y donde

h : altura del agua (m)

h_0 : altura inicial (10 m)

g : aceleración de la gravedad (9.8 m/s^2)

A : sección del depósito (5 m^2)

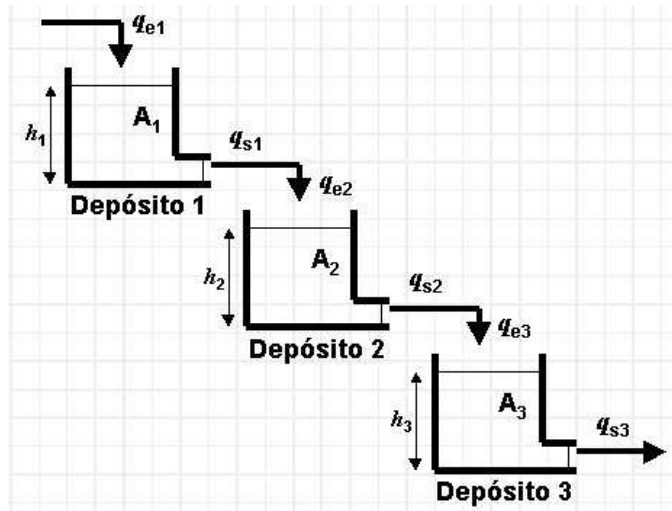


q_e : caudal de entrada ($0.5 \text{ m}^3/\text{s}$)

q_s : caudal de salida (m^3/s)

K : constante

1. Obtener un modelo Simulink para el sistema anterior. Ejecutar la simulación añadiendo posibles mejoras.
2. Crear un subsistema con el modelo. La entrada deberá ser el caudal de entrada y la salida el caudal de salida.
3. Enmascarar el subsistema, añadiendo una ventana que pida al usuario, por ejemplo, sección A y altura inicial.
4. Copiar el sistema anterior para representar la dinámica de tres depósitos que vierten cada uno en el siguiente.



PRACTICA 3

MODELADO Y SIMULACIÓN CON ACSL

OBJETIVOS: Se trata de una toma de contacto con uno de los entornos de simulación más utilizados en ámbitos industriales y académicos durante las dos últimas décadas: ACSL. En ella se pondrán de manifiesto las características de este lenguaje, haciendo especial énfasis en las diferencias existentes con el resto de entornos de prácticas anteriores. Se va a utilizar una antigua versión de estudiante que funciona bajo DOS.

REALIZACIÓN: Se van a repetir varias de las prácticas anteriores, pero en este caso utilizando ACSL.

1. Repetir el problema del cañón codificado en ACSL. Detectar el evento de llegada al suelo del proyectil con el comando term
2. Idem con el programa de la bola. Utilizar una instrucción schedule que modifique la velocidad de la bola cada vez que rebote. Recordar que en ACSL no es necesario realizar simulaciones concatenadas como se hacía con MATLAB.
3. Codificar con ACSL el problema de los depósitos acoplados de clase. Fijar un punto de trabajo en el que estabilizar los niveles. Utilizar el comando analyz para obtener un modelo lineal en torno a ese punto de trabajo. Comprobar que es el mismo que el calculado teóricamente. Para facilitar los resultados, y con el modelo de clase, utilizar los valores de $k_g=2$, $D=1 \text{ m}^2$, y $q=1 \text{ m}^3/\text{h}$.

ACSL VERSIÓN DE ESTUDIANTE PARA MS-DOS

A continuación se describe cómo instalar en casa la versión de estudiante de ACSL:

1. Ejecutar el programa "instalar" del disquete. Puede hacerse perfectamente desde Windows.
2. Modificar el autoexec.bat de forma que incluya las siguientes instrucciones:

```
SET ACSLSYS = C:\SIMAV\ACSL
SET ACSLPLT = VGACOLOR
SET ACSLFOR = MS
SET TMP = C:\SIMAV\ACSL
SET LIB = C:\SIMAV\FORTRAN
SET INCLUDE = C:\SIMAV\FORTRAN
PATH =(AÑADIR AL PATH) C:\SIMAV\ACLS:C:\SIMAV\FORTRAN
```

3. Reiniciar el equipo

CARACTERÍSTICAS

- La versión de estudiante para MS-DOS acepta la mayoría de los comandos de la versión de Windows del manual, con algunas salvedades:
- Cuando se utiliza especificaciones de los comandos, como plot, no se ponen detrás de una barra \ sino entre comillas. Ejemplo plot 'xaxis'=x1,x3 que especifica que se utilice la variable x1 como eje x (por defecto siempre es la variable independiente t la que se utiliza).
- Si se quiere utilizar un fichero de comandos hay que utilizar el comando siguiente set cmd=10, a lo que el compilador solicitará el nombre del fichero *.cmd que se utilizará. Esto es práctico cuando se repiten muchas veces los mismos comandos.
- Si se quieren guardar los nombres de las variables en un fichero ASCII, para posteriormente utilizar, por ejemplo, MATLAB, hay que definir con output las variables que se desean guardar, utilizar el comando set dis=1, para que la salida estándar sea un fichero. Al ejecutar start posteriormente, se solicita el fichero donde se guardarán los datos. si se quieren sacar los datos por pantalla. utilizar set dis=0.
- No se permiten sentencias IF..THEN..ELSE. Para simular éstas ver alguno de los ejemplos de a continuación.
- Todos los comentarios deben estar entre comillas
- No se aceptan comandos parameter ni dimension, por lo que no se pueden utilizar vectores.
- Otras diferencias se pueden apreciar en los ejemplos siguientes.

PRACTICA 4

DOMINIOS DE ESTABILIDAD

OBJETIVOS: El objetivo principal de esta práctica es que el alumno sepa evaluar el dominio de estabilidad de un algoritmo de simulación cualquiera. Para ello debe calcular la matriz discreta de evolución de estados (matriz F) para un sistema lineal, y a partir de ella estudiar sus autovalores, que deberán estar contenidos en el círculo unidad (su módulo, recordemos que pueden ser número complejos, debe ser por tanto menor que uno).

REALIZACIÓN: Como sistema lineal se va a utilizar el dado por las matrices siguientes. Como puede observarse, el tamaño de estas matrices impide un cálculo analítico inmediato.

$$A = \begin{pmatrix} 1250 & -25113 & -60050 & -42647 & -23999 \\ 500 & -10068 & -24057 & -17092 & -9613 \\ 250 & -5060 & -12079 & -8586 & -4826 \\ -750 & 15101 & 36086 & 25637 & 14420 \\ 250 & -4963 & -11896 & -8438 & -4756 \end{pmatrix} \quad B = \begin{pmatrix} 5 \\ 2 \\ 1 \\ -3 \\ 1 \end{pmatrix}$$
$$C = (-1 \quad 26 \quad 59 \quad 43 \quad 23) \quad D = 0 \quad x_0 = \begin{pmatrix} 1 \\ -2 \\ 3 \\ -4 \\ 5 \end{pmatrix}$$

Para el desarrollo de la práctica se seguirán los siguientes apartados:

1. Crear un modelo simulink con el sistema continuo anterior, utilizando un bloque que permita la introducción del modelo en espacio de estados. Introducir la matriz anterior en el espacio de estados, no en el modelo y utilizar nombres de variables en el interior de la máscara. Lo mismo con el estado inicial x_0 .
2. Excitarlo con una entrada escalón unitario, pero con amplitud nula. Esto es simplemente para que exista una entrada nula. También puede excitarse con una entrada constante nula, pero el anterior da más generalidad.
3. El sistema tendrá dos salidas. Una un Scope normal y otra un port, que permitirá que las salidas pasen al espacio de trabajo. Habilitar en los parámetros de simulación el

que salgan al espacio de trabajo tanto el vector de tiempos de simulación como el de la salida del sistema.

4. Efectuar una simulación con un método cualquiera para ver que el sistema funciona bien, y ploteando desde el espacio de trabajo la variable de salida.
5. En el menú de parámetros de simulación elegir un algoritmo de paso fijo y el método de integración el de Euler.
6. Realizar en MATLAB una función que permita calcular el paso máximo que garantice la estabilidad del método. Para ello, el fichero calculará la función F y analizará para cada paso h los autovalores. El programa se detendrá cuando, al aumentar el valor de h , aparezcan autovalores fuera del círculo unitario. En ese instante el programa devolverá el valor de h , que se tomará como el h máximo que garantiza la estabilidad. La función debe tener seis entradas, h mínimo y máximo entre los que buscará la solución, y las matrices A , B , C y D . Y debe tener una salida $hest$ (el paso h que garantiza la estabilidad).
7. Simular el sistema con distintos valores de h , por ejemplo $h=0.1$ hest, $h=0.9$ hest, $h=1.1$ hest, y $h=3$ hest. Analizar los resultados.
8. Volver al repetir los apartados 5, 6 y 7 para los algoritmos de Heun (recordar que es un RK2) y el de RK4. Esas dos nuevas funciones serán idénticas a la anterior, salvo en el cálculo de la matriz F .

PRÁCTICA 5

MODELADO Y SIMULACIÓN CON ECOSIMPRO

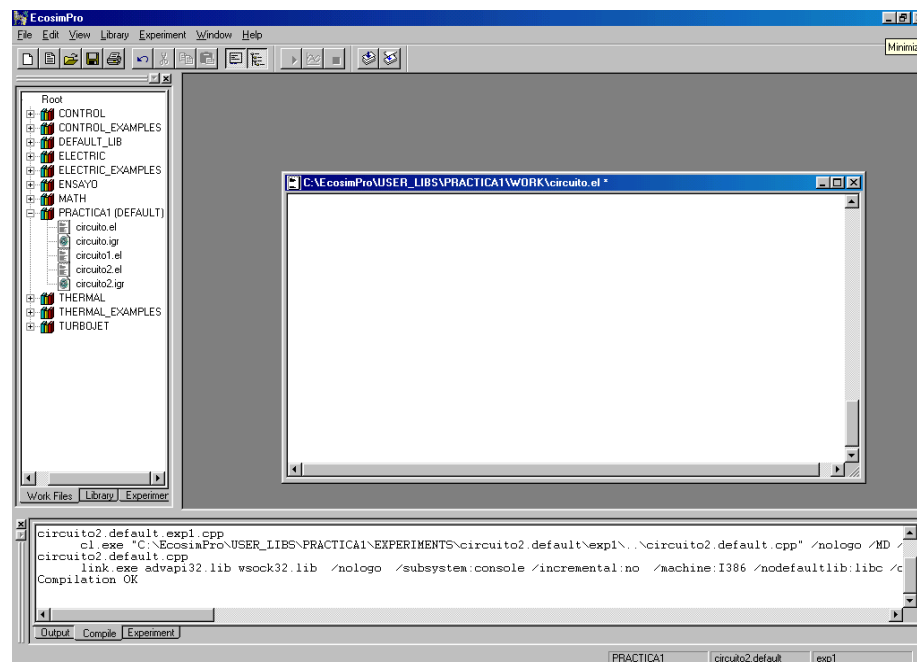
OBJETIVOS: El objetivo de esta práctica es dar una introducción sobre los lenguajes de modelado orientado a objetos (MOO), utilizando la herramienta EcosimPro. Para ello se creará una librería de componentes eléctricos simples, que se irán uniendo para crear otros componentes más complejos.

DESARROLLO: La práctica se desarrolla en dos sesiones. En la primera se realizará una librería de componentes eléctricos a partir de la cual se generarán sistemas más completos. En la segunda, se creará una librería gráfica basada en la anterior, y a partir de ella se volverán a generar otros sistemas.

PROBLEMA 5.1: MOO CON LIBRERÍAS TEXTUALES

Para llevar a cabo esta primera sesión, se realizarán los siguientes pasos:

1. Abrir el entorno ECOSIM. Aparecerá una ventana como la siguiente, donde se muestran los distintos campos, en concreto, la ventana de librerías, la de compilación y la de trabajo.



2. Utilizando el menu **Library/New**, crear una nueva librería que se denomine, por ejemplo, PRACTICA1.
3. Crear un fichero de texto nuevo (el segundo icono de la barra de menús). En éste, y utilizando los apuntes de clase, crear una librería de componentes simples, en concreto, una resistencia, un condensador, una fuente de tensión constante y una toma de tierra. Todos, salvo el último, serán SUBCLASES de la clase DosPines, creada previamente.
4. Crear en el mismo fichero un circuito serie RC, con la fuente de tensión y la toma de tierra. Denominar a éste con el nombre de circuitoRC. La resistencia y el condensador deben tener $R=1000$ y $C=0.0001$ para que la cte de tiempo sea de 0.1 segundo.
5. Una vez escrito el fichero, salvarlo dentro de la librería anterior.
6. Colocar el ratón sobre el nombre del fichero en la ventana de librerías y pulsar el botón derecho, para compilarlo. Debe estar abierta la hoja (menú inferior de la ventana de librerías) de **work files**.
7. Una vez compilado con éxito, abrir en la ventana de librerías la hoja de **librarys** para ver que se han creado con éxito todos los componentes.
8. Abrir en la ventana de librerías la hoja de **experiments**. Colocar el ratón sobre el objeto circuitoRC y con el botón derecho crear una **Default Partition**. Esto creará una secuencia de ecuaciones ordenadas y con asignación de causalidad.
9. Una vez creada la partición (aparece un nuevo elemento llamado f(x) Default) se pueden ver las ecuaciones ordenadas con el botón derecho y seleccionando **Mathematical View**. Además se habrá creado un fichero en el directorio `C:\ECOSIM\USERLIBS\PRACTICA1\EXPERIMENTS\CIRCUITO1.DEFAULT` denominado **circuito1.default.txt** que contiene esta información.
10. Colocando de nuevo el ratón sobre f(x)Default y pulsando el botón derecho, seleccionar esta vez, New Experiment, para crear un experimento. Aparece un nuevo fichero de texto en la pantalla de trabajo. Modificar las condiciones iniciales, el tiempo de simulación y el periodo de comunicación. Es recomendable no tocar las primeras, poner a 0.3 el segundo y a 0.001 el tercero, si se han utilizado los anteriores valores de R y C.
11. Al salvarlo aparece debajo de f(x)Default la palabra exp1. Con el botón derecho de ratón, colocado sobre esta palabra seleccionar **Simulate in Monitor**, aunque se puede utilizar el icono de la barra de menús superior.
12. Aparece una nueva pantalla, después de unos instantes en los que se ha compilado en C++ el fichero del modelo (o en concreto, su partición) y el de la simulación.

13. En esta pantalla, utilizar el icono superior **Edit Plot**, para seleccionar las variables que se deseen dibujar, por ejemplos las tensiones v de todos los componentes.
14. Pulsar **start** y analizar la gráfica, para ver que se corresponda a la típica curva de carga de un condensador.
15. Volver a repetir el mismo proceso para un circuito RCC con los dos condensadores en paralelo. Comparar los dos ficheros con la partición.

PROBLEMA 5.2: MOO CON LIBRERÍAS GRÁFICAS

En esta segunda sesión se realizarán los siguientes pasos:

1. Abrir el entorno ECOSIM. Recompilar el fichero de la sesión anterior para generar los distintos componentes.
2. En la ventana de librerías, y en la hoja Library, seleccionar cada uno de los componentes creados (la fuente, la toma de tierra, la resistencia y el condensador), hacer click con el botón derecho y elegir, en el menú contextual que se abre, la opción *Symbol -> Generate Symbol*. Si la opción no estuviera habilitada hacerlo desde la entrada de menú *Library -> Component Type -> Generate Symbol*. A continuación se abrirá *Ecodiagram* y permitirá trazar el dibujo correspondiente al componente para que, posteriormente, se pueda añadir a cualquier esquema.
3. Modificar a voluntad el dibujo de cada componente (emplear la barra de herramientas vertical de la derecha de la ventana) y guardar el trazado resultante. Los cambios aparecerán reflejados en el panel izquierdo llamado *Ecosim Libraries*, en la pestaña correspondiente a la librería sobre la que se está trabajando.
4. Después de crear el símbolo para cada componente, es necesario crear un nuevo esquema en el que se conectarán los componentes gráficos que se deseen y que, posteriormente, *Ecodiagram* se encargará de traducir al lenguaje EL. Para ello, hacer click en el botón *New* de *Ecodiagram*, situado en la esquina superior izquierda de la ventana, en la barra de herramientas.
5. Ir seleccionando los distintos componentes del panel de la izquierda para colocarlos en la ventana de trabajo de la derecha, donde se encuentra el esquema.
6. En cada uno de los componentes insertados se puede hacer doble click para entrar en un editor de datos, en el que se pueden especificar valores para las variables públicas del componente.

7. Crear un circuito fuente-resistencia-condensador-tierra, como el de la práctica anterior, uniendo los terminales con la herramienta *Draw Connector* de la barra de herramientas vertical de la derecha.
8. Una vez creado el circuito, salvarlo con el nombre circuito1.
9. Hacer click en el botón *Compile Schematic* de la segunda barra de herramientas de *Ecodiagram*.
10. El control pasa a la pantalla de Ecosim. Comprobar que se ha creado un fichero llamado Circuito1.el. Editarlo y comprobar que se ha generado correctamente la topología deseada.
11. Proceder en Ecosim como en la práctica anterior, para crear una partición, un experimento y una simulación, en este orden.
12. Comprobar que el fichero de Mathematical View coincide con el generado en la práctica anterior utilizando librerías de texto.
13. Volver a realizar en mismo proceso con un circuito con una fuente, una resistencia, dos condensadores en paralelo y una toma de tierra. Comprobar los resultados con los de la práctica anterior.

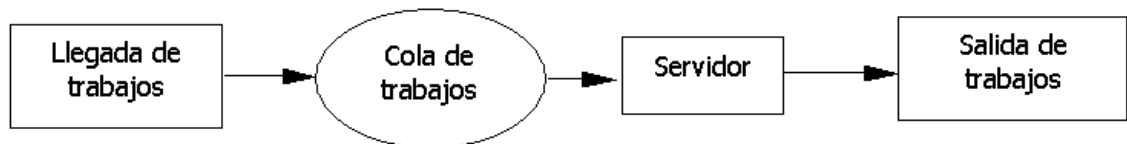
PRÁCTICA 6

MODELADO Y SIMULACIÓN CON ARENA

OBJETIVOS: Con esta práctica se pretende que el alumno se familiarice con los entornos de simulación de sistemas de producción, una de las aplicaciones principales de los simuladores de eventos discretos.

PROBLEMA 6.1: EL PROBLEMA DE LA COLA SIMPLE

Vamos a construir el siguiente modelo simple de servidor:



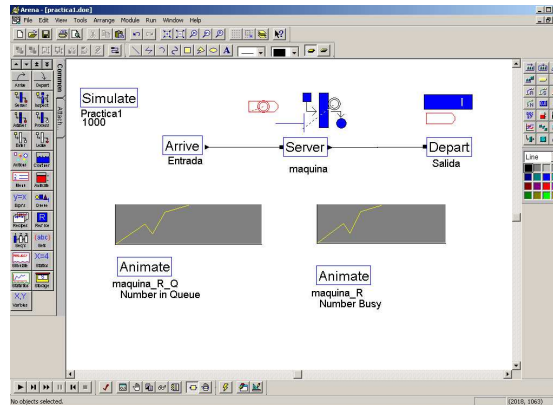
Este servidor representa un sistema de fabricación con las siguientes características.

- Dispone de una única máquina.
- Las piezas llegan con una distribución exponencial de media 5.
- No existe tiempo de transferencia a la máquina.
- El tiempo de procesado tiene una distribución triangular (1, 4,8) minutos.


Se pide:

- Modelar el sistema.
- Poner un contador del número de piezas que salen.
- Mostrar un gráfica de la longitud de la cola y el nivel de ocupación del recurso.
- Hacer una simulación de 15 minutos.

El módulo final debe resultar como se muestra en la figura:



PROCEDIMIENTO

1. Abrir una nueva ventana del modelo. (**File-New**).
2. Si no estuviera el **Common template** añadirlo en la barra de templates (plantillas).
3. Arrastrar el módulo **Arrive**  a la ventana del modelo. Este módulo permite la creación de entidades. Transfiere las entidades de llegada a otro módulo o estación.
4. Hacer doble click sobre el manejador del módulo arrive y rellenar los campos. Hay tres zonas. Salvo que se indique lo contrario, dejar los campos en sus valores por defecto.

Enter Data. Define cómo entran en un módulo las entidades.

Station: Nombre simbólico de la estación asociada al módulo. Tiene un valor por defecto pero se puede cambiar. Llamarla *Entrada*

Arrive Data. Es lo que ocurre durante la acción de salida.

Batch Size: Número de unidades creadas en cada lote, por defecto 1.

First Creation: Tiempo de creación de la primera entidad, por defecto 0.

Time Between: Tiempo entre creación de entidades. Si está en blanco sólo hay una creación. Normalmente se elige entre una distribución. En este caso, escribir un *EXPO(5)*, como indica el enunciado.

Max Batches: Número máximo de lotes a crear.

Mark Time Attribute: Para guardar el instante en el que se produce cada llegada hay que darle un nombre. En este caso, escribir *T_llegada*.

Assign: Cambia otros atributos.

Leave Data. Dónde y cómo van las entidades cuando salen del módulo.

Tran Out: Si el movimiento requiere un medio (camión, AGV, persona, cinta transportadora).

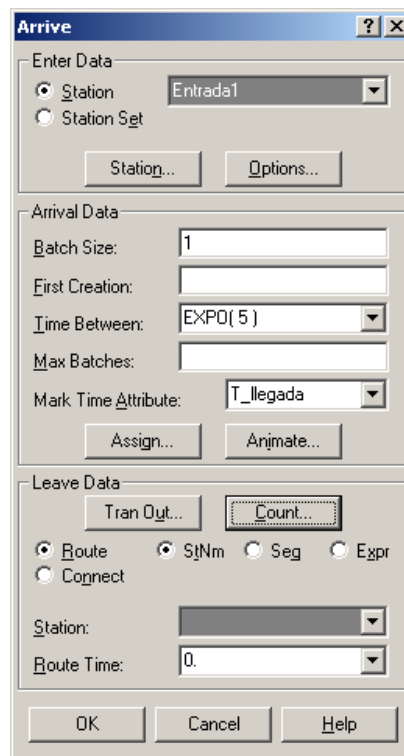
Count: Actualiza un contador por cada entidad que deja el módulo.



Route: Si se requiere tiempo para el movimiento (se muestra en la animación).

Connect: El movimiento a la siguiente estación es instantáneo. *Seleccionarlo.*

Next label: Etiqueta del siguiente módulo al que se va a transferir la entidad.

Debe quedar como se muestra en la figura:



5. Arrastrar el módulo **Server**  a la ventana del modelo. Define una estación donde ocurre un proceso. Las entidades entran en la estación, se apoderan del recurso del servidor, experimentan un proceso de retardo y se transfieren a otra estación o módulo. Algunas características son: representa un recurso, tiene una duración del servicio, dispone de una cola, tiene control de estadísticas, se pueden modelar averías, desconexiones, capacidad variable, disciplinas de cola.
6. Pulsar el icono connect  y unir los dos módulos anteriores, arrive y server.
7. Hacer doble click sobre el manejador del módulo server (la caja con el nombre módulo) y rellenar los campos como se indica:

Enter Data.

Label: Etiqueta asociada. Dejar en blanco.

Station: Se requiere un nombre para la estación (en el menú se ven todos los nombres ya utilizados) asociada con este módulo. Llamarla *maquina*

Tran In: Tiempos de descarga o transferencia de la entidad a este módulo.

Server Data.

Resource: nombre del recurso. Por defecto, sale *maquina_R*

Capacity type: Tipo de capacidad del recurso: capacity (fija) o schedule (que varía en el tiempo de acuerdo a un schedule). Dependiendo de la opción elegida se tendrá uno de los siguientes campos:

Capacity: Capacidad del servidor, número de elementos que sirve cada vez.

Schedule: Nombre del schedule.

Resource Statistic: Si se recogen datos estadísticos del recurso. Marcarlo.

Process Time: Expresión que indica el tiempo de proceso. Como indicaba el enunciado, escribir, *TRIA(1,4,8)*

Botones:

Options: Opciones asociadas con el servidor.

Resource: Acceso a datos de los elementales de recursos.


Queue: Acceso a datos de la cola. Por defecto, el entorno ha denominado a esta variable, *maquina_R_Q*.


Animate: Permite animar la estación.

Leave Data. Igual que el del arrive. Seleccionar *connect*.

The screenshot shows a 'Server' configuration dialog box. It contains the following fields and controls:

- Enter Data:** Label (text box), Station (dropdown menu showing 'maquina'), Tran In... (button).
- Server Data:** Resource (dropdown menu showing 'maquina_R'), Capacity Type (dropdown menu showing 'Capacity'), Capacity (text box with '1'), Resource Statistics (checked checkbox), Process Time (dropdown menu showing 'TRIA(1,4,8)'), Options... (button), Resource... (button), Queue... (button), Animate... (button).
- Leave Data:** Tran Out... (button), Count... (button), Route (radio button), Connect (radio button, selected), Next Label (text box).
- Buttons:** OK, Cancel, Help.

8. Arrastrar el módulo **Depart**  a la ventana del modelo. Se utiliza para recoger estadísticas y eliminar entidades del modelo.

9. Pulsar el icono connect  y unir los dos módulos anteriores, depart y server.

10. Picar en el módulo y rellenar los distintos campos:

Enter data Requiere el nombre de la estación.

Label Nombre de etiqueta. Dejar en blanco.

Station Nombre de la estación. Escribir *salida*.

Count Se puede especificar un contador que se actualiza cada vez que una entidad deja el modelo. Se usa para determinar la producción, etc.

Individual counter: un contador individual. Seleccionarlo.

Counter Set Member: El contador es un miembro de un conjunto de contadores.

None: No hay contador.

Counter: Nombre del contador si se ha seleccionado la opción Individual Counter. Cambiar el nombre a *Produccion*. Esta variable nos indica el número de unidades producidas.

Tally Cálculos estadísticos sobre algunos datos.

Tipo de Tally: Individual tally, tally set member, none. Seleccionar el primero.

Tally: Nombre del tally. Llamarlo *Procesado*.

Type of Statistic: tipo de estadística.

Interval: Intervalo de tiempo sobre algún atributo definido. Se calcula como el instante actual menos dicho atributo. El atributo se indica en el pop-up de debajo.

Between: Tiempo entre salidas.

Expr: Expresión genérica.

Attribute: Elegir T_llegada. De esta forma la cola Procesado nos muestra el tiempo de procesado de una pieza, desde que llega hasta que sale.



11. Arrastrar el módulo **Simulate** a la ventana del modelo. Controla la ejecución de la simulación. Es un módulo de datos, por lo que no está conectado a otros módulos. Especifica el nombre del proyecto, nombre del analista, fecha, el número de replicas de simulación a realizar, el tiempo de comienzo, longitud máxima o condición de terminación de cada replica.

Project:

Title: Escribir *Practica1*.

Analyst: Nombre del analista.

Date: Fecha.

Replicate:

Number of Replication: Número de repeticiones. Cada una se ejecuta hasta (Beginning Time + Length of Replication). Escribir un *1*.

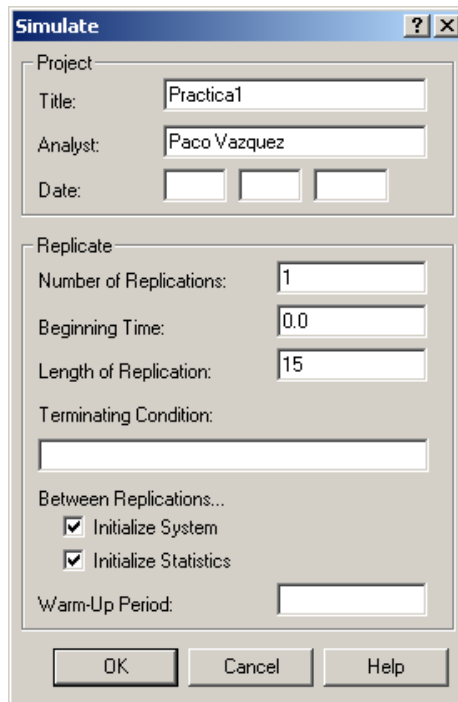
Beginning Time: Tiempo de comienzo de la primera repetición. Escribir un *0*.

Length of Replications: Longitud máxima de repetición. Sirven para detener la simulación. Se especifica un tiempo en **minutos**. Escribir un *15*.

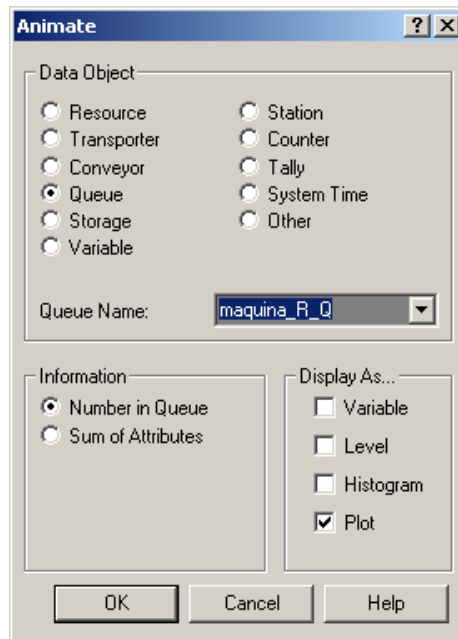
Terminating Condition: Condición de terminación de la replicación.

Initialize System: Determina si el estado del sistema se inicializa entre sucesivas replications. Marcarlo.

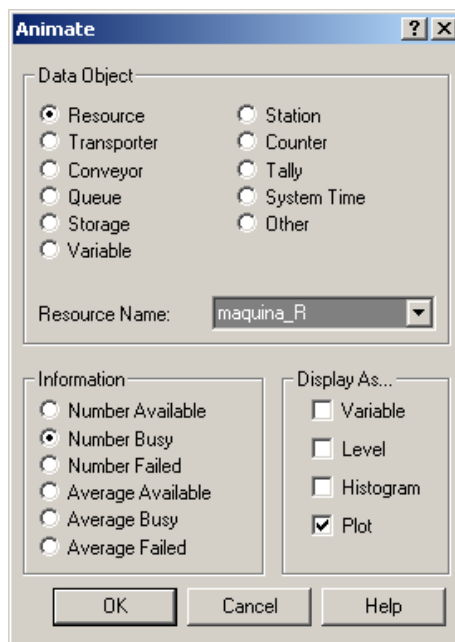
Initialize Statistics: Indica si los acumuladores estadísticos se limpian entre sucesivas replications. Marcarlo.



12. Arrastrar dos módulos **Animate** a la ventana del modelo. Se utiliza para mostrar el estado de la simulación del sistema de forma gráfica. Dibuja la evolución de variables, recursos, colas, etc. de distintas formas: como un gráfico, un nivel, una variable o un histograma.
13. Abrir el primero de ellos. Pulsar en el cuadrado azul que pone Animate. Rellenar los campos tal como se indica en la figura. De esta forma se va a mostrar en una gráfica el número de elementos existentes en la cola, *maquina_R_Q*, creada anteriormente en el módulo server.



14. Con el segundo se procede de la misma forma, como indica la figura, para conseguir mostrar una gráfica



15. Ejecutar el modelo. Para ejecutar el modelo disponemos del menú run o de la barra



de herramientas run. Para ejecutar el modelo tenemos el botón **Run** o la opción **Go** del menú Run. La primera vez que se ejecuta un modelo se cheque y compila. Si no hay errores se entra en el modo de ejecución del modelo (donde no se puede editar). Cuando finaliza la simulación se pregunta si se quiere ver un sumario de resultados (fichero Notepad). Para salir del modo de ejecución, se debe pulsar el botón Stop. Para detener la ejecución, se pulsa el botón pause. Se

puede ejecutar la animación más rápidamente. Se puede volver al estado inicial. Se puede ejecutar paso a paso.

- Viendo los resultados. Cuando finaliza la simulación se muestra un mensaje indicando si se desea ver los resultados. Si se responde afirmativamente se abre el editor Notepad y nos muestra los resultados. Se pueden guardar en un fichero.out.

```

ARENA Simulation Results
. - License #9400000

Summary for Replication 1 of 1

Project: Practical1                      Run execution date : 8/14/2002
Analyst: Paco Vazquez                    Model revision date: 8/14/2002

Replication ended at time      : 15.0

TALLY VARIABLES
Identifier      Average  Half width  Minimum  Maximum  Observations
-----
maquina_R_Q Queue Time .18608    (Insuf)    .00000   .55824   3
Procesado      3.7669   (Insuf)    2.9552   4.5786   2

DISCRETE-CHANGE VARIABLES
Identifier      Average  Half width  Minimum  Maximum  Final Value
-----
# in maquina_R_Q .27271   (Insuf)    .00000   2.0000   2.0000
maquina_R Available 1.0000   (Insuf)    1.0000   1.0000   1.0000
maquina_R Busy .84926   (Insuf)    .00000   1.0000   1.0000

COUNTERS
Identifier      Count  Limit
-----
Produccion      2     Infinite

Simulation run time: 0.00 minutes.
Simulation run complete.
  
```

- Los resultados tienen tres partes:

Variables Tally. Contienen estadísticas de tiempo de ciclo y tiempo de cola.

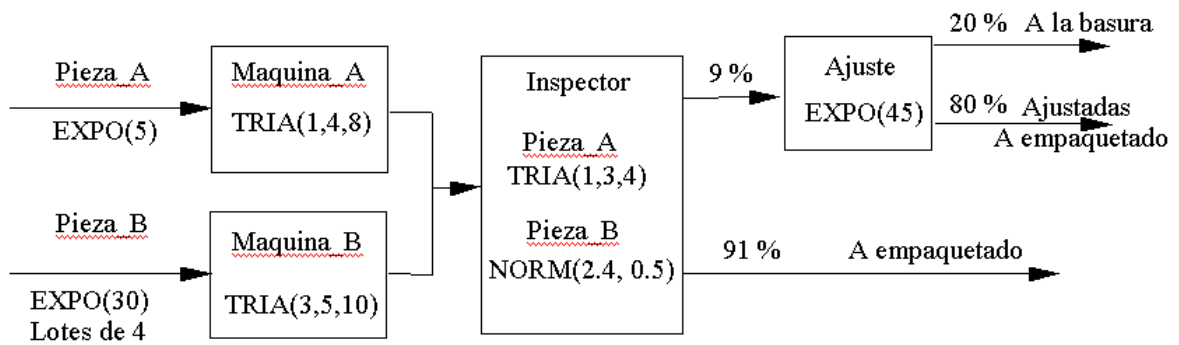
Variables de cambio discreto. Contiene estadísticas de utilización de recursos (Busy), disponibilidad de recursos y número en colas.

Contadores. Son variables contadoras.

- Modificar el sistema para que el tiempo de procesado sea una triangular (10,14,18). Observar cómo se incremente la cola de la entrada.
- Sustituir las conexiones por un enrutado. Para ello elegir Route en el Leave Data de los dos primero módulos, debiendo escribir tanto la estación a la que va conectado, como el tiempo de transporte entre ambas. Observar cómo se eliminan de forma automática las conexiones.

PROBLEMA 6.2: MODELO DE ENSAMBLAJE

Se va a modelar el siguiente sistema de ensamble:



Se trata de un sistema donde llegan dos tipos de piezas, A y B (que son placas donde se van a insertar componentes). Estas piezas se preparan en diferentes áreas: maquina_A y maquina_B. Una vez preparadas ambas piezas van al puesto de ensamble y control de calidad (inspector). Del inspector pueden ir directamente a empaquetar o se vuelven a reprocesar si se detecta un error. Finalmente las piezas reprocesadas se pueden aceptar de nuevo o eliminar. El tiempo de tránsito entre todas las estaciones es de 2 minutos.

Se pide:

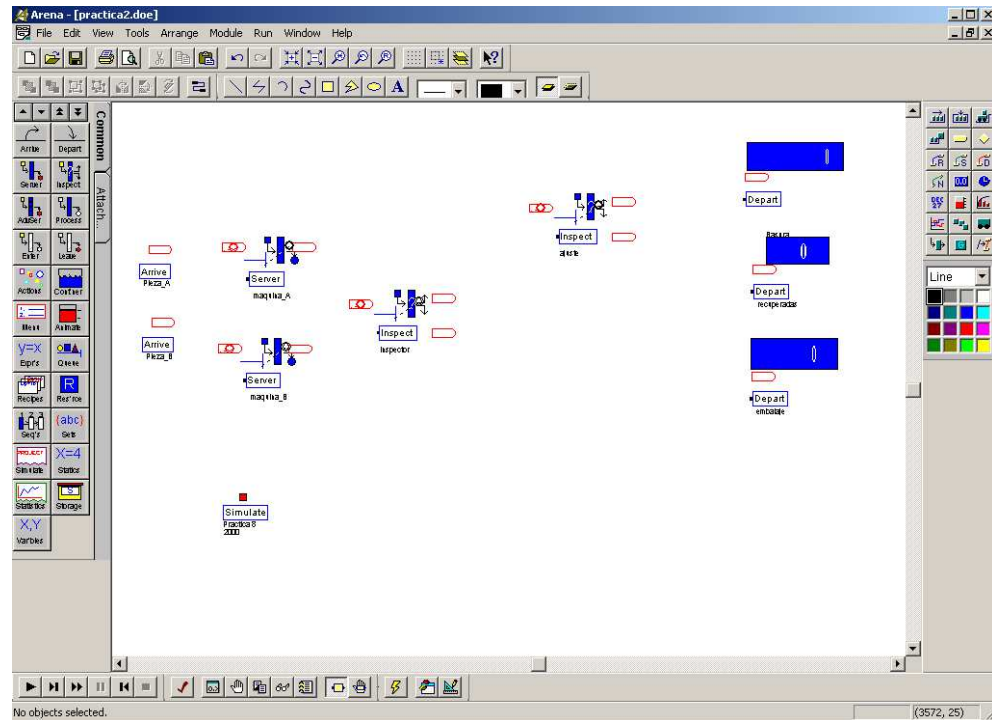
1. Construir el modelo del sistema anterior.
2. Simular el sistema durante 2.000 minutos.
3. Obtener las siguientes salidas:
 - Utilización de todos los recursos.
 - Número en cada cola.
 - Tiempo en cada cola.
 - Tiempo de ciclo de las para piezas empaquetadas, ajustadas/empaquetadas y tiradas.
4. Obtener las siguientes animaciones
 - Colas, recursos ocupados/desocupados
 - Movimiento entre estaciones.

Para construir este modelo necesitamos:

- Dos módulos **arrive**, uno para cada tipo de pieza.
- Dos módulos **server** distintos para procesado.
- Dos módulos **inspect**, uno para ensamble (inspector) y otro para reprocesado (ajuste).
- Tres módulos **depart**, uno para empaquetado, otro para ajustadas/empaquetadas y otro para tirar.

- El tiempo de transferencia se especifica con Route en lugar de Connect en el Leave Data de los módulos y con tiempo 2 minutos.
- El tiempo de procesado de ensamblaje al ser distinto para cada pieza, se asigna en una variable (T_ajuste) en el modulo Arrive de cada pieza.

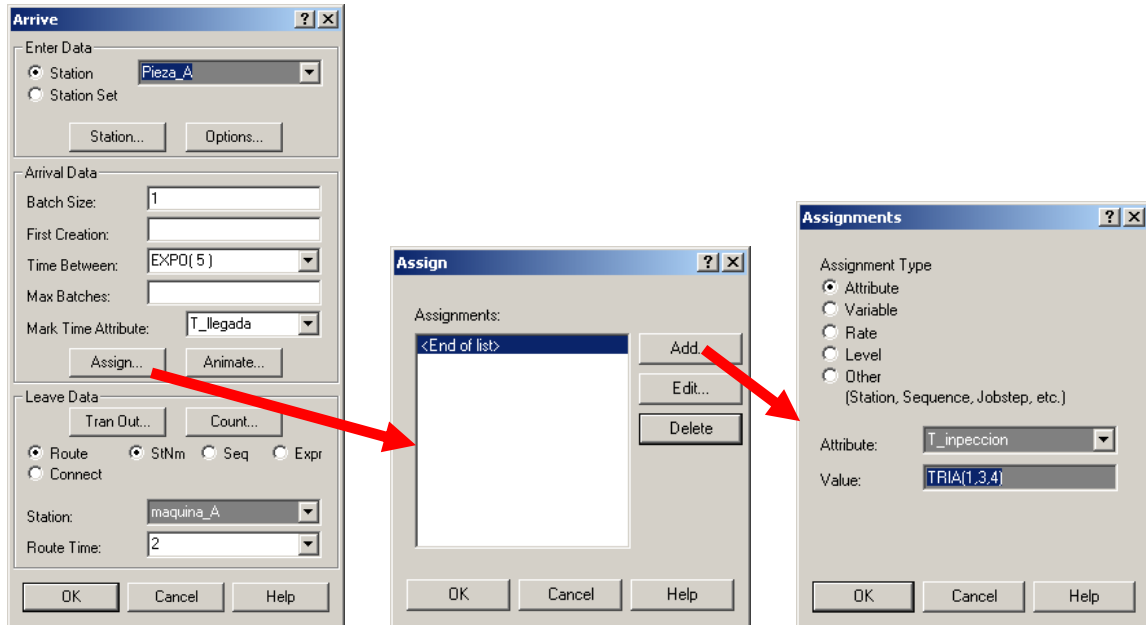
El aspecto del modelo terminado sería:



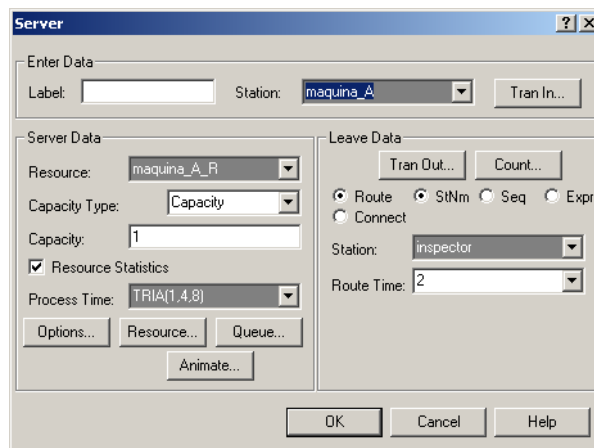
PROCEDIMIENTO

1. Arrastrar primero todos los módulos y ponerles su nombre. Llamar a los dos **arrive** Pieza_A y Pieza_B, a los **server**, maquina_A y maquina_B, a los **depart**, basura, recuperadas y embalaje. Abrir cada módulo **inspect**. llamando al primero inspector y al segundo ajuste. Notar que ara cerrar estos módulos es necesario escribir un valor en el campo **Failure Probability**. Escribir un 0.8 de momento, y cerrarlos.
2. Abrir y configurar los módulos **Arrive** como en la práctica anterior, teniendo en cuenta las características actuales, es decir, incluyendo un tiempo de enrutado de 2, y eligiendo la estación de destino adecuada para cada uno. En **Mark Time Attribute**, incluir el nombre de la variable que almacena el instante de llegada de cada evento, en este caso, T_llegada.
3. Para asignar a cada una de las piezas que llegan el tiempo que tardará posteriormente en el bloque del inspector se puede hacer mediante un atributo. Para realizar esto, abrir

las ventanas del botón **assign** de cada **arrive** y asignar el atributo correspondiente para cada una de las dos piezas, es decir, TRIA(1,3,4) y NORM(2.4, 0.5). Llamar al atributo T_inspeccion. La asignación de un tiempo aleatorio, con dichas distribuciones, será tenido en cuenta posteriormente.

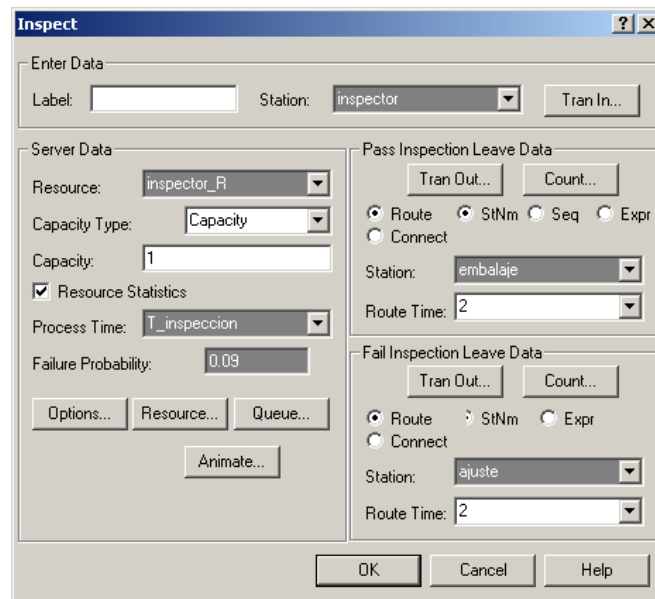


4. Abrir el primer módulo **server**, llamado maquina_A. Rellenar los campos necesarios como indica el enunciado. En la figura siguiente se muestra cómo queda. Hacer lo correspondiente con la maquina_B.

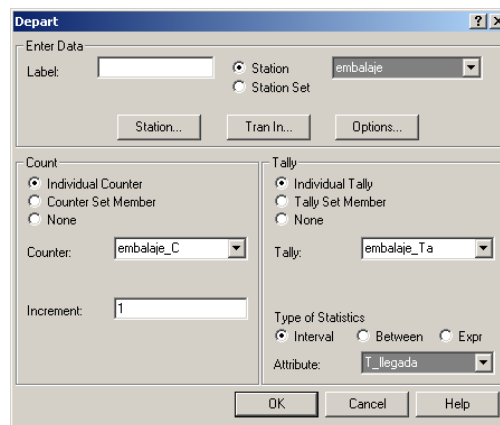


5. Abrir el módulo primer **Inspect**, llamado inspector. Es similar al módulo Server excepto en que indica la fracción de entidades que pasan o no la inspección y permite especificar diferentes destinos para las entidades que pasan y para las que no pasan la inspección. En **Failure Probability** escribir un 0.09, que indica que un 9% de las piezas no pasan la inspección, como indica el enunciado. En **Process Time** hay que indicar cuánto tiempo tarda en efectuar la inspección. Recordar que este tiempo ha sido previamente almacenado en una variable, configurada en el módulo **arrive**. La sección de salida del

inspector tiene dos partes. La primera para configurar dónde y cómo van las piezas que pasan la inspección, y la segunda para las que no la pasan. El módulo queda:



6. Configurar el módulo **inspect** denominado ajuste de la misma forma.
7. Abrir el módulo de salida denominado embalaje. Configurararlo como indica la figura siguiente. Se ha generado una variable, embalaje_C que cuenta las piezas que salen, otra variable de cola, embalaje_Ta que guarda el tiempo de de procesamiento desde que la pieza entró (por eso hay que escribir el campo inferior como T_llegada, configurada en el bloque de entrada del sistema).



8. De la misma forma, configurar los otros dos módulos de salida.
9. Incluir un bloque de simulación, con una longitud de repetición de 2000.
10. Simular. El resultado tiene que se el siguiente:

practica2.out - Bloc de notas

Archivo Edición Formato Ayuda

Summary for Replication 1 of 1

Project: Practica 8 Run execution date : 8/16/2002
Analyst: Paco Vazquez Model revision date: 8/16/2002

Replication ended at time : 2000.0

TALLY VARIABLES

Identifiser	Average	Half width	Minimum	Maximum	Observations
recuperadas-Ta	540.48	(Insuf)	101.91	874.90	31
embalaje-Ta	35.911	(Corr)	9.7800	117.85	548
maquina_A_R_Q Queue Ti	6.0259	(Corr)	.00000	28.804	365
ajuste_R_Q Queue Time	429.95	(Insuf)	.00000	782.19	36
maquina_B_R_Q Queue Ti	42.308	(Insuf)	.00000	105.34	240
inspector_R_Q Queue Ti	1.9247	(Corr)	.00000	12.850	603
Basura-Ta	334.79	(Insuf)	76.343	821.27	4



DISCRETE-CHANGE VARIABLES

Identifiser	Average	Half width	Minimum	Maximum	Final value
maquina_A_R Busy	.79647	(Corr)	.00000	1.0000	1.0000
ajuste_R Busy	.99029	(Insuf)	.00000	1.0000	1.0000
# in ajuste_R_Q	11.615	(Insuf)	.00000	22.000	19.000
inspector_R Busy	.77139	(Corr)	.00000	1.0000	.00000
# in maquina_B_R_Q	5.0770	(Insuf)	.00000	19.000	.00000
# in maquina_A_R_Q	1.0999	(Corr)	.00000	6.0000	2.0000
maquina_B_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
maquina_A_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
ajuste_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
inspector_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
# in inspector_R_Q	.58031	(Corr)	.00000	4.0000	.00000
maquina_B_R Busy	.72774	(Insuf)	.00000	1.0000	.00000

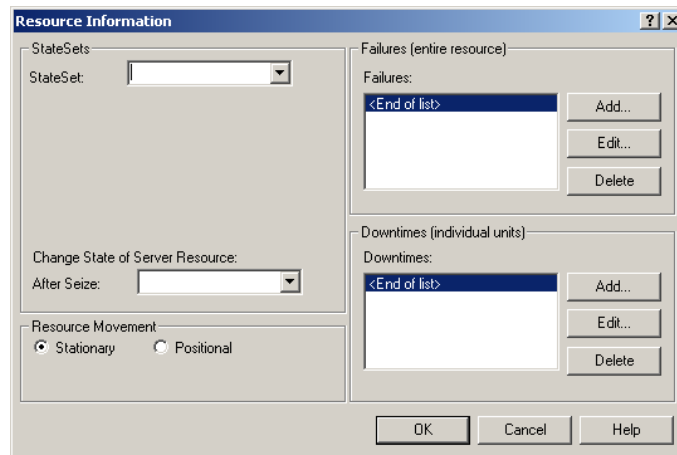
COUNTERS

Identifiser	Count	Limit
embalaje_C	548	Infinite
recuperadas_C	31	Infinite
Basura_C	4	Infinite

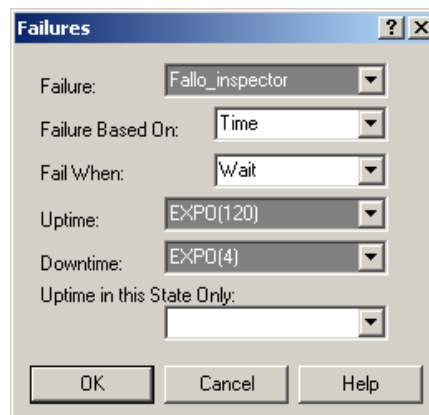
Simulation run time: 0.03 minutes.
Simulation run complete.

- El siguiente proceso consistirá en incluir animaciones en la simulación, y en concreto en las rutas.
- Sólo hay que animar las rutas y no conectar los módulos, ya que se ha indicado Route en lugar de **connect** y el origen y destino de la ruta se ha especificado utilizando el campo **Station** de los **Enter** y **Leave Data**, que indican respectivamente la estación origen y la estación destino de la ruta. Para animar una ruta, pulsar sobre el botón route  del Animate toolbar. El cursor cambia a una cruz. Conectar el cuadrado rojo del módulo inicial  con el cuadrado rojo del módulo final. Repetir con todos los módulos a conectar.
- Volver a simular, observando la evolución de las piezas. La velocidad de simulación puede aumentarse o disminuirse mediante los signos > o <. Observar la entrada de las piezas B, que llegan en paquetes de cuatro, pero que se procesan individualmente (la cola crece en cuatro unidades).
- A continuación se van a realizar algunas modificaciones en el modelo de ensamblaje.

- a. Como uno de los cuellos de botella se encuentra en el ajuste, dicha estación tendrá dos turnos. El primero igual que antes, pero el segundo tendrá dos operadores. Cada turno es de 480 minutos.
 - b. Un modelo más realista supone que el puesto de inspección falla a veces. Las observaciones indican que el tiempo entre fallos es de unos 120 minutos y el tiempo de reparación a una exponencial de media 4 minutos.
 - c. Almacenar las siguientes estadísticas: el número de piezas en las colas de maquina_B, inspector y ajuste, el tiempo del ciclo de todas las piezas que no pasan por ajuste y el porcentaje de tiempo que la cola ajuste es de longitud 0, entre 0 y 10 y entre 10 y 20.
15. Para poder realizar las prestaciones anteriores necesitamos utilizar nuevas características de SIMAN, como son la planificación (schedules), el estados de los recursos (Resource States), los fallos en un recurso (Resource Failures) y almacenar datos estadísticos (Statistics).
16. Planificación (schedules): Permiten modificar la capacidad (capacity) o número de unidades de un recurso a lo largo del tiempo. Se puede modificar esta opción en los módulos server, inspect y process. En nuestro caso se va a modificar el número de empleados por turno. Pulsamos sobre el módulo **Inspect** denominado ajuste. En **Server Data**, modificar el campo **Capacity Type**, sustituyendo Schedule en lugar de Capacity.
17. Lo anterior implica que hemos de poner nombre a la planificación de los turnos, en este caso en **Schedule** escribir Turno_ajuste. Elegir Ignore de entre Preempt/Ignore/Wait, que indica qué ocurre si el recurso está ocupado cuando queda fuera de servicio. Además hay que definir estos turnos. Para ello, pulsar en el botón **schedule** y aparece el subdiálogo schedule. Pulsar **add**, para añadir el par capacidad-duración. **Capacity: 1, Duration: 480** para el primer turno y **Capacity: 2 Duration: 480** para el segundo.
18. Fallos de recursos (Resource Failures). Una unidad de recurso puede ponerse fuera de servicio. En este caso se produce en el módulo de inspección. Abrir dicho módulo. Pulsar el botón **Resource** y aparece el diálogo de Resource Information.



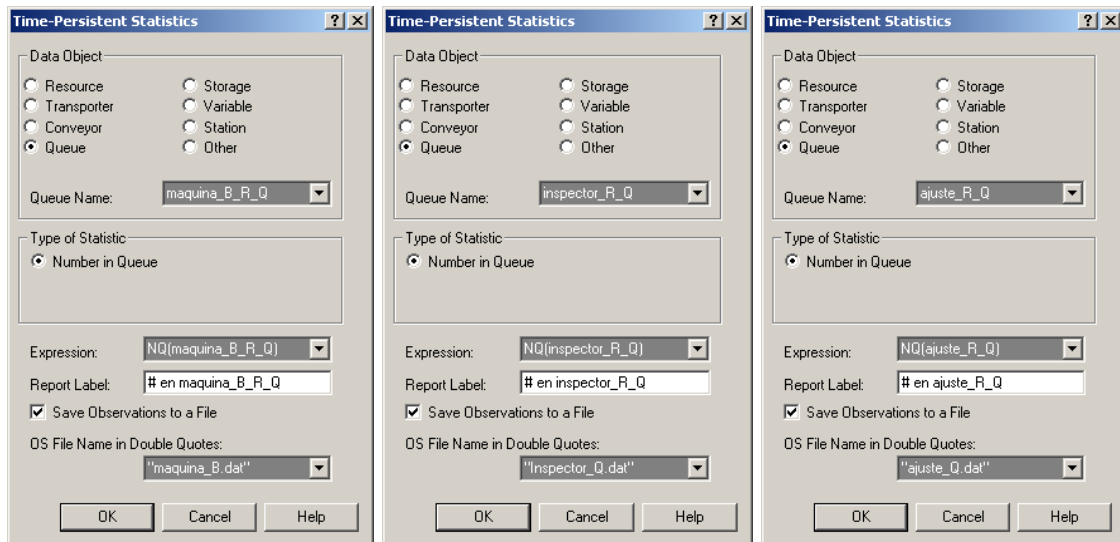
19. Esta ventana tiene tres secciones. La primera es para definir los estados. La diferencia entre downtimes y failures se encuentra en cómo el fallo afecta a la unidad, si lo hace a una de sus unidades o a todas las del recurso. Cuando la capacidad del recurso es la unidad, no hay diferencia en ambas. Añadir en Failures un fallo denominado Fallo_inspector, basado en tiempo, y con las características antes comentadas, y tal como s muestra en la figura.



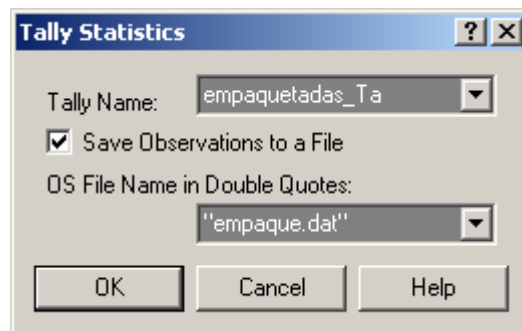
20. Almacenado datos estadísticos. Los datos estadísticos se recogen durante la simulación y se pueden almacenar luego en un fichero para utilizarlo con el **Analizador de Salidas (Output Analyzer)**. Para ello utilizamos el módulo estadístico desde donde se pueden recopilar distintos tipos de estadísticas.



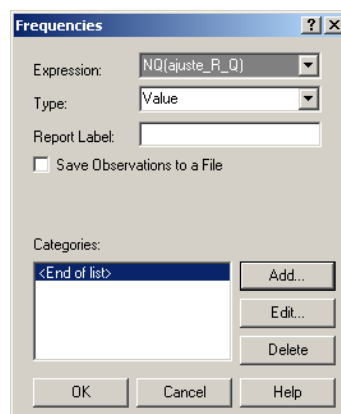
21. Arrastramos el **Módulo Statictics**. Pulsamos sobre él para que aparezca el diálogo **Statictics**. Pulsamos sobre el botón **Add** de **Time-Persistent** y se abre el diálogo de estadísticas de este tipo. Rellenar los siguientes datos para calcular las piezas en la cola de la maquina_B, del inspector y del ajuste:



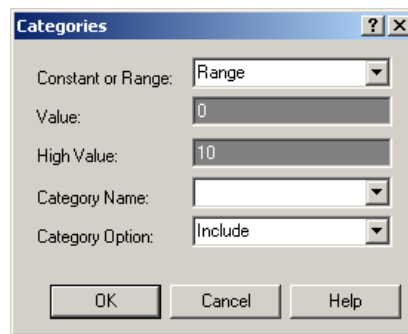
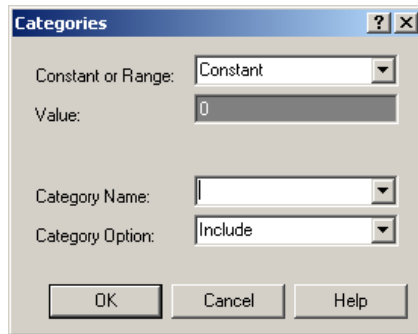
22. Pulsamos sobre el botón **Add** de **Tally** y se abre el diálogo de estadísticas de Tally. Rellenar los siguientes datos para calcular el tiempo de ciclo de las piezas que no pasan por ajuste:



23. Pulsamos sobre el botón **Add** de **Frequencies** y se abre el diálogo de estadísticas de Tally. Rellenar los siguientes datos para calcular el porcentaje de tiempo que la cola ajuste es de tamaño 0, 0-10, 10-20, etc:



24. Se pulsa el **botón add de categories**, Para añadir el primero de los tres rangos que nos interesan. El de tamaño 0. Luego los distintos rangos, hasta 40-50.



25. Cambiar la simulación a 50.000 minutos y ejecutarla.

26. El resultado es

Identifier	Average	Half width	Minimum	Maximum	observer
recuperadas-Ta	302.17	(Corr)	19.439	1229.2	1199
embalaje-Ta	57.472	(Corr)	9.0197	564.91	15031
maquina_A_R_Q queue T1	15.183	2.5646	.00000	99.536	9876
empaquetadas-Ta					0
ajuste_R_Q Queue Time	193.93	48.888	.00000	674.58	1467
maquina_B_R_Q queue T1	70.972	(Corr)	.00000	541.65	6624
inspector_R_Q queue T1	6.2386	1.5828	.00000	50.795	16499
Basura-Ta	285.39	(Insuf)	25.366	1159.2	267

DISCRETE-CHANGE VARIABLES					
Identifier	Average	Half width	Minimum	Maximum	Final Value
maquina_A_R Busy	.85391	.02036	.00000	1.0000	1.0000
ajuste_R Busy	1.3732	.06088	.00000	2.0000	1.0000
# in ajuste_R_Q	5.6899	1.6284	.00000	28.0000	.00000
inspector_R Busy	.84177	.02003	.00000	1.0000	.00000
# in maquina_B_R_Q	9.4023	(Corr)	.00000	88.0000	.00000
# in maquina_A_R_Q	2.9990	.61504	.00000	24.0000	1.0000
maquina_B_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
maquina_A_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
ajuste_R Available	1.4992	(Insuf)	1.0000	2.0000	1.0000
inspector_R Available	1.0000	(Insuf)	1.0000	1.0000	1.0000
# in inspector_R_Q	2.0586	.59611	.00000	21.0000	.00000
maquina_B_R Busy	.79305	.04708	.00000	1.0000	.00000

COUNTERS			
Identifier	Count	Limit	
embalaje_C	15031	Infinite	
recuperadas_C	1199	Infinite	
Basura_C	267	Infinite	

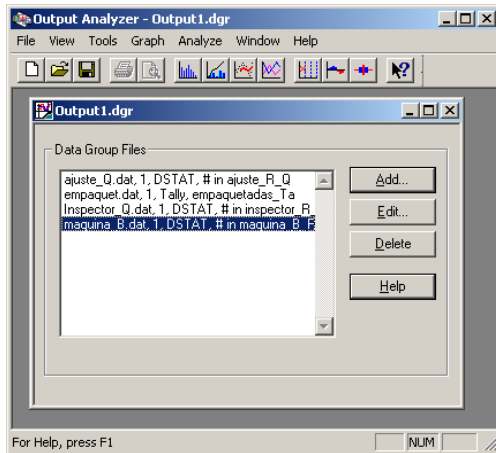
FREQUENCIES					
Identifier	Category	--Occurrences-- Number	AvgTime	Standard Percent	Restrict Percent
STATE(inspector_R)	BUSY	3635	11.578	84.18	84.18
	IDLE	3339	1.9188	12.81	12.81
	FAILED	387	3.8871	3.01	3.01
Value(NQ(AJUSTE_R_Q))	Constant(0,)	112	81.559	18.27	18.27
	Range(0,10)	163	197.97	64.54	64.54
	Range(10,20)	63	120.45	15.18	15.18

27. Observar las estadísticas de área de ajuste. El nº máximo en cola fue de 28, y estuvo vacía sólo al principio de la ejecución. La disponibilidad del ajuste fue de 1.4992 y estuvo ocupado un 1.3732. Esto refleja que su utilización real fue de un 91.6 % ($100 \times 1.3732 / 1.4992$).

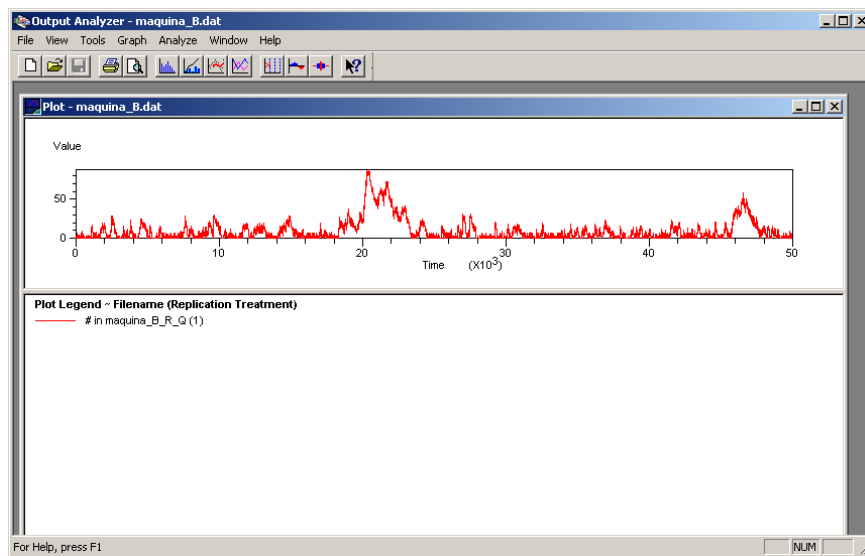
28. Las estadísticas del inspector se muestran más abajo, en frecuencias, así como el número de piezas en cola del ajuste agrupadas por rangos.

29. El analizador de las salidas, proporciona la posibilidad de ejecutar un análisis posterior a la ejecución de una simulación, cuyos datos han sido guardados en ficheros. Normalmente se interactúa con el analizador de salidas definiendo grupos, que es un conjunto de ficheros que proceden de la misma simulación. Abrir esta herramienta en Tool/Output analyzer.

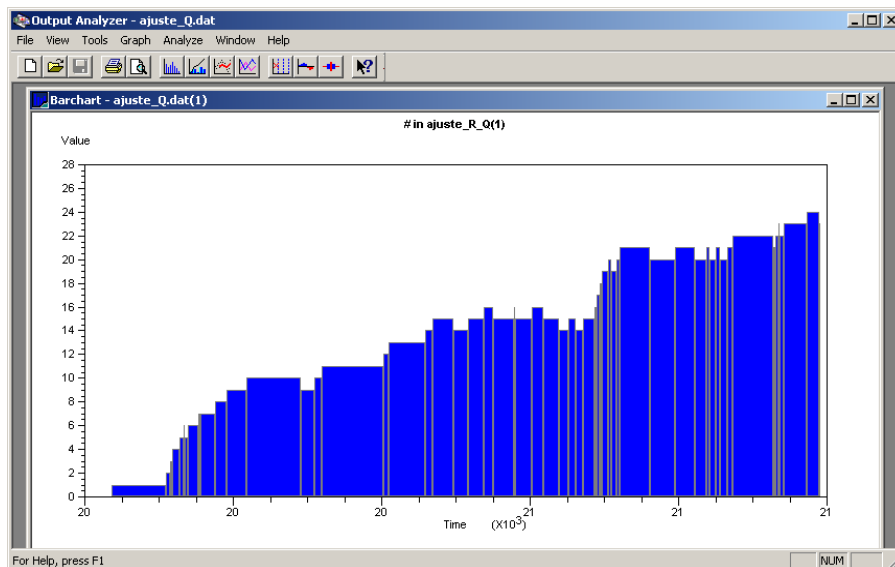
30. Con file/new crear un grupo con los cuatro ficheros almacenados anteriormente.



31. Ahora se pueden hacer gráficas, por ejemplo de las piezas en cola de la máquina B, con el icono plot de la barra de herramientas.



32. Dibujar un diagrama de barras de las piezas en la cola de ajuste, entre los instantes 20000 y 21000. El resultado será:



33. Dibujar un histograma de las piezas en la cola de ajuste, con un nº de celdas de 3 y un límite inferior de 0.5. El resultado será:

