

Uma revisão da simulação de Monte Carlo enviesada

Guilherme de Aquino Torres, Laboratório de Análise, Avaliação e Gerenciamento de Risco – LabRisco,
Universidade de São Paulo

Danilo Taverna Martins Pereira de Abreu, Laboratório de Análise, Avaliação e Gerenciamento de Risco –
LabRisco, Universidade de São Paulo

Marcelo Ramos Martins, Laboratório de Análise, Avaliação e Gerenciamento de Risco – LabRisco,
Universidade de São Paulo

RESUMO

A simulação de Monte Carlo (MCS, do inglês, *Monte Carlo Simulation*) é um método estatístico que busca por meio de uma geração massiva de dados apresentar uma solução para um problema multidimensional complexo. Quando abordamos a análise e gerenciamento de risco, a MCS pode ser usada como uma ferramenta para estimar a probabilidade de ocorrência de eventos indesejados por meio da simulação do comportamento do sistema ao longo do tempo. Entretanto, quando se aborda problemas cujas probabilidades são muito baixas, é necessário um número relativamente grande de simulações para gerar ao menos uma observação do evento desejado ao longo da simulação. Em alguns casos, o número de simulações necessárias se torna impeditivo em termos computacionais. Como alternativa para viabilizar a aplicação da MCS a tais casos, este trabalho apresenta a aplicação da simulação de Monte Carlo enviesada (BMCS, do inglês, *Biased Monte Carlo Simulation*) à análise de confiabilidade e disponibilidade de sistemas de engenharia.

A BMCS constitui uma forma de abordar sistemas com probabilidades de falha muito baixas e reduzir o número de rodadas necessárias em uma simulação. Esta redução é possível devido ao fato de que um procedimento de enviesamento força a ocorrência dos eventos de interesse, tais como a falha de componentes. Em seguida, o viés introduzido pela abordagem é retirado pela aplicação de um fator de atualização, que ajusta o resultado à probabilidade real daquele evento acontecer. Desta maneira, sistemas nos quais a consequência de uma falha é catastrófica, e que, portanto, necessitam de componentes cujas taxas de falha sejam significativamente baixas, podem ter seus riscos analisados por vias menos custosas do ponto de vista do processamento computacional.

1. Introdução

Sistemas industriais complexos apresentam probabilidades inerentes de falha, que potencialmente resultam em consequências graves para a vida, patrimônio e meio ambiente [1]. Desta forma é essencial mensurar as probabilidades de falha dos sistemas por meio de análises quantitativas, para auxiliar o gerenciamento de seus riscos. Uma forma de fazê-lo é por meio das simulações de Monte Carlo, do inglês, *Monte Carlo Simulation* (MCS) que é um método estatístico de se obter respostas numéricas para a modelagem de problemas multidimensionais complexos na perspectiva probabilística.

Apesar da popularidade da MCS, existem algumas limitações para a sua aplicação. Em particular, a flexibilidade proporcionada pelo método implica numa alta demanda de processamento e memória ao passo que se deseja refinar a precisão numérica [2]. Isto culmina numa alta exigência de tempo para processar uma simulação e, eventualmente, inviabiliza a estimativa de resultados numéricos que envolvem probabilidades de ocorrência significativamente baixas. Uma alternativa para otimizar as simulações é por meio do uso da simulação de Monte Carlo enviesada [3], do inglês, *Biased Monte Carlo Simulation* (BMCS), que é um método

de se obter a resposta numérica mediante à substituição das distribuições de probabilidades originais por outra distribuição de probabilidade que favoreça o evento de interesse (em geral, a falha do sistema).

Com o intuito de executar os avanços realizados pela literatura sobre a simulação de Monte Carlo enviesada, este trabalho visa apresentar uma formulação do BMCS à análise RAM (*Reliability, Availability, and Maintainability*) de sistemas de engenharia e uma aplicação-piloto.

2. Formulação típica da MCS para a análise RAM

O modelo de simulação indireta consiste no sorteio de um tempo T_{s+1} das transições do sistema a partir de uma função acumulada de probabilidade de transição do sistema $F_j(T_{s+1}|T_s)$ [2], onde T_s é o instante de tempo atual no qual o sistema se encontra. A função de densidade de probabilidade do sistema é derivada a partir da função acumulada de probabilidade de transição de cada componente i do sistema no estado j_i dentre todos os N_c componentes, de forma que a função acumulada de probabilidade pode ser obtida por meio de:

$$F_j(T_{s+1}|T_s) = 1 - \prod_{i=1}^{N_c} [1 - F_{j_i}(T_{s+1}|T_s)] \quad (2.1)$$

Por definição, temos a função de densidade de probabilidade derivando a equação 2.1:

$$f_j(T_{s+1}|T_s) = \frac{\partial F_j(T_{s+1}|T_s)}{\partial T_{s+1}} = \exp \left\{ - \sum_{n=1}^{N_c} [g_{j_n}(T_{s+1}) - g_{j_n}(T_s)] \right\} \cdot \sum_{l=1}^{N_c} \gamma_{j_l}(T_{s+1}) \quad (2.2)$$

Onde γ_{j_i} é a taxa que governa a transição do componente i a partir de seu estado j_i . Por sua vez, $g_{j_i}(t) = \int_0^t \gamma_{j_i}(\tau) d\tau$.

A estrutura de cada rodada de simulação baseada na formulação acima pode ser dividida em 3 partes: (I) sorteio temporal para a próxima transição de estado do sistema, (II) sorteio do componente k que sofrerá a transição de estado, (III) sorteio do novo estado i do componente k sorteado.

- (I) O sorteio temporal T_{s+1} , que define o momento em que o sistema sofrerá uma transição de estado, é obtido por meio do sorteio de uma variável aleatória $\xi_t \in [0,1[$ uniformemente distribuída, atribuída à solução da equação 2.1, de forma que:

$$F_j(T_{s+1}|T_s) = \xi_t \quad (2.3)$$

- (II) O sorteio do componente k que sofrerá a transição será realizado por meio do sorteio de uma variável aleatória $\xi_k \in [0,1[$ uniformemente distribuída, de forma que o componente em transição será aquele que satisfaz a equação 2.4., onde $p_{j_i} = \gamma_{j_i}(T_{s+1}) / \sum_{n=1}^{N_c} \gamma_{j_n}(T_{s+1})$.

$$\sum_{i=0}^{k-1} p_{j_i} \leq \xi_k < \sum_{i=1}^k p_{j_i} \quad (2.4)$$

- (III) O sorteio do estado m do componente k selecionado será realizado por meio do sorteio de uma variável aleatória $\xi_m \in [0,1[$ uniformemente distribuída, de forma que o estado do componente em transição será aquele que satisfaz a equação 2.5, na qual $l_k = 1, \dots, N_s^k$ representa um dos N_s^k

estados do componente k , isto é, l_k trata-se do novo estado do componente k , e j_k o estado de partida do componente k . Desta forma temos que $\pi_{j_k \rightarrow l_k} = \gamma_{j_k \rightarrow l_k}(T_{s+1})/\gamma_{j_k}(T_{s+1})$.

$$\sum_{l_k=1}^{m-1} \pi_{j_k \rightarrow l_k} \leq \xi_m < \sum_{l_k=1}^m \pi_{j_k \rightarrow l_k} \quad (2.5)$$

A estrutura é repetida enquanto $T_{s+1} \leq T_M$, onde T_M é o tempo de missão do sistema. Para todo instante em que o sistema encontra o evento topo, é registrado uma falha nos canais temporais ao longo do tempo de missão da simulação, ilustrado na Figura 2.1.

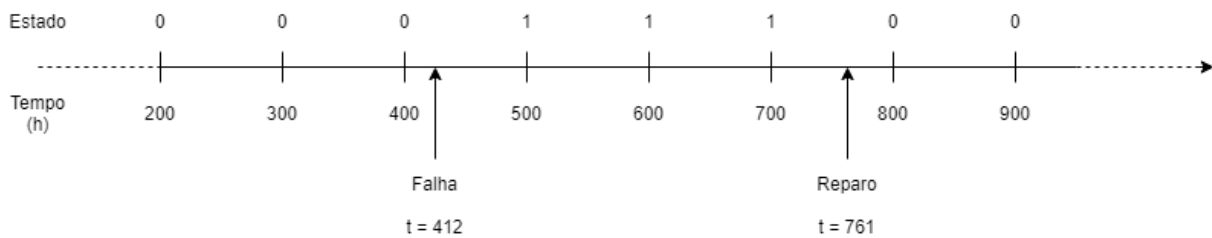


Figura 2.1 – Registro de falha no canal temporal

A indisponibilidade é obtida por meio da soma de todas as contagens de falha nos canais temporais, e dividindo pelo número de rodadas.

3. Aplicação da BMCS para a análise RAM

A BMCS é um modelo estendido da MCS governado por uma função de distribuição de probabilidade (fdp) que favoreça um evento topo no lugar de uma fdp original. O enviesamento da simulação oriunda da troca de distribuição é seguido por uma correção do resultado quantitativo por um fator de atualização (FA). A BMCS na perspectiva da análise RAM visa substituir as fdps originais do sistema por aquelas que favorecem os estados de indisponibilidade. Isto ocorre por meio da mudança da distribuição contínua de transição no tempo e/ou nas distribuições discretas na escolha do componente em transição de estado. O enviesamento de componentes de estados binários, isto é, operante e em falha, podem acontecer de duas formas: (I) modificando as distribuições de probabilidade temporais para favorecer as transições dentro do tempo de missão; ou (II) modificar as distribuições de probabilidade discretas (vide as equações 2.4 e 2.5) a fim de favorecer as transições para estados de falha. Ambos os métodos visam aumentar o número de registros de falhas do sistema, estimando o valor real da indisponibilidade por meio da inserção do fator de atualização (FA). A aplicação de cada um dos métodos citados ocorre como descrito a seguir:

- (I) O avanço temporal será substituído por uma distribuição uniforme contínua entre o tempo T_s e T_M , cuja distribuição acumulada de probabilidade é dada pela equação 3.1.

$$F_j^*(t|T_s) = \frac{t - T_s}{T_M - T_s} \quad (3.1)$$

O tempo T_{s+1} é obtido através do sorteio de uma variável aleatória $\xi_t \in [0,1[$ uniformemente distribuída, atribuída à solução da equação 3.1, de forma que $F_j^*(T_{s+1}|s) = \xi_t$. Desta forma T_{s+1} é dado por pela equação 3.2.

$$T_{s+1} = T_s + (T_M - T_s) \cdot \xi_t \quad (3.2)$$

Vale a pena reparar que devido às propriedades da distribuição uniforme, a simulação descrita pela equação 4.2 possui infinitas iterações. Para solucionar esse problema, essa distribuição é condicionada ao sistema operante exclusivamente, caso contrário, a distribuição de probabilidade acumulada que descreve os tempos de transição do sistema é a equação 2.1 (isto é, sem aplicação do enviesamento). Desta forma, as transições de reparo (cujos tempos até a ocorrência geralmente são muito menores do que os tempos até a falha) viabilizam o término da rodada de simulação.

- (II) O sorteio do componente que sofrerá a transição é obtido de forma condicionada. Caso o sistema não esteja em estado de falha, a distribuição de probabilidade é substituída por uma distribuição discreta uniforme entre todos os componentes do sistema, expressa pela equação 3.3.

$$\sum_{i=0}^{k-1} \frac{i}{N_c} \leq \xi_k < \sum_{i=1}^k \frac{i}{N_c} \quad (3.3)$$

Caso contrário, o sistema não necessita de enviesamento. Logo este não ocorrerá.

Por fim, para toda rodada onde foi necessário a introdução do enviesamento, será necessário corrigi-los por meio do fator de atualização (FA), que é um valor que forçará o termo a assumir a indisponibilidade caso ela não fosse enviesada. O fator de atualização é dado pela razão entre a fdp original, f_{ji} , e a fdp enviesada, \tilde{f}_j^* expresso na equação 3.4.

$$FA = \frac{f_{ji}(T_s|T_{s+1})}{\tilde{f}_j^*(T_s|T_{s+1})} \quad (3.4)$$

Desta forma, para aplicar o fator de atualização na (I) modificação das distribuições de probabilidade temporais para favorecer as transições dentro do tempo de missão temos a fdp da distribuição enviesada na equação 3.5.

$$\tilde{f}_j^*(T_{s+1}|T_s) = \frac{dF_j^*(t|T_s)}{dt} = \frac{1}{T_M - T_s} \quad (3.5)$$

Com o auxílio da equação 3.1, temos que o FA de (I) avanços temporais são dados por 3.6, de um dado sistema com um componente i transitando do estado inicial j_i para um estado final k_i num tempo T_s para um tempo T_{s+1} dentro de um tempo de missão T_M .

$$FA = \frac{f_{ji}(T_s|T_{s+1})}{\tilde{f}_j^*(T_s|T_{s+1})} = \sum_{i=0}^{N_c} \gamma_{ji \rightarrow ki}^i \exp \left[\sum_{i=0}^{N_c} \gamma_{ji \rightarrow ki}^i \cdot (T_{s+1} - T_s) \right] \cdot (T_M - T_s) \quad (3.6)$$

Por fim, por meio da equação 3.4, temos que o FA de (II) sorteio dos componentes em transição é dado por 3.7.

$$FA = \frac{f_{ji}(T_s|T_{s+1})}{\tilde{f}_j^*(T_s|T_{s+1})} = \frac{\gamma_{ji \rightarrow ki}^i}{\sum_{i=0}^{N_c} \gamma_{ji \rightarrow ki}^i} \cdot N_c \quad (3.7)$$

A indisponibilidade por fim é obtida realizando o produto entre o FA acumulado no tempo T_{s+1} conjuntamente com o produto de todos os FA em tempos anteriores dentro da rodada.

4. Aplicação-piloto

A aplicação-piloto considerando um sistema composto por dois componentes idênticos e reparáveis instalados em série, cujas transições de estado pode variar em disponível e indisponível com distribuições exponenciais. Os componentes apresentam uma taxa de falha constante $\lambda_{0i \rightarrow 1i}^i = 10^{-6} \text{ ano}^{-1}$ e taxa de reparo constante $\lambda_{1i \rightarrow 0i}^i = 10^{-2} \text{ ano}^{-1}$. É possível determinar a disponibilidade esperada deste sistema de maneira analítica por meio de um modelo Markoviano e, com isso, validar os resultados obtidos a partir da simulação. O sistema tem um tempo de missão de T_m , a nível de N_r rodadas, que são geradas em um computador com processador Intel® Core™ i5-3550 CPU@3.30GHz 3.70 GHz, memória instalada (RAM) 8,00 GB e sistema operacional Windows 10 x 64.

A MCS, para tempos de missão curtos em relação ao tempo médio de falha dos componentes, não consegue capturar as transições de falha. Em virtude disto, o resultado da MCS obtido na Figura 4.1 ao nível de 10^3 rodadas dentro de um tempo de missão de 10 anos é constante e nulo, o que não provê discernimento em termos de análise de risco. Como solução, a BMCS ao mesmo nível de rodadas apresenta um comportamento semelhante ao resultado analítico. Desta forma a simulação enviesada apresenta a capacidade de alcançar um nível satisfatório no refino dos resultados sem a necessidade de um número extenso de rodadas. A Figura 4.2 destaca o refinamento dos resultados em função do aumento do número de rodadas de simulação.

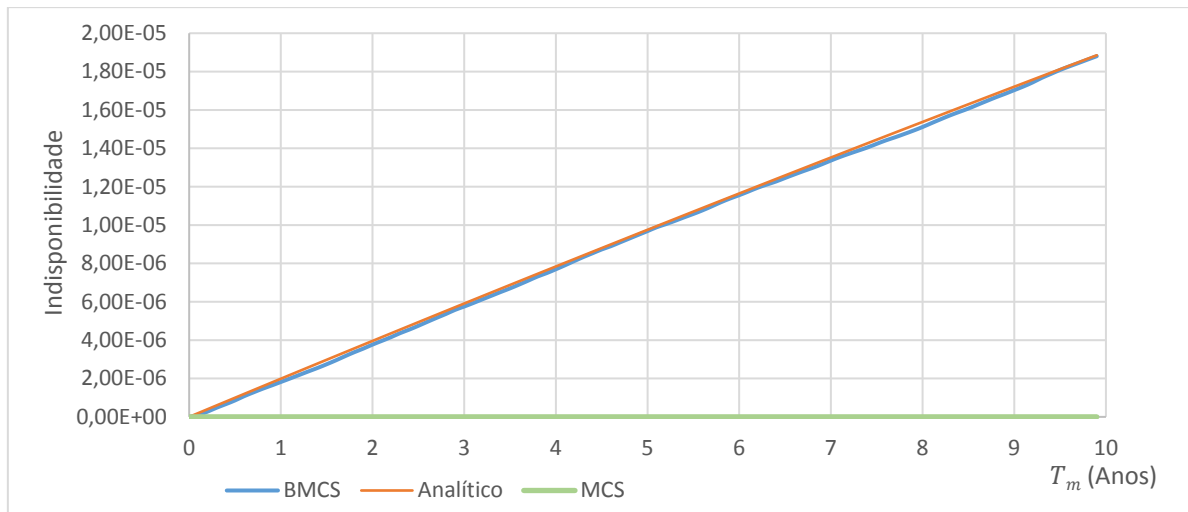


Figura 4.1 – Comparação entre métodos ao nível de 10^3 rodadas

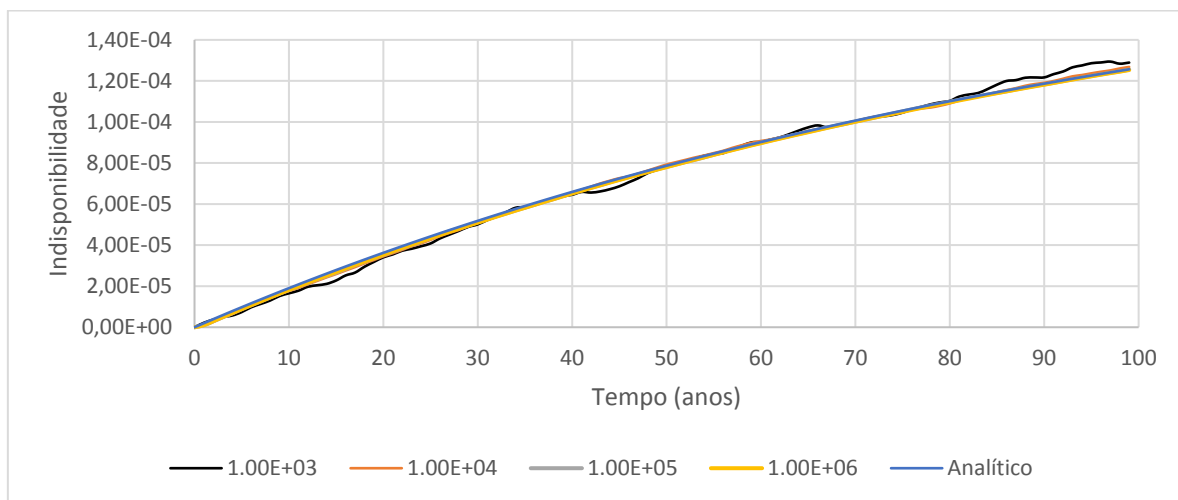


Figura 4.2 – MCS por número de rodadas vs. resultado analítico

Na tabela 4.1 é possível observar que a um mesmo nível de rodadas, a BMCS possui um tempo de processamento superior a MCS a um mesmo nível de rodadas. Isso ocorre devido às transações de estado serem forçados a manter a simulação em um *looping* enquanto o sistema não entrar em estado de falha ao menos uma vez. Desta forma, cada rodada da BMCS se estende por mais tempo em relação ao método tradicional. Por outro lado, a BMCS necessita de um volume reduzido de rodadas tendo em vista que o *looping* força pelo menos uma falha ao longo da simulação, o que se reflete em um número de falhas por rodada superior ao MCS. A consequência do maior número de registro de falhas é a convergência mais rápida para zero do desvio entre o resultado da simulação ao resultado analítico.

Tabela 4.1 – Comparação MCS e BMCS

Rodadas	Tempo (s)		Desvio		Falhas por rodada	
	MCS	BMCS	MCS	BMCS	MCS	BMCS
10^3	0.0857667	0.314461	2.45E-04	2.57E-06	0.001	1.42
10^4	0.648089	1.86263	7.69E-05	9.82E-07	0.0003	1.4204
10^5	6.70952	17.2603	2.81E-05	6.73E-07	0.00024	1.42672
10^6	67.1263	185.152	2.39E-05	7.14E-07	0.000174	1.428566

Embora para uma simulação simples exista solução analítica, casos complexos com distribuições diferentes da exponencial, ou que possuam necessidades de interferências na simulação, isto é, uma substituição do equipamento por taxas de falha que dependem do tempo, ou simular o comportamento humano no reparo de peças, torna os resultados analíticos extremamente complexos. As simulações, portanto, são alternativas práticas na abordagem da análise de risco perante a análise RAM. A Figura 4.3 ilustra a possibilidade de simulação de dois componentes idênticos e reparáveis associados em série, com distribuições de transição Weibull. Os parâmetros adotados para o mesmo sistema descrito anteriormente são $\alpha_{0i \rightarrow 1i}^i = 10^6$ anos, $\beta_{0i \rightarrow 1i}^i = 1$ para a distribuição do tempo até a falha e $\alpha_{1i \rightarrow 0i}^i = 10^2$ ano, $\beta_{1i \rightarrow 0i}^i = 2$ para a distribuição do tempo até o reparo.

Na Figura 4.3 a MCS tradicional apresenta resultados pouco refinados, com variações bruscas na estimativa da indisponibilidade ao longo do tempo. O baixo refino na resposta numérica acontece por conta da baixa capacidade da MCS de capturar eventos cuja probabilidade é da ordem de grandeza de $1/N_{rod}$ ou menores. Logo, embora o aumento do número de rodadas por simulação aproxime a estimativa da indisponibilidade ao resultado analítico, a convergência é lenta, o que culmina num alto custo de processamento computacional. A BMCS, por outro lado apresenta uma convergência acelerada ao resultado analítico, sem limitações efetivamente no refino da estimativa da indisponibilidade. Isso porque a indisponibilidade é multiplicada por FA que não possui restrições de valor.

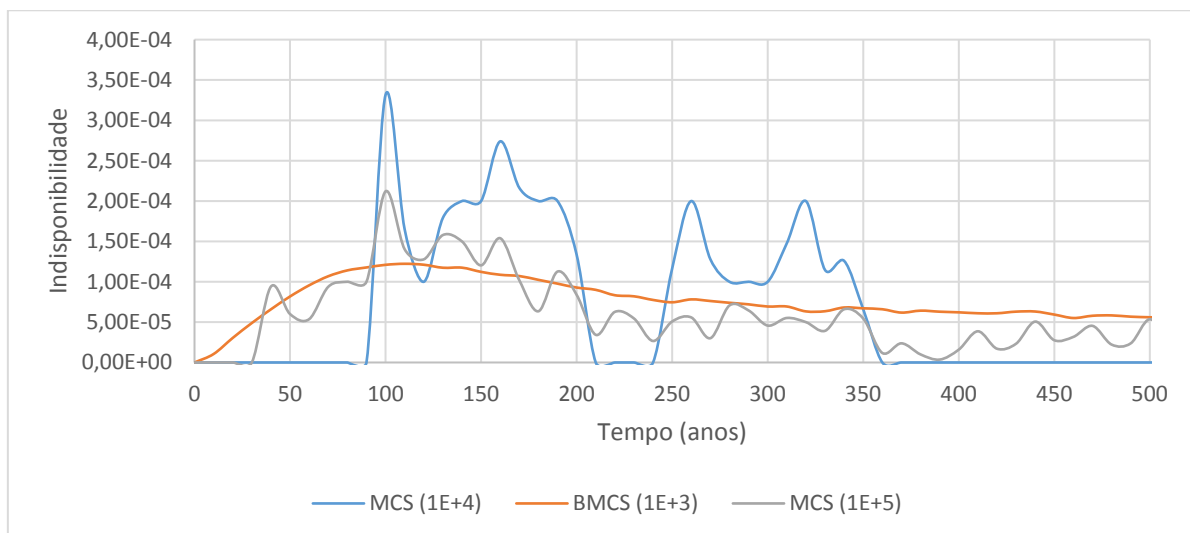


Figura 4.3 – Comportamento dos resultados de MCS e BMCS perante a distribuição Weibull

Embora os resultados do BMCS apresentem um desvio do resultado analítico inferior aos resultados do MCS ao mesmo nível de rodadas, o tempo de processamento por rodada do BMCS é maior. Isso acontece pela estrutura da BMCS que se mantém no loop forçadamente até que haja ao menos uma falha, o que garante um número de falhas por rodada substancialmente maior do que o MCS, o que viabiliza o menor desvio. Embora o tempo de processamento por rodada do BMCS seja superior ao MCS, é necessário um número muito maior de rodadas para o MCS alcançar o grau de refino alcançado pela simulação enviesada. Portanto a BMCS alcança seu objetivo de eficiência em reduzir a necessidade de processamento.

5. Considerações finais

A revisão da literatura proposta neste trabalho focou-se na execução do BMCS. A revisão sugere que a simulação reduz consideravelmente a necessidade do uso de memória e processamento computacional, otimizando o desempenho de uma simulação por meio da redução da necessidade do número de rodadas para refinar a resposta numérica da Simulação de Monte Carlo. Se conclui que para todas os processos em que há uma distribuição acumulada de probabilidade, é possível inserir o enviesamento por meio de uma função invertível.

6. Agradecimentos

O primeiro autor agradece ao financiamento oriundo da fundação de apoio à Universidade de São Paulo (FUSP).

7. Referências

- [1] MARTINS, M. R. *Considerações sobre análise de confiabilidade e risco*. São Paulo: Escola Politécnica da Universidade de São Paulo, 2013.
- [2] ZIO, E. *The Monte Carlo simulation method for system reliability and risk analysis*. London: Springer, 2013.
- [3] MARSEGUERRA, M.; ZIO, E.; CADINI, F. *Biased Monte Carlo unavailability analysis for systems with time-dependent failure rates*. Italy: Elsevier, 2001
- [4] MODARRES, M.; KAMINSKIY, M.; KRIVSTOV, V. *Reliability engineering and risk analysis: a practical guide*. New York: Marcel Dekker, 1999.
- [5] RUBINO, G.; TUFFIN, B. *Rare event simulation*. United Kingdom: Wiley, 2009.