Módulo Montecarlo
Software RMES

Centro de Desarrollo de Gestión Empresarial.
1 Oriente 1097 - Viña del Mar, Chile.
Fono:(56) (32)688987 - Fax:(56) (32)2684079
empresa@mes.cl

Abril, 2010

Resumen

El presente informe pretende clarificar y documentar la implementación lógica del módulo Montecarlo para RMES, por lo que se presentarán explicaciones exhaustivas de cada algoritmo implementado. El fin de este documento es que pueda ser comprendido por cualquier persona que tenga conocimientos básicos sobre Ingeniería en Mantención y esté familiarizado con el software RMES.

Índice

1. Introducción

1.1. El Método Montecarlo

El método de Montecarlo es un método no determinístico o estadístico numérico usado para aproximar expresiones matemáticas complejas y costosas de evaluar con exactitud. El método se llamó así en referencia al Casino de Montecarlo (Principado de Mónaco) por ser "la capital del juego de azar", al ser la ruleta un generador simple de números aleatorios. El nombre y el desarrollo sistemático de los métodos de Monte Carlo datan aproximadamente de 1944 y se mejoraron enormemente con el desarrollo de la computadora.

El uso de los métodos de Monte Carlo como herramienta de investigación, proviene del trabajo realizado en el desarrollo de la bomba atómica durante la segunda guerra mundial en el Laboratorio Nacional de Los Álamos en EE.UU. Este trabajo conllevaba la simulación de problemas probabilísticos de hidrodinámica concernientes a la difusión de neutrones en el material de fusión, la cual posee un comportamiento eminentemente aleatorio. En la actualidad es parte fundamental de los algoritmos de trazado de rayos para la generación de imágenes sintéticas. Monte carlo

En la primera etapa de estas investigaciones, John von Neumann y Stanislaw Ulam refinaron esta ruleta rusa y los métodos de "división" de tareas. Sin embargo, el desarrollo sistemático de estas ideas tuvo que esperar al trabajo de Harris y Herman Kahn en 1948. Aproximadamente en el mismo año, Enrico Fermi, Metropolis y Ulam obtuvieron estimadores para los valores característicos de la ecuación de Schrödinger para la captura de neutrones a nivel nuclear usando este método.

El método de Monte Carlo proporciona soluciones aproximadas a una gran variedad de problemas matemáticos posibilitando la realización de experimentos con muestreos de números pseudoaleatorios en una computadora. El método es aplicable a cualquier tipo de problema, ya sea estocástico o determinista. A diferencia de los métodos numéricos que se basan en evaluaciones en N puntos en un espacio M-dimensional para producir una solución aproximada, el método de Monte Carlo tiene un error absoluto de la estimación que decrece como $\frac{1}{\sqrt{N}}$ en virtud del teorema del límite central.

1.2. Aplicación sobre RMES

La idea que inspiró la creación de este módulo fue la de aplicar el método Montecarlo al problema de mantención utilizando los modelos presentes en el software RMES.

Es bien sabido que la representación de una planta se realiza utilizando un modelo de árbol, en donde el nodo principal es la planta completa y los hijos son los subsistemas constituyentes de la planta. Las hojas, que son la unidad mínima de partición lógica de una planta, generalmente, son equipos o máquinas.

El comportamiento común en un problema de mantención es que ciertos equipos detienen su funcionamiento, provocando detenciones sobre los subsistemas a nivel superior y así sucesivamente, llegando en algunas ocasiones a la planta, lo que implicaría una detención completa de ella. Estas detenciones pueden ser o no programadas, lo que se traduce en que algunas detenciones se producen aleatoriamente y otras siguen algún plan de mantención.

Aplicando el Método de Montecarlo al problema anterior, lo que se busca es simular situaciones reales de funcionamiento de la planta de forma de evaluar el comportamiento de algún indicador de mantención. A pesar de que la idea suena sencilla, no es fácil esgrimir procedimientos para realizar esta tarea, debido a las muchas consideraciones que se deben tener presentes.

Después de varios intentos en los cuales no se obtuvieron resultados convincentes, se decidió por revisar por completo el módulo creando casos de pruebas para comprobar la correcta ejecución de las tareas. Luego de corregir los errores encontrados y mejorar algunas funciones, el módulo aseguró su correcta ejecución. Aún así, los resultados no eran satisfactorios, por lo que, dado que la probabilidad de que existiera alguna implementación erronea era baja luego de la revisión exhaustiva, sólo quedaba investigar la correctitud de cómo se estaba realizando la simulación.

Luego de un tiempo de investigación sobre la simulación, se llegó a la conclusión de que ésta debía reimplementarse nuevamente de forma "ciega", en el sentido de no tener presente los conocimientos previos sobre mantención. De esta forma, el problema se reduce a simular un conjunto de números o sumatorias.

Uno de los objetivos de este documento es explicar los distintos pasos realizados para llegar a la simulación final actualmente utilizada en el módulo y algunas otras ideas que han surgido en el camino.

2. Implementación

2.1. Variables, Indicadores y Distribuciones

La implementación del Método Montecarlo sobre RMES necesita de datos de entrada, más precisamente las variables a simular y también de indicadores que indiquen un resultado respecto a las variables. Las primeras, debe ser distribuciones de probabilidad que permitan generar números aleatorios, de forma que un indicador tome estos valores y genere un resultado.

Las variables básicas del módulo son:

- El TBF: la distribución de tiempos entre fallas.
- El TTR: la distribución de tiempos de detención.

Estas variables deben estar asociadas a todos los equipos existentes en la planta, dado que son ellos las unidades básicas de entrada de información (generalmente sobre ellos se cargan detenciones) y se obtienen directamente del conjunto de detenciones existentes. También es posible asignar una distribución cualquiera con parámetros arbitrarios.

Teniendo esto, es posible simular detenciones, entrelazando los tiempos entre fallas y los tiempos de detención, creando así una lista de *tuplas*, en donde *tupla* es un concepto acuñado a nivel informático que representa una detención.

Luego, por cada equipo, es posible cargar estas detenciones simuladas como si fueran reales, posteriormente utilizar el módulo de cálculos históricos de RMES y finalmente elegir los indicadores de interés que sean necesarios para estudiar el nuevo compartamiento de la planta.

El indicador que maneja el módulo actualmente es Disponibilidad. Anteriormente, también se consideraba también el Costo de Mantención y el Costo de la Falta. El primero, debido a que es un cálculo no aleatorio, no se incluye dado que no aporta información. El segundo, debido a su alto costo computacional será considerado en las próximas liberaciones del módulo, cuando RMES implemente las herramientas necesarias para obtenerlo de forma rápida.

El estudio puede ser realizado sobre cualquier nivel dentro de una planta.

A cada variable, ya sea TBF o TTR, se le puede asignar una de las siguientes distribuciones:

- Weibull
- Exponencial
- Normal
- LogNormal
- Triangular
- Uniforme
- Punto Fijo (Dirac)

A cada distribución se le puede configurar sus parámetros y sus bordes. Estos bordes se refieren al menor y mayor número que puede abarcar la distribución. Matematicamente, estos bordes son $-\infty$ y $+\infty$, pero ésto computacionalmente no existe, por lo cual, los bordes por defecto se asignan respecto a la distribución a los puntos x tales que la probabilidad evaluada en ellos sea 0,01 y 0,99, respectivamente. El usuario puede ajustar estos bordes siempre y cuando no violen ninguna de las siguientes restricciones:

1. **Suficiencia**: La probabilidad entre el borde inferior y superior debe ser mayor que 40 % (con el fin de que el generador de números aleatorios no entre en loop que tomen demasiado tiempo).
2. **Positividad**: La probabilidad entre 0 y el borde superior debe ser al menos el doble que la probabilidad entre el borde inferior y 0 (con el objeto de que la sumatoria total de los números positivos sea mayor que la sumatoria de los numeros negativos generados).

Los parámetros también afectan a estas restricciones, por lo cual, diremos que una distribución configurada de forma **suficiente** y **positiva** será aceptada por el módulo.

2.2. Generación de Números Aleatorios

Para la generación de números aleatorios se utilizó el Algoritmo de la Transformada Inversa y se puede definir, de forma general, de la siguiente manera:

> Sea U una variable aleatoria uniforme en (0,1). Para cualquier función de distribución continua F, invertible, la variable aleatoria X definida como
> $$X = F^{-1}(U)$$
> tiene distribución F, donde $F^{-1}(u)$ se define como el valor de x tal que $F(x) = u$

Ejemplificando, sea una variable X que se puede modelar con una exponencial de media 0,007, Exp(0,007). Para obtener valores de la variable X acorde con su comportamiento exponencial, se realiza lo siguiente:

1. Se obtiene un número aleatorio uniformemente distribuido entre 0 y 1, digamos, el número p.
2. Luego, con el número p, se utiliza la función inversa de la función de distribucion acumulada para obtener el punto x asociado.

Por ejemplo, suponiendo que el número aleatorio obtenido fue 0.4. Sabemos que la función de distribución acumulada de una exponencial es:

$$F(x) = 1 - e^{-\lambda x}$$

La inversa se puede obtener directamente

$$F^{-1}(y) = \frac{e^{(1-y)}}{\lambda}$$

donde y es una variable aleatoria uniformemente distribuida entre 0 y 1.

Ahora, simplemente se reemplazan los valores: $y = 0{,}4$; $\lambda = 0{,}007$

$$F^{-1}(0{,}4) = 260{,}3$$

Lo que se realizó entonces es un mapeo de una función uniforme entre 0 y 1 a una exponencial con media 0.007. Y en general, este procedimiento es similar para toda distribución que posea una función inversa de probabilidad acumulada definida.

Dado que las distribuciones Normal y LogNormal no poseen inversa, se debe utilizar un generador de números aleatorios normalmente distribuido de manera estándar, $N(0, 1)$. Para obtener un número aleatorio respecto a una $N(\mu, \sigma^2)$ simplemente, se realiza un cambio de variables: sea x un número aleatorio normalmente distribuido $N(0, 1)$, entonces

$$z = g(x) = \sigma x + \mu$$

donde z es número uniformemente distribuido de forma $N(\mu, \sigma^2)$

Para obtener un valor aleatorio asociado a una LogNormal con media μ y desviación estándar σ se realiza algo similiar: sea x un número aleatorio normalmente distribuido $N(0, 1)$, entonces

$$z = g(x) = e^{\sigma x + \mu}$$

donde z es un número lognormalmente distribuido de forma $LN(\mu, \sigma^2)$.

Cabe destacar que dado que existe la posibilidad de que el usuario configure los bordes de la distribución, se hace necesario el uso de distribuciones truncadas, que básicamente es un cambio de variables, más específicamente, una regla de 3, sobre el valor del número aleatorio uniformemente distribuído de forma de transformar este valor de un rango $[0, 1]$ a un rango $[r_1, r_2]$ que es el rango de probabilidad permitido por la distribución respecto a sus bordes. Por ello, se debe entender que, inicialmente, los valores de $[r_1, r_2]$ son $[0{,}01, 0{,}99]$, como se explicó anteriormente. Además, se puede decir que toda distribución del módulo es una distribución truncada.

> Sea r el valor de un número aleatorio uniformeme distribuído entre $[0, 1]$, el valor r^* asociado a una distribución truncada con bordes $[b_1, b_2]$ es:
>
> $$r^* = \left(F^{-1}(b_2) - F^{-1}(b_1)\right) \cdot r + F^{-1}(b_1)$$
>
> donde $F^{-1}(x)$ es la inversa de la función de probabilidad acumulada.

Dado que para las distribuciones Normal y LogNormal no existe $F^{-1}(x)$ se lanzan números aleatorios indefinidamente distribuidos como $N(0, 1)$, sea uno de estos números x, hasta que alguno de ellos cumpla que $b_1 \leq F(g(x)) \leq b_2$, donde $g(x)$ es la transformación del número x asociado a la distribución no invertible anteriormente explicada.

Debido a estas últimas consideraciones es que se restinguió la configuración de bordes a abarcar al menos un 40 % de probabilidad de la distribución, con el fin de que el lanzamiento de números aleatorios para distribuciones no invertibles tenga fin. Si hipotéticamente, se diese el caso contrario para la Normal y la LogNormal, esta restricción no sería necesaria, dado que existiría una función directa entre un número aleatorio y su punto x en la distribución.

2.3. Asignación de Distribuciones

Como se explicó anteriormente, el módulo permite la configuración de la distribución asignada a cada variable.

El módulo asigna una distribución por defecto que se obtiene a través de la información que los equipos contenga. Inicialmente, la distribución del TBF se obtenía directamente de RMES. La estimación de parámetros de esta distribución se realiza considerando sólo las detenciones de tipo MP y MC, y con ellas, utiliza el algoritmo de censura a la derecha para generar los tiempos y las correcciones necesarias para ajustar a sólo dos distribuciones: Weibull o Exponencial.

Dado que el módulo fue re-pensado por completo, lo que se realiza actualmente es, teniendo un equipo y sus detenciones, una estimación de parámetros para las 6 distribuciones soportadas (no se considera la distribucion de Dirac) de los tiempos entre fallas y tiempos de detención **sin considerar el tipo de detención** y mediante Test de Hipótesis Kolmogorov-Smilrnov se eligen aquellas que posee mejor cercanía con los datos.

Esto produce que el módulo pueda asignar una distribución al TBF distinta a la que RMES asignó, dado que el módulo no realiza ninguna consideración de tipo de tupla, sólo le interesa modelar de forma más feasiente posible a las variables. RMES no considera otras distribuciones por 2 razones: uno, es complejo obtener indicadores probabilisticos para distribuciones distintas a Weibull y Exponencial, y por simplicidad, sólo se utilizan éstas que están muy bien documentadas en la literatura de la Ingeniería en Mantención; y dos, a RMES le interesa saber el comportamiento de la detenciones tipo MC, dado que son ellas las representativas del comportamiento del equipo, a diferencia del módulo, que intentan recrear una situación real en donde ocurren detenciones de todo tipo.

Cuando los equipos no poseen datos (más especificamente, no poseen tuplas), el ajuste se realiza asignando distribuciones Dirac según los valores presentes en los campos del equipo asociados a MTBF y MTTR.

2.4. Proceso de simulación

Teniendo la configuración de las distribuciones de cada variable para todos los equipos de la planta, es posible comenzar con la simulación.

Lo que el módulo pretende es simular detenciones, cargárselas a los equipos y luego utilizar el módulo de cálculos históricos para obtener los indicadores de interés, por ejemplo, la disponibilidad.

El proceso de simulación de detenciones se divide en 4 partes:

1. Asignación del intervalo de simulación: se refiere a definir las fechas inicial y final de generación de tiempos.
2. Simulación de Tiempos: se refiere a cómo generar los tiempos para las variables TBF y TTR.
3. Simulación de Tipos de Tupla: se refiere a cómo asignar los tipos de tuplas después que se hayan creado los tiempos.
4. Creación y Asignación de Tuplas Simuladas: se refiere a la creación de las listas de tuplas que finalmente serán cargadas a los equipos.

Para los 3 primeros pasos, existen varias formas diferentes de cómo realizar la tarea, que está asociado a lo que se quiere obtener como resultado de la simulación. A continuación, sólo se explicarán los modelos que utiliza el módulo.

La Asignación del intervalo de simulación se define desde un fecha inicial fija. Esta fecha es el **01/01/2000 00:00:00 hrs**. Esto quiere decir que, en términos de simulación, todos los equipos partirán su funcionamiento desde ese día. Para efectos de cálculo, utilizar otra fecha de inicio no influye en lo más mínimo el resultado, la elección ha sido completamente arbitraria. Ahora, la fecha de término dependerá de la cantidad de años que el usuario quiera simular. El módulo soporta el ingreso de hasta 5 años de simulación, por lo que la fecha final dependará de aquello. El valor por defecto para los años de simulación es 1.

La Simulación de Tiempos fue una de las partes lógicas del módulo que más sufrió cambios y estudios exhaustivos. Uno de los errores iniciales que el módulo tenía era que la generación de tiempos no consideraba números negativos, esto implica que las distribuciones siempre se truncaban en el valor cero, y por ende, los resultados finales perdían el comportamiento inicial de la distribución debido a que no se conservan sus propiedades, como la esperanza. Lo que se implementó entonces fue un algoritmo de corrección de negatividad, que se puede resumir en los siguientes pasos:

1. Para una variable, se generan los tiempos según su distribución. Se almacenan y suman los números positivos y los negativos sólo se suman.

$$x_{pos} = \sum_i x_i^+ \quad \text{sumatoria de numeros positivos}$$

$$x_{neg} = \sum_i x_i^- \quad \text{sumatoria de numeros negativos}$$

2. Se chequea que la sumatoria de los números positivos sea mayor que la de los números negativos, de lo contrario, implicaría que finalmente se generarían tiempos negativos.

$$x_{pos} > x_{neg}$$

3. Se encuentra un valor de corrección c tal que:

$$c \cdot x_{pos} = x_{pos} - x_{neg}$$

$$c = \frac{x_{pos} - x_{neg}}{x_{pos}}$$

4. Cada xi se corrige multiplicando su valor por c:

$$x_i = c \cdot x_i \quad \forall_i$$

La generación inicial de tiempos agrega una tolerancia que impide la generación de tiempos positivos muy pequeños, de esta forma, no se permiten tiempos que estén en el rango de 0 a 36 segundos.

Además, luego de aplicar la corrección a los tiempos de cada variable, se chequea que la sumatoria de todos los tiempos, TBF's y TTR's, sea superior que el tiempo de simulación ingresado por el usuario. Si esto no es así, se vuelven a generar tiempos aleatorios simplemente mayores que la tolerancia, 36 segundos, hasta completar el tiempo de simulación. Cabe destacar que esta última generación de datos afectará la distribución de tiempos simulados en menor medida que la antigua forma de simular.

La Simulación de Tipos de Tupla posee varias implementaciones. La forma original simplemente asignaba un tipo MCM a todo detención simulada. La forma actual utiliza una asignación de tipo de tupla vía "Ruleta Rusa". Para cada equipo, se cuentas la cantidad de detenciones pertenecientes a uno de los siguientes tipos: MP, MC, DO, DONP. Teniendo esta información, se obtiene la frecuencia relativa de tipos de tuplas y finalmente se crea la frecuencia relativa acumulada, que representa probabilidad. De esta forma, se lanza un número aleatorio uniformemente distribuído y según su valor caerá en uno de los rangos asociados a un tipo de tupla. Este tipo de tupla se asignará a la detención en proceso.

Finalmente, la Creación y Asignación de Tuplas Simuladas junta toda la información y genera la lista de tuplas simuladas que se ingresarán a los equipos. Esto se maneja cuidadosamente, dado que el módulo guarda una copia de los datos originales de cada equipo. La simulación sobreescribe una y otra vez las tuplas de un equipo, calcula los indicadores y luego restaura al equipo los datos originales. Con esto se logra un menor uso de memoria y ofrece mayor rapidez a la simulación, respecto a la implementación anterior.

Entonces, se generarán detenciones desde la fecha de inicio antes descrita hasta completar la cantidad de años de simulación que el usuario indique tantas veces como la cantidad de iteraciones que el usuario desee. Este valor está acotado entre 1 y 10.000. Para cada iteración, se obtiene un punto del indicador, por ejemplo, 100 iteraciones representan 100 puntos de disponibilidad. Estos puntos se ordenan, luego, se dividide el intervalo de existencia de datos en varios intervalos creando sacos en los que cada punto según su valor será asignado. Con ello, es posible crear un histograma de frecuencia absoluta, luego, dividiendo cada saco por el total de puntos (en este caso, 100), se crear el histograma de frecuencias relativas, con el cual es posiblee realizar un estudio probabilístico.

3. Trabajo Futuro

Este documento representa un punto de congelación para el módulo, lo que implica que el desarrollo será aplazado momentáneamente.

Uno de los temas importantes de éste y cualquier desarrollo informático es el diseño de casos de pruebas que aseguren la más adecuada, sino perfecta (en absoluta presencia de factores externos), ejecución de las tareas que implementen los algoritmos. Lamentablemente, por problemas de configuración, el módulo no puede utilizar la biblioteca de casos de prueba que RMES usa, lo que implica que las casos de prueba implementados para el módulo son menos robustos, por lo cual, pueden existir fugas en ciertos puntos dada la gran cantidades de modificaciones. Otra razón mucho más medular es que un caso de prueba tiene por definición un valor objetivo al cual el algoritmo, dado cierto input, debe dar como resultado. En la simulación, este valor objetivo no existe, dado que no es sencillo diseñar un caso de prueba para, por ejemplo, dos equipos en fraccionamiento, o una planta con varios niveles. Queda como tarea futura finiquitar este punto del módulo.

Pendiente también queda el estudio de las nuevas formas de simulación implementadas en el módulo que fueron desechadas dado que la implementación actual es la de mejor comportamiento con los casos de prueba que se cuentan. No se descarta algún error de implementación en alguna de ellas que no fue detectada.

Nuevas ideas de simulación surgieron mientras se avanzaba en el módulo. Una interasente era realizar una versión modificada que simulara sólo la duración de las detenciones. La posición temporal (cuando comienza la detención) y el tipo se obtendría directamente de los datos originales, de esta forma, la lista de detenciones simuladas sería en extremo similar a la lista de detenciones originales, por lo que finalmente los indicadores probabilísticos sería muy parecidos y cercanos a lo real. La deficiencia de ésto es que la variabilidad de los resultados sería escasa y se perdería la idea de aleatoriedad de las detenciones no programadas. A pesar de ello, esta idea si es factible para detenciones programadas, dado que ellas siguen un patrón del cual no se tiene ninguna información.

Otra idea es tambíen crear distribuciones para tiempos de duración para cada tipo de tupla. De este modo, se lanzarían tiempos simulados por varios canales, cada canal asociado a un tipo de tupla, y luego estos canales se unirían en un solo conformando la lista de tuplas final en una iteración. El defecto de esto es que separar los datos en varios subconjuntos implica que se tendrá menos información para realizar el ajuste por cada tipo, considerando que los datos en Ingeniería en Mantención no son numerosos, lo que podría llevar a que el ajuste sea defectuoso y simulaciones poco precisas.

Más interesante sería unir estas dos ideas, de forma de abstraer las buenas propiedades de cada una, lo cual no sería muy dificil de idear. Por ejemplo, ajustar los tiempos para detenciones de tipo MC y DONP (detenciones de tipo no programadas y de caracter aleatorio), luego, por un canal, generar los tiempos aleatorios no programados, mientras que por otro, generar los tiempos para todo lo programado utilizando la primera idea. Por supuesto, ambas ideas asumen que existen suficientes datos, partiendo de la base de que ellos existen.

4. Conclusiones

El trabajo que duró aproximadamente 2 meses desde el día en que se retomó el módulo hasta ahora fue el de investigar el porqué los resultados no eran satisfactorios y si esto era correcto o no.

De forma exhaustiva, se revisó gran parte del código y se llego a la conclusión que las partes básicas y medulares del módulo realizan correctamente su función. Es así como se tiene gran certeza de que las implementaciones de los distintos algoritmos y de las distribuciones se han hecho de forma correcta. Además, no sólo se corrigieron errores existentes en la implementación anterior, sino que también se depuraron asperezas que hacían lenta la simulación, por lo que se puede decir que optimizar más aún el módulo será una tarea muy dificil, por lo que es conveniente recomendar que los esfuerzos futuros no se dirigan a esta área antes de completar otras de mayor importancia, como Testing.

El módulo Montecarlo para RMES finalmente se transforma en una compleja calculadora de indicadores probabilísticos para Ingeniería en Mantención, la cual utiliza el poderoso algoritmo de cálculo histórico, implementado únicamente en este software, para obtener un indicador ”real” del comportamiento del subsistema en estudio. Entiéndase ”real” como el hecho de no utilizar fórmulas probabilísticas para obtener un indicador, si no el de utilizar sumas y ponderaciones aritméticas sobre las detenciones que han subido sobre un subsistema.

En este sentido, el indicador probabilístico que entrega el módulo es mucho más preciso y optimizable que el actual implementado en el software, dado que es posible mejorar el valor estadístico mientras más tiempo se esté dispuesto a correr la simulación. Además, presenta un histograma que permite ver no sólo el valor promedio (que es lo que se presenta en RMES) si no que también la varianza del indicador, en donde también es posible realizar estudios estocásticos gracias a la interfaz gráfica implementada, junto con presentar información variada de los resultados (media, mediana, moda, cuartiles, etc).

Además, el módulo permite al usuario simular el comportamiento de una planta o un subsistema sin necesidad de tener datos, sólo configurando las distribuciones que él estime convenientes para cada uno de los equipos en juego y de esta forma obtener escenarios probables de lo que podría ocurrir a futuro.

Por ello, su utilidad radica en ser una herramienta con una base matemática fuertemente usada en todos los campos de la investigación que servirá de apoyo y soporte a la toma de decisiones en la Ingeniería en Mantención.