



Fundação Universidade Federal
do ABC Pró reitoria de pesquisa
Av. dos Estados, 5001, Santa Terezinha, Santo
André/SP, CEP 09210-580, Bloco L, 3ºAndar, Fone
(11) 3356-7617

Projeto de Iniciação Científica do
PIBIC referente ao Edital:
01/2021

Nome do Aluno: Vitória Marques da Silva

Nome do Orientador: Marcelo Araujo da Silva

Título do projeto: Confiabilidade do Pouso em Marte do Rover Perseverance

Palavras-chave do projeto: Otimização, Confiabilidade, Rover Perseverance, Probabilidade.

Área do conhecimento do projeto: Engenharia Aeroespacial.

Bolsista: Sim.

São Bernardo do Campo

2022

São Bernardo do Campo, 05 de outubro de 2022.

À Pró-Reitora de Pesquisa
Prof. Dr. Wagner Alves Carvalho

Ref.: Relatório Final da Aluna Vitória Marques da Silva (RA: 11202020231)

Prezado Professor:

Encaminho o relatório do(a) aluno(a) ***Vitória Marques da Silva*** referente ao projeto de pesquisa junto ao programa de Iniciação Científica na modalidade (***PIBIC***) no edital 01/2021.

Como Professor do Curso de Engenharia Aeroespacial, do Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas - CECS, da Universidade Federal do ABC - UFABC, informo que orientei as atividades de pesquisa da Aluna no período de vigência do referido edital, cujo tema foi “Confiabilidade do Pouso em Marte do Rover Perseverance”.

A aluna foi assídua, pontual, proativa, de ótimo relacionamento e extremamente dedicada ao projeto de pesquisa, obtendo bons resultados. A aluna contemplou com êxito todas as etapas inicialmente propostas.

Do ponto de vista acadêmico, a aluna teve um bom desempenho nas disciplinas cursadas.

Atenciosamente,



Prof. Dr. Marcelo Araujo da Silva

Sumário

Resumo	4
1. Introdução	5
2. Fundamentação teórica	6
3. Metodologia	7
3.1 Materiais e métodos	9
3.1.1 Otimização com o GRG	9
3.1.2 Processo de Monte Carlo	10
3.2 Etapas da pesquisa	10
4. Resultado e discussão dos resultados	10
4.1 Distribuições do tipo normal	10
4.2 Distribuição não normal	11
4.3 Simulação do MC em Python	13
5. Conclusão e perspectivas de trabalhos futuros	17
Referências	18

Resumo

Este projeto de pesquisa tem o propósito de estudar a confiabilidade do pouso do Rover Perseverance em Marte com uso do Processo de Monte Carlo (MC) e o Método *Generalized Reduced Gradient* (GRG). Foram usados o Solver do Excel para o MC e GRG e bibliotecas do Python para o MC a fim de comparação com os resultados já obtidos com o Solver. O MC gera projetos, através de amostras aleatórias, que serão processados para se determinar a probabilidade de falha. O GRG consiste em utilizar técnicas de otimização para se determinar a maior probabilidade de falha de um modelo matemático que representa o processo mecânico do equilíbrio de forças durante o pouso. Neste método, a função objetivo é o índice de confiabilidade e a restrição de igualdade é a função de performance. A função de performance é a equação de equilíbrio de forças durante o pouso. O índice de confiabilidade deve ser minimizado, considerando que a função de performance seja nula no ponto ótimo, denominado também de ponto mais provável. Os dados obtidos serão discutidos a fim de se entender e estudar o pouso do Rover Perseverance em Marte à partir do ponto de vista da confiabilidade.

Palavras-chave: Otimização, Confiabilidade, Rover Perseverance, Probabilidade.

1. Introdução

Missões espaciais estão sendo cada vez mais discutidas e planejadas, permitindo o avanço das comunicações via satélite e novas descobertas científicas. É evidente a importância da exploração espacial para a sociedade mundial e empresas desse setor avançam cada vez mais cientificamente na busca de meios para futuras colonizações de outros planetas, começando pelo planeta Marte. Pesquisas voltadas para uma possível viagem humana a Marte são feitas a partir de estações análogas que simulam seu habitat em alguns países e através do envio de robôs chamados de Rovers para explorar amostras físicas do planeta.

Nesse ano a NASA (National Aeronautics and Space Administration) obteve sucesso na missão do lançamento do Rover Perseverance ao planeta vermelho, com o objetivo de explorar sua habitabilidade passada e testar tecnologias para pavimentar o caminho para a futura missão tripulada. Percebe-se assim a importância dessa iniciativa para os próximos passos da humanidade no setor espacial e a necessidade de uma metodologia científica estruturada e automática na determinação de soluções ótimas de otimização, que auxilie projetistas e engenheiros na exploração mais eficiente das diversas soluções de um projeto com tanto investimento como esse em um ambiente tão adverso da Terra.

Os algoritmos de otimização se enquadram nessa situação, introduzindo estratégias numéricas na busca por soluções ótimas de engenharia. Neste aspecto as bibliotecas do Python permitem a solução de inúmeros problemas científicos com sua sintaxe acessível, excelentes ferramentas de depuração e extensas oportunidades de manipulação de gráficos que permitem ao usuário que se concentre no contexto físico e matemático do problema de otimização sem se distrair com problemas de implementação técnica. Como citado, o Python será utilizado para a solução do problema da confiabilidade do pouso em Marte na programação do MC. O Solver do Excel será utilizado tanto para o MC quanto para o GRG para igualmente calcular a confiabilidade.

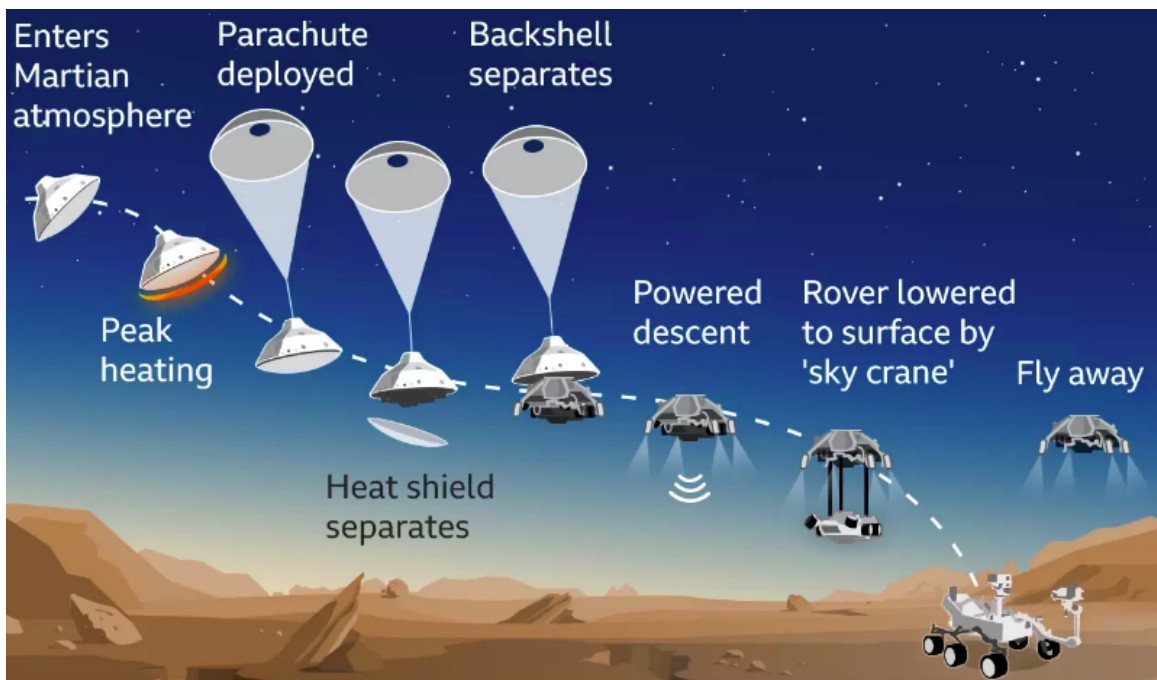


Figura 1 – Rover Perseverance pousando em Marte – Fonte: BBC

A Figura 1 apresenta os passos do processo do pouso do Rover Perseverance em Marte.

Com base nos fatos supracitados a presente pesquisa visa descobrir, utilizando o MC e o GRG, a probabilidade de falha, ou de sucesso, do pouso do Rover Perseverance em Marte. A soma da probabilidade de falha com a probabilidade de sucesso é igual a 100%. Portanto, ao se conhecer uma das duas, conhece-se a outra.

2. Fundamentação teórica

Otimização pode ser definida como um procedimento pelo qual é possível encontrar uma solução ou um conjunto de soluções ótimas para uma determinada função ou um conjunto de funções, que regem um problema específico, submetido a restrições. Para uma maior abrangência de conjunto de dados e um resultado com maior eficácia é necessária uma metodologia científica estruturada e automática na determinação de soluções ótimas. O

procedimento utilizado no presente projeto é a adoção do MC e GRG para se obter a confiabilidade do pouso do Rover Perseverance em Marte. Serão utilizados o Solver Excel e o Python.

A confiabilidade é entendida como a habilidade de equipamentos ou de seres humanos de executarem suas funções esperadas apropriadamente sob condições específicas durante um determinado período de tempo, na ausência de quebras ou falhas. É uma área de estudo que tem o objetivo de avaliar e otimizar a confiabilidade de sistemas através de técnicas provenientes da probabilidade para descrever processos e experimentos aleatórios. O estudo da confiabilidade engloba diversas disciplinas, tais como a estatística para analisar os dados e definir a média, o desvio padrão e a curva de densidade de probabilidade das variáveis aleatórias, a ciência da computação para a implementação eficiente de algoritmos e a programação matemática para formular e resolver problemas de otimização. A geração de números aleatórios (MC) com métodos computacionais é bastante utilizada para resolver problemas em finanças, engenharia e estatística. Um dos principais desafios do presente projeto é escrever os códigos relativos à confiabilidade, livrando-se de eventuais instabilidades numéricas e tempo computacional excessivo.

3. Metodologia

Para a realização deste projeto foi feito um estudo sobre os conceitos fundamentais que seriam utilizados durante a pesquisa tais como a determinação da confiabilidade utilizando-se o MC e o GRG.

Inicialmente foram definidas a função de performance que é a equação de equilíbrio em função das variáveis aleatórias. A função de performance é a diferença entre as forças que evitam do deslocamento vertical descendente do Rover e aquelas que impulsionam a nave contra o solo. Tem-se a seguinte função de performance:

$$F = D - W + E = (1/2) p S V^2 C_{d-W+E} \quad (1)$$

Observe que se a função de performance for positiva o pouso será seguro, pois se tem mais frenagem que aceleração, caso seja negativa, o pouso será inseguro e caso seja nula o Rover descerá para o solo numa velocidade constante que pode ser nociva ao processo. Nesta equação, tem-se:

- D , força de arrasto;
- W , força peso do Rover;
- ρ , densidade atmosférica;
- S , área de projeção do paraquedas;
- V , velocidade do pouso;
- E , força dos retrofoguetes;
- C_d , coeficiente de arrasto do paraquedas.

A função de performance pode ser escrita em função das variáveis aleatórias X_i como sendo:

$$F(X) = (1/2) X_1 \cdot X_2 \cdot X_3^2 \cdot X_4 - X_5 + X_6 \quad (2)$$

Na Tabela 1 então são descritos os valores adotados para cada uma das variáveis aleatórias.

As variáveis aleatórias X podem ser escritas em termos das variáveis aleatórias reduzidas (PACHÁS, 2009), que são denominadas Y . A relação entre elas é escrita como:

$$X_i = \sigma_i Y_i + \mu_i \quad (3)$$

ou

$$Y_i = (X_i - \mu_i) / \sigma_i \quad (4)$$

onde μ_i e σ_i são respectivamente a média e o desvio padrão da variável

aleatória X_i .

Tabela 1 - Dados Numéricos

Variável	Descrição	Unidade	Média	Desvio Padrão	Coef de Variação
X1	Densidade Atmosférica	kg/m³	0,014555	0,000011	0,08%
X2	Area do paraquedas	m²	363,05	1,82	0,50%
X3	Velocidade de pouso	m/s	0,75	0,00375	0,50%
X4	Coeficiente de arrasto	-	1,072	0,05	4,66%
X5	Peso em Marte	N	3792,5	19	0,50%
X6	Retrofoguetes	N	3800	19	0,50%

Utilizando-se os valores da Tabela 1, aplicou-se o GRG e o MC no Excel e também o MC em Python para o cálculo da confiabilidade. É válido ressaltar que os valores da Tabela 1 foram obtidos através de pesquisa nos sites descritos nas referências bibliográficas. Pesquisas mais rigorosas devem ser feitas para aprimorar a precisão destes dados de entrada.

3.1 Materiais e métodos

3.1.1 Otimização com o GRG

O índice de confiabilidade (β) representa a menor distância da origem até um ponto do espaço das variáveis aleatórias reduzidas (Brasil e Silva, 2018), sobre a função de performance nula.

O problema de otimização pode ser definido como:

Determine $\mathbf{Y} \in \mathbb{R}^n$ que minimize a função

$$\beta = (\mathbf{Y}^T \mathbf{Y})^{1/2} \quad (5)$$

Sujeito a

$$F(\mathbf{Y}) = 0 \quad (6)$$

A probabilidade de falha é $P_f = \Phi(-\beta)$, onde Φ é a função de probabilidade acumulada da distribuição normal padrão. Como citado, utiliza-se o GRG no Solver do Excel para resolver o problema de otimização. É válido lembrar que o

problema definido pelas equações (5) e (6) pode também ser formulado em função das variáveis \mathbf{X} .

3.1.2 Processo de Monte Carlo

O Processo de Monte Carlo gera vários projetos com a utilização da geração de números aleatórios. Para um dado projeto \mathbf{X} , o valor de $F(\mathbf{X})$ é calculado. Para m projetos, calcula-se a média da função de performance μ_F e o desvio padrão da função de performance σ_F . O índice de confiabilidade é então definido como a razão:

$$\beta = \mu_F / \sigma_F \quad (7)$$

A probabilidade de falha é dada por

$$P_f = n(F < 0) / m, \quad (8)$$

ou seja, o número de vezes que a função de performance for negativa, dividida pelo número de projetos gerados. O MC será rodado no Solver do Excel e também em Python.

3.2 Etapas da pesquisa

Para realização deste projeto primeiramente foi realizada uma revisão bibliográfica. Posteriormente a implementação dos códigos no Excel e em Python. Foram resolvidos exemplos numéricos e por último escrito o relatório.

4. Resultado e discussão dos resultados

4.1 Distribuições do tipo normal

Inicialmente consideramos **todas as distribuições como normais** e foi utilizado o GRG e MC no Excel. Nesta primeira formulação foram gerados 1000 projetos no MC. **Numa primeira análise, a variável X_6 (retrofoguetes) foi negligenciada.** Os resultados obtidos são mostrados a seguir.

Utilizando o Método GRG os seguintes resultados foram obtidos:

$$(\beta) = -200$$

$$(Pf) = 100\%$$

Utilizando o Método de Monte Carlo foram obtidos os seguintes resultados:

$$(\beta) = -210$$

$$(Pf) = 100\%$$

Os resultados obtidos nesta abordagem mostram que sem os retrofoguetes o projeto falharia absurdamente, não tendo nenhuma chance de sucesso.

Numa segunda formulação, agora **considerando a contribuição dos retrofoguetes**, tem-se os seguintes resultados.

Utilizando o Método GRG:

$$(\beta) = 0,34$$

$$(Pf) = 37\%$$

Utilizando o Método de Monte Carlo:

$$(\beta) = 0,37$$

$$(Pf) = 35\%$$

Observa-se que com o uso dos retrofoguetes, o projeto mostra uma probabilidade de sucesso de 65%. Embora ainda seja baixa, mas já mostra um caminho relativamente viável.

Os resultados obtidos com e sem os retrofoguetes são mostrados na Tabela 2.

Tabela 2 – Comparação dos resultados com e sem os retrofoguetes

	Análise Inicial		Utilizando Retrofoguetes	
	GRG	Monte Carlo	GRG	Monte Carlo
β	-200	-210	0,34	0,37
Pf	100%	100%	37%	35%

4.2 Distribuição não normal

Nesta segunda simulação, para velocidade de pouso (X_3) foi adotada a distribuição de **extremos inferiores do tipo I** e para o coeficiente de arrasto

(X_4), foi adotada a distribuição de **extremos superiores do tipo I**. As variáveis X_2 , X_5 e X_6 continuaram normais. Essas simulações foram realizadas para se verificar a sensibilidade do problema em relação ao tipo de distribuição adotada. Foram gerados 10.000 projetos no MC para melhorar a precisão. Tanto o MC quanto o GRG ainda estão sendo modelados no Excel até este ponto do projeto.

As expressões para as distribuições de Extremos Tipo I são mostradas a seguir, onde $p(x)$ é a função densidade de probabilidade e $P(x)$ é a probabilidade acumulada.

Máximo:

$$p(x) = \alpha \exp^{(-\alpha(x-u) - \exp(-\alpha(x-u)))} \text{ Equação 5}$$

$$P(x) = \exp(-\exp(\alpha(x-u))) \text{ Equação 6}$$

Mínimo:

$$p(x) \propto \exp(\alpha(x-u) - \exp(\alpha(x-u))) \text{ Equação 7}$$

$$P(x) = 1 - \exp(-\exp(\alpha(x-u))) \text{ Equação 8}$$

Neste caso é necessário o cálculo da normal equivalente (PACHÁS, 2009) no GRG.

Aplicando as novas alterações no GRG e no Monte Carlo obtivemos estes resultados.

Utilizando o Método GRG os seguintes resultados foram obtidos:

$$(\beta) = 0,34$$

$$(Pf) = 37\%$$

Utilizando o Método de Monte Carlo foram obtidos os seguintes resultados:

$$(\beta) = 0,36$$

$$(Pf) = 37\%$$

É observável que os valores do beta de ambos ficaram bem próximos e também batem com a formulação considerando todas as variáveis como sendo normal. Ou seja, o problema não se mostrou sensível com a alteração de algumas variáveis do tipo normal para extremos tipo I.

Os projetos obtidos no MC nas simulações do Excel propiciaram a criação das curvas de densidade de probabilidade mostradas a seguir.

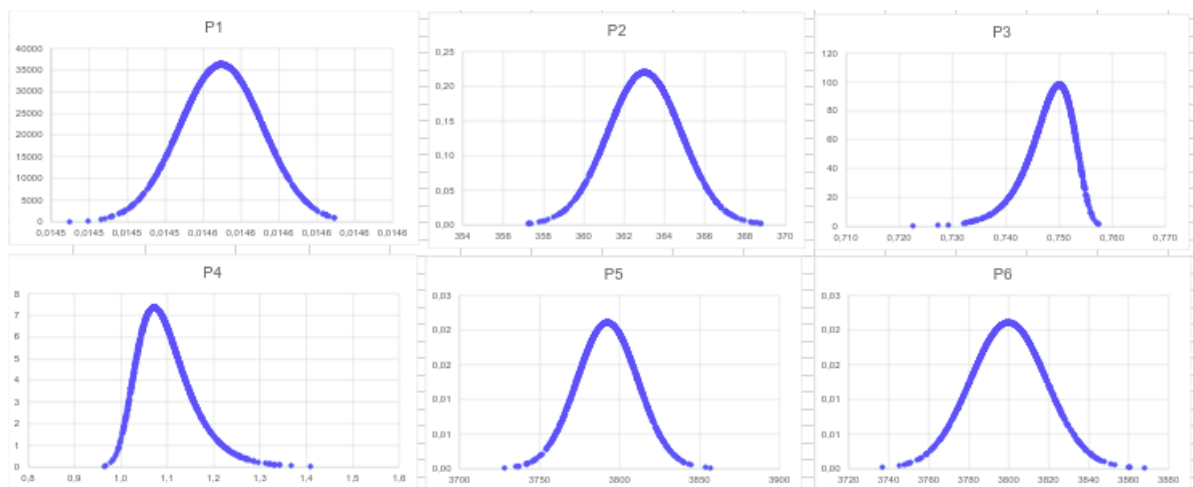


Figura 2 – Curvas de densidade de probabilidade obtidas à partir dos projetos gerados no MC no Excel (10.000 projetos gerados)

Observe que no caso de p3 e p4 as curvas são assimétricas, como era de se esperar no caso das distribuições de extremos tipo I. As demais permanecem simétricas e bem coerente com a distribuição normal.

4.3 Simulação do MC em Python

Para se ter mais dados de comparação, a mesma formulação apresentada para o MC na Seção 4.2 é agora programa em Python com 100.000 projetos gerados.

O código de programação para o MC é mostrado nas Figura 3 a 6.

```
[ ] def dist_tipol_min(valor, media, desvio):
    a = 1/desvio
    b = math.e
    c = ((valor - media)/desvio)
    resultado = a*(b**(c-b**c))
    return(resultado)

def dist_tipol_max(valor, media, desvio):
    a = 1/desvio
    b = math.e
    c = ((valor - media)/desvio) * -1
    resultado = a*(b**(c-b**c))
    return(resultado)
```

Figura 3 – Código em Python para a geração de números aleatórios das variáveis de extremos tipo I no MC

```
[ ] def dist_normal(valor, media, desvio):
    a = 1 / (desvio*math.sqrt(2*math.pi))
    b = math.e
    c = -(1/2)*(((valor-media)/desvio)**2)
    resultado = a*(b**c)
    return(resultado)
```

Figura 4 – Código em Python para a geração de números aleatórios das variáveis normal no MC

```
[ ] # Calcula a função de falha para cada ponto (cada pouso) considerando
# as variáveis aleatórias simuladas
vetor_func_falha = []
for i in range(0, qtd_pontos):
    x1, x2, x3 = pontos_x1[i], pontos_x2[i], pontos_x3[i]
    x4, x5, x6 = pontos_x4[i], pontos_x5[i], pontos_x6[i]
    f = (0.5*x1) * x2 * x3 * x4 - x5 + x6
    vetor_func_falha.append(f)

[ ] # Calcula a média, o desvio padrão e o  $\beta$  das funções de falha
media_f = numpy.mean(vetor_func_falha)
dvp_f = numpy.std(vetor_func_falha)
b = media_f/dvp_f
print(f'μ = {media_f:.3f}')
print(f'σ = {dvp_f:.3f}')
print(f'β = {b:.3f}')
```

Figura 5 – Código em Python para o cômputo da função de performance e o índice de confiabilidade β no MC

```
[ ] # Calcula a probabilidade de falha
cont = 0
for i in range(0, len(vetor_func_falha)):
    if vetor_func_falha[i] < 0:
        cont += 1
prob_falha = cont/qtd_pontos*100
print(f"Pf = {prob_falha:.2f}%")
```

Figura 6 – Código em Python para o cômputo da probabilidade de falha no MC

As curvas de densidade de probabilidade obtidas na programação em Python são mostradas na Figura 7. Observa-se que estas curvas são bem semelhantes àsquelas mostradas na Figura 2.

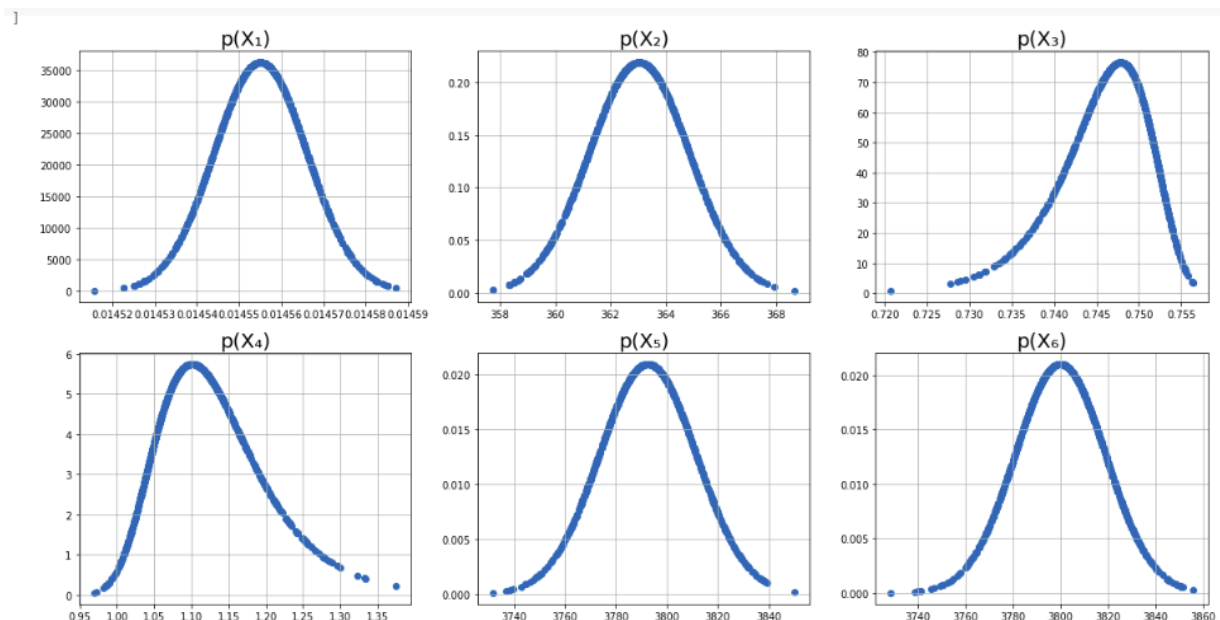


Figura 7 – Curvas de densidade de probabilidade obtidas à partir dos projetos gerados no MC no Python (100.000 projetos gerados)

Utilizando o Python no MC os seguintes resultados foram obtidos:

$(\beta) = 0,36$

$(Pf) = 36\%$

Dados que se aproximam muito do resultado obtido pelo GRG e MC (com 10.000 projetos gerados) no Excel.

Na tabela a seguir é possível visualizar todos os resultados obtidos no decorrer da pesquisa e compará-los de forma visual:

Tabela 3 – Comparação de todos os resultados obtidos

	Sem Retrofoguetes		Com Retrofoguetes	
	Excel			
	Distribuição Normal			
	GRG	MC	GRG	MC*
β	-200	-210	0,34	0,37
P_f	100%	100%	37%	35%
	Excel			
	Distribuição Não Normal			
			GRG	MC**
β			0,34	0,36
P_f			37%	37%
	Python			
	Distribuição Não Normal			
				MC***
β				0,34
P_f				37%
* COM 1.000 PROJETOS				
** COM 10.000 PROJETOS				
*** COM 100.000 PROJETOS				

5. Conclusão e perspectivas de trabalhos futuros

Ao longo da pesquisa foram implementados o MC, tanto no Excel quanto no Python, e o GRG, implementado no Excel. Foram realizadas várias simulações e os resultados foram bem coerentes de uma para a outra. Observou-se que o problema não apresentou uma sensibilidade significativa ao se considerar as variáveis normal ou não normal. Outra questão foi o número de projetos gerados no MC. No Excel, gerou-se 1.000 e 10.000 projetos, enquanto em Python 100.000 projetos. Mais uma vez, a diferença entre os valores obtidos com diferentes números de projetos não variou significativamente.

O trabalho mostrou, baseando-se nos dados adotados, que sem os retrofoguetes o pouso seguro em Marte seria praticamente impossível na prática. Já com os retrofoguetes, a probabilidade de sucesso é da ordem de 63%: Mesmo assim, entende-se que esta probabilidade de sucesso é ainda baixa. Uma pesquisa para se apurar com mais precisão os dados de input obtidos fica como sugestão de futuros trabalhos, assim como uma nova formulação para a equação de performance.

Referências

- [1] ERICKSON, K., **The Mars Rovers**. NASA Space Place, [s. /], 23 mar. 2021. Disponível em: <https://spaceplace.nasa.gov/mars-rovers/en/#:~:text=The%20names%20of%20the%20five,%2C%20gullies%2C%20and%20flat%20plains>. Último Acesso em: 30 out 2022.
- [2] **NASA SCIENCE MARS**. 2020 Mission Perseverance Rover. EUA, 2021. Disponível em: <https://mars.nasa.gov/mars2020>. Último Acesso em: 30 out 2022
- [3] SHINZATO, D., ARMANI, E., STRACERI, L. S. e ETTORI, R., Seminário apresentado na disciplina de Confiabilidade de Componentes e Sistemas no primeiro quadrimestre de 2021.
- [4] AMOS, J., **Nasa Mars rover: Perseverance robot all set for big test**. BBC News, [s. /], 18 fev. 2021. Disponível em: <https://www.bbc.com/news/science-environment-56103231>. Último Acesso em: 30 out 2022.
- [5] GONTIJO, I., **A Caminho de Marte**. Rio de Janeiro, Sextante, 2018.
- [6] HALL, N., **Mars Atmosphere Model**. NASA, [s. /], 13 maio 2021. Disponível em: <https://www.grc.nasa.gov/www/k-12/airplane/atmosmrm.html>. Último Acesso em: 30 out 2022.
- [7] Verges, A. and Braun, R., **“Evaluation of the Mars Pathfinder Parachute Drag Coefficient”**, 2007.
- [8] PACHÁS, M. A. C., **Análise Limite com Otimizador de Grande Escala e Análise de Confiabilidade**, Tese de Doutorado, PUC/RJ, 2009.
- [9] MARKLUND, H., **Supersonic Retro Propulsion Flight Vehicle Engineering of a Human Mission to Mars**. Luleå University of Technology. Department of Computer Science, Electrical and Space Engineering, 2019.
- [10] Nelessen, A., **“Mars 2020 Entry, Descent, and Landing System Overview,”** 2019 IEEE Aerospace Conference, Big Sky, MT, USA, pp. 1-20, 2019.

[11] BRASIL, R. M. L. R. F e SILVA, M. A., **Otimização de Projetos de Engenharia**. Editora Edgard Blucher, São Paulo, 2018.

[12] Jin, G., **Monte Carlo Finite Element Method of Structure Reliability Analysis**. Reliability engineering & systems safety [0951-8320] Ano:1993 v.:40 n.:1 p.:77 -83, 1993.