

Capítulo 4

Tratamiento de los resultados

4.1. Introducción

Generalmente, el mayor esfuerzo a realizar en un proyecto de simulación se centra en la implementación del modelo. De hecho, podemos considerar el desarrollo de un simulador como un ejercicio más o menos complicado de programación. Sin embargo, un proyecto de simulación no finaliza con la implementación del simulador. Todo proyecto de simulación debe incluir además una etapa de *experimentación*, la cual consiste en el tratamiento estadístico de los datos de salida del simulador. Recordemos que estos datos están directamente relacionados con los *objetivos* del modelo de simulación. Sin esta fase experimental, las conclusiones que extraigamos carecerán de fundamento y posiblemente sean erróneas.

Desafortunadamente, sobre los datos de salida de un simulador no podemos aplicar directamente las técnicas clásicas de análisis estadístico, debido principalmente a que estas técnicas se basan en que las muestras sean **IID** (Independientes e Identicamente Distribuidas). Como veremos, los datos de salida de un simulador suelen contener fuertes correlaciones y sesgos, y por lo tanto no se cumple esta propiedad.

Pero antes de tratar los resultados de salida del simulador, debemos asegurarnos de que éste sea correcto y válido con respecto al modelo del sistema. El proceso de *validación* consiste en comprobar que las aproximaciones realizadas en el modelo de simulación se ajustan lo más posible a la realidad. Por otro lado, el proceso de *verificación* consiste en comprobar que dichas aproximaciones se implementan de forma correcta. En el proceso de validación deben intervenir expertos en el sistema real que avalen la proximidad de los

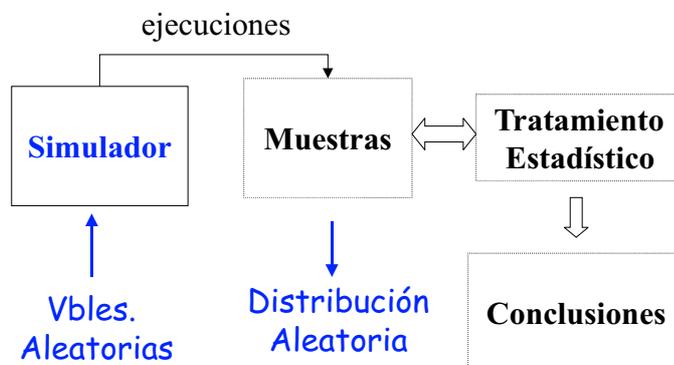


Figura 4.1: Tratamiento de resultados

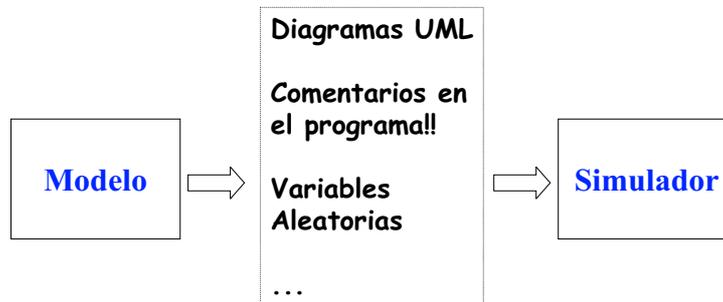


Figura 4.2: Herramientas de validación de modelos

resultados obtenidos con el modelo de simulación y el comportamiento del sistema real. En el proceso de verificación entra en juego todas las técnicas de depuración y verificación de programas que hemos visto a lo largo de la carrera.

4.2. Validación del modelo

Validar el modelo supone comprobar que las asunciones hechas para simplificar el funcionamiento del sistema son aceptables. En ese caso, y si el modelo se implementa correctamente, los resultados del simulador y del sistema real deberían ser parecidos.

Para validar el modelo, se deben comprobar tres aspectos del mismo: las asunciones que simplifican el sistema real, los parámetros de entrada del simulador y las distribuciones utilizadas, y por último los valores obtenidos y las conclusiones que se pueden extraer.

Estos tres aspectos deberían contrastarse con tres fuentes de información: la opinión de expertos en sistemas como el que se está simulando, mediciones sobre el sistema real y datos obtenidos con otros tipos de modelos.

En realidad se debería contrastar los tres aspectos a validar con las tres fuentes de información, aunque esto generalmente no es posible, ya que se suele utilizar técnicas de simulación cuando no se dispone de otras posibilidades. En cambio, suele ser posible la comparación con algunas de las fuentes de información para casos simplificados.

4.3. Verificación o depurado del modelo

Se trata de comprobar que el modelo ha sido implementado correctamente en el programa informático. Para verificar la implementación del modelo se pueden utilizar técnicas típicas del desarrollo de software. Además, existen otras técnicas más específicas para programas de simulación. Destacaremos las siguientes:

Diseño jerárquico y modular. Los modelos de simulación pueden resultar en proyectos software bastante complejos. Por tanto las técnicas generales de desarrollo de software deben ser utilizadas. Si el modelo de simulación se divide en módulos, éstos podrán ser comprobados independientemente. Además se puede modificar el funcionamiento de un módulo sin afectar al resto del programa. Por otro lado, los módulos son más fáciles de verificar, ya que son partes más pequeñas. En la división del modelo en módulos debe quedar clara las variables y funciones (la interfaz) que conecta a los módulos. El diseño jerárquico supone que para desarrollar cada módulo, se identifique una estructura en la que el problema se divide en subproblemas más sencillos. De esta forma se llegaría a partes cuya verificación sería directa. Para aplicar estos principios muchas bibliotecas de simulación se basan en la metodología orientada a objetos.

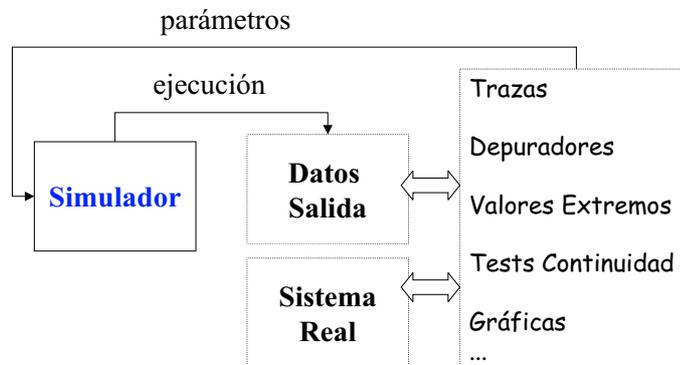


Figura 4.3: Verificación de simuladores

Comprobaciones en el programa. Consiste en poner comprobaciones en el programa que podrán detectar ciertos tipos de error. Por ejemplo, comprobar que una variable supera un valor imposible (por ejemplo, una probabilidad mayor que 1) o que la diferencia entre clientes creados y destruidos es incoherente.

Exposición del programa. Una manera de depurar el programa consiste en explicarlo detalladamente (línea a línea) a una pequeña audiencia. Simplemente el hecho de tener que ordenar las ideas para exponerlas hace que el programador detecte errores. Además, la audiencia también puede detectar errores o aportar sugerencias.

Modelos determinísticos. El hecho de utilizar variables aleatorias en los programas difulta la detección de errores porque su comportamiento es menos predecible. Una posibilidad es estudiar el funcionamiento sustituyendo las variables aleatorias por comportamientos deterministas (por ejemplo sustituir una distribución aleatoria por la media de la distribución). De esta manera, los resultados son más fáciles de calcular y deben coincidir con el resultado de la simulación.

Ejecución de casos simples. Para encontrar respuestas previsibles y comprobar que coinciden con el resultado de la simulación se puede simplificar los casos a simular. Por ejemplo, ver la respuesta si se simula un único cliente, o reducir el número de servidores en una red. Esto no asegura que el simulador funcione bien en el caso general, pero muchos errores ya se producen en simulaciones muy simplificadas.

Estudio de trazas. En una traza se puede ver la secuencia de cambios en el simulador de forma ordenada temporalmente (en tiempo de la simulación). Las trazas permiten hacer un seguimiento de la simulación y entender los cambios que se producen y los valores que toman las variables de estado. Las bibliotecas de simulación pueden estar preparadas para generar trazas. En otro caso, el programador puede introducirlas en el programa, pero debería ser muy fácil de activar o desactivar su funcionamiento. Una vez que el programa ha sido depurado ya no se utilizan trazas. Las trazas pueden tener diferente nivel de información. Por ejemplo se puede incluir el tipo de evento y el tiempo en que se crea y procesa o también se puede indicar el cliente asociado, los valores de variables de estado modificadas, etc. Otra posibilidad es presentar en la traza solamente información de algún tipo de cliente particular o de algunas partes del sistema simulado. En ocasiones es interesante mostrar gráficamente algún parámetro de la simulación al mismo tiempo que ésta se desarrolla. En este caso

se dispone de menor información que con una traza pero puede ser fácil detectar que la simulación es errónea.

Test de continuidad. Se basan en que pequeños cambios en la entrada de las simulaciones deben traducirse en pequeños cambios en la salida (generalmente con un tendencia coherente respecto a la modificación introducida). Si el efecto en la salida es brusco o no sigue la tendencia esperada, puede haber errores en el programa.

Comprobación de casos extremos. Consiste en comprobar los casos extremos que se consideran en la simulación. Estos casos extremos incluirían tanto los más sencillos de configuración o de carga como los más complicados. Las pruebas pueden no representar casos de funcionamiento típico en el sistema real pero ayudan a detectar errores y situaciones no contempladas en el programa.

Test de consistencia. Consiste en modificar un conjunto de parámetros de la simulación de manera que el resultado deba ser equivalente. Por ejemplo, en una simulación cambiar un servidor por dos servidores idénticos con la mitad de potencia que el inicial. El resultado de la simulación debería ser equivalente. Las semillas de los generadores de números aleatorios tampoco deberían variar el resultado de la simulación. De todas formas esto podría detectarse si se replican los experimentos de simulación y se calcula un intervalo de confianza para los resultados. Los conjuntos de pruebas que se han utilizado para probar un simulador es conveniente almacenarlos con sus resultados. Así se pueden volver a aplicar cada vez que se modifica el simulador.

4.4. Eliminación del transitorio

Una vez validado y verificado el simulador, para extraer los resultados de las simulaciones, surgen otras cuestiones: si la simulación no arranca de un estado estable, ¿en qué momento se alcanza el estado estable? Esta cuestión es interesante porque necesitamos estar seguros de tomar datos en un funcionamiento representativo del sistema. A esta operación se la conoce como *eliminación del transitorio*. Para tener suficientes datos de una simulación, también se debería evaluar la longitud de la simulación. Si es demasiado corta, los datos pueden tener una gran variabilidad y se podrían sacar conclusiones erróneas. Si la simulación es demasiado larga, se puede perder demasiado tiempo en las simulaciones, sin que se aumente la calidad de los resultados. Veamos en primer lugar el problema del transitorio.

En la figura 4.4 se muestra la evolución de la salida de un simulador para distintas ejecuciones (cada curva es una ejecución distinta).

Como puede observarse, para cada ejecución podemos identificar un periodo transitorio, también denominado de calentamiento, producido por la situación inicial del simulador. En muchas ocasiones, este periodo transitorio se produce porque las colas de una red de estaciones están inicialmente vacías. Tras este periodo de transición, el simulador puede llegar a un estado estable (y representativo del sistema), y la varianza de los resultados entre las distintas ejecuciones se reduce notablemente.

Nótese que este periodo transitorio produce una distorsión muy importante en las medidas globales de la salida, tanto en la media como en la varianza, sobre todo si la simulación es demasiado corta. Por consiguiente, es conveniente eliminar de alguna forma este periodo transitorio a la hora de realizar el tratamiento de los resultados.

No hay un método exacto para eliminar el transitorio. Se utilizan diferentes métodos aproximados, los cuales normalmente se basan en que la varianza de los resultados es menor cuando se alcanza el funcionamiento normal que durante el transitorio (esto es cierto si el sistema simulado es estable pero no se cumple en sistemas inestables).

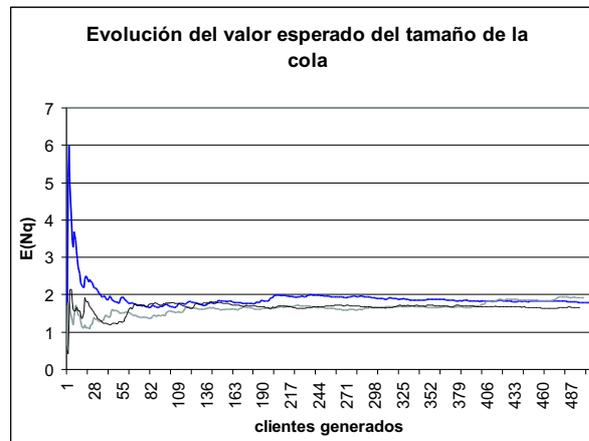


Figura 4.4: Periodo transitorio

4.4.1. Alargar la simulación

Una técnica comúnmente utilizada consiste en alargar la simulación más de lo necesario. De esta forma la influencia de los datos del transitorio se diluye al tomar un mayor número de datos.

Esta forma de actuar tiene dos inconvenientes. Por un lado se desperdicia tiempo y recursos porque la simulación es más larga de lo necesario y por otro no se tiene garantía de estar actuando bien, porque nada indica que se han tomado suficientes datos.

4.4.2. Seleccionar un buen estado inicial

Si se encuentra un estado típico del régimen estable de la simulación, se puede programar directamente ese estado como el inicio de la simulación. En ese caso no sería necesario eliminar el transitorio. El estado inicial representativo se podría buscar a partir de algunas simulaciones previas.

4.4.3. Método del truncado

Como los valores en estado estable oscilarán menos que durante el periodo inicial, una forma de estimar el transitorio es ir comprobando alguna variable de la simulación hasta que el valor de esta variable no sea ni un máximo ni un mínimo del conjunto de valores muestreados.

Ejemplos de variables que se pueden utilizar en un sistema estable, podrían ser el tiempo de respuesta de los clientes o la población media que hay en el sistema. Inicialmente el sistema estará vacío y el número de clientes en el sistema irá aumentando hasta quedar oscilando en un conjunto de valores (la media de ese conjunto será el número medio de clientes en el sistema en el funcionamiento estable; en cambio si el sistema es inestable, el transitorio no terminaría nunca).

4.4.4. Impacto de eliminar datos iniciales

Este método consiste en eliminar datos iniciales hasta que se establezca la media de los resultados. Es decir, que se va calculando la media de los resultados al eliminar las i primeras muestras (empezando por $i = 0$). Llegaremos a un valor de i a partir del cual la media calculada del resultado no se modifica.

Como la media del resultado de una simulación tiene variaciones incluso en estado estable, se utiliza la media de varias réplicas de la simulación en las que lo único que cambia son las semillas de los generadores de variables aleatorias.

Supongamos que se hacen m simulaciones y en cada simulación se toman n muestras. x_{ij} es la muestra j de la simulación i . Entonces podemos seguir el siguiente algoritmo:

1. La trayectoria media: $\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij}$ para $j = 1, 2, \dots, n$
2. Media total: $\bar{x} = \frac{1}{n} \sum_{j=1}^n \bar{x}_j$
3. Desde $l = 1$ calculamos las medias de los $n - l$ valores: $\bar{x}_l = \frac{1}{n-l} \sum_{j=l+1}^n \bar{x}_j$
4. Calcular el cambio relativo al eliminar l valores: $\frac{\bar{x}_l - \bar{x}}{\bar{x}}$
5. Repetir los pasos de 3 a 4 variando l desde 1 a $n - 1$. Si se va representando el cambio relativo en función de l , se observará que a partir de un valor de l se estabiliza el cambio relativo.

Un inconveniente de este método es que se descarta el transitorio de varias simulaciones.

4.4.5. Medias Batch

En principio es la misma idea que el método anterior. En cambio, en vez de replicar la simulación, lo que se hace es hacer una simulación muy larga y dividirla en pedazos. Cada pedazo se toma como una réplica de la simulación. Únicamente el primer pedazo tendrá transitorio, el resto parten de un estado representativo. El método estudia la varianza de las medias de cada pedazo en función de la talla del pedazo. Una simulación (N observaciones) se divide en m pedazos de talla n . x_{ij} es la muestra j del batch i . Empezando con un n pequeño se siguen los siguientes pasos:

1. Se calcula la media de las muestras de cada batch: $\bar{x}_i = \frac{1}{m} \sum_{j=1}^n x_{ij}$ para $i = 1, 2, \dots, m$.
2. Se calcula la media total: $\bar{x} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i$
3. Se calcula la varianza de las medias de cada batch en función del tamaño de los batches: $Var(\bar{x}) = \frac{1}{m-1} \sum_{i=1}^m (\bar{x}_i - \bar{x})^2$

Incrementando n se repiten los pasos 1 a 3. Se representa la varianza en función de n . El valor de n a partir del cual la varianza siempre decrece, indica el transitorio (hay que tener en cuenta que la curva que forma el transitorio puede tener picos).

4.5. Criterio de parada

Si una simulación es demasiado corta, no nos podremos fiar de los resultados obtenidos porque pueden representar condiciones particulares. Si es demasiado larga, se utilizan muchos recursos y se puede perder mucho tiempo en los experimentos.

Para tener una idea de la longitud correcta de la simulación lo que se hace es fijar un intervalo de confianza de los resultados y calcular el número de muestras (la longitud de la simulación) para obtener ese intervalo de confianza.

La fórmula del *intervalo de confianza*, dado un nivel de rechazo α para una población de n muestras se calcula como sigue:

$$\bar{x} \pm z_{1-\alpha/2} \sqrt{\frac{Var(x)}{n}}$$

donde \bar{x} es la media de las n muestras, y $Var(x)$ es la varianza de las muestras.

Sin embargo, el uso de la función z para el cálculo del intervalo de confianza solo es válido cuando el número de muestras es grande. En nuestros experimentos estas muestras se extraen de las distintas réplicas de la simulación, que suele ser un número bajo (menos de 10). Por esta razón, se debe calcular los intervalos de confianza con una distribución que se aproxime a z , y ésta es la distribución t_γ (t student) con γ grados de libertad. Esta distribución tiende a la distribución normal $N(0, 1)$ cuando γ tiende a infinito.

Con todo esto, el intervalo de confianza se calcula como sigue:

$$\bar{x} \pm t_{n-1, 1-\alpha/2} \sqrt{\frac{Var(x)}{n}}$$

La siguiente tabla contiene un fragmento de la tabla de valores críticos de la función t con respecto al número de muestras n y el nivel de rechazo α .

$n - 1$	$1 - \alpha/2$		
	0.90	0.95	0.99
3	1.638	2.353	4.541
4	1.533	2.132	3.747
5	1.476	2.015	3.365
6	1.440	1.943	3.291
∞	1.282	1.645	2.326

Desgraciadamente, la fórmulas anteriores solo se pueden aplicar si las muestras son independientes. En el caso de las simulaciones las muestras pueden estar correlacionadas. Por ejemplo, si se calcula el tiempo de respuesta que sufren los clientes de una simulación, sus tiempos de respuesta dependen unos de otros si coinciden en las estaciones.

Para solucionar este problema se han propuesto diferentes soluciones.

4.5.1. Réplicas independientes

Se basa en la idea de que los datos de diferentes réplicas son independientes.

Se realizan m réplicas de tamaño $n + n_o$ de la simulación, donde n_o representa el transitorio. Los n_o datos iniciales de cada réplica son descartados. Luego se actúa de la siguiente manera:

1. Calcular la media de cada réplica: $\bar{x}_i = \frac{1}{n} \sum_{j=n_o+1}^{n_o+n} x_{ij}$ para $i = 1, 2, \dots, m$
2. Calcular la media de todas las réplicas: $\bar{\bar{x}} = \frac{1}{m} \sum_{i=1}^m \bar{x}_i$
3. Calcular la varianza de las medias de las réplicas:
 $Var(\bar{\bar{x}}) = \frac{1}{m-1} \sum_{i=1}^m (\bar{x}_i - \bar{\bar{x}})^2$
4. El intervalo de confianza para la media: $\bar{\bar{x}} \pm t_{m-1, 1-\alpha/2} \sqrt{\frac{Var(\bar{\bar{x}})}{m}}$

El intervalo de confianza es inversamente proporcional a \sqrt{mn} . Como se pierden n_o datos de cada réplica, el número de réplicas, m , normalmente no es mayor que 10. La longitud de las simulaciones, n , se aumenta hasta tener un intervalo de confianza de la media tan estrecho como sea necesario.

4.5.2. Medias Batch

Se realiza una simulación con $n_o + N$ muestras. Se descartan las n_o muestras iniciales (el transitorio) y las N muestras restantes se dividen en $m = \frac{N}{n}$ partes (lo que serían las réplicas de la simulación).

Empezando por un valor de n pequeño (por ejemplo 1) se actúa igual que en el caso anterior.

Al igual que antes, el intervalo de confianza de la media es inversamente proporcional a \sqrt{mn} .

Sin embargo hay un inconveniente. En el método de las réplicas se acepta que cada réplica es independiente de las otras. Esto no es cierto en el caso de usar las medias de partes de una misma simulación, de hecho los pedazos pueden ser más dependientes cuanto más pequeños sean (las muestras que contienen están más cercanas entre sí).

Para resolver este problema se va aumentando n y se recalcula los datos igual que en las réplicas pero además se calcula la covarianza entre medias consecutivas. Se actúa así hasta que la autocovarianza es pequeña comparada con la varianza.

La fórmula de la covarianza es la siguiente:

$$Cov(\bar{x}_i, \bar{x}_{i+1}) = \frac{1}{m-2} \sum_{i=1}^{m-1} (\bar{x}_i - \bar{\bar{x}})(\bar{x}_{i+1} - \bar{\bar{x}})$$

4.5.3. Método de la regeneración

El método se basa en encontrar estados de una simulación, a partir de los cuales la simulación se puede tomar como una simulación independiente que empieza en un mismo estado inicial. Este método no siempre se puede aplicar, ya que los sistemas no siempre presentan puntos de regeneración o bien éstos tienden a presentarse cada vez con menor probabilidad a medida que avanza la simulación.

El método para calcular la varianza con el método de regeneración es un poco más complicado debido a que cada réplica tiene un tamaño diferente.

Supongamos que se dispone de m ciclos de tallas n_1, n_2, \dots, n_m . Para calcular el intervalo de confianza:

1. Calcular la suma de cada ciclo: $y_i = \sum_{j=1}^{n_i} x_{ij}$
2. Calcular la media total: $\bar{\bar{x}} = \frac{\sum_{i=1}^m y_i}{\sum_{i=1}^m n_i}$
3. Calcular la diferencia entre las sumas de ciclos esperadas y observadas: $w_i = y_i - n_i \bar{\bar{x}}$ para $i = 1, 2, \dots, m$
4. Calcular la varianza de las diferencias: $Var(w) = s_w^2 = \frac{1}{m-1} \sum_{i=1}^m w_i^2$
5. Calcular el tamaño medio de los ciclos: $\bar{n} = \frac{1}{m} \sum_{i=1}^m n_i$
6. El intervalo de confianza es: $\bar{\bar{x}} \pm t_{m-1, 1-\alpha/2} \frac{s_w}{\bar{n}\sqrt{m}}$

Con el método de la regeneración no se eliminan datos iniciales, en cambio presenta algunos inconvenientes que dificultan su aplicación. Puede ser difícil definir los puntos de regeneración. Además en los estados de la simulación se debe ir comprobando si se trata de un punto de regeneración. Otras desventajas es que no se puede predecir el tiempo de la simulación y que los datos estadísticos son menos exactos.

4.6. Bibliografía

- A. M. Law, W. D. Kelton "Simulation Modeling & Analysis". Ed. McGraw-Hill (1984).
- R. Jain "The Art of Computer Systems Performance Analysis". Ed. Wiley (1991).