



Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Engenharia Elétrica



Trabalho de Conclusão de Curso

Análise da Confiabilidade na Rede de Distribuição de Energia Elétrica

Miriele dos Santos Campos

João Monlevade, MG
2024

Miriele dos Santos Campos

Análise da Confiabilidade na Rede de Distribuição de Energia Elétrica

Trabalho de Conclusão de curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.

Orientador: Prof. Dr. Thainan Santos Theodoro

Universidade Federal de Ouro Preto
João Monlevade
2024

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

C198a Campos, Miriele dos Santos.
Análise da confiabilidade na rede de distribuição de energia elétrica.
[manuscrito] / Miriele dos Santos Campos. - 2024.
81 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Thainan Santos Theodoro.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia
Elétrica .

1. Energia elétrica - Distribuição. 2. Monte Carlo, Método de. 3.
Processo estocástico. 4. Redes elétricas - Analisadores. 5. Simulação
(Computadores). 6. Sistemas de energia elétrica - Confiabilidade. 7.
Sistemas de energia elétrica - Estabilidade. I. Theodoro, Thainan Santos.
II. Universidade Federal de Ouro Preto. III. Título.

CDU 621.31:519.21

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



FOLHA DE APROVAÇÃO

Miriele dos Santos Campos

Análise da confiabilidade na rede de distribuição de energia elétrica

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica

Aprovada em 14 de outubro de 2024

Membros da banca

Prof. Dr. Thainan Santos Theodoro - Departamento de Engenharia Elétrica - Ufop – Orientador
Prof. Dr. Wilington Guerra Zvietcovich - Departamento de Engenharia Elétrica - Ufop - Convidado
Prof. Dr. Francisco Ricardo Abrantes Couy Baracho - Departamento de Engenharia Elétrica - Ufop - Convidado

Thainan Santos Theodoro, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 17/10/2024.



Documento assinado eletronicamente por **Thainan Santos Theodoro, PROFESSOR DE MAGISTERIO SUPERIOR**, em 18/10/2024, às 17:12, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0797792** e o código CRC **9D283E11**.

Agradecimentos

Primeiramente, elevo meus agradecimentos a Deus, fonte de toda a sabedoria e amor. Agradeço de forma especial ao meu orientador, Thainan, suas orientações precisas e seu apoio constante foram pilares fundamentais em todo o processo de aprendizagem, cuja expertise, personalidade e dedicação foram essenciais para a realização deste trabalho. Agradeço à banca examinadora, pelo tempo dedicado, pela avaliação cuidadosa e contribuições que enriqueceram este estudo. Agradeço meu Marido, pelo amor, compreensão e apoio. Aos meus pais, irmãos, primos e tios por cada sacrifício feito e cada palavra para que eu não desistisse e pudesse alcançar este grande objetivo. A todos vocês, meu sincero obrigado!

"A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original."

– Albert Einstein

Resumo

O sistema de distribuição de energia elétrica exige alta confiabilidade, ou seja, fornecimento contínuo, seguro e de qualidade, com mínima interrupção. Neste cenário, estudos preditivos matemáticos são essenciais para avaliar a confiabilidade, destacando-se os métodos analíticos e estocásticos. O objetivo deste trabalho é realizar uma análise preliminar das vantagens e desvantagens da Simulação Método Analítico (SMA) e da Simulação Monte Carlo Sequencial (SMCS) para a confiabilidade de sistemas de distribuição, aplicados a um sistema elétrico teste de 5 barras. Foram desenvolvidos algoritmos para ambas as metodologias, implementados computacionalmente, gerando índices de confiabilidade que permitem comparações de precisão. Os resultados confirmam a efetividade dos métodos e sugerem a relevância das informações fornecidas pelo SMCS no planejamento de redes de distribuição. Dos resultados, a SMA apresenta menor custo computacional, porém não é recomendada para grandes sistemas, enquanto o SMCS consegue simular diversas funções de probabilidade, embora tenha maior custo computacional.

Palavras-chave: Confiabilidade, Rede Elétrica de Distribuição, Monte Carlo sequencial, Sistemas de Distribuição, Método analítico.

Abstract

The Electrical Power System Sistema Elétrico de Potência (SEP) requires high reliability, that is, continuous, safe, and high-quality supply, with minimal interruptions. In this scenario, mathematical predictive studies are essential for evaluating availability, with analytical and stochastic methods standing out. The objective of this work is to carry out a preliminary analysis of the advantages and disadvantages of the SMA and SMCS for distribution system reliability, applied to a 5-bus test system. Algorithms for both methodologies were developed and computationally implemented, generating reliability indices that allow for accuracy comparisons. The results confirm the effectiveness of the methods and suggest the relevance of the information provided by SMCS in distribution network planning. SMA has a lower computational cost but is not recommended for large systems, while SMCS can simulate various probability functions, although with a higher computational cost.

Keywords: Reliability, Electrical Power System, Sequential Monte Carlo, Distribution Systems, Analytical Method.

Lista de ilustrações

Figura 1 – Níveis hierárquicos dos SEPs.	8
Figura 2 – Probabilidade de funcionamento $R(t)$ para alguns valores de taxa de falha (λ) ao longo de 20 anos. Os valores de λ estão em ocorrências/ano.	10
Figura 3 – Modelo a dois estados de um componente - tempo de operação e tempo de reparo.	10
Figura 4 – Representação do sistema de distribuição radial para as 4 configurações.	20
Figura 5 – Diagrama de blocos da SMA para análise de confiabilidade.	21
Figura 6 – Diagrama de blocos da SMCS para análise de confiabilidade.	25
Figura 7 – Histogramas de índices de confiabilidade para a SMCS.	32
Figura 8 – Representação do sistema radial como grafo orientado.	46
Figura 9 – Fluxograma para criação da matriz <code>seccionadora_bds</code>	49
Figura 10 – Fluxograma para criação da matriz <code>alimentadora_bda</code>	52

Lista de Siglas

AENS	Energia Média Não Fornecida (do inglês, <i>Average Energy Not Supplied</i>)
ANEEL	Agência Nacional de Energia Elétrica
ASAI	Índice Médio de Disponibilidade de Serviço (do inglês, <i>Average Service Availability Index</i>)
ASUI	Índice Médio de Indisponibilidade de Serviço (do inglês, <i>Average Service Unavailability index</i>)
BDP	Barramento dos protetores
BDS	Barramento das seccionadoras
BDA	Barramento das alimentações
CAIDI	Índice Médio de Duração de Interrupção por Cliente (do inglês, <i>Customer Average Interruption Duration Index</i>)
CDF	Função de Distribuição Acumulada (do inglês, <i>Cumulative Distribution Function</i>)
DEC	Duração Equivalente de Interrupção por Consumidor
DMIC	Duração Máxima de Interrupção por Consumidor
ENS	Energia Não Fornecida (do inglês, <i>Energy Not Supplied</i>)
FEC	Frequência Equivalente de Interrupção por Consumidor
MTTR	Tempo Médio para Reparo (do inglês, <i>Mean Time to Repair</i>)
MTTF	Tempo Médio para a falha (do inglês, <i>Mean Time to Failure</i>)
NH	Níveis Hierárquicos
NH0	Nível Hierárquico 0
NH1	Nível Hierárquico 1
NH2	Nível Hierárquico 2
NH3	Nível Hierárquico 3
SAIDI	Índice Médio de Duração de Interrupção do Sistema (do inglês, <i>System Average Interruption Duration Index</i>)

SAIFI	Índice Médio de Frequência de Interrupção do Sistema (do inglês, <i>System Average Interruption Frequency Index</i>)
SEP	Sistema Elétrico de Potência
SMA	Simulação Método Analítico
SMC	Simulação Monte Carlo
SMCS	Simulação Monte Carlo Sequencial

Sumário

1	INTRODUÇÃO	1
1.1	Contextualização e Motivação	1
1.2	Problema de Pesquisa	2
1.3	Estado da arte	2
1.4	Objetivos	4
1.5	Justificativa	4
1.6	Estrutura do Trabalho	5
2	FUNDAMENTOS TEÓRICOS	6
2.1	Sistemas de Distribuição de Energia Elétrica	6
2.2	Método Analítico	8
2.2.1	Método analítico de sistemas de distribuição para um único componente	9
2.2.2	Método analítico para sistemas de distribuição com múltiplos componentes	11
2.3	Método Monte Carlo	13
2.3.1	Método de Monte Carlo Sequencial	14
2.3.2	Índices de confiabilidade	16
2.3.3	Funções-testes	17
2.4	Conclusões parciais	18
3	DESENVOLVIMENTO	19
3.1	Sistema teste	19
3.2	Algoritmo analítico	21
3.3	Algoritmo de Monte Carlo Sequencial	25
3.4	Conclusões parciais	28
4	RESULTADOS	29
4.1	Dados do sistema	29
4.2	Resultados de simulação	30
4.3	Discussão dos resultados	31
4.4	Conclusões parciais	33
5	CONCLUSÃO	34
	REFERÊNCIAS	36

APÊNDICES **38**

APÊNDICE A – ÍNDICES DE CONFIABILIDADE DO MÉTODO ANALÍTICO **39**

A.1	Cálculo dos índices de confiabilidade para o caso 1 usando método analítico	39
A.2	Calculo dos índices de confiabilidade para o caso 2 usando método analítico	40
A.3	Calculo dos índices de confiabilidade para o caso 3 usando método analítico	41
A.4	Calculo dos índices de confiabilidade para o caso 4 usando método analítico	42
A.5	Validação da literatura	42

APÊNDICE B – ALGORITMO MÉTODO ANALÍTICO **45**

B.1	Detalhes de implementação	45
B.1.1	Inserção dos dados do sistema teste	45
B.1.2	Matrizes auxiliares	45
B.1.2.0.1	Matriz de Adjacência A	45
B.1.3	Matriz de Alcance R	47
B.1.4	Matriz de protetor	48
B.1.5	Matriz de seccionadora	49
B.1.6	Matriz de alimentação alternativa	51
B.2	Código	54

APÊNDICE C – ALGORITMO MÉTODO DE MONTE CARLO . . **61**

C.1	Passo 1: Iniciar todos os componentes no estado de sucesso	61
C.2	Passo 2: Gerar a duração no estado de sucesso	61
C.3	Passo 3: Identificar o menor tempo e o respectivo componente . . .	61
C.4	Passo 4: Avaliar o estado do componente e o impacto para os consumidores	62
C.4.1	Identificação do estado anterior	62
C.4.2	Identificação dos nós	62
C.4.3	Inicialização das Matrizes BDP, BDS e BDA	62
C.4.4	Atualização das Matrizes	63
C.4.5	Cálculo do Estado da Carga	64
C.4.6	Impacto da falha no sistema	64
C.5	Passo 5: Incrementar as funções-teste dos índices de confiabilidade	65
C.5.1	Indicadores de confiabilidade	65
C.5.2	Acumulação de Impactos ao Longo da Simulação	65

C.5.3	Passo 6: Gerar uma nova duração de estado se o período de análise de 8760 horas não acabou, caso contrario, calcular os índices de confiabilidade para o período correspondente	66
C.5.4	Passo 7: Avaliar o coeficiente de variação para todos os índices de confiabilidade	66
C.6	Código	66

1 Introdução

1.1 Contextualização e Motivação

A energia elétrica é essencial na sociedade moderna, sendo crucial para o desenvolvimento e a eficiência de diversos setores. Os sistemas elétricos, projetados para converter, transportar e distribuir energia de forma confiável, atendem às necessidades de residências, comércios e indústrias. Eles garantem o fornecimento contínuo de energia, fundamental para o crescimento econômico, a inovação tecnológica e o bem-estar social. Dessa forma, a energia elétrica se consolida como um pilar insubstituível em nosso dia a dia.

O Sistema Elétrico de Potência (SEP) é composto por um conjunto de equipamentos e circuitos interligados que operam de maneira coordenada para atender as demandas de carga em todo o sistema. Seu objetivo principal é garantir um fornecimento de energia com qualidade, confiabilidade e continuidade, ao menor custo possível. Para otimizar sua operação, o SEP é segmentado em três zonas funcionais: geração, transmissão e distribuição. Essa divisão, conforme descrito por Monticelli e Garcia (2011), facilita a organização e aumenta a eficiência operacional. As zonas variam em função do papel que desempenham, dos equipamentos utilizados e da quantidade de energia que processam. A rede de distribuição, especificamente, responde por cerca de 90% dos problemas de confiabilidade enfrentados pelos consumidores, segundo Brown (2017).

A crescente demanda por energia, junto com a necessidade de um fornecimento confiável, impõe grandes desafios aos sistemas de distribuição. Não basta apenas dispor de energia; é essencial que seu fornecimento atenda a padrões técnicos regulados e fiscalizados pela Agência Nacional de Energia Elétrica (ANEEL), conforme destacado por Mattos et al. (2008). Confiabilidade, em termos de sistemas elétricos, refere-se à probabilidade de um dispositivo operar corretamente durante um período de tempo sob condições normais de funcionamento. Na rede de distribuição de energia, isso implica fornecer energia continuamente, minimizando interrupções que possam gerar perdas econômicas, comprometer a qualidade do serviço ou, em casos extremos, representar riscos à saúde e segurança da população (BILLINTON; ALLAN, 1992). Para avaliar a confiabilidade de um sistema de distribuição, é necessário analisar detalhadamente o desempenho de seus componentes, identificando possíveis pontos de falha. Isso é feito utilizando-se índices que medem a frequência e a duração das interrupções. Esses indicadores são cruciais para identificar áreas que necessitam de melhorias e para avaliar o impacto das falhas no desempenho geral do sistema, conforme apontado por Billinton e Allan (1984).

Apesar dos avanços na confiabilidade dos sistemas de distribuição, a escolha do método mais adequado para sua avaliação ainda apresenta desafios. Embora os métodos

analíticos sejam eficientes em termos de custo e simplicidade, eles podem não refletir a complexidade real do sistema. Em contrapartida, os métodos estocásticos, como o Simulação Monte Carlo (SMC), oferecem simulações mais detalhadas e precisas, mas exigem um custo computacional elevado. O escopo deste trabalho consiste em uma comparação preliminar entre essas duas abordagens, aplicando-as a um sistema teste radial de distribuição de 5 barras. Vale ressaltar que, devido a limitações de tempo, não foi possível testar os algoritmos em um sistema real, o que representa uma oportunidade para trabalhos futuros.

1.2 Problema de Pesquisa

A análise da confiabilidade é fundamental para disponibilidade do sistema elétrico de potência. Existem dois principais métodos computacionais para resolver esse problema: método analítico e método de Monte Carlo. Neste contexto, este trabalho busca responder quais as vantagens e desvantagens da SMA e da SMCS para análise da confiabilidade de sistemas elétricos de distribuição.

1.3 Estado da arte

Na literatura existem diversos livros, teses, dissertações e artigos que apresentam diferentes métodos e análises referente ao estudo da confiabilidade. Nesta seção, são apresentadas as principais características dos trabalhos correlatos consultados.

O artigo de Billinton e Wang (1999) tem o objetivo de melhorar o ensino da avaliação de confiabilidade de sistemas de distribuição de energia elétrica utilizando a simulação de Monte Carlo. O artigo destaca a importância da compreensão da variabilidade inerente aos índices de confiabilidade, para isso, é utilizado um programa computacional que simula o desempenho de sistemas de distribuição complexos, considerando diferentes elementos do sistema e suas configurações operacionais. Este estudo é relevante para a análise de custo/benefício da confiabilidade em clientes industriais e comerciais com funções de custo não lineares.

A monografia de Silva (2008), investiga a influência dos parâmetros de qualidade da energia elétrica, como interrupções e afundamentos de tensão, na confiabilidade de sistemas elétricos industriais. Utiliza diagramas de blocos de confiabilidade e simulações para avaliar a disponibilidade e confiabilidade do sistema, considerando diferentes níveis hierárquicos. O estudo também busca formas de aumentar a confiabilidade sem recorrer a redundâncias excessivas de equipamentos ou alterações significativas na topologia do sistema.

A tese de Nascimento (2009) propõe uma metodologia para avaliar a confiabilidade de sistemas de distribuição de energia elétrica, empregando a SMC e técnicas de computação distribuída. A tese tem como principal objetivo otimizar a eficiência computacional para

análises de confiabilidade em sistemas de distribuição de grande porte, usando uma rede de computadores com diferentes configurações, permitindo análises mais eficazes, essenciais para cumprir padrões regulatórios e evitar penalidades. A metodologia é aplicada em sistemas de teste e em redes de distribuição reais.

A dissertação de Zardo (2014) apresenta uma metodologia para estimar indicadores de confiabilidade em redes de distribuição ativas. Este estudo foca em redes que incluem geração distribuída, microgeração, e a possibilidade de formação de microrredes. A metodologia emprega a análise estocástica de Monte Carlo Sequencial, que simula eventos no sistema de forma temporal e cronológica, permitindo representar a aleatoriedade das variáveis envolvidas. O trabalho explora quatro modelos para representar as fontes de geração de energia, incluindo fontes despacháveis e intermitentes como eólica e solar fotovoltaica.

O trabalho de Centenaro, Catapani e Rios (2017) aborda a avaliação de confiabilidade em redes de distribuição elétrica. O estudo se concentra em uma rede de distribuição real de uma concessionária brasileira, utilizando para isso métodos analíticos e simulação de Monte Carlo sequencial. O trabalho implementa algoritmos computacionais que analisam a confiabilidade, gerando índices relevantes para o planejamento de redes de distribuição. A pesquisa valida efetivamente os métodos e algoritmos propostos e destaca a importância do método de Monte Carlo sequencial na análise preditiva de confiabilidade de redes elétricas.

O artigo de Kafle, Bhandari e Rana (2022) oferece uma revisão abrangente das técnicas de análise de confiabilidade em sistemas de distribuição. Este estudo destaca diferentes índices de confiabilidade e seus significados, além de realizar um estudo comparativo entre diversos componentes da análise de confiabilidade. O artigo sugere as melhores técnicas para manter a confiabilidade do sistema, enfatizando a importância da avaliação durante a fase de design e desenvolvimento de componentes complexos e equipamentos para eliminar fraquezas de confiabilidade.

O artigo de Aschidamini et al. (2022) oferece uma revisão abrangente dos métodos de planejamento da expansão de sistemas de distribuição de energia com foco na confiabilidade. O estudo analisa modelos de otimização que consideram critérios de confiabilidade, como índices de falhas e interrupções. Ele compara as principais contribuições de vários artigos e destaca as tendências futuras na inclusão de confiabilidade no planejamento. O artigo também explora os desafios da integração de geração distribuída e o uso de redes inteligentes para melhorar a confiabilidade.

O artigo de Selim et al. (2021) desenvolve uma abordagem analítica para a avaliação da confiabilidade de sistemas híbridos de geração solar fotovoltaica e sistemas de armazenamento de energia. A saída do sistema é modelada por um modelo de múltiplos estados, enquanto os sistemas de armazenamento de energia representado por um modelo de dois estados de Markov. A metodologia integra a Análise de Modos de Falha e Efeitos para considerar as falhas dos componentes do sistema híbrido. O estudo também propõe

um modelo para o posicionamento e dimensionamento ótimos do sistema híbrido em redes de distribuição, com o objetivo de minimizar perdas elétricas e garantir níveis aceitáveis de confiabilidade e perfil de tensão. A aplicação em um estudo de caso valida o modelo proposto, demonstrando melhorias significativas na confiabilidade da rede de distribuição. Esse trabalho é especialmente relevante pela sua abordagem inovadora para lidar com a intermitência da energia solar e as falhas dos componentes, ao mesmo tempo em que melhora a confiabilidade dos sistemas de distribuição de energia.

Com isso, percebemos que cada um oferece abordagens e metodologias significativas para a avaliação da confiabilidade em sistemas de distribuição de energia elétrica, destacando a aplicabilidade prática em diferentes cenários.

1.4 Objetivos

O objetivo geral do trabalho é realizar uma análise preliminar das vantagens e desvantagens da SMA e da SMCS para a confiabilidade de sistemas de distribuição, aplicados a um sistema teste de 5 barras. Para isso, são desenvolvidos algoritmos para ambas as metodologias, implementados computacionalmente, gerando índices de confiabilidade que permitem comparações de precisão.

Para atingir o objetivo geral, pode-se listar os seguintes objetivos específicos:

- Realizar uma revisão bibliográfica dos principais componentes do sistema de distribuição de energia elétrica;
- Realizar uma revisão bibliográfica das ferramentas matemáticas da análise de confiabilidade aplicadas ao sistema de distribuição;
- Desenvolver os dois algoritmos de cálculos de confiabilidade: a SMA e a SMCS.
- Aplicar os dois algoritmos desenvolvidos em um sistema teste radial de 5 barras.

1.5 Justificativa

A disponibilidade da rede elétrica exerce um impacto significativo na vida das pessoas, afetando desde aspectos financeiros até questões de segurança e saúde pública. Este trabalho tem o potencial de contribuir para a academia, ao incentivar novas pesquisas e estudos no campo. Além disso, oferece à população em geral, mesmo sem formação técnica, um acesso mais simples a materiais técnicos relevantes. Por fim, profissionais da área podem utilizar este conteúdo como uma oportunidade para revisar conhecimentos e impulsionar seu crescimento profissional.

1.6 Estrutura do Trabalho

Este trabalho de conclusão de curso é constituído de 4 capítulos, além deste capítulo introdutório. No Capítulo 2 são apresentados os principais tópicos referentes a análise do sistemas de distribuição de energia e de técnicas de análise dos índices de confiabilidade. No Capítulo 3 é apresentado o detalhamento sobre o método analítico empregado na análise dos índices de desempenho. Além disso, é feita uma simulação computacional para o método analítico. No Capítulo 4 são apresentados os resultados obtidos diante da análise dos dados. Finalmente, no Capítulo 5 são avaliadas as conclusões e propostas para a segunda etapa do trabalho.

2 Fundamentos Teóricos

Neste capítulo, são delineados os princípios teóricos essenciais dos sistemas de distribuição de energia elétrica, acompanhados de uma discussão sobre a análise dos índices de desempenho e confiabilidade desses sistemas.

2.1 Sistemas de Distribuição de Energia Elétrica

Nas últimas décadas, os sistemas de distribuição tem recebido menos atenção em termos de modelagem e avaliação de confiabilidade em comparação com os sistemas de geração e transmissão, segundo Aschidamini et al. (2022), no entanto, os sistemas de distribuição desempenham um papel crítico na garantia de um fornecimento de eletricidade confiável aos consumidores finais, e suas falhas contribuem significativamente para as interrupções dos clientes. A crescente demanda por confiabilidade, juntamente com a necessidade de equilibrar os investimentos de capital, tornou a avaliação de confiabilidade dos sistemas de distribuição um elemento crucial no planejamento da expansão dos sistemas de energia. Embora as falhas em sistemas de geração e transmissão possam resultar em impactos catastróficos trazendo elevados prejuízos, os sistemas de distribuição apesar de menos onerosos, desempenham um papel fundamental, já que contribuem significativamente para a indisponibilidade do fornecimento ao cliente. As estatísticas indicam a necessidade de avaliar a confiabilidade desses sistemas, considerando a eficiência do capital investido em reforços e melhorias. É importante manter um equilíbrio na confiabilidade entre todas as partes do sistema de energia (geração, transmissão e distribuição) e explorar diferentes alternativas para aprimorar a confiabilidade ao cliente. A avaliação quantitativa da confiabilidade é essencial para comparar efetivamente as alternativas e seus custos (BILLINTON; ALLAN, 1984).

A importância da confiabilidade em SEPs é destacada devido aos seus impactos financeiros e sociais, o que tem impulsionado pesquisas na área. Tradicionalmente, a maioria das pesquisas se concentra na geração e na transmissão, pelo fato de que nessas etapas as falhas podem ocasionar enormes prejuízos para os agentes de mercado, principalmente para os consumidores. Nas últimas décadas tem se observado um maior investimento nas redes de distribuição, uma vez que estudos revelaram que a maior parte das falhas ocorre justamente no segmento de distribuição, responsável por aproximadamente 90% das indisponibilidades de energia. Esse reconhecimento tem motivado um aumento significativo nas pesquisas sobre a confiabilidade em redes de distribuição (BROWN, 2017). Esse reconhecimento tem motivado um aumento significativo nas pesquisas sobre a confiabilidade em redes de distribuição, especialmente em relação ao planejamento da expansão e à integração

de novas tecnologias, como fontes de energia distribuída e renováveis, que impactam diretamente a confiabilidade dos sistemas de distribuição (ASCHIDAMINI et al., 2022).

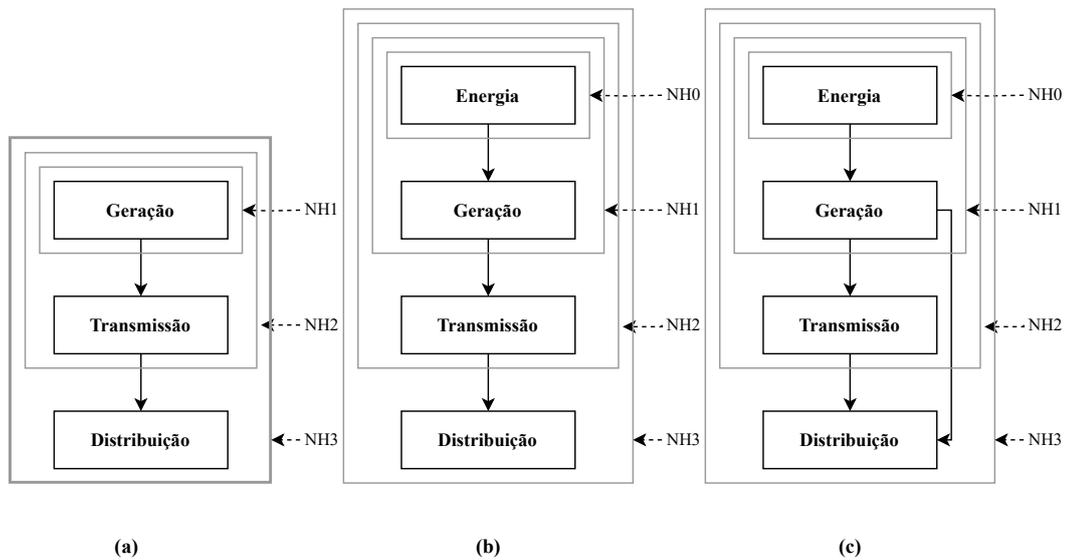
A principal responsabilidade de um sistema elétrico é gerar e entregar energia elétrica aos consumidores de maneira eficiente, constante e de adequada qualidade, garantindo a operação do sistema dentro de padrões desejáveis e sob condições específicas. As técnicas para verificar se um sistema de energia elétrica atende aos requisitos necessários são categorizadas com base na parte do sistema a que se referem, abrangendo as áreas de geração, transmissão e distribuição de energia.

Esta divisão é útil, pois muitas empresas do setor energético organizam suas operações e análises com base nessas áreas específicas, ou se concentram exclusivamente em uma delas. Avaliações são feitas em cada uma dessas áreas para garantir que as necessidades de energia estejam sendo atendidas. Devido ao tamanho e à complexidade dos sistemas de potência, eles foram categorizados em três zonas funcionais. Tendo em mente estudos de confiabilidade, essas zonas ajudam a conduzir cada avaliação de adequação individualmente seguindo os níveis hierárquicos. A Figura 1 mostra as últimas evoluções propostas nas conhecidas. Esta divisão cria uma estrutura organizacional que facilita o planejamento e a operação do sistema elétrico, bem como a realização de estudos de adequação nessas zonas. Essas zonas funcionais são integradas para formar os níveis hierárquicos do sistema, conforme descrito por Billinton e Li (1994) e ilustrado na Figura 1 (a).

Nos últimos anos, os conceitos de Níveis Hierárquicos (NH) foram revisitados principalmente devido às mudanças importantes no mercado de energia elétrica de acordo com o trabalho de Rosa (2009), em que foi introduzida uma nova zona funcional no contexto de sistemas de energia, chamada de energia de Nível Hierárquico 0 (NH0), como ilustrado na Figura 1 (b), esta zona incorpora recursos renováveis e oferece uma representação mais precisa da capacidade de geração ao longo do tempo. Com o avanço tecnológico e o crescimento da integração da geração distribuída na rede de distribuição, tornou-se essencial uma reestruturação do modelo tradicional de níveis hierárquicos. Isso é evidenciado na Figura 1 (c), que inclui a capacidade de geração distribuída nas redes de distribuição (SANTOS, 2019).

O Nível Hierárquico NH0, introduzido por Rosa, Issicaba e Lopes (2011), refere-se à geração de energia diretamente conectada ao sistema de distribuição, como as fontes de energia distribuída. Esse nível está relacionado às micro e pequenas gerações que operam de maneira mais descentralizada em relação aos níveis superiores. A inclusão do NH0 reflete as mudanças recentes no setor energético, especialmente com a crescente adoção de geração distribuída, que impacta diretamente a confiabilidade e o planejamento das redes de distribuição. Para a área de geração de Nível Hierárquico 1 (NH1), os estudos de adequação concentram-se em estimar a probabilidade de interrupções no fornecimento de energia considerando a demanda prevista do sistema em um determinado horizonte de planejamento. Os estudos focados na área de transmissão de Nível Hierárquico 2 (NH2)

Figura 1 – Níveis hierárquicos dos SEPs.



Legenda: zonas funcionais (a) primeira proposta, (b) incorporação de recursos renováveis e (c) inclusão da geração distribuída.

Fonte: Adaptado de Santos (2019).

buscam calcular a probabilidade de interrupções nos diversos barramentos do sistema, levando em conta possíveis falhas nos sistemas de geração e transmissão. Por fim, os estudos na área de distribuição de Nível Hierárquico 3 (NH3) abrangem o sistema como um todo, utilizando modelos estocásticos de falha e reparo para os elementos da rede de distribuição, além de modelos aproximados para os sistemas de geração e transmissão (BILLINTON; LI, 1994).

2.2 Método Analítico

O método analítico é tido como a primeira forma utilizada para se avaliar índices de desempenho do SEP. A abordagem analítica pode ser executada manualmente, sem o auxílio de dispendiosos recursos computacionais. Um procedimento comum para avaliar a confiabilidade de um sistema é dividi-lo em seus componentes, estimar a confiabilidade de cada um e, em seguida, combinar essas estimativas utilizando métodos numéricos para calcular a confiabilidade do sistema completo (BILLINTON; ALLAN, 1992).

Dessa forma, para quantificar o desempenho de sistemas de distribuição, foram desenvolvidos indicadores específicos para os pontos de entrega de serviço, que são avaliados com base nos conceitos clássicos, como o tempo médio de interrupção, dado por,

$$r = \frac{U}{\lambda} \quad [\text{horas/ocorrência}], \quad (2.1)$$

em que U é a duração da interrupção U , dada em horas/ano; λ é a taxa de falha, dada em ocorrências/ano; e r é o tempo médio de interrupção, dado em horas/ocorrência.

A partir da ponderação desses indicadores pelo número de consumidores ou pela carga no ponto de entrega, são calculados índices sistêmicos que permitem medir o desempenho dos sistemas de distribuição. (BILLINTON; ALLAN, 1984).

2.2.1 Método analítico de sistemas de distribuição para um único componente

Para entender como avaliar analiticamente as falhas em um sistema de distribuição diante da falha de seus componentes, observa-se que caso a taxa de falha esteja relacionada a um único componente, então a taxa de falha pode ser definida como,

$$\lambda(t) = \frac{\text{taxa de variação da probabilidade de falha no tempo } t}{\text{probabilidade de sobrevivência até o tempo } t} \quad (2.2)$$

Esta equação expressa a taxa de falha $\lambda(t)$ em função do tempo, considerando a probabilidade do componente continuar operacional em um determinado momento. Pode-se expandir essa análise transformando a definição em uma equação diferencial de primeira ordem:

$$R(t)\lambda(t) = -\frac{dR(t)}{dt} \quad (2.3)$$

em que $-\frac{dR(t)}{dt}$ representa a taxa de mudança na probabilidade de sobrevivência do componente. O sinal negativo indica que, conforme o tempo passa, a probabilidade do componente continuar funcionando normalmente, diminui.

Finalmente resolvendo essa equação diferencial, a solução é dada por,

$$R(t) = e^{-\int_0^t \lambda(u) du} \quad (2.4)$$

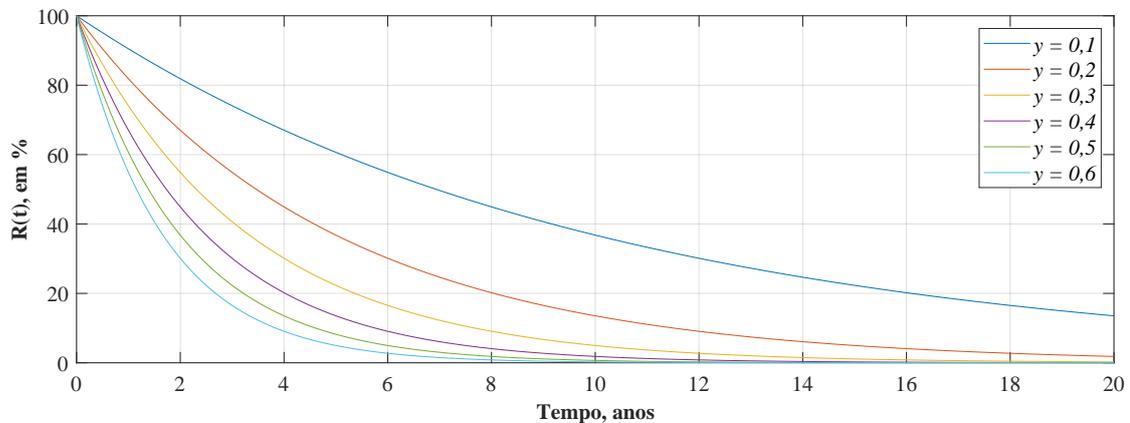
em que $R(t)$ representa a probabilidade de que o componente não falhe até o tempo t . Essa equação considera a taxa de falha ao longo do tempo e possibilita o cálculo da chance do componente continuar funcionando conforme o esperado. Supondo que a taxa de falha seja constante, a confiabilidade do componente pode ser descrita pela distribuição exponencial, resultando em,

$$R(t) = e^{-\lambda(t)} \quad (2.5)$$

Na maioria das vezes, as falhas de equipamentos elétricos seguem uma distribuição exponencial, caracterizando um processo estocástico (BILLINTON; ALLAN, 1992).

A Figura 2 apresenta a probabilidade de funcionamento $R(t)$ ao longo de 20 anos, para diferentes valores de taxa de falha (λ), conforme (2.5). Vale ressaltar que os valores de λ estão em unidades de falhas por ano (ocorrências/ano) e precisam ser convertidos para falhas por hora (ocorrências/h) para garantir consistência nas análises. Observa-se que a confiabilidade do componente começa em 100% no início de sua vida útil, mas decresce gradualmente com o passar do tempo. Por exemplo, com um ano de operação, um equipamento com $\lambda = 0,1$ tem 90,5% de chance de funcionar sem falhas, enquanto para $\lambda = 0,5$, essa probabilidade cai para 55,0%.

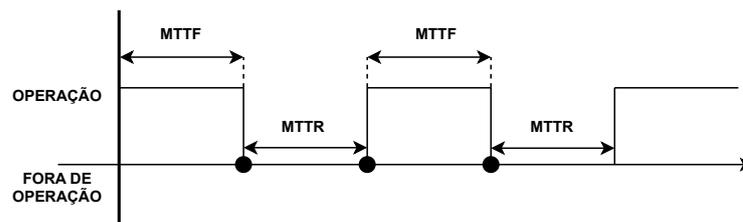
Figura 2 – Probabilidade de funcionamento $R(t)$ para alguns valores de taxa de falha (λ) ao longo de 20 anos. Os valores de λ estão em ocorrências/ano.



Fonte: Elaborado pelo autor (2024).

De acordo com essas definições, os componentes de um sistema podem ser representados por um modelo de Markov com dois estados: operando e fora de operação, conforme descrito por Billinton e Allan (1992) e pode ser observado na Figura 3. Esse modelo permite a análise das transições entre os estados de funcionamento e falha, considerando as taxas de falha e reparo. O modelo contínuo de Markov descreve um componente com dois estados e as transições possíveis entre eles, sendo amplamente utilizado em estudos de confiabilidade. Segundo Brown (2017), o tempo é usualmente medido em horas, e os resultados são avaliados em termos anuais.

Figura 3 – Modelo a dois estados de um componente - tempo de operação e tempo de reparo.



Fonte: Adaptado de Billinton e Allan (1992).

Nesse contexto, além de considerar a taxa de falha λ , é fundamental introduzir o conceito de Tempo Médio para a falha (do inglês, *Mean Time to Failure*) (MTTF) e Tempo Médio para Reparo (do inglês, *Mean Time to Repair*) (MTTR), permitindo que as concessionárias monitorem e aprimorem o desempenho da rede, planejem manutenções preventivas e reduzam o impacto de interrupções para os consumidores.

O MTTF estima o tempo médio esperado até que ocorra uma falha no componente

e sua relação com λ é dada por,

$$\lambda = \text{taxa de falha} = \frac{1}{\text{MTTF}} \quad (2.6)$$

em que MTTF é o tempo médio para falha, em anos/ocorrência, enquanto λ é dada em ocorrências por ano.

De forma similar, MTTR estima o tempo médio em que uma falha é reparada e tem relação com a taxa de reparo,

$$\mu = \text{taxa de reparo} = \frac{1}{\text{MTTR}} \quad (2.7)$$

em que MTTR é o tempo médio de reparo, em horas/reparo; e μ é a taxa de reparo, geralmente dada em reparos/hora.

Para ilustrar a utilização de (2.6) e (2.7), considere um componente com $\lambda = 0,2$ ocorrências/ano e um MTTR = 4 horas. A partir desses dados, pode-se calcular o MTTF = $1/0,2 = 5$ anos/ocorrência, indicando que o componente, em média, funcionará por 5 anos antes de apresentar uma falha. Em contrapartida, a taxa de reparo é, $\mu = 1/4 = 0,25$ reparos/hora, o que significa que é possível realizar, em média, 0,25 reparos a cada hora.

2.2.2 Método analítico para sistemas de distribuição com múltiplos componentes

Para analisar o desempenho de sistemas de distribuição mais complexos foram estabelecidos indicadores de ponto de entrega de serviço, esses indicadores permitem calcular a probabilidade de falha do sistema e quanto tempo o sistema ficará inativo devido a falhas. O cálculo das taxas de falha para cada ponto é realizado utilizando o princípio de sistema em série. Especifica-se que todos os componentes entre o ponto consumidor e pelo menos uma fonte de energia devem estar operando adequadamente para que o serviço seja possível. Três elementos fundamentais são empregados no estudo da confiabilidade de uma rede, dado por,

$$\lambda_s = \sum_i \lambda_i \quad (2.8)$$

$$U_s = \sum_i \lambda_i r_i \quad (2.9)$$

$$r_s = \frac{U_s}{\lambda_s} = \frac{\sum_i \lambda_i r_i}{\sum_i \lambda_i} \quad (2.10)$$

em que λ_s é a taxa de falha do sistema como um todo, e é calculada como a soma das taxas de falhas individuais de cada componente, U_s é a indisponibilidade anual que permite obter uma medida de quanto tempo, em média, o sistema estará indisponível devido a falhas em qualquer um dos componentes, enquanto que r_s é a duração da interrupção no ponto de carga, sendo possível observar de quanto tempo, em média, o sistema ficará

inativo devido a reparos necessários após uma falha. Analogamente, λ_i é a taxa de falha de cada componente e r_i é o tempo de restabelecimento médio de cada componente.

Utilizando o princípio de associação em série é possível calcular a taxa de falha em cada ponto do sistema somando as taxas de falha individuais dos componentes que influenciam diretamente a falha desse ponto. Estes índices podem ser utilizados para avaliar o desempenho histórico ou prever o desempenho futuro do sistema. Os índices adicionais que são mais comumente usados são definidos a seguir, de acordo com Billinton e Allan (1984):

- **Índice Médio de Frequência de Interrupção do Sistema (do inglês, *System Average Interruption Frequency Index*) (SAIFI) do indicador brasileiro Frequência Equivalente de Interrupção por Consumidor (FEC)**

$$SAIFI = \frac{\text{total de interrupções de clientes}}{\text{total de clientes atendidos}} = \frac{\sum \lambda_i N_i}{\sum N_i} \quad (2.11)$$

em que λ_i é a taxa de falha e N_i é o número de clientes no ponto de carga i . Esse índice é calculado como a razão entre o número total de interrupções dos clientes pelo total de clientes atendidos.

- **Índice Médio de Duração de Interrupção do Sistema (do inglês, *System Average Interruption Duration Index*) (SAIDI) do indicador brasileiro Duração Equivalente de Interrupção por Consumidor (DEC)**

$$SAIDI = \frac{\text{somatório das durações de interrupção dos clientes}}{\text{número total de clientes}} = \frac{\sum U_i N_i}{\sum N_i} \quad (2.12)$$

em que U_i é o tempo de interrupção anual e N_i é o número de clientes no ponto de carga i .

- **Índice Médio de Duração de Interrupção por Cliente (do inglês, *Customer Average Interruption Duration Index*) (CAIDI)**

$$CAIDI = \frac{\text{somatório das durações de interrupção dos clientes}}{\text{número total de interrupções de clientes}} = \frac{\sum U_i N_i}{\sum \lambda_i N_i} \quad (2.13)$$

- **Índice Médio de Disponibilidade de Serviço (do inglês, *Average Service Availability Index*) (ASAI)**

$$ASAI = \frac{\text{horas de serviço disponível ao cliente}}{\text{horas de serviço solicitadas pelo cliente}} = \frac{\sum N_i \times 8760 - \sum U_i N_i}{\sum N_i \times 8760} \quad (2.14)$$

- **Índice Médio de Indisponibilidade de Serviço (do inglês, *Average Service Unavailability index*) (ASUI)**

$$ASUI = \frac{\text{horas de serviço não disponível ao cliente}}{\text{horas de serviço solicitadas pelo cliente}} = \frac{\sum U_i N_i}{\sum N_i \times 8760} \quad (2.15)$$

em que 8760 é o número de horas em um ano calendário.

- **Energia Não Fornecida (do inglês, *Energy Not Supplied*) (ENS)**

$$ENS = \text{energia total não fornecida pelo sistema} = \sum L_{ai}U_i \quad (2.16)$$

em que L_{ai} é a carga média conectada ao ponto de carga i .

- **Energia Média Não Fornecida (do inglês, *Average Energy Not Supplied*) (AENS)**

$$AENS = \frac{\text{energia total não fornecida}}{\text{número total de clientes atendidos}} = \frac{\sum L_{ai}U_i}{\sum N_i} \quad (2.17)$$

A abordagem analítica envolve o uso de modelos matemáticos com soluções numéricas diretas de modo a calcular valores esperados dos índices dos componentes do sistema elétrico, conforme discutido por Billinton e Li (1994). Esses índices focam na avaliação de interrupções sustentadas, considerando um período de apuração de um ano e englobando todos os consumidores da rede, as variáveis SAIFI e SAIDI correspondem aos indicadores brasileiros FEC e DEC, respectivamente. Por serem estimativas médias, esses indicadores podem ser calculados diretamente por métodos analíticos. Entretanto, outros índices, como a Duração Máxima de Interrupção por Consumidor (DMIC), requerem para sua estimativa o uso de métodos de simulação (ANEEL, 2021).

Para entender como falhas em componentes afetam um sistema de distribuição de energia, o uso desse método é bem prático. Ele permite ver como problemas em um único componente podem influenciar todo o sistema, afetando a sua confiabilidade e eficiência. Com o avanço tecnológico, surgiram métodos mais modernos, como simulações computacionais e técnicas de inteligência artificial, que oferecem análises mais detalhadas. Contudo, o método analítico ainda é relevante, principalmente em situações com recursos limitados ou para ensinar os conceitos básicos sobre sistemas de energia.

2.3 Método Monte Carlo

O método de SMC é amplamente conhecido como uma técnica de simulação estocástica que utiliza números aleatórios para modelar e analisar sistemas complexos. Essa técnica usa a geração de números aleatórios para replicar o comportamento de um sistema real e suas variações aleatórias, com o objetivo de calcular indicadores de desempenho em diferentes cenários segundo Billinton e Allan (1992). No contexto de sistemas, essas técnicas estimam índices de confiabilidade ao simular repetidamente o comportamento dos componentes, reproduzindo suas operações e falhas de maneira aleatória. A probabilidade de eventos é determinada contabilizando o número de vezes que esses eventos ocorrem durante as várias simulações realizadas, permitindo uma estimativa do desempenho do sistema. (BILLINTON; LI, 1994).

Diferentemente do método analítico, o ambiente de simulação computacional permite o cálculo de índices relativos. Os resultados de métodos probabilísticos em ambientes controlados podem ser confiáveis, pois são avaliados usando a média para medir o valor central, e a variância para verificar a variação dos dados (BILLINTON; ALLAN, 1984).

Segundo Billinton e Allan (1992), ao invés de comparar os resultados com metas ou valores genéricos, essa abordagem faz uma análise relativa, o que significa que as variações presentes nos dados de entrada e às condições do sistema são levadas em consideração, pois são avaliados usando a média para medir o valor central, e a variância para verificar a variação dos dados. Com isso, é possível obter uma avaliação mais realista e confiável das diferenças nos resultados antes e após as alterações.

Uma das principais vantagens dessa técnica é a possibilidade de definir parâmetros dos componentes, como taxa de falha e de reparo e variáveis aleatórias utilizando funções de distribuição de probabilidade (BROWN, 2017). A SMC modela a ocorrência de eventos estocásticos em três etapas: seleção de estados, avaliação dos estados, e estimativa dos índices, segundo Billinton e Li (1994). Assim, essas ferramentas são aplicadas para estimar índices de confiabilidade, simulando o comportamento da rede frente a contingências ou mudanças operacionais, permitindo uma análise mais realista e precisa dos resultados. (BILLINTON; ALLAN, 1992).

2.3.1 Método de Monte Carlo Sequencial

Atualmente, os métodos de simulação podem ser divididos em três categorias principais: sequencial, não sequencial e pseudo-sequencial, sendo esta última uma combinação das duas primeiras. A SMCS modela eventos em ordem cronológica, permitindo a análise de como a sequência de eventos influencia os resultados. Em contrapartida, a simulação não-sequencial gera cenários independentes, onde os eventos são tratados de forma isolada. A abordagem pseudo-sequencial busca aliar a eficiência computacional do método não-sequencial com a flexibilidade e precisão do sequencial Silva (2016). A escolha do método a ser empregado deve considerar a importância de relacionar eventos em uma linha do tempo e seu impacto nos índices de confiabilidade (BILLINTON; ALLAN, 1992).

Neste trabalho, foi utilizada a SMCS, pois é a mais empregada na análise de confiabilidade de sistemas de distribuição elétrica. Isso se deve ao fato de que a sequência de falhas pode impactar diretamente a qualidade do serviço; por exemplo, uma falha na rede pode levar à sobrecarga/falha de um transformador e conseqüentemente impactar nos índices de confiabilidade. Embora a SMC não-sequencial possa oferecer maior velocidade, ela ignora a relação temporal entre eventos, o que pode ser crítico para a confiabilidade do sistema. A análise pseudo-sequencial é indicada quando a complexidade do sistema torna a simulação computacionalmente intensa, oferecendo uma alternativa viável. No entanto, essa abordagem não é o foco deste trabalho.

O método sequencial é utilizado para gerar uma sequência aleatória de estados de sucesso ou falha, que avança conforme a simulação no tempo. Essas sequências são criadas com base na probabilidade de o componente permanecer em cada estado, sendo que o tempo em que o componente permanece em cada estado segue uma distribuição exponencial (BILLINTON; LI, 1994).

As equações utilizadas para calcular o tempo de falha e o tempo de reparo dos componentes são derivadas da transformada inversa da Função de Distribuição Acumulada (do inglês, *Cumulative Distribution Function*) (CDF) da distribuição exponencial. Esse método é amplamente aplicado em SMCS para gerar variáveis aleatórias a partir de uma distribuição específica, utilizada no estudo de confiabilidade. (LAW; KELTON, 1991)

A distribuição exponencial é conhecida por modelar o tempo entre eventos, como falhas e reparos. A CDF de uma variável aleatória T , que segue uma distribuição exponencial com uma taxa de falha ou reparo λ , é dada por:

$$F(T) = P(T \leq t) = 1 - e^{-\lambda t} \quad (2.18)$$

Neste caso, $F(T)$ representa a probabilidade de que o tempo até a falha ou reparo, T , seja igual ou inferior a um determinado valor t . Para gerar valores aleatórios que sigam essa distribuição, utilizamos o método da transformada inversa. O processo é descrito da seguinte maneira:

- Gerar um número aleatório U que segue uma distribuição uniforme no intervalo $(0, 1)$.
- Utilizar a função inversa da CDF para transformar U no valor de T que segue a distribuição exponencial.

Igualando a CDF a U , onde λ é a taxa de falha, e T é o tempo até a próxima falha ou reparo. Resolvendo essa equação para T , temos:

$$U = 1 - e^{-\lambda T} \quad (2.19)$$

$$T = -\frac{1}{\lambda} \ln U \quad (2.20)$$

Esse método de geração de tempos aleatórios via transformada inversa é amplamente aplicado em estudos de confiabilidade e simulações estocásticas, permitindo que se simule a ocorrência de falhas e reparos de componentes com base nas suas taxas associadas. As equações a seguir são derivadas desse método, que é usada para modelar o tempo de permanência em cada estado de um componente. Nos modelos de Markov, assume-se que as falhas e os reparos ocorrem de forma aleatória e contínua no tempo (BILLINTON; LI, 1994).

A duração no estado de operação (tempo até a falha) é dada por:

$$T^{sucesso} = -\frac{1}{\lambda_i} \ln U_i \quad (2.21)$$

E o tempo no estado de falha (tempo até o reparo) é calculado por:

$$T^{falha} = -\frac{1}{\mu_i} \ln U_i \quad (2.22)$$

onde U_i é um número aleatório que segue uma distribuição uniforme, λ_i é a taxa de falha do componente, μ_i é a taxa de reparo, e i identifica o componente, conforme descrito por Calado et al. (2015)

2.3.2 Índices de confiabilidade

No ambiente de simulação, os dados são convertidos em taxas de falha e de reparo e os índices de confiabilidade não são valores fixos, mas sim estimativas baseadas em uma distribuição de probabilidade, calculadas ao longo do tempo. Esses índices são determinados por meio de funções-teste G , conforme descrito por Silva et al. (2005). Dessa forma, $E[G]$ representa uma estimativa do valor esperado do índice.

$$E[G] = \frac{1}{N} \sum_{n=1}^N G(y_n) \quad (2.23)$$

onde, N é o número de amostras anuais, e $G(y_n)$ é a função-teste aplicada a cada ano n . Há uma incerteza associada a $E[G]$, uma vez que a função-teste depende de variáveis aleatórias, o que resulta em valores diferentes para cada ano Calado et al. (2015).

A variância do estimador é utilizada para medir essa incerteza, dessa forma, quanto maior for a quantidade de amostras, mais precisos serão os resultados da simulação. As variâncias da funções-testes $Var(G)$ são calculados por,

$$Var(E[G]) = \frac{Var(G)}{N}. \quad (2.24)$$

A incerteza nas estimativas obtidas pela simulação é medida pelo coeficiente de variação β , representado pela Equação 2.25.

$$\beta = \frac{\sqrt{Var(E[G])}}{E[G]}. \quad (2.25)$$

em que β é uma medida da quantidade de variação ou dispersão das estimativas obtidas pela simulação em relação à média.

Valores baixos de β indicam que as estimativas estão próximas da média, refletindo baixa variabilidade e, conseqüentemente, maior confiabilidade nos resultados. Em contrapartida, valores altos de β sugerem que as estimativas estão dispersas. Geralmente, β é utilizado como critério parada/convergência do método. Neste casos valores típicos são

inferiores a 2% (alguns casos 5%) (CALADO et al., 2015; CENTENARO; CATAPANI; RIOS, 2017). Vale ressaltar que, por simplicidade, o critério de parada adotado neste trabalho foi o número de iterações.

2.3.3 Funções-testes

São apresentados na Tabela 1 os índices de confiabilidade, que podem ser estimados empregando as funções-testes G , como descrito em (ROSA; ISSICABA; LOPES, 2011).

Tabela 1 – Funções-testes para calcular os índices de confiabilidade

Funções-teste G
$G_{SAIFI}(y_r) = \frac{n^\circ \text{ de interrupções consumidoras em } y_r}{n^\circ \text{ total de consumidores}}$
$G_{SAIDI}(y_r) = \frac{\text{Total de duração das interrupções consumidoras em } y_r}{n^\circ \text{ total de consumidores}}$
$G_{CAIDI}(y_r) = \frac{SAIDI \text{ em } y_r}{SAIFI \text{ em } y_r}$
$G_{ASAI}(y_r) = 1 - \frac{G_{SAIDI}(y_r)}{n^\circ \text{ de horas em } y_r}$
$G_{ASUI}(y_r) = 1 - G_{ASAI}(y_r)$
$G_{ENS}(y_r) = \text{energia não suprida em } y_r$
$G_{AENS}(y_r) = \frac{G_{ENS}(y_r)}{n^\circ \text{ total de consumidores}}$

Fonte: Adaptado de (CALADO et al., 2015)

2.4 Conclusões parciais

Neste capítulo, são explorados os fundamentos dos sistemas de distribuição, com ênfase na importância da confiabilidade nesses sistemas. Introduzem-se conceitos essenciais, como os níveis hierárquicos que compõem o sistema elétrico. O texto também aborda técnicas analíticas, definições de taxa de falha, duração de interrupção e tempo médio de interrupção, além de explorar os métodos estocásticos, que aplica o métodos de SMC para avaliação da confiabilidade do sistema de distribuição, os conceitos básicos desse método são apresentados e discutidos do ponto de vista da avaliação da confiabilidade.

3 Desenvolvimento

Neste capítulo, é delineada a metodologia analítica para a análise da confiabilidade de redes de distribuição. Um sistema elétrico simples é utilizado como tutorial para ilustrar a metodologia, com a exploração de quatro casos distintos, os quais diferem em complexidade.

3.1 Sistema teste

Considere o sistema teste radial representado na Figura 4. Ele é representado por nove pontos de conexão chamado de nós, possui quatro linhas de distribuição representando o tronco principal da fonte de alimentação e quatro ramais conduzindo energia aos pontos de cargas A, B, C e D, conforme mostrado na Tabela 2. Os parâmetros de confiabilidade do sistema são mostrados na Tabela 3.

Tabela 2 – Clientes e cargas conectadas ao sistema da Figura 4.

Pontos de Cargas	Número de Consumidores	Carga Média (kW)
A	1000	5000
B	800	4000
C	700	3000
D	500	2000

Fonte: Adaptado de Billinton e Allan (1984).

Tabela 3 – Parâmetros de confiabilidade para o sistema teste.

Componente	Comprimento (km)	$\lambda((falhas/ano))$	$r(horas)$
Seção			
1	2	0,2	4
2	1	0,1	4
3	3	0,3	4
4	2	0,2	4
Distribuição			
a	1	0,2	2
b	3	0,6	2
c	2	0,4	2
d	1	0,2	2

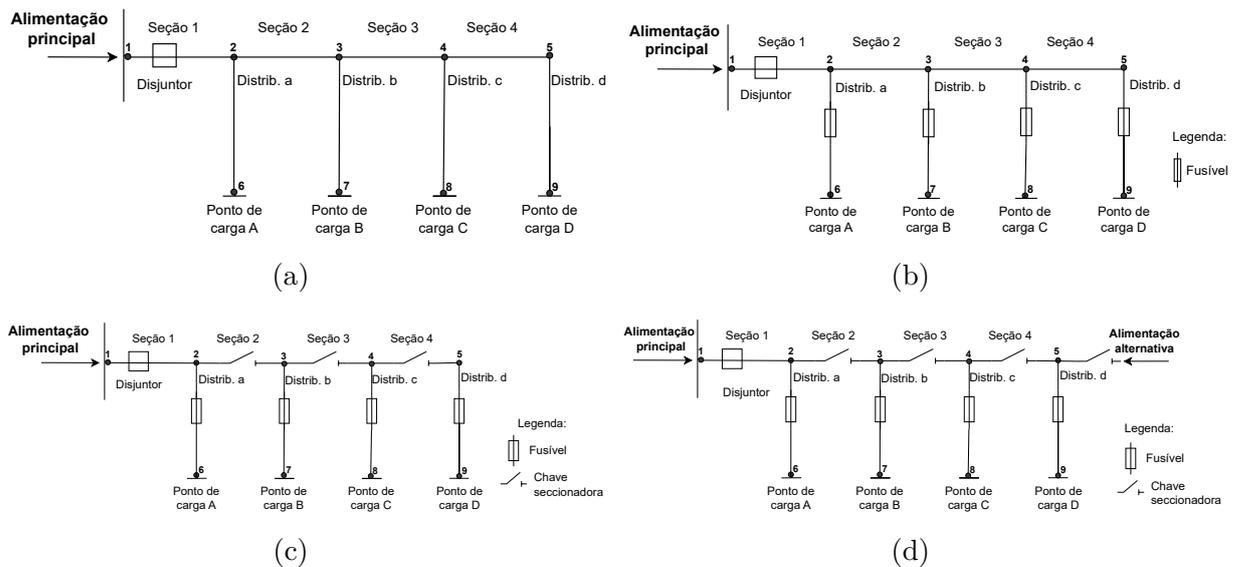
Fonte: Adaptado de Billinton e Allan (1984).

Na prática, observa-se que a probabilidade de falha em linhas e cabos tende a ser proporcional ao comprimento desses elementos. Neste caso específico, considera-se que o alimentador principal (composto pelas Seções 1, 2, 3, 4) apresenta uma taxa de falha de 0,1 falhas por quilômetro por ano, enquanto que os ramais (identificados como a, b, c, d)

exibem uma taxa de falha de 0,2 falhas por quilômetro por ano. Com base nestes dados iniciais e nos comprimentos das linhas apresentados na Tabela 3, é possível calcular os parâmetros de confiabilidade, que também são listados. Foram considerados quatro casos como objeto do estudo de confiabilidade, utilizando assim todos os recursos explicitados no sistema teste.

- **Configuração 1:** sistema protegido por um disjuntor na saída do alimentador,
- **Configuração 2:** sistema protegido pelo disjuntor, acrescentando fusíveis nos ramais,
- **Configuração 3:** sistema protegido por disjuntor, fusíveis nos ramais e seccionadoras no tronco principal,
- **Configuração 4:** sistema protegido pelo disjuntor, fusíveis, seccionadora, e considerando uma alimentação alternativa no final do tronco principal.

Figura 4 – Representação do sistema de distribuição radial para as 4 configurações.



Legenda: (a) configuração 1, (b) configuração 2, (c) configuração 3 e (d) configuração 4.

Fonte: Elaborado pelo autor (2024).

No primeiro cenário analisado, o sistema é representado por uma linha única, então qualquer falha em qualquer ponto do sistema acarretará na operação do disjuntor. Após a ocorrência de uma falha, o disjuntor é aberto, o operador localiza a falha, realiza o reparo e somente depois ele rearma o disjuntor para que o sistema todo seja realimentado novamente.

No segundo cenário, foi considerado uma proteção adicional, inclui-se fusíveis nos ramais do sistema como dispositivos de proteção adicionais aos disjuntores. Nessa situação,

um curto em um dos distribuidores acarretará na queima do fusível associado, fazendo com que o ponto de carga associado seja isolado até que o reparo seja feito, sem a necessidade de abertura do disjuntor. Com isso os pontos de cargas permanecem alimentados.

Para o terceiro cenário, além de considerar disjuntores e fusíveis nos ramais de distribuição e assumindo que os fusíveis são acionados quando ocorrem falhas nos ramais, ainda assim, a ocorrência de uma falha no tronco principal acarretará na atuação do disjuntor, nesse caso o sistema pode ser melhorado incluindo as seccionadoras ao longo do alimentador principal. No caso de falhas, o operador procede com a abertura da seccionadora para isolar a parte defeito, permitindo assim religar o disjuntor para continuar alimentando os outros pontos que estão entre a fonte de alimentação e o ponto de isolamento da chave, ainda que o reparo não tenha sido feito.

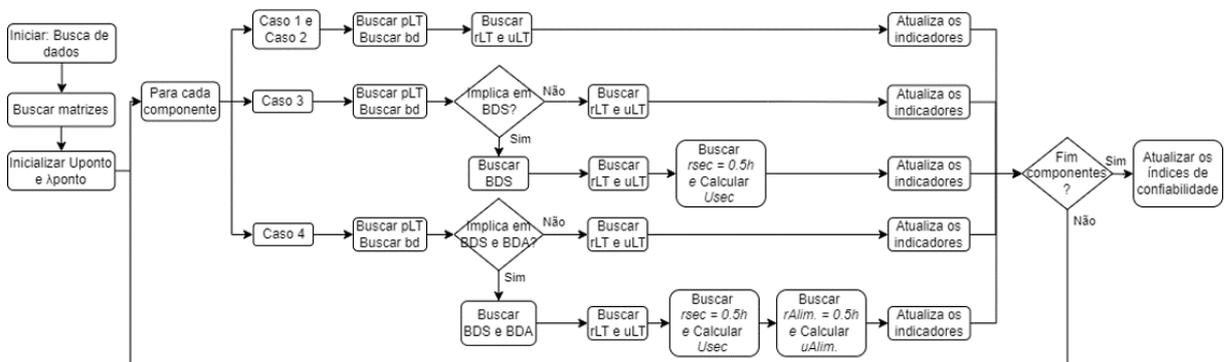
No quarto cenário, considera-se a existência de uma fonte de alimentação alternativa, com isso as cargas que seriam desconectadas podem ser redirecionadas para outras áreas do sistema. Esse método impacta significativamente nos índices de confiabilidade, os maiores efeitos são observados nos pontos de carga mais próximo do ponto novo de transferência. Nessa circunstancia, os pontos de carga são menos afetadas por falhas ocorridas no tronco principal por um longo tempo.

Com as taxas de falha e a indisponibilidade calculadas para cada ponto de carga, em cada cenário, procede-se ao cálculo dos indicadores sistêmicos. Após calcular os índices para cada caso, o código exibe os resultados.

3.2 Algoritmo analítico

A Figura 5 mostra o fluxograma da implementação da SMA para melhorar a visualização do método aplicado

Figura 5 – Diagrama de blocos da SMA para análise de confiabilidade.



Fonte: Elaborado pelo autor (2024).

Utilizando a taxa de falha e a indisponibilidade de cada ponto de carga, os indicadores sistêmicos são calculados para cada caso com base nas equações (2.11), (2.12),

(2.13), (2.14), (2.15), (2.16) e (2.17). Os cálculos e análises de cada caso estão disponíveis no Apêndice A.

O pseudocódigo a seguir apresenta o algoritmo implementado. Esse algoritmo calcula e apresenta os índices gerais de confiabilidade para o sistema como um todo, de acordo com o trabalho de Centenaro, Catapani e Rios (2017).

1. Leitura de dados: buscar todos os dados relevantes conforme Tabela 2 e Tabela 3;
2. Definir a matriz de adjacência e matriz de alcance, conforme disposto Apêndice B, Equação B.2 : configuração da matriz a partir do conjunto de dados para definição dos ramos do sistema, representado na Figura 8;
3. Inicializar a indisponibilidade e a taxa de falha de cada ponto, representando as variáveis vetoriais $\mathbf{U}_{\text{ponto}}$ e λ_{ponto} respectivamente;
4. Para cada componente/linha:
 - Buscar a taxa de falha λ_{LT} do componente na matriz de dados. **Para o caso 1:**
 - Buscar protetor associado na matriz de dados.
 - Buscar o vetor de indicadores de **barras desconectadas** (**bd**) pela falta através da matriz de alcance.
 - Buscar o tempo médio de reparo r_{LT} do componente na matriz de dados.
 - Calcular a indisponibilidade $U_{LT} = \lambda_{LT} \times r_{LT}$.
 - Atualizar os indicadores de ponto:

$$\lambda_{\text{ponto}} = \lambda_{\text{ponto}} + \mathbf{bd} \times \lambda_{LT}$$

$$\mathbf{U}_{\text{ponto}} = \mathbf{U}_{\text{ponto}} + \mathbf{bd} \times U_{LT}$$

Para o caso 2:

- Buscar protetor na matriz de dados, considerando fusíveis além do disjuntor.
- Repetir os passos de busca de **bd**, r_{LT} , e cálculo de U_{LT} como no caso 1.
- Atualizar os indicadores de ponto com os novos valores considerando os fusíveis.

Para o caso 3:

- Buscar protetor na matriz de dados, considerando disjuntores, fusíveis e seccionadoras.
- Repetir os passos de busca de **bd** como nos casos anteriores.

Se a falha não implica em abertura da seccionadora:

- Buscar o tempo médio de reparo r_{LT} do componente;
- Calcular U_{LT} como nos casos anteriores.

Se a falha implica em abertura da seccionadora:

- Buscar o vetor de **barras desconectadas pela seccionadora (bds)** através da matriz de alcance;
- Buscar o tempo médio de reparo r_{LT} ;
- Calcular a indisponibilidade $U_{LT} = \lambda_{LT} \times r_{LT}$;
- Buscar r referente á seccionadora $r_{sec} = 0,5horas$;
- Calcular a indisponibilidade da seccionadora $U_{sec} = \lambda_{LT} \times r_{sec}$;
- Atualizar os indicadores de ponto com as novas indisponibilidades considerando a abertura da seccionadora, usando as equações:

$$\lambda_{\text{ponto}} = \lambda_{\text{ponto}} + \mathbf{bd} \times \lambda_{LT}$$

$$U_{\text{ponto}} = U_{\text{ponto}} + \mathbf{bds} \times U_{LT} + (\mathbf{bd} - \mathbf{bds}) \times U_{sec}$$

Para o caso 4:

- Buscar protetor na matriz de dados, considerando todos os instrumentos de proteção e alimentação alternativa.
- Repetir os passos de busca de **bd** como nos casos anteriores.

Se a falha não implica em abertura da seccionadora e acionamento da alimentação alternativa:

- Buscar o tempo médio de reparo r_{LT} do componente;
- Calcular U_{LT} como nos casos anteriores e atualizar os indicadores de ponto.

Se a falha implica em acionamento da seccionadora e da alimentação alternativa:

- Se existe abertura da seccionadora: Buscar as barras desconectadas pela seccionadora **bds** através da matriz de alcance; caso contrário, **bds** é vetor de zeros;
- Se acionamento da alimentação alternativa: Buscar as **barras supridas pela alimentação alternativa (bda)** pela através da matriz de alcance; caso não haja acionamento da alimentação alternativa: **bda** é vetor de zeros;
- Buscar o tempo médio de reparo r_{LT} do componente;
- Calcular a indisponibilidade $U_{LT} = \lambda_{LT} \times r_{LT}$;
- Buscar r referente á seccionadora $r_{sec} = 0,5horas$;
- Calcular a indisponibilidade da seccionadora $U_{sec} = \lambda_{LT} \times r_{sec}$;
- Buscar r referente á alimentação alternativa $r_{alim.alt} = 0,5horas$;
- Calcular a indisponibilidade da alimentação $U_{alim.alt} = \lambda_{LT} \times r_{alim.alt}$;

- Atualizar os indicadores de cada ponto com as novas indisponibilidades considerando a abertura da seccionadora e o acionamento da alimentação alternativa usando as equações:

$$\lambda_{\text{ponto}} = \lambda_{\text{ponto}} + \mathbf{bd} \times \lambda_{LT} \quad (3.1)$$

$$\begin{aligned} \mathbf{U}_{\text{ponto}} = \mathbf{U}_{\text{ponto}} + (\mathbf{bd} - \mathbf{bds}) \times U_{sec} + \\ + (\mathbf{bds} - \mathbf{bda}) \times U_{LT} + \mathbf{bda} \times U_{alim.alt} \end{aligned} \quad (3.2)$$

5. Aplicar o cálculo dos indicadores sistêmicos SAIFI, SAIDI, CAIDI, ASAI, ASUI, ENS e AENS.

Com todos esses dados disponíveis, o código agora pode calcular os índices para cada ponto de carga no sistema, determinando qual caso será avaliado. Ao analisar os diferentes casos de falha com o auxílio da variável `caseToAccess`, o código processa diversos cenários para entender como diferentes condições afetam a confiabilidade do sistema. Para cada caso, os índices de carga são recalculados, levando em conta as especificidades daquele cenário. Isso permite uma análise comparativa entre diferentes situações de falha e estão dispostos no Apêndice B.

No primeiro cenário analisado, é identificado o dispositivo de proteção associado a cada componente e verifica-se, por meio de uma matriz de alcance, quais nós seriam desconectados em caso de falha nesse componente. A taxa de falha de cada nó é aumentada pela taxa de falha do componente correspondente. Da mesma forma, a indisponibilidade de cada nó é incrementada pelo produto da taxa de falha do componente e seu tempo médio de reparo.

No segundo cenário, a análise segue um padrão semelhante ao do primeiro. Isso altera as taxas de falha e, conseqüentemente, a indisponibilidade nos pontos, uma vez que os fusíveis impactam os nós desconectados nos ramais de distribuição.

Para o terceiro cenário, além de considerar disjuntores e fusíveis nos ramais de distribuição. Aqui, é avaliado o impacto da atuação da seccionadora. Se a seccionadora não atuar, o cálculo segue como no cenário anterior. Caso contrário, considera-se o novo tempo de abertura e isolamento da seccionadora, multiplicando-o pela taxa de falha para determinar a indisponibilidade dos pontos.

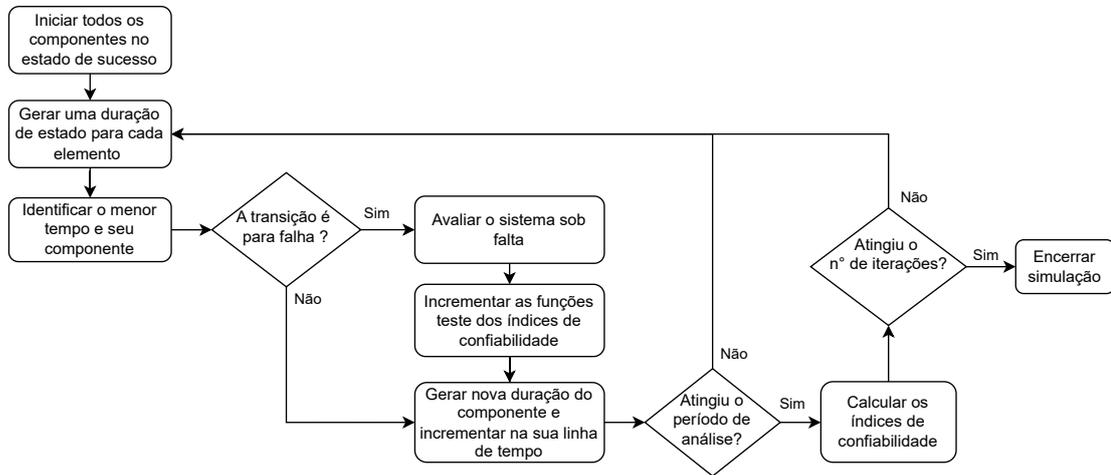
No quarto cenário, considera-se a existência de uma fonte de alimentação alternativa. Isso inclui um elemento adicional ao cálculo da indisponibilidade, levado em conta na análise do cenário anterior. A taxa de falha dos nós continua sendo incrementada pela taxa de falha do componente.

Após calcular os índices para cada caso, o código exibe os resultados. Isso é feito através de tabelas que mostram os índices de desempenho para diferentes partes do sistema.

3.3 Algoritmo de Monte Carlo Sequencial

A Figura 6 mostra o fluxograma da implementação da SMCS.

Figura 6 – Diagrama de blocos da SMCS para análise de confiabilidade.



Fonte: Elaborado pelo autor (2024).

A metodologia aplicada neste trabalho utiliza a SMCS para avaliar a confiabilidade de sistemas elétricos. Este método permite modelar o comportamento estocástico de componentes do sistema, levando em consideração o ciclo falha, que é o tempo que o componente falha até o momento em que ele se recupera, o que é característico de sistemas de distribuição de energia. A seguir, descrevem-se os passos necessários para a implementação do método, associando-os à estrutura do código desenvolvido. Neste estudo, o critério de convergência adotado foi um número fixo de iterações por simulação, determinado empiricamente com base nas características do sistema. Devido a limitações de tempo, decidiu-se não utilizar critérios de convergência mais rigorosos.

1. Inicializar todos os componentes no estado de sucesso;

A primeira etapa dessa análise começa com a inicialização de todos os componentes do sistema em estado de sucesso, ou seja, funcionando sem falhas.

2. Gerar a duração no estado de sucesso;

Nessa etapa, utiliza-se uma função para gerar números aleatórios e calcular o tempo que cada componente permanecerá no estado de sucesso. Esse tempo é gerado por uma distribuição exponencial, como mostra (3.3). A função gera um número aleatório para o componente U_i , e λ_i é a taxa de falha do componente i .

$$T_i^{sucesso} = -\frac{1}{\lambda_i} \ln U_i \quad (3.3)$$

Se o componente está no estado de falha, a função calcula a duração no estado de falha usando a taxa de reparo μ_i , como mostrado (3.4),

$$T_i^{falha} = -\frac{1}{\mu_i} \ln U_i \quad (3.4)$$

Essas equações são aplicadas a todos os componentes do sistema, e os tempos são armazenados para determinar quando ocorrerá a próxima falha.

3. Identificar o menor tempo e o respectivo componente; Para cada componente, calcula-se o menor tempo até a próxima transição entre os estados de sucesso e falha. O componente com o menor tempo é o primeiro a sair do estado de sucesso e entrar no estado de falha. A simulação identifica esse componente comparando os tempos gerados para cada um ao longo das 8760 horas de simulação.
4. Avaliar o estado do componente e o impacto para os consumidores;

Nessa etapa, verifica-se se o componente falhou ou se ainda está em estado de sucesso. Se o componente estiver em estado de sucesso, gera-se um novo tempo de falha. Se o componente falhou, avalia-se o impacto dessa falha no sistema e nos consumidores afetados.

O impacto é avaliado usando matrizes de alcance, que identificam os consumidores afetados pela falha do componente. Essas matrizes ajudam a determinar a resposta do sistema, como a possibilidade de seccionamento de carga ou a utilização de uma fonte alternativa. Os detalhes dessa análise podem ser vistos no Apêndice C, na seção C.3. A seguir, estão os passos para essa análise.

- Identificação do estado anterior;
 - Identificação dos nós;
 - Inicialização das Matrizes Barramento dos protetores (BDP), Barramento das seccionadoras (BDS) e Barramento das alimentações (BDA);
 - Atualização das Matrizes;
 - Cálculo do Estado da Carga;
 - Impacto da falha no sistema.
5. Incrementar as funções-teste dos índices de confiabilidade; Após a análise da falha e a identificação dos consumidores afetados, atualizam-se os índices de confiabilidade do sistema. As funções de teste desses índices são incrementadas e calculam-se o número de interrupções para os consumidores ($N_{\text{int_cons}}$), a duração total das interrupções ($D_{\text{int_cons}}$) e a energia não suprida ($E_{\text{int_cons}}$), todos em um período de um ano,

$$N_{\text{int_cons}} = N_{\text{int_cons}} + N_{ca} \quad (3.5)$$

$$D_{\text{int_cons}} = D_{\text{int_cons}} + N_{ca} \times D_{\text{int}} \quad (3.6)$$

$$E_{\text{int_cons}} = E_{\text{int_cons}} + P_c \times D_{\text{int}} \quad (3.7)$$

em que N_{ca} é o número de consumidores afetados, D_{int} duração da interrupção e P_c potência cortada.

O SAIFI é calculado com base no número de consumidores afetados pela falha em relação ao total de consumidores no sistema.

$$SAIFI(y_k) = \frac{N_{\text{int_cons}}(y_k)}{\text{Número total de consumidores}} \quad (3.8)$$

O SAIDI representa a duração média das interrupções para os consumidores impactados.

$$SAIDI(y_k) = \frac{D_{\text{int_cons}}(y_k)}{\text{Número total de consumidores}} \quad (3.9)$$

A ENS calcula a quantidade de energia que deixou de ser fornecida devido à falha.

$$ENS(y_k) = E_{\text{int_cons}}(y_k) \quad (3.10)$$

Durante a simulação, os índices de confiabilidade são atualizados continuamente a cada falha no sistema. Os resultados acumulados após cada iteração são armazenados para análise posterior, permitindo a obtenção de métricas consolidadas ao final da simulação. Esse procedimento é repetido várias vezes, caracterizando a simulação de Monte Carlo, que captura a variabilidade dos tempos de falha e de reparo.

6. Gerar uma nova duração de estado; Após a identificação da falha do componente e a avaliação dos impactos sobre os consumidores, repete-se o processo de transição entre os estados de sucesso e falha. Esse procedimento continua até que o período de análise seja concluído, registrando todas as falhas e reparos ocorridos. Assim, é possível calcular os índices de confiabilidade para o período correspondente, conforme,

$$\tilde{E}[SAIFI] = \frac{1}{N_y} \sum_{i=1}^{N_y} SAIFI(y_k) \quad (3.11)$$

$$\tilde{E}[SAIDI] = \frac{1}{N_y} \sum_{i=1}^{N_y} SAIDI(y_k) \quad (3.12)$$

$$\tilde{E}[ENS] = \frac{1}{N_y} \sum_{i=1}^{N_y} ENS(y_k) \quad (3.13)$$

7. Avaliar o coeficiente de variação para todos os índices de confiabilidade;

Esse método permite uma análise eficiente dos índices de confiabilidade em sistemas complexos. Ao final da simulação, o coeficiente de variação e a incerteza é calculado para cada índice de confiabilidade, conforme,

$$Var(\tilde{E}[SAIFI]) = \frac{Var(SAIFI)}{\text{Número de amostras}} \quad (3.14)$$

$$Var(\tilde{E}[SAIDI]) = \frac{Var(SAIDI)}{\text{Número de amostras}} \quad (3.15)$$

$$Var(\tilde{E}[ENS]) = \frac{Var(ENS)}{\text{Número de amostras}} \quad (3.16)$$

$$\beta[SAIFI] = \frac{\sqrt{Var[\tilde{E}[SAIFI]]/N_s}}{\tilde{E}[SAIFI]} \quad (3.17)$$

$$\beta[SAIDI] = \frac{\sqrt{Var[\tilde{E}[SAIDI]]/N_s}}{\tilde{E}[SAIDI]} \quad (3.18)$$

$$\beta[ENS] = \frac{\sqrt{Var[\tilde{E}[ENS]]/N_s}}{\tilde{E}[ENS]} \quad (3.19)$$

Com isso, a simulação de Monte Carlo é finalizada após atingir o número de iterações pré-determinado.

Mais detalhes do SMCS implementado são discutidos e apresentados no Apêndice C. Os códigos para as duas abordagens anteriores foram implementados em Python. Eles estão disponíveis tanto nos Apêndices B e C, quanto no repositório [MirieleCampos/AnáliseDeConfiabilidade](#) no GitHub.

3.4 Conclusões parciais

Neste capítulo, é apresentada a metodologia analítica e de Monte Carlo Sequencial para a análise de confiabilidade de redes de distribuição. O caso estudado envolve um sistema radial com cinco barras, um disjuntor, quatro cargas, fusíveis, chaves seccionadoras e duas fontes de energia, abrangendo quatro cenários com diferentes níveis de complexidade. O algoritmo pode lidar com sistemas radiais, realizar manobras de isolamento de falhas com seccionadoras e incluir fontes de energia manobráveis. Por fim, os códigos em Python são disponibilizados abertamente, facilitando o acesso e possíveis melhorias por profissionais e estudantes.

4 Resultados

Neste capítulo, são apresentados os cálculos e discussões derivados da aplicação da metodologia analítica ao sistema. Além disso, são exibidos os índices de confiabilidade obtidos por meio do programa apresentado no capítulo anterior, com comparações realizadas com a literatura de Billinton e Allan (1984).

4.1 Dados do sistema

A Tabela 4 mostra os índices de desempenho para cada caso do sistema, da Figura 4 (BILLINTON; ALLAN, 1984). Tais índices servem de base para a validação do algoritmo implementado e incluem,

- taxa de falha no ponto de carga (λ_s);
- duração da interrupção no ponto de carga (r_s);
- indisponibilidade de serviço no ponto de carga (U_s).

Tabela 4 – Taxa de falha, duração da interrupção de serviços e indisponibilidade nos pontos de carga para cada caso.

	Caso 1	Caso 2	Caso 3	Caso 4
Ponto de carga A				
$\lambda(\text{falhas/ano})$	2,2	1,0	1,0	1,0
$r(\text{horas})$	2,73	3,6	1,5	1,5
$U(\text{horas/ano})$	6,0	3,6	1,5	1,5
Ponto de carga B				
$\lambda(\text{falhas/ano})$	2,2	1,4	1,4	1,4
$r(\text{horas})$	2,73	3,14	1,89	1,39
$U(\text{horas/ano})$	6,0	4,4	2,65	1,95
Ponto de carga C				
$\lambda(\text{falhas/ano})$	2,2	1,2	1,2	1,2
$r(\text{horas})$	2,73	3,33	2,75	1,88
$U(\text{horas/ano})$	6,0	4,0	3,3	2,25
Ponto de carga D				
$\lambda(\text{falhas/ano})$	2,2	1,0	1,0	1,0
$r(\text{horas})$	2,73	3,6	1,5	1,5
$U(\text{horas/ano})$	6,0	3,6	1,5	1,5

Fonte: Adaptado de Billinton e Allan (1984).

Ambas as simulações, SMA e SMCS, foram executadas para cada um dos casos da Figura 4. No caso da SMCS, não foi aplicado nenhum critério de convergência devido à limitação de tempo. Assim, foram realizadas 2000 iterações para cada caso, valor escolhido de forma empírica para o sistema analisado. Uma discussão mais detalhada sobre critérios de convergência é um tópico relevante para trabalhos futuros.

4.2 Resultados de simulação

As Tabela 5, Tabela 6, Tabela 7 e Tabela 8 mostram os resultados de simulação para os casos 1, 2, 3 e 4 respectivamente do sistema teste. Estão inclusos os índices da literatura, SMA e SMCS. Uma abordagem mais detalhada dos cálculos analíticos pode ser encontrada no Apêndice A. Os resultados mostram a boa precisão dos resultados obtidos. Os índices de confiabilidade possuem correspondência exata entre os dados da literatura e a SMA, enquanto a SMCS apresenta um valor ligeiramente diferente, mas ainda próximo. Esses achados confirmam que a SMCS fornecem índices de confiabilidade comparáveis a SMA, evidenciando a precisão do algoritmo desenvolvido.

Tabela 5 – Resultados para os índices de confiabilidade do Caso 1.

Índices	Unidade	Dados da Literatura	Resultados da SMA	Resultados da SMCS	Incerteza β (%)
SAIFI	oc/ano	2,20	2,20	2,1915	0,0154
SAIDI	h/ano	6,00	6,00	6,0263	0,0220
CAIDI	h	2,73	2,73	2,4349	0,0204
ASAI	pu	0,9993	0,9993	0,9993	0,0000
ASUI	pu	0,0007	0,0007	0,0007	0,0220
ENS	kWh/ano	84.000,00	84.000,00	84.369,00	0,0220
AENS	kWh/c.ano	28,00	28,00	28,1229	0,6186

Fonte: Elaborado pelo autor (2024).

Na Tabela 5, e nas tabelas seguintes, as unidades “oc” representam o número de ocorrências de interrupções, enquanto “pu” é uma unidade adimensional, utilizada para indicar frações relativas de grandezas. Já “kWh/c.ano” refere-se à energia não suprida por consumidor por ano.

Tabela 6 – Resultados para os índices de confiabilidade do Caso 2.

Índices	Unidade	Dados da Literatura	Resultados da SMA	Resultados da SMCS	Incerteza β (%)
SAIFI	oc/ano	1,15	1,15	1,1124	0,0184
SAIDI	h/ano	3,91	3,91	3,6699	0,0285
CAIDI	h	3,39	3,39	2,6059	0,0215
ASAI	pu	0,9996	0,9996	0,9996	0,0000
ASUI	pu	0,0004	0,0004	0,0004	0,0285
ENS	kWh/ano	54.800,00	54.800,00	51.432,00	0,0285
AENS	kWh/c.ano	18,30	18,30	17,1441	0,4886

Fonte: Elaborado pelo autor (2024).

Tabela 7 – Resultados para os índices de confiabilidade do Caso 3.

Índices	Unidade	Dados da Literatura	Resultados da SMA	Resultados da SMCS	Incerteza β (%)
SAIFI	oc/ano	1,15	1,15	1,1669	0,0184
SAIDI	h/ano	2,58	2,58	2,3882	0,0290
CAIDI	h	2,23	2,23	1,8147	0,0222
ASAI	pu	0,9997	0,9997	0,9997	0,0000
ASUI	pu	0,0003	0,0003	0,0003	0,0290
ENS	kWh/ano	35.200,00	35.200,00	32.353,00	0,0292
AENS	kWh/c.ano	11,70	11,70	10,7844	0,3149

Fonte: Elaborado pelo autor (2024).

Tabela 8 – Resultados para os índices de confiabilidade do Caso 4.

Índices	Unidade	Dados da Literatura	Resultados da SMA	Resultados da SMCS	Incerteza β (%)
SAIFI	oc/ano	1,15	1,15	1,1205	0,0182
SAIDI	h/ano	1,80	1,80	1,4536	0,0233
CAIDI	h	1,56	1,56	1,2809	0,0210
ASAI	pu	0,9998	0,9998	0,9998	0,0000
ASUI	pu	0,0002	0,0002	0,0002	0,0233
ENS	kWh/ano	25.100,00	25.100,00	20.227,00	0,0236
AENS	kWh/c.ano	8,40	8,40	6,7424	0,1589

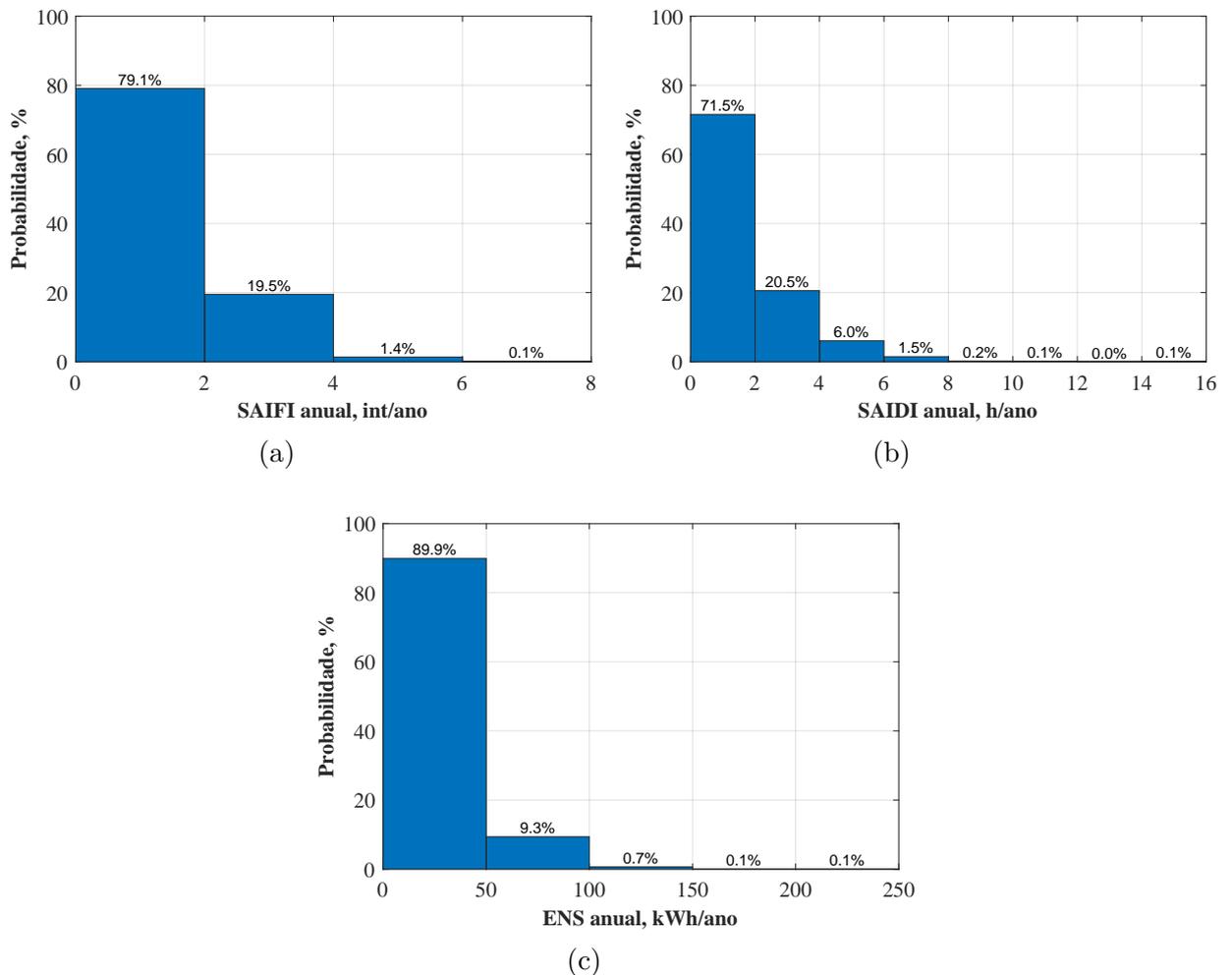
Fonte: Elaborado pelo autor (2024).

4.3 Discussão dos resultados

Para obter mais informações sobre o comportamento do sistema, foram agrupados valores por meio da distribuição de frequências, gerando histogramas para os índices SAIFI, SAIDI e ENS, de acordo com as iterações da simulação de Monte Carlo para o Caso 4. Esses histogramas representam estimativas das distribuições de probabilidade dos índices

de confiabilidade e estão apresentados nas Figura 7a, Figura 7b e Figura 7c, mostrando a distribuição dos índices calculados ao longo das 2000 iterações realizadas em cada caso.

Figura 7 – Histogramas de índices de confiabilidade para a SMCS.



Legenda: (a) SAIFI, (b) SAIDI, (c) ENS. Fonte: Elaborado pelo autor (2024).

Os histogramas estimam as distribuições de probabilidade, permitindo visualizar algumas dessas probabilidades. De acordo com o histograma da Figura 7a, o risco de o SAIFI exceder 2,0 ocorrências/ano é de aproximadamente 20,9%. Na Figura 7b, observa-se que o SAIDI apresenta um risco de ser superior a 2,0 horas/ano em torno de 28,5%. Já o indicador ENS, mostrado na Figura 7c, indica um risco próximo de 10,1% de ultrapassar 50 kWh/ano.

Os resultados apresentados, como os histogramas de SAIFI, SAIDI e ENS, só podem ser obtidos por meio da abordagem de Monte Carlo. A SMCS gera uma série de simulações ao longo de várias iterações, capturando a distribuição de probabilidade das falhas de cada componente e proporcionando uma análise detalhada do comportamento probabilístico do sistema. A principal vantagem da simulação de Monte Carlo em relação ao

método analítico é sua capacidade de lidar com a complexidade e a incerteza dos sistemas reais de maneira mais flexível e precisa. Enquanto os métodos analíticos geralmente se baseiam em suposições simplificadas e modelos determinísticos, que não refletem a variabilidade e os comportamentos estocásticos dos componentes, a simulação de Monte Carlo permite modelar diretamente essas incertezas, oferecendo uma representação mais realista e abrangente do desempenho do sistema.

4.4 Conclusões parciais

Neste capítulo, foram apresentados os resultados obtidos a partir da metodologia analítica e da simulação de Monte Carlo para avaliação da confiabilidade de redes de distribuição, conforme descrito no capítulo anterior. Os índices de desempenho do sistema de teste foram calculados e analisados. Em seguida, os resultados gerados pelos algoritmos implementados foram comparados entre si, bem como com os dados da literatura existente, especificamente os de Billinton e Allan (1984). Essas comparações permitiram avaliar a precisão e a robustez das metodologias aplicadas, evidenciando as vantagens do método de Monte Carlo em termos de flexibilidade e capacidade de modelar a incerteza de sistemas reais.

5 Conclusão

A qualidade do serviço de distribuição de energia elétrica e o desempenho do sistema podem ser avaliados por meio do monitoramento de interrupções e dos indicadores de continuidade. Existem duas principais abordagens para avaliar a confiabilidade das redes: métodos analíticos e de simulação. Os métodos analíticos são rápidos, mas fornecem apenas estimativas médias. Já os métodos de simulação, como o SMC, estimam propriedades estocásticas, incluindo distribuições de probabilidade. Este estudo tem como objetivo comparar as vantagens e desvantagens dessas abordagens na análise da confiabilidade de sistemas de distribuição.

Neste contexto, é feita uma revisão bibliográfica sobre a confiabilidade de sistemas de distribuição, incluindo os principais indicadores de desempenho e confiabilidade. Além disso, são revisadas as ferramentas matemáticas utilizadas na análise de confiabilidade. A SMA e a SMCS. O sistema base é radial, composto por 5 barras, um disjuntor, 4 cargas, fusíveis, chaves seccionadoras e duas fontes de energia (uma principal e outra alternativa), foram calculados os principais índices de confiabilidade, para cada uma das metodologias. Foram analisados 4 casos com variações de complexidade, permitindo a identificação de pontos que afetam a confiabilidade dos sistemas.

Foi realizada a análise de viabilidade para a implementação das metodologias. Os resultados obtidos mostram que o SMA apresenta um custo computacional significativamente menor, sendo uma abordagem eficiente para sistemas de pequeno porte ou que exigem respostas rápidas. No entanto, para sistemas mais complexos ou com maior quantidade de variáveis estocásticas, mostrou-se limitado na captura de informações importantes do comportamento do sistema, uma vez que não simula diretamente a aleatoriedade das falhas e reparos. Por outro lado, a SMCS, apesar de demandar um maior esforço computacional, forneceu uma análise mais detalhada e precisa, capaz de simular eventos de falhas de forma cronológica e realista. Isso permitiu uma melhor avaliação da confiabilidade em cenários onde a aleatoriedade é um fator determinante. A SMCS se destacou, principalmente, pela capacidade de lidar com uma ampla gama de distribuições de probabilidade, oferecendo maior flexibilidade em comparação ao SMA.

Como contribuição principal, este trabalho oferece uma análise comparativa preliminar das duas metodologias, explicada em diferentes cenários que refletem situações reais. Profissionais da área de planejamento de redes de distribuição podem utilizar essas informações para estudo e renovação de conteúdos importantes para a suas carreiras. Adicionalmente, o código desenvolvido para ambas as metodologias pode ser utilizado como base para futuras pesquisas e aprimoramentos.

As seguintes linhas de pesquisas podem ser exploradas em trabalhos futuros:

- explorar métodos de convergência para o SMCS;
- testar outras distribuições de probabilidades, diferentes da exponencial;
- aplicar as metodologias propostas a sistemas reais e de maior escala;

Referências

- ANEEL. *Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional - PRODIST, Módulo 8 – Qualidade de Energia Elétrica*. 2021. Acessado em: 11 de janeiro de 2024. Disponível em: <https://www2.aneel.gov.br/cedoc/aren2021956_2_7.pdf>. 13
- ASCHIDAMINI, G. L. et al. Expansion planning of power distribution systems considering reliability: A comprehensive review. *Energies*, v. 15, n. 6, 2022. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/15/6/2275>>. 3, 6, 7
- BILLINTON, R.; ALLAN, R. N. *Reliability Evaluation Of Power Systems*. New York: Plenum Publishing Corp., New York, NY, 1984. 1, 6, 9, 12, 14, 19, 29, 33, 42, 43, 44
- BILLINTON, R.; ALLAN, R. N. *Reliability evaluation of engineering systems*. New York: Plenum press: Springer, 1992. 1, 8, 9, 10, 13, 14
- BILLINTON, R.; LI, W. *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods*. Address of the Publisher: Springer Science & Business Media, 1994. 7, 8, 13, 14, 15
- BILLINTON, R.; WANG, P. Teaching distribution system reliability evaluation using monte carlo simulation. *IEEE Transactions on Power Systems*, v. 14, n. 2, p. 397–403, May 1999. Disponível em: <<https://ieeexplore.ieee.org/document/761856>>. 2
- BROWN, R. E. *Electric power distribution reliability*. Boca Raton: CRC press, 2017. 1, 6, 10, 14
- CALADO, D. E. D. et al. *Representação de curto-circuito probabilístico na avaliação da confiabilidade de sistemas de distribuição*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Brasil, 2015. Disponível em: <<https://repositorio.ufsc.br/xmlui/handle/123456789/169406>>. 16, 17
- CENTENARO, B.; CATAPANI; RIOS, G. F. *Avaliação da confiabilidade de rede de distribuição de uma concessionária brasileira via método analítico e monte carlo sequencial*. Brasil, 2017. Monografia (Graduação). Universidade Tecnológica Federal do Paraná. Disponível em: <<http://repositorio.utfpr.edu.br/jspui/handle/1/10060>>. 3, 17, 22
- KAFLE, P.; BHANDARI, M.; RANA, L. B. Reliability analysis techniques in distribution system: A comprehensive review. *International Journal of Engineering and Manufacturing*, v. 12, n. 2, p. 11–24, April 2022. Disponível em: <<https://www.mecs-press.org/ijem/ijem-v12-n2/v12n2-2.html>>. 3
- LAW, A. M.; KELTON, W. D. Simulation modeling and analysis. mcgraw-hili. *Google Scholar Google Scholar Digital Library Digital Librar*, <https://faculty.ksu.edu.sa/sites/default/files/index.pdf>, 1991. 15
- MATTOS, R. S. d. et al. Integração de modelos econométrico e de insumo-produto para previsões de longo prazo da demanda de energia no brasil. *Estudos Econômicos (São Paulo) - SciELO Brasil*, v. 38, n. 1, p. 675–699, December 2008. Disponível em: <<https://www.scielo.br/j/ee/a/5wF9zbjs9b4k3NsZW3gQ9Nv/?lang=pt>>. 1

- MONTICELLI, A.; GARCIA, A. *Introdução a Sistemas de Energia Elétrica*. Campinas - SP: Editora Unicamp, 2011. 1
- NASCIMENTO, L. C. d. *Avaliação da confiabilidade de sistemas de distribuição utilizando processamento distribuído*. Tese (Doutorado) — Universidade Federal de Itajubá, Brasil, 2009. Disponível em: <<https://repositorio.unifei.edu.br/jspui/handle/123456789/1545>>. 2
- ROSA, M.; ISSICABA, D.; LOPES, J. P. Distribution systems performance evaluation considering islanded operation. *CPEs - Indexed Articles in Conferences*, <http://repositorio.inesctec.pt/handle/123456789/2209>, 2011. 7, 17
- ROSA, M. A. da. *Agent-based Technology Applied to Power Systems Reliability*. Tese (Doutorado) — University of Porto, Portugal, 2009. Disponível em: <<https://www.proquest.com/openview/7eddf363d049a16eea01d812e1c757ed/1?pq-origsite=gscholar&cbl=2026366>>. 7
- SANTOS, E. P. d. *Estratégias de análise em regime permanente para avaliação de confiabilidade composta de sistemas de energia elétrica*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Brasil, 2019. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/219182>>. 7, 8
- SELIM, F. et al. Flexible active power control for pv-ess systems: A review. *Energies*, MDPI, v. 14, n. 21, p. 7388, 2021. 3
- SILVA, A. L. D. et al. Analytical and monte carlo approaches to evaluate probability distributions of interruption duration. *IEEE Transactions on Power Systems*, <https://ieeexplore.ieee.org/document/1490585>, v. 20, n. 3, p. 1341–1348, 2005. 16
- SILVA, J. L. Reis e. *Avaliação da confiabilidade em sistemas elétricos com base nos parâmetros de qualidade da energia*. Brasil, 2008. Monografia (Graduação). Universidade Federal de Minas Gerais. Disponível em: <<http://hdl.handle.net/1843/BUBD-9DLGF6>>. 2
- SILVA, M. C. F. A. d. *Metodologia de solução aplicada ao planejamento da expansão de sistemas de distribuição de energia elétrica*. Brasil, 2016. Monografia (Graduação). Universidade Federal de Santa Maria. Disponível em: <<http://repositorio.ufsm.br/handle/1/15066>>. 14
- SPERANDIO, M. *Planejamento da automação de sistemas de manobra em redes de distribuição*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Brasil, 2008. Disponível em: <<http://repositorio.ufsc.br/xmlui/handle/123456789/91427>>. 45, 47
- ZARDO, F. *Metodologia de análise de confiabilidade em redes ativas de distribuição de energia elétrica*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Brasil, 2014. Disponível em: <<https://repositorio.ufsc.br/xmlui/handle/123456789/123382>>. 3

Apêndices

APÊNDICE A – Índices de confiabilidade do método analítico

A.1 Cálculo dos índices de confiabilidade para o caso 1 usando método analítico

Os índices de desempenho do sistema são definidos por (2.8), (2.9) e (2.10). Com base nas informações, nota-se que os indicadores de confiabilidade dos pontos de carga são idênticos. Para melhorar o entendimento, a seguir os índices da “carga A” são calculados de acordo com os dados, é possível calcular os parâmetros de confiabilidade, que também são listados na ??, como se segue,

$$\begin{aligned}\lambda_A &= 0,2 + 0,1 + 0,3 + 0,2 + 0,2 + 0,6 + 0,4 + 0,2 \\ &= 2,2 \text{ interrupções/cliente ano}\end{aligned}$$

$$\begin{aligned}U_A &= 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 4 + 0,2 * 2 + 0,6 * 2 + 0,4 * 2 + 0,2 * 2 \\ &= 6 \text{ horas/cliente ano}\end{aligned}$$

$$\begin{aligned}r_A &= \frac{6,0}{2,2} \\ &= 2,73 \text{ horas}\end{aligned}$$

Os cálculos dos índices de confiabilidade para todos os casos são obtidos segundo as equações (2.11), (2.12), (2.13), (2.14), (2.15), (2.16) e (2.17).

Assim, para Caso 1, tem-se,

$$\begin{aligned}
 \text{SAIFI} &= \frac{2,2 \times 1000 + 2,2 \times 800 + 2,2 \times 700 + 2,2 \times 500}{1000 + 800 + 700 + 500} \\
 &= 2,2 \text{ interrupções/cliente ano} \\
 \text{SAIDI} &= \frac{6 \times 1000 + 6 \times 800 + 6 \times 700 + 6 \times 500}{1000 + 800 + 700 + 500} \\
 &= 6 \text{ horas/cliente ano} \\
 \text{CAIDI} &= \frac{2,2 \times 1000 + 2,2 \times 800 + 2,2 \times 700 + 2,2 \times 500}{6 \times 1000 + 6 \times 800 + 6 \times 700 + 6 \times 500} \\
 &= 2,73 \text{ horas/interrupção cliente} \\
 \text{ASAI} &= \frac{3000 \times 8760 - (6 \times 1000 + 6 \times 800 + 6 \times 700 + 6 \times 500)}{3000 \times 8760} \\
 &= 0,999315 \\
 \text{ASUI} &= \frac{6 \times 1000 + 6 \times 800 + 6 \times 700 + 6 \times 500}{3000 \times 8760} \\
 &= 0,000685 \\
 \text{ENS} &= 6 \times 5000 + 6 \times 4000 + 6 \times 3000 + 6 \times 2000 \\
 &= 84000 \text{ kWh/ano} \\
 \text{AENS} &= \frac{84000}{3000} \\
 &= 28 \text{ kWh/cliente ano}
 \end{aligned}$$

A Tabela 9 resume os índices do sistema para o Caso 1.

A.2 Cálculo dos índices de confiabilidade para o caso 2 usando método analítico

Procedendo de forma análoga para o caso 2, para as “Cargas A”, “B”, “C” e “D”, tem-se:

$$\begin{aligned}
 \lambda_A &= 0,2 + 0,1 + 0,3 + 0,2 + 0,2 = 1,0 \text{ interrupções/cliente ano} \\
 U_A &= 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 4 + 0,2 * 2 = 3,6 \text{ horas/cliente ano} \\
 r_A &= \frac{3,6}{1,0} = 3,6 \text{ horas} \\
 \\
 \lambda_B &= 0,2 + 0,1 + 0,3 + 0,2 + 0,6 = 1,4 \text{ interrupções/cliente ano} \\
 U_B &= 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 4 + 0,6 * 2 = 4,4 \text{ horas/cliente ano} \\
 r_B &= \frac{4,4}{1,4} = 3,14 \text{ horas}
 \end{aligned}$$

$$\lambda_C = 0,2 + 0,1 + 0,3 + 0,2 + 0,4 = 1,2 \text{ interrupções/cliente ano}$$

$$U_C = 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 4 + 0,4 * 2 = 4,0 \text{ horas/cliente ano}$$

$$r_C = \frac{4,0}{1,2} = 3,33 \text{ horas}$$

$$\lambda_D = 0,2 + 0,1 + 0,3 + 0,2 + 0,2 = 1,0 \text{ interrupções/cliente ano}$$

$$U_D = 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 4 + 0,2 * 2 = 3,6 \text{ horas/cliente ano}$$

$$r_D = \frac{3,6}{1,0} = 3,6 \text{ horas}$$

Os índices de confiabilidade são melhorados para todos os pontos de carga, embora a quantidade de melhoria seja diferente para cada um. O ponto de carga com menor confiabilidade é o B, o que se deve principalmente ao impacto significativo das falhas no seu distribuidor lateral. A Tabela 10 resume os índices do sistema.

A.3 Cálculo dos índices de confiabilidade para o caso 3 usando método analítico

A taxa de falha é calculada da mesma forma que anteriormente, já no caso da indisponibilidade de serviço, o cálculo deverá acompanhar o novo tempo de reparo estabelecido para comutação das chaves e tempo de reparo, sob esta nova configuração operacional, podem ser recalculados da maneira mostrada no algoritmo da seção 3.2.

$$\lambda_A = 0,2 + 0,1 + 0,3 + 0,2 + 0,2 = 1,0 \text{ interrupções/cliente ano}$$

$$U_A = 0,2 * 4 + 0,1 * 0,5 + 0,3 * 0,5 + 0,2 * 0,5 + 0,2 * 2 = 1,5 \text{ horas/cliente ano}$$

$$r_A = \frac{1,5}{1,0} = 1,5 \text{ horas}$$

$$\lambda_B = 0,2 + 0,1 + 0,3 + 0,2 + 0,6 = 1,4 \text{ interrupções/cliente ano}$$

$$U_B = 0,2 * 4 + 0,1 * 4 + 0,3 * 0,5 + 0,2 * 0,5 + 0,6 * 2 = 2,65 \text{ horas/cliente ano}$$

$$r_B = \frac{2,65}{1,4} = 1,89 \text{ horas}$$

$$\lambda_C = 0,2 + 0,1 + 0,3 + 0,2 + 0,4 = 1,2 \text{ interrupções/cliente ano}$$

$$U_C = 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 5 + 0,4 * 2 = 3,3 \text{ horas/cliente ano}$$

$$r_C = \frac{3,3}{1,2} = 2,75 \text{ horas}$$

$$\lambda_D = 0,2 + 0,1 + 0,3 + 0,2 + 0,2 = 1,0 \text{ interrupções/cliente ano}$$

$$U_D = 0,2 * 4 + 0,1 * 4 + 0,3 * 4 + 0,2 * 4 + 0,2 * 2 = 3,6 \text{ horas/cliente ano}$$

$$r_D = \frac{3,6}{1,0} = 3,6 \text{ horas}$$

Neste caso, houve melhora nos índices de confiabilidade dos pontos de “Carga A”, “B” e “C”, observando-se um aprimoramento mais significativo nos pontos mais próximos à fonte de alimentação. Em contrapartida, o ponto de “Carga D” não registra melhorias, visto que o processo de isolamento não atenua as consequências das falhas que ocorrem especificamente nessa área. A Tabela 11 mostra os novos índices de confiabilidade do sistema.

A.4 Cálculo dos índices de confiabilidade para o caso 4 usando método analítico

Para o caso 4 e seguindo o algoritmo da seção 3.2, os índices de ponto de carga podem ser calculados como segue,

$$\lambda_A = 0,2 + 0,1 + 0,3 + 0,2 + 0,2 = 1,0 \text{ interrupções/cliente ano}$$

$$U_A = 0,2 * 4 + 0,1 * 0,5 + 0,3 * 0,5 + 0,2 * 0,5 + 0,2 * 2 = 1,5 \text{ horas/cliente ano}$$

$$r_A = \frac{1,5}{1,0} = 1,5 \text{ horas}$$

$$\lambda_B = 0,2 + 0,1 + 0,3 + 0,2 + 0,6 = 1,4 \text{ interrupções/cliente ano}$$

$$U_B = 0,2 * 4 + 0,1 * 4 + 0,3 * 0,5 + 0,2 * 0,5 + 0,6 * 2 = 1,95 \text{ horas/cliente ano}$$

$$r_B = \frac{1,95}{1,4} = 1,39 \text{ horas}$$

$$\lambda_C = 0,2 + 0,1 + 0,3 + 0,2 + 0,4 = 1,2 \text{ interrupções/cliente ano}$$

$$U_C = 0,2 * 0,5 + 0,1 * 0,5 + 0,3 * 4 + 0,2 * 0,5 + 0,4 * 2 = 2,25 \text{ horas/cliente ano}$$

$$r_C = \frac{2,25}{1,2} = 1,88 \text{ horas}$$

$$\lambda_D = 0,2 + 0,1 + 0,3 + 0,2 + 0,2 = 1,0 \text{ interrupções/cliente ano}$$

$$U_D = 0,2 * 0,5 + 0,1 * 0,5 + 0,3 * 0,5 + 0,2 * 4 + 0,2 * 2 = 1,5 \text{ horas/cliente ano}$$

$$r_D = \frac{1,5}{1,0} = 1,5 \text{ horas}$$

Os índices de confiabilidade do sistema para o quarto caso são alterados e apresentados na Tabela 12.

A.5 Validação da literatura

Com o objetivo de validação do algoritmo apresentado, os índices obtidos na bibliografia de Billinton e Allan (1984), foram dispostos em conjunto com índices obtidos

Tabela 9 – Índices de confiabilidade do Caso 1: referência e algoritmo próprio.

Índices	Resultados Billinton e Allan (1984)	Resultados da SMA	Erro (%)
SAIFI	2,2	2,2	0,00
SAIDI	6	6	0,00
CAIDI	2,73	2,727	0,11
ASAI	0,999315	0,999315	0,00
ASUI	0,000685	0,000685	0,00
ENS	84.000	84.000	0,00
AENS	28	28	0,00

Fonte: Adaptado de Billinton e Allan (1984).

Tabela 10 – Índices de confiabilidade do Caso 2: referência e algoritmo próprio.

Índices	Resultados Billinton e Allan (1984)	Resultados da SMA	Erro (%)
SAIFI	1,15	1,153	0,26
SAIDI	3,91	3,907	0,08
CAIDI	3,39	3,387	0,09
ASAI	0,999554	0,999554	0,00
ASUI	0,000446	0,000446	0,00
ENS	54.800	54.800	0,00
AENS	18,3	18,27	0,16

Fonte: Adaptado de Billinton e Allan (1984).

Tabela 11 – Índices de confiabilidade do Caso 3: referência e algoritmo próprio.

Índices	Resultados Billinton e Allan (1984)	Resultados da SMA	Erro (%)
SAIFI	1,15	1,153	0,26
SAIDI	2,58	2,577	0,12
CAIDI	2,23	2,234	0,18
ASAI	0,999706	0,999706	0,00
ASUI	0,000294	0,000294	0,00
ENS	35.200	35.200	0,00
AENS	11,7	11,73	0,26

Fonte: Adaptado de Billinton e Allan (1984).

no programa implementado para que pudessem ser comparados. Os resultados de cada caso podem ser encontrados na Tabela 9, Tabela 10, Tabela 11 e Tabela 12.

O erro percentual foi calculado a partir de,

$$e = \frac{V_b - V_x}{V_b} \times 100\% \quad (\text{A.1})$$

em que V_b é o valor base, neste caso o valor de Billinton e Allan (1984) e V_x é o valor

Tabela 12 – Índices de confiabilidade do Caso 4: referência e algoritmo próprio.

Índices	Resultados Billinton e Allan (1984)	Resultados da SMA	Erro (%)
SAIFI	1,15	1,153	0,26
SAIDI	1,8	1,795	0,28
CAIDI	1,56	1,556	0,26
ASAI	0,999795	0,999795	0,00
ASUI	0,000205	0,000205	0,00
ENS	25.100	25.050	0,20
AENS	8,4	8,35	0,60

Fonte: Adaptado de Billinton e Allan (1984).

encontrado pelo algoritmo do método analítico implementado.

A maior diferença entre os valores esperados e os alcançados causou um erro de 0,60%. Este valor pode ser considerado pequeno e pode ser atribuída aos arredondamentos e à quantidade de casas decimais usadas na bibliografia consultada. Assim, com base nos resultados apresentados, é possível concluir que o método analítico implementado fornece as soluções esperadas.

APÊNDICE B – Algoritmo Método Analítico

B.1 Detalhes de implementação

Seguindo o estudo de confiabilidade do sistema de teste, foi desenvolvido um algoritmo para automatizar e facilitar os cálculos dos indicadores de confiabilidade, considerando disjuntores, chaves fusíveis, seccionadoras e chaves de seção para a alimentação alternativa. O desenvolvimento desse algoritmo foi realizado na linguagem de programação Python, conhecida por sua eficiência e versatilidade em tarefas de computação e análise de dados, além de ser valorizado por sua biblioteca rica e sua capacidade de integrar-se facilmente com outras ferramentas e linguagens, o que o torna ideal para resolver problemas complexos de engenharia.

B.1.1 Inserção dos dados do sistema teste

Para dar início ao desenvolvimento do programa, é essencial incluir todas as informações referentes à rede do sistema. Foi criada uma lista de instruções, na qual foram incluídos os dados referentes a cada componente do sistema. Foram criadas duas listas de dados: (*i*) uma contendo as taxas de falhas, indisponibilidade de serviço e o tempo de reparo para cada seção do sistema radial; (*ii*) outra contendo instruções referentes aos pontos de carga do circuito, número de consumidores e ponto de carga em kW. Em seguida busca-se matrizes auxiliares para identificação dos nós e dos ramais de alimentação do sistema.

B.1.2 Matrizes auxiliares

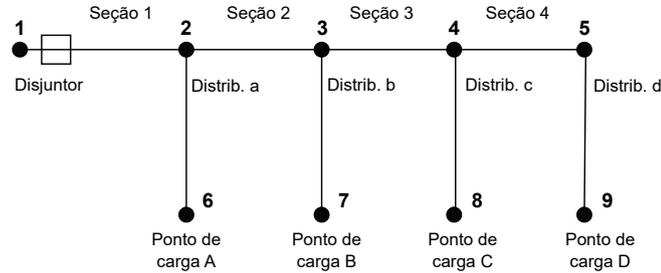
Para uma melhor identificação dos nós e dos ramais de alimentação do sistema, foi necessário utilizar matrizes auxiliares de modo a facilitar os cálculos. A primeira matriz necessária foi a **matriz de alcance**. Essa matriz é determinada a partir de uma outra matriz denominada **matriz de adjacência**, sendo essa, uma ferramenta matemática usada em teoria dos grafos, particularmente na análise de redes, para representar as conexões possíveis entre os pontos ou nós de um sistema. (SPERANDIO, 2008)

B.1.2.0.1 Matriz de Adjacência A

O primeiro passo para montar a matriz de adjacência é conhecer os nós e os demais ramos do sistema, o modelo de distribuição radial do sistema estudado foi representado através de um grafo orientado, conforme mostrado na Figura 8, com a fonte sendo

representada pelo nó de origem em 1 e os demais equipamentos sendo nós conectados por ramos considerando o sentido da corrente elétrica.

Figura 8 – Representação do sistema radial como grafo orientado.



Fonte: Elaborado pelo autor (2024).

Os dados do grafo orientado podem ser representados em formato de tabela para melhor identificação das conexões entre os nós. A Tabela 13 possui: (i) duas colunas, **DE** e **PARA**, e (ii) $(n - 1)$ linhas, em que n representa o total de nós. Nesse caso, o grafo possui nove nós, então a matriz terá oito linhas. Esta matriz define os ramos do grafo: a primeira coluna identifica o nó de partida de cada ramo, referido como **DE**, e a segunda coluna indica o nó de chegada, referido como **PARA**. Essa estrutura permite visualizar as conexões possíveis entre os nós, obedecendo o sentido do fluxo de potência.

Tabela 13 – Conjunto de dados para as definições de ramos para grafo orientado da Figura 8.

DE	PARA
1	2
2	3
3	4
4	5
2	6
3	7
4	8
5	9

Fonte: Elaborado pelo autor (2024).

A partir dos ramos definidos, é possível construir uma matriz de adjacência **A**, de dimensão $n \times n$, que identifica quais nós são adjacentes respeitando a direção dos ramos. Portanto, o elemento A_{ij} da matriz será:

- igual a 1 caso haja um ramo direcionado do nó (i) para o nó (j) ,
- 0 caso contrário.

A matriz de adjacência **A** que foi definida a partir da Tabela 13 pode ser observada em (B.1). Ela indica que existe um caminho entre um nó em uma linha ((i) ou **DE**) e o outro nó

em uma coluna (j ou **PARA**). Para a seção 1 que é o alimentador, por exemplo, o fluxo de energia é do nó (1) para o nó (2), dessa forma a posição $\mathbf{A}_{1,2} = 1$. Porém, não é considerado o fluxo de energia do nó (2) para o nó (1), portanto, $\mathbf{A}_{2,1} = 0$. Analogamente, não existe conexão do nó (1) para os nós (3) a (9), dessa forma, $\mathbf{A}_{1,3} = \mathbf{A}_{1,4} = \dots = \mathbf{A}_{1,9} = 0$.

$$[A] = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}}_{\text{PARA}} \left. \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} \right\} \text{DE} \quad (\text{B.1})$$

B.1.3 Matriz de Alcance R

Ao realizar a multiplicação da matriz de adjacência por ela mesma, é possível identificar caminhos que podem ser percorridos em dois passos, ou seja, utilizando dois ramos consecutivos. Se essa operação for estendida até o número total de ramos e adicionarmos a matriz identidade, obtemos a matriz de alcance R ,

$$R = (I + A)^{n-1} \quad (\text{B.2})$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.3})$$

Esta matriz tem um elemento $R_{ij} = 1$ se for possível ir do nó da linha i até o nó da coluna j , independentemente do número de etapas necessárias para isso. Ou seja, dado que o sistema de distribuição é radial, pode-se constatar pela matriz R que o nó (1), correspondente à subestação, conecta-se a todos os outros nós. Por outro lado, o nó (2) se conecta a todos os nós, exceto ao nó (1), devido à ausência de fluxo reverso. Os nós (5), (6), (7) e (8) da matriz \mathbf{R} têm alcance limitado a si mesmos, pois são terminais (SPERANDIO, 2008).

Com base nessas informações, nota-se a importância do uso da matriz de alcance para a análise da proteção de sistemas de distribuição, pois ela indica quais nós estão diretamente afetados por um nó específico. Isso implica quais cargas estão localizadas a jusante de um equipamento de proteção, como uma chave fusível, por exemplo. Para verificar isso, é suficiente inspecionar as colunas que apresentam elementos iguais a 1 na linha correspondente.

B.1.4 Matriz de protetor

A **matriz de protetores** informa em qual trecho está a proteção que atuará caso o componente falhe. Todos os componentes devem estar nesta matriz e o protetor associado dependerá do caso analisado. Para o caso 1 da ??, por exemplo, tem-se apenas um protetor que é o disjuntor. Logo, a matriz `protetorComponent` é composta apenas de 1s. Por outro lado, para os casos 2 a 4, a matriz de protetores será diferente para os distribuidores “a” até “d” pois devem levar em conta o fusível.

A partir dessa análise foi criada a matriz `protetorComponent` que auxilia na busca dessas informações, conforme trecho de código a seguir. Observe que cada coluna dessa matriz corresponde a um determinado caso, sendo a primeira coluna para identificação dos componentes, a segunda para o caso 1, a terceira para o caso 2, a quarta para o caso 3 e a quinta coluna corresponde ao 4º caso.

```
protetorComponent = [
["Section 1", 1, 1, 1, 1],
["Section 2", 1, 1, 1, 1],
["Section 3", 1, 1, 1, 1],
["Section 4", 1, 1, 1, 1],
["Distrib. a", 1, 6, 6, 6],
["Distrib. b", 1, 7, 7, 7],
["Distrib. c", 1, 8, 8, 8],
["Distrib. d", 1, 9, 9, 9],
]
```

Na execução do código, ambas matrizes (de alcance \mathbf{R} e `protetorComponent`) ajudam no cálculo dos índices de desempenho (2.8), (2.9) e (2.10).

Elas ajudam a determinar se a falha em um determinado componente impacta ou não no índice de desempenho de um ponto de carga. Especificamente, se o valor de $\mathbf{R}_{\text{protetor,carga}}$ é 1 então a falha impacta, se for 0 a falha não impacta.

Por exemplo, deseja-se saber como a falha em cada componente do circuito impacta nos índices de desempenho da “carga A” para o caso 2. Sabe-se que a “carga A” está associada ao nó (6) e a matriz `protetorComponent` informa que, para o caso 2, o protetor da “Seção 1” está associado ao nó (1). Então $\mathbf{R}_{\text{protetor,carga}} = \mathbf{R}_{1,6} = 1$, ou seja, uma falha na “Seção 1” impacta no índice da “carga A”. Analogamente:

- falha na “Seção 1”, protetor (1) e $\mathbf{R}_{1,6} = 1$,

- falha na “Seção 2”, protetor (1) e $\mathbf{R}_{1,6} = 1$,
- falha na “Seção 3”, protetor (1) e $\mathbf{R}_{1,6} = 1$,
- falha na “Seção 4”, protetor (1) e $\mathbf{R}_{1,6} = 1$,
- falha no “Distrib. a”, protetor (6) e $\mathbf{R}_{6,6} = 1$,
- falha no “Distrib. b”, protetor (7) e $\mathbf{R}_{7,6} = 0$,
- falha no “Distrib. c”, protetor (8) e $\mathbf{R}_{8,6} = 0$,
- falha no “Distrib. d”, protetor (9) e $\mathbf{R}_{9,6} = 0$,

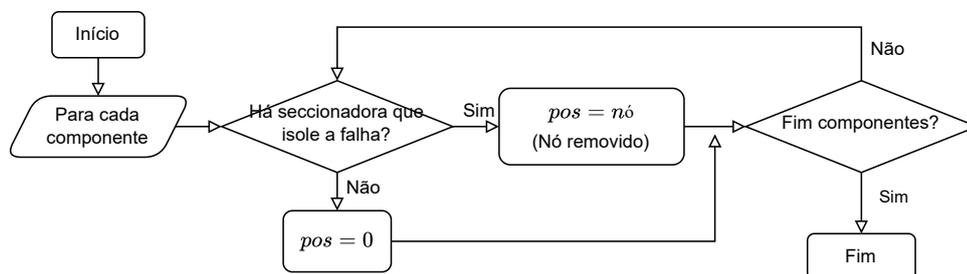
Dessa forma, percebe-se que falhas em qualquer componente impactam nos índices da “carga A”, exceto as falhas nos distribuidores “Distrib. b”, “Distrib. c” e “Distrib. d”. Isto faz sentido já que no caso de falha nestes componentes, o respectivo fusível atuará.

B.1.5 Matriz de seccionadora

A **matriz de seccionadora** informa qual nó do sistema será removido caso a seccionadora isole a falha em um determinado componente. Todos os componentes devem estar nesta matriz e o nó associado dependerá do caso analisado. Para os casos 1 e 2 não há seccionadoras, porém, para os casos 3 a 4, a matriz de seccionadora terá valores.

O fluxograma de formação de `seccionadora_bds` é mostrada na Figura 9. Para cada componente avalia-se se, no caso de falha, existe chave seccionadora que isole o defeito. Caso não haja, o valor será zero. Porém, caso haja chave seccionadora que isole o defeito, o valor na célula da matriz recebe o nó que será removido na manobra.

Figura 9 – Fluxograma para criação da matriz `seccionadora_bds`.



Fonte: Elaborado pelo autor (2024).

A partir dessa análise foi criado a matriz `seccionadora_bds` para cada caso, conforme mostrado na ???. Observe que cada coluna dessa matriz corresponde a um determinado caso, sendo a primeira coluna para identificação dos componentes, a segunda para o caso 1, a terceira para o caso 2, a quarta para o caso 3 e a quinta coluna corresponde ao 4º caso.

```

seccionadora_bds = [
  ["Section 1", "", "", 0, 2],
  ["Section 2", "", "", 3, 3],
  ["Section 3", "", "", 4, 4],
  ["Section 4", "", "", 5, 5],
  ["Distrib. a", "", "", 0, 0],
  ["Distrib. b", "", "", 0, 0],
  ["Distrib. c", "", "", 0, 0],
  ["Distrib. d", "", "", 0, 0],
]

```

Para esclarecer melhor o funcionamento, suponha que deseja-se construir `seccionadora_bds` para o caso 3, assim os passos serão:

- na “Seção 1”, caso haja falha, não há seccionadora que isole. Logo, $bds = 0$;
- na “Seção 2”, caso haja falha, a chave (1) é aberta e o disjuntor religado. Desta forma, o nó (3) é removido. Logo, $bds = 3$;
- na “Seção 3”, caso haja falha, a chave (2) é aberta e o disjuntor religado. Desta forma, o nó (4) é removido. Logo, $bds = 4$;
- na “Seção 4”, caso haja falha, a chave (3) é aberta e o disjuntor religado. Desta forma, o nó (5) é removido. Logo, $bds = 5$;
- nos distribuidores, caso haja falha, não há seccionadora que isole. Logo, $bds = 0$;

Na execução do código, ambas matrizes (de alcance \mathbf{R} e `seccionadora_bds`) ajudam nos cálculos. Elas ajudam a determinar como as manobras no sistema, no caso de falha em um determinado componente, impactam ou não na indisponibilidade um ponto de carga. Especificamente,

- se o valor de $\mathbf{R}_{\text{seccionadora,carga}}$ é 1, então o isolamento da falha para manutenção **retira a carga** e **há impacto** na indisponibilidade; Por outro lado,
- se o valor de $\mathbf{R}_{\text{seccionadora,carga}}$ é 0, então o isolamento da falha para manutenção **não retira** a carga e, portanto, **não há impacto** na indisponibilidade.

Por exemplo, deseja-se saber como a falha em cada componente do circuito impacta na indisponibilidade “carga A” para o caso 3. Para uma falha da “Seção 1”, a matriz `seccionadora_bds` informa que, não há seccionadora que isole a falha, portanto, haverá impacto na indisponibilidade. Porém, para falha na “Seção 1”, a matriz `seccionadora_bds` informa que a manobra da seccionadora (1) isola a falha, removendo o nó (3) e impactando apenas com o tempo de manobra. Assim, sabendo que a “carga A” está associada ao nó (6), tem-se $\mathbf{R}_{\text{seccionadora,carga}} = \mathbf{R}_{3,6} = 0$. Analogamente:

- falha na “Seção 1”, não há seccionadora que isole o defeito, há impacto total na indisponibilidade;
- falha na “Seção 2”, nó (3) é removido com a manobra da chave (1) e $\mathbf{R}_{3,6} = 0$, há impacto parcial na indisponibilidade, devido a manobra;
- falha na “Seção 3”, nó (4) é removido com a manobra da chave (2) e $\mathbf{R}_{4,6} = 0$, há impacto parcial na indisponibilidade, devido a manobra;
- falha na “Seção 4”, nó (5) é removido com a manobra da chave (3) e $\mathbf{R}_{5,6} = 0$, há impacto parcial na indisponibilidade, devido a manobra;
- falhas no “Distrib. a”, não há seccionadora que isole o defeito, há impacto na indisponibilidade;
- falhas nos “Distribs. b” até “d”, não há seccionadora que isole o defeito, não há impacto na indisponibilidade;

Dessa forma, percebe-se que falhas em qualquer trecho do alimentador principal impactam nos na indisponibilidade da “carga A”, parcial ou totalmente. Falhas nos distribuidores “Distrib. b”, “Distrib. c” e “Distrib. d” não impactam na indisponibilidade da “carga A”. Isto faz sentido já que no caso de falha nestes componentes, o respectivo fusível atuará.

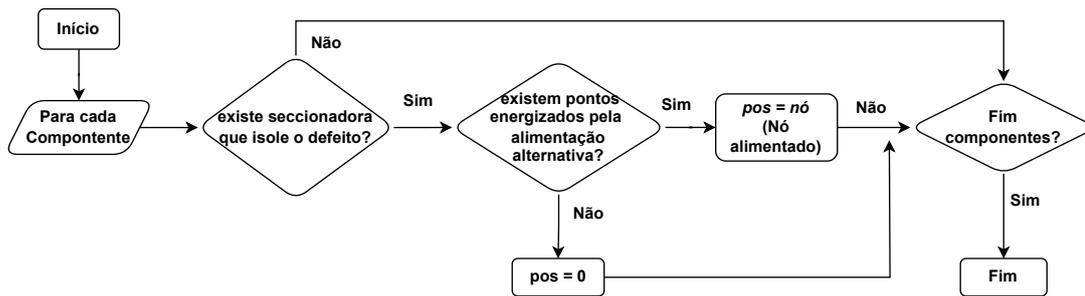
B.1.6 Matriz de alimentação alternativa

A **matriz de alimentação** informa o nó que manterá energizado quando a seccionadora for manobrada para isolamento da falha. Todos os componentes devem estar nesta matriz e o nó associado dependerá do caso analisado. Para os casos 1, 2 e 3 não há acionamento da alimentação alternativa, porém, para o caso 4, a matriz de alimentação terá valores.

O fluxograma de formação de `alimentadora_bda` é mostrado na Figura 10. Para cada componente avalia-se se, no caso de falha, existe chave seccionadora que isole o defeito e se existem pontos que podem ser supridos pela segunda alimentação. Caso afirmativo, o valor será zero. Porém, caso contrário, o valor na célula da matriz recebe o nó que será alimentado na manobra.

A partir dessa análise foi criado a matriz `alimentadora_bda` para cada caso. Da mesma forma que analisado `seccionadora_bds`, cada coluna dessa matriz corresponde a um determinado caso, sendo a primeira coluna para identificação dos componentes, a segunda para o caso 1, a terceira para o caso 2, a quarta para o caso 3 e a quinta coluna corresponde ao 4º caso.

Para esclarecer melhor o funcionamento, suponha que deseja-se construir uma matriz `alimenntadora_bda` para o caso 4, assim os passos serão:

Figura 10 – Fluxograma para criação da matriz `alimentadora_bda`.

Fonte: Elaborado pelo autor (2024).

```

alimentadora_bda = [
  ["Section 1", "", "", "", 3],
  ["Section 2", "", "", "", 4],
  ["Section 3", "", "", "", 5],
  ["Section 4", "", "", "", 0],
  ["Distrib. a", "", "", "", 0],
  ["Distrib. b", "", "", "", 0],
  ["Distrib. c", "", "", "", 0],
  ["Distrib. d", "", "", "", 0],
]
  
```

- na “Seção 1”, caso haja falha realiza-se a manobra de abertura da seccionadora e o nó desconectado é o nó 2, então as cargas B,C e D agora serão alimentados pela alimentação alternativa, que suprirá as demandas de energia alimentando até o nó 3. Logo, $bda = 3$;
- na “Seção 2”, caso haja falha, as chaves (1) e (2) serão abertas e o disjuntor religado restabelecendo A. Desta forma, o nó (3) é removido e pode ser restabelecido a alimentação para as cargas C e D, então o nó que terá demandas de energia supridas é o nó 4. Logo, $bda = 4$;
- na “Seção 3”, caso haja falha, as chaves (2) e (3) serão aberta e o disjuntor religado, restabelecendo A e B. Desta forma, o nó (4) é removido e pode ser restabelecido a alimentação para a carga D, então o nó que terá demandas de energia supridas é o nó 5. Logo, $bds = 5$;
- na “Seção 4”, caso haja falha, a chave (3) e aberta e o disjuntor religado, restabelecendo A B e C, Desta forma, o nó (5) é removido, mas não houve acionamento da alimentação alternativa. Logo, $bda = 0$;
- nos distribuidores, caso haja falha, não há seccionadora que isole pois, o fusível atuará. Logo, $bda = 0$;

Na execução do código, a matriz `alimentadora_bda` ajuda nos cálculos determinando como as manobras no sistema, no caso de falha, impactam ou não na indisponibilidade um ponto de carga. Especificamente,

- se o valor de $\mathbf{R}_{\text{alimentadora,carga}}$ é 1, então houve alimentação alternativa no ponto de carga em análise e **há impacto** na indisponibilidade; Por outro lado,
- se o valor de $\mathbf{R}_{\text{alimentadora,carga}}$ é 0, então não houve alimentação alternativa no ponto de carga em análise, portanto, **não há impacto** na indisponibilidade.

Por exemplo, deseja-se saber como a falha em cada componente do circuito impacta na indisponibilidade “carga A” para o caso 4. Para uma falha da “Seção 1”, a matriz seccionadora_bds informa que há manobra da seccionadora (1) que isole a falha, removendo o nó (2), fazendo com que a alimentadora_bda seja acionada até o nó (3), portanto, haverá impacto apenas com o tempo de manobra e acionamento da alimentação no índice de indisponibilidade. Assim, sabendo que a “carga A” está associada ao nó (6), tem-se $\mathbf{R}_{\text{alimentadora,carga}} = \mathbf{R}_{3,6} = 0$. Analogamente:

- falha na “Seção 1”, nó (2) é removido com a manobra da chave (1) , alimentadora é acionada até o nó (3) e $\mathbf{R}_{3,6} = 0$, há impacto parcial na indisponibilidade, devido a manobra;
- falha na “Seção 2”, nó (3) é removido com a manobra da chave (1) e (2) , alimentadora é acionada até o nó (4) e $\mathbf{R}_{4,6} = 0$, há impacto parcial na indisponibilidade, devido a manobra;
- falha na “Seção 3”, nó (4) é removido com a manobra da chave (2) e (3) , alimentadora é acionada até o nó (5) e $\mathbf{R}_{5,6} = 0$, há impacto parcial na indisponibilidade, devido a manobra;
- falha na “Seção 4”, nó (5) é removido com a manobra da chave (3), mas a alimentadora não é acionada, há impacto parcial na indisponibilidade, devido a manobra;
- falhas no “Distrib. a”, “Distrib. b”, “Distrib. c” e “Distrib. d”, não há seccionadora que isole o defeito, há impacto na indisponibilidade;

Dessa forma, percebe-se que falhas em qualquer trecho do alimentador principal impactam nos na indisponibilidade da “carga A”, parcialmente. Falhas nos distribuidores “Distrib. a”, “Distrib. b”, “Distrib. c” e “Distrib. d” não impactam na indisponibilidade da “carga A”. Isto faz sentido já que no caso de falha nestes componentes, o respectivo fusível atuará.

B.2 Código

```

## Assets
from tabulate import tabulate      # print tables
import numpy as np

def showReliabilitySystemData(reliabilityData, name):
    for k in range( np.shape(reliabilityData)[0] ):
        tab_text.append(
            [(reliabilityData[k][0]),
             "{:.2f}".format(reliabilityData[k][1]),
             "{:.2f}".format(reliabilityData[k][2]),
             "{:.2f}".format(reliabilityData[k][3])
            ] )
    print(name)
    print(tabulate(tab_text))

def showLoadSystemData(loadSystemData, name):
    tab_text = [["Name", "Circuit node", "Consumers", "Load (kW)"]]
    for k in range( np.shape(loadSystemData)[0] ):
        tab_text.append(
            [(loadSystemData[k][0]),
             "{:.0f}".format(loadSystemData[k][1]),
             "{:.0f}".format(loadSystemData[k][2]),
             "{:.0f}".format(loadSystemData[k][3])
            ] )
    print(name)
    print(tabulate(tab_text))

def showLoadSystemIndexes(loadSystemIndexes, name):
    #   0       1       2       3       4   5   6
    #["name", node, consumers, load, lamb, U, r]
    tab_text = [["Name Load", "Circuit node", "N_consumers", "Load (kW)",
                 "      ", "U", "r"]]
    for kL in range( np.shape(loadSystemIndexes)[0] ):
        tab_text.append([
            loadSystemIndexes[kL][0]),
            "{:.0f}".format(loadSystemIndexes[kL][1]),
            "{:.0f}".format(loadSystemIndexes[kL][2]),
            "{:.0f}".format(loadSystemIndexes[kL][3]),
            "{:.3f}".format(loadSystemIndexes[kL][4]),
            "{:.3f}".format(loadSystemIndexes[kL][5]),
            "{:.3f}".format(loadSystemIndexes[kL][6])
        ] )
    print(name)
    print(tabulate(tab_text))

```

```

def calcIndexes(loadSystemIndexes):
    N_vet = np.array([line[2] for line in loadSystemIndexes])
    L_vet = np.array([line[3] for line in loadSystemIndexes])
    y_vet = np.array([line[4] for line in loadSystemIndexes])
    U_vet = np.array([line[5] for line in loadSystemIndexes])
    r_vet = np.array([line[6] for line in loadSystemIndexes])
    indexes = {
        'SAIFI': "{:.3f}".format( sum(y_vet*N_vet)/sum(N_vet) ),
        'SAIDI': "{:.3f}".format(sum(U_vet*N_vet)/sum(N_vet) ),
        #'CAIFI': "Not defined",
        'CAIDI': "{:.3f}".format(sum(U_vet*N_vet)/sum(y_vet*N_vet) ),
        'ASAI' : "{:.6f}".format((sum(N_vet*8760) -
            sum(U_vet*N_vet))/sum(N_vet*8760) ),
        'ASUI' : "{:.6f}".format(1 - (sum(N_vet*8760) -
            sum(U_vet*N_vet))/sum(N_vet*8760) ),
        'ENS'  : "{:.3f}".format(sum(L_vet*U_vet) ),
        'AENS' : "{:.2f}".format(sum(L_vet*U_vet)/sum(N_vet) ),
    }
    return indexes

def showSystemIndexes(loadSystemIndexes, indexes, name):
    tab_text = [["Index", "Value"]]
    indexes_matrix = [[key, value] for key, value in indexes.items()]
    for line in indexes_matrix:
        tab_text.append( line )
    print(name)
    print(tabulate(tab_text))
    return indexes

def getAdjacencyAndRangeMatrix(fromToData):
    n_nodes = np.max(fromToData)
    n_lines = np.shape(fromToData)[0]

    # Adjacency Matrix
    A = np.zeros((n_nodes, n_nodes))
    for line in range( n_lines ):
        node_from = fromToData[line][0] - 1
        node_to = fromToData[line][1] - 1
        A[node_from][node_to] = 1

    # Range Matrix
    A = A.astype(int)
    I = np.eye(n_nodes)
    R_aux = np.linalg.matrix_power(A + I, n_nodes - 1)
    R = (R_aux >= 1).astype(int)
    return A, R

```

```
# Code
# Reading data
import numpy as np
from tabulate import tabulate

reliabilitySystemData = [
    ["Section 1", 2, 0.2, 4],
    ["Section 2", 1, 0.1, 4],
    ["Section 3", 3, 0.3, 4],
    ["Section 4", 2, 0.2, 4],
    ["Distrib. a", 1, 0.2, 2],
    ["Distrib. b", 3, 0.6, 2],
    ["Distrib. c", 2, 0.4, 2],
    ["Distrib. d", 1, 0.2, 2],
]

# ["Name", "Circuit node", "Consumers", "Load (kW)"]
loadSystemData = [
    ["A", 6, 1000, 5000],
    ["B", 7, 800, 4000],
    ["C", 8, 700, 3000],
    ["D", 9, 500, 2000],
]

# Montando Matriz de adjacencia
fromToData = np.array([
    [1, 2],
    [2, 3],
    [3, 4],
    [4, 5],
    [2, 6],
    [3, 7],
    [4, 8],
    [5, 9]
])

dataTable_name = "Par metros de Confiabilidade do sistema - Bilington,
    1991 - P g 234"
showReliabilitySystemData(reliabilitySystemData, dataTable_name)

dataTable_name = "N mero de Consumidores e Carga Conectada ao sistema
    - Bilington, 1991 - P g 234"
showLoadSystemData(loadSystemData, dataTable_name)

## 2. Getting Adjacency and Range Matrices
A, R = getAdjacencyAndRangeMatrix(fromToData)
#print(A)
#print(R)
```

```
# [ "component", "case 1", "case 2", "case 3", "case 4"]

protectorComponent = [
  ["Section 1", 1, 1, 1, 1],
  ["Section 2", 1, 1, 1, 1],
  ["Section 3", 1, 1, 1, 1],
  ["Section 4", 1, 1, 1, 1],
  ["Distrib. a", 1, 6, 6, 6],
  ["Distrib. b", 1, 7, 7, 7],
  ["Distrib. c", 1, 8, 8, 8],
  ["Distrib. d", 1, 9, 9, 9],
]

seccionadora_bds = [
  ["Section 1", "", "", 0, 2],
  ["Section 2", "", "", 3, 3],
  ["Section 3", "", "", 4, 4],
  ["Section 4", "", "", 5, 5],
  ["Distrib. a", "", "", 0, 0],
  ["Distrib. b", "", "", 0, 0],
  ["Distrib. c", "", "", 0, 0],
  ["Distrib. d", "", "", 0, 0],
]

alimentadora_bda = [
  ["Section 1", "", "", "", 3],
  ["Section 2", "", "", "", 4],
  ["Section 3", "", "", "", 5],
  ["Section 4", "", "", "", 0],
  ["Distrib. a", "", "", "", 0],
  ["Distrib. b", "", "", "", 0],
  ["Distrib. c", "", "", "", 0],
  ["Distrib. d", "", "", "", 0],
]

## 3. Algorithm

# Definir o caso a ser avaliado
caseToAccess = int(input("Definir o caso a ser avaliado: "))

# Inicializar variáveis:

# 0 1 2 3 4 5 6
#["name", node, consumers, load, lamb, U, r]
loadSystemIndexes = []
```

```

for k in range( len(loadSystemData) ):
    loadSystemIndexes.append([
        loadSystemData[k][0],
        loadSystemData[k][1],
        loadSystemData[k][2],
        loadSystemData[k][3],
        0, 0, 0])

# Para cada componente/linha:
n_components = len(reliabilitySystemData)
for kLT in range( n_components ):
    # print("Olhando componente: {}".format(reliabilitySystemData[kLT][0]))
    # Buscar a taxa de falha do componente na matriz
    # de dados;
    yLT = reliabilitySystemData[kLT][2]
    # Buscar o tempo médio de reparo do componente na
    # matriz de dados;
    rLT = reliabilitySystemData[kLT][3]
    # Calcular a indisponibilidade =
    # , e atualizar

    ULT = yLT*rLT

    # Buscar protetor na matriz de dados;
    pLT = protectorComponent[kLT][caseToAccess]

    idx_seccionadora = seccionadora_bds[kLT][caseToAccess]
    idx_alimentadora_bda = alimentadora_bda[kLT][caseToAccess]

    # Para cada ponto de carga
    for kLoad in range( len(loadSystemIndexes) ):
        node_load = loadSystemIndexes[kLoad][1]
        # print("Olhando componente: {}".format(node_load))

        bd = R[pLT-1][node_load-1]
        #print("bd = {} para (pLT, Load) ({} , {})".format(db, pLT,
            node_load))

#condition for case 1:
    if caseToAccess == 1:
        lambLoad = loadSystemIndexes[kLoad][4]
        ULoad = loadSystemIndexes[kLoad][5]

        lambLoad = lambLoad + bd*yLT
        ULoad = ULoad + bd*ULT

        loadSystemIndexes[kLoad][4] = lambLoad

```

```
loadSystemIndexes[kLoad][5] = ULoad

#condition for case 2:
if caseToAccess == 2:
    lambLoad = loadSystemIndexes[kLoad][4]
    ULoad     = loadSystemIndexes[kLoad][5]

    lambLoad = lambLoad + bd*yLT
    ULoad     = ULoad     + bd*ULT

    loadSystemIndexes[kLoad][4] = lambLoad
    loadSystemIndexes[kLoad][5] = ULoad

#condition for case 3:
if caseToAccess == 3:

    if idx_seccionadora == 0: #Falha n o implica em abertura da
        seccionadora
        lambLoad = loadSystemIndexes[kLoad][4]
        ULoad     = loadSystemIndexes[kLoad][5]

        lambLoad = lambLoad + bd*yLT
        ULoad     = ULoad     + bd*ULT

        loadSystemIndexes[kLoad][4] = lambLoad
        loadSystemIndexes[kLoad][5] = ULoad

    if idx_seccionadora != 0: # Falha implica em abertura da
        seccionadora
        Usec = yLT * 0.5 # Indisponibilidade da seccionadora
        bd = R[pLT-1][node_load-1]
        bds = R[idx_seccionadora-1][node_load-1]

        lambLoad = loadSystemIndexes[kLoad][4]
        ULoad     = loadSystemIndexes[kLoad][5]

        lambLoad = lambLoad + bd*yLT
        ULoad     = ULoad     + bds*ULT + (bd - bds)*Usec

        loadSystemIndexes[kLoad][4] = lambLoad
        loadSystemIndexes[kLoad][5] = ULoad

#condition for case 4:
if caseToAccess == 4: # se caso 4
    # Se a Falha n o implica em abertura da seccionadora e
    alimenta o alternativa
```

```

if (idx_seccionadora == 0):
    lambLoad = loadSystemIndexes[kLoad][4]
    ULoad    = loadSystemIndexes[kLoad][5]

    lambLoad = lambLoad + bd*yLT
    ULoad    = ULoad    + bd*ULT

    loadSystemIndexes[kLoad][4] = lambLoad
    loadSystemIndexes[kLoad][5] = ULoad

if (idx_seccionadora != 0): # abertura da seccionadora

    bds = R[idx_seccionadora-1][node_load-1]

    if (idx_alimentadora_bda == 0):
        bda = 0
    else:
        bda = R[idx_alimentadora_bda-1][node_load-1]

    Usec = yLT * 0.5 # Indisponibilidade da seccionadora
    Ualim = yLT *0.5
    lambLoad = loadSystemIndexes[kLoad][4]
    ULoad    = loadSystemIndexes[kLoad][5]

    lambLoad = lambLoad + bd*yLT
    ULoad    = ULoad    + (bd - bds)*Usec + (bds - bda)*ULT +
        (bda)*Ualim

    loadSystemIndexes[kLoad][4] = lambLoad
    loadSystemIndexes[kLoad][5] = ULoad

for kLoad in range( len(loadSystemIndexes) ):
    lambLoad = loadSystemIndexes[kLoad][4]
    ULoad    = loadSystemIndexes[kLoad][5]
    loadSystemIndexes[kLoad][6] = ULoad/lambLoad

name = "LoadSystemIndexes p/ Caso {}".format(caseToAccess)
showLoadSystemIndexes(loadSystemIndexes, name)

indexes = calcIndexes(loadSystemIndexes)
name = "SystemIndexes p/ Caso {}".format(caseToAccess)
indexes = showSystemIndexes(loadSystemIndexes, indexes, name)

```

APÊNDICE C – Algoritmo Método de Monte Carlo

C.1 Passo 1: Iniciar todos os componentes no estado de sucesso

No início da simulação, todos os componentes do sistema são inicialmente colocados no estado de sucesso, neste trecho, os dados são carregados, definindo as taxas de falha, dados de carga, conexões entre componentes.

```
dataCase = SystemData(reliabilityData, loadData, fromToData,
protectorData, switchData, alternativeFeederData)
```

C.2 Passo 2: Gerar a duração no estado de sucesso

No código, a classe `MonteCarloSimulation` utiliza a função de geração de duração de tempo `generateStateDurationTime` para gerar números aleatórios e calcular a duração em que cada componente permanecerá no estado de sucesso. As entradas representam:

- `componentState`: Indica o estado atual do componente, sendo 1 para sucesso e 0 para falha.
- `failureRate`: A taxa de falha do componente.
- `repairRate`: A taxa de reparo do componente.

Para o componente está no estado de sucesso `componentState == 1`, a função calcula a duração no estado de sucesso. Se o componente está no estado de falha `componentState == 0`, a função calcula a duração no estado. A nova duração no estado de falha é calculada com base na taxa de reparo `repairRate`.

C.3 Passo 3: Identificar o menor tempo e o respectivo componente

Para cada componente, a simulação identifica aquele que possui o menor tempo no estado de sucesso e que, portanto, será o primeiro a falhar, no código isso pode ser observado na variável `minTime`, onde irá conter o valor desse menor tempo e a variável `idxComp` irá conter o índice do componente que tem o menor tempo, ou seja, o componente que deve ser o próximo a mudar de estado

```
minTime = np.min(timeDurationMatrix)
idxComp = np.argmin(timeDurationMatrix)
```

C.4 Passo 4: Avaliar o estado do componente e o impacto para os consumidores

No código, o impacto da falha é avaliado utilizando as matrizes de alcance, que determinam quais consumidores serão afetados com base na falha de determinado componente. A matriz de alcance é calculada usando a função `_calculateReachMatrix()`, que determina os nós conectados e o impacto da falha através da topologia da rede. Os valores dos BDP, BDS, e BDA são utilizados para determinar a resposta do sistema à falha.

C.4.1 Identificação do estado anterior

Nesse passo é necessário verificar se o componente estava no estado de sucesso antes da falha. Se o componente estava no estado de sucesso `COMPONENT_UP_STATE`, ele agora é identificado e passa para o estado de falha `COMPONENT_DOWN_STATE`. No código isso é feito da seguinte forma:

```
previousState = stateMatrix[idxComp][-1]
if previousState == MonteCarloSimulation.COMPONENT_UP_STATE :
stateMatrix[idxComp][0] = MonteCarloSimulation.COMPONENT_DOWN_STATE
```

C.4.2 Identificação dos nós

Após identificar a falha, o código busca os nós relevantes no sistema, tais como o BDP, BDS e BDA, que estão associados ao componente em falha. Esses nós são essenciais para entender o impacto da falha no sistema. No código foi criado três elementos que desempenham um papel de controle de falhas e na continuidade do fornecimento de energia. O protetor protege o componente, a seccionadora isola a falha, e o alimentador alternativo tenta garantir que a energia continue sendo fornecida por outra fonte na ocorrência de uma falha.

```
pLT = self._systemData.getProtector(idxComp)
sLT = self._systemData.getSwitch(idxComp)
aLT = self._systemData.getAlternativeFeeder(idxComp)
```

C.4.3 Inicialização das Matrizes BDP, BDS e BDA

Em seguida, três matrizes são inicializadas para monitorar o comportamento do sistema em relação às cargas loads conectadas:

- BDP: Representa se a proteção relacionada ao componente está ativa ou não para cada carga.
- BDS: Indica se a seccionadora consegue isolar a falha.
- BDA: Indica se há uma alimentação alternativa disponível para a carga.

No código, essas matrizes são vetores que armazenam a situação de cada carga em relação à falha do componente.

```
BDP = np.zeros((numberOfLoads, 1))
BDS = np.zeros((numberOfLoads, 1))
BDA = np.zeros((numberOfLoads, 1))
```

O código então faz um loop sobre todas as cargas do sistema, verificando a situação de cada uma delas com base na matriz de alcance *reach matrix* do sistema. A matriz de alcance define como os nós estão conectados entre si, permitindo que o código determine quais consumidores são afetados pela falha:

```
for idxLoad in range(numberOfLoads):
    nodeLoad = self._systemData.getNodeLoad(idxLoad)
    R = self.systemData.reachMatrix
```

Aqui, a matriz de alcance R mapeia as conexões entre os componentes do sistema e as cargas, ajudando a identificar os consumidores impactados.

C.4.4 Atualização das Matrizes

Para cada carga, a matriz de proteção BDP é atualizada com base na matriz de alcance, verificando se o protetor do componente falhado está protegendo aquela carga específica.

```
BDP[idxLoad][0] = R[pLT-1][nodeLoad-1]
```

O código também verifica se existe uma seccionadora conectada ao componente. Se houver, a matriz de alcance é usada para determinar se a seccionadora pode isolar a carga da falha. Caso não exista seccionadora, o valor da matriz BDS é ajustado para 1, indicando que não houve seccionamento e a carga será afetada diretamente pela falha.

```
if sLT != 0:
    BDS[idxLoad][0] = R[sLT-1][nodeLoad-1]
else:
    BDS[idxLoad][0] = 1
```

Por fim, verifica-se a existência de um alimentador alternativo disponível para as cargas. Se houver um alimentador alternativo, a matriz de alcance é utilizada para verificar se ele pode atender a carga. Se não houver, o valor de BDA é ajustado para 0.

```
if aLT != 0:
    BDA[idxLoad][0] = R[aLT-1][nodeLoad-1]
else:
    BDA[idxLoad][0] = 0
```

Para consolidação dos Resultados, após varrer todas as cargas e atualizar as matrizes, essas são convertidas em inteiros (0 ou 1) para simplificar os cálculos subsequentes:

```
BDP = BDP.astype(int)
BDS = BDS.astype(int)
BDA = BDA.astype(int)
```

C.4.5 Cálculo do Estado da Carga

O cálculo do estado da carga é calculado com base nessas matrizes, onde o estado da carga será impactado se a proteção e o seccionamento falharem, ou se não houver alimentação alternativa disponível. O código combina as três matrizes para determinar se a carga será impactada pela falha. O resultado final é que `loadState` determina se a carga será ou não impactada, com base nos mecanismos de proteção, seccionamento e alimentação alternativa. Este processo avalia o impacto da falha em cada carga do sistema, permitindo que a simulação calcule os índices de confiabilidade com precisão.

```
loadState = np.abs(BDP - BDS) | BDA
```

C.4.6 Impacto da falha no sistema

Após identificar que um componente falhou, é preciso determinar quantos consumidores foram afetados pela falha e calcular a energia não fornecida devido à interrupção. O próximo passo agora, será multiplicar a matriz BDP pelos consumidores, resultando no total de consumidores impactados pela falha. O `loadState` indica o estado de cada carga se foi impactada ou não pela falha, a subtração (`loadConsumersOnes - loadState`) resulta em 1 para as cargas que foram impactadas pela falha, e `sum((loadConsumersOnes - loadState) * loadConsumers)` multiplica o vetor de cargas impactadas pelo número de consumidores, resultando no total de consumidores realmente afetados pela falha. No final o resultado será o número de consumidores afetados após a falha, considerando o estado de cada carga.

```
impacted_consumers_fault = sum(BDP*loadConsumers)
impacted_consumers = sum((loadConsumersOnes - loadState)*loadConsumers)
shed_load = sum((loadConsumersOnes - loadState)*loadCarga)
```

Para cada carga impactada, `loadCarga` representa a carga elétrica em cada nó de carga e calcula a potência cortada devido à falha, e para cada carga impactada, basta multiplicar a carga elétrica impactada, resultando na potência total não fornecida. No final, a potência cortada será a energia não suprida devido à falha.

C.5 Passo 5: Incrementar as funções-teste dos índices de confiabilidade

Após determinar quais cargas foram impactadas pela falha no passo anterior, o código calcula os índices de confiabilidade do sistema com base na falha identificada. O código utiliza funções para calcular esses índices com base no número de falhas e no tempo de interrupção causado pela falha.

C.5.1 Indicadores de confiabilidade

O algoritmo é apresentado para a estimação dos indicadores SAIFI, SAIDI e ENS, sendo a estimação dos demais indicadores implementada de forma semelhante. O SAIFI é calculado com base no número de consumidores afetados pela falha em relação ao total de consumidores no sistema. No SAIDI representa a duração média das interrupções, onde o código acumula a duração da falha multiplicada pelo número de cargas impactadas e divide pelo número total de consumidores. A ENS calcula a quantidade de energia que deixou de ser fornecida devido à falha. O código multiplica a potência interrompida pela duração da falha e acumula esse valor.

C.5.2 Acumulação de Impactos ao Longo da Simulação

O código acumula o impacto de cada interrupção sobre os consumidores, permitindo que, ao final da simulação, esses índices reflitam a confiabilidade do sistema ao longo de todo o período simulado. O processo contabiliza todos os impactos das falhas em cada componente, garantindo que os efeitos sobre os consumidores sejam considerados de forma precisa.

Esse procedimento é repetido diversas vezes, caracterizando a simulação de Monte Carlo, isso assegura que os índices de confiabilidade representem as condições reais e estocásticas do sistema, fornecendo uma visão global da sua confiabilidade.

C.5.3 Passo 6: Gerar uma nova duração de estado se o período de análise de 8760 horas não acabou, caso contrário, calcular os índices de confiabilidade para o período correspondente

O próximo passo é Gerar e adicionar uma nova duração de estado, que pode ser tanto o tempo que o componente permanecerá no estado de falha quanto o tempo de sucesso após ser reparado. Quando o tempo de falha é calculado, ele é armazenado para ser utilizado no controle da simulação. De forma semelhante, ao retornar ao estado de sucesso, o tempo de permanência no estado de sucesso é recalculado com base na taxa de falha do componente, conforme descrito no Passo 2.

C.5.4 Passo 7: Avaliar o coeficiente de variação para todos os índices de confiabilidade

Ao final da simulação, o coeficiente de variação é calculado para cada índice de confiabilidade, conforme equações (2.24) e (2.25) Com isso, a simulação de Monte Carlo é finalizada após atingir o número de iterações pré-determinado.

C.6 Código

```
# Classes definition
## class SystemData:
from tabulate import tabulate
import numpy as np
import random
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import statistics
from statistics import pvariance
import pprint

class SystemData:

    HOURS_IN_YEAR = 8760

    def __init__(self, reliabilityData, loadData, fromToData,
                 protectorData, switchData, alternativeFeederData):
        self._reliabilityData = reliabilityData
        self._loadData = loadData
        self._fromToData = fromToData
        self._protectorData = protectorData
        self._switchData = switchData
        self._alternativeFeederData = alternativeFeederData
```

```
self._calculateAdjacencyMatrix()
self._calculateReachMatrix()

@property
def switchData(self):
    return self._switchData

@property
def alternativeFeederData(self):
    return self._alternativeFeederData

@property
def reliabilityData(self):
    return self._reliabilityData

@property
def loadData(self):
    return self._loadData

@property
def fromToData(self):
    return self._fromToData

@property
def adjacencyMatrix(self):
    return self._A.copy()

@property
def reachMatrix(self):
    return self._R.copy()

@property
def protectorData(self):
    return self._protectorData

def getNumberComponents(self):
    return len(self._reliabilityData)

def getNodeLoad(self, indexLoad):
    return self._loadData[indexLoad][1]

def getConsumersLoad(self, indexLoad):
    return self._loadData[indexLoad][2]

def getCargaLoad(self, indexLoad):
    return self._loadData[indexLoad][3]
```

```

def getNumberOfLoads(self):
    return len(self._loadData )

def getFailureRateHours(self, indexComponent):
    return
        self._reliabilityData[indexComponent][2]/SystemData.HOURS_IN_YEAR

def getMeanTimeRepairHours(self, indexComponent):
    return self._reliabilityData[indexComponent][3]

def getProtector(self, indexComponent):
    return self._protectorData[indexComponent][1]

def getSwitch(self, indexComponent):
    return self._switchData[indexComponent][1]

def getAlternativeFeeder(self, indexComponent):
    return self._alternativeFeederData[indexComponent][1]

def _calculateAdjacencyMatrix(self):
    n_nodes = np.max(self._fromToData)
    n_lines = np.shape(self._fromToData)[0]
    A = np.zeros((n_nodes, n_nodes))
    for line in range( n_lines ):
        node_from = self._fromToData[line][0] - 1
        node_to = self._fromToData[line][1] - 1
        A[node_from][node_to] = 1
    self._A = A

def _calculateReachMatrix(self):
    n_nodes = np.max(fromToData)
    # Range Matrix
    A = self._A.astype(int)
    I = np.eye(n_nodes)
    R_aux = np.linalg.matrix_power(A + I, n_nodes - 1)
    R = (R_aux >= 1).astype(int)
    self._R = R

def showReliabilitySystemData(self, title):
    tab_text = [["Component", "Length (km)", "Failure rate
        (failure/year)", "Mean Time to repair r (h)"]]
    for k in range( np.shape(self._reliabilityData)[0] ):
        tab_text.append(
            [(self._reliabilityData[k][0]),
             "{:.2f}".format(self._reliabilityData[k][1]),
             "{:.2f}".format(self._reliabilityData[k][2]),
             "{:.2f}".format(self._reliabilityData[k][3])
            ]
        )

```

```

        ] )
    print(title)
    print(tabulate(tab_text))

def showLoadSystemData(self, title):
    tab_text = [["Name", "Circuit node", "Consumers", "Load (kW)"]]
    for k in range( np.shape(self._loadData)[0] ):
        tab_text.append(
            [(self._loadData[k][0]),
             "{:.0f}".format(self._loadData[k][1]),
             "{:.0f}".format(self._loadData[k][2]),
             "{:.0f}".format(self._loadData[k][3])
            ] )
    print(title)
    print(tabulate(tab_text))

def getIndexes(output, isPrint=False):

    SAIFI = output['Num_int_cons'] / output['NumConsumersTotal']
    SAIDI = output['NumHoursInterrup'] / output['NumConsumersTotal']
    if SAIFI == 0:
        CAIDI = 0
    else:
        CAIDI = SAIDI / SAIFI
    ASAI = 1 - (SAIDI / 8760)
    ASUI = 1 - ASAI
    ENS = output['Eint_cons']
    AENS = output['Eint_cons'] / output['NumConsumersTotal']

    # Dicionário de índices
    indexes = {
        'SAIFI': SAIFI,
        'SAIDI': SAIDI,
        'CAIDI': CAIDI,
        'ASAI': ASAI,
        'ASUI': ASUI,
        'ENS': ENS,
        'AENS': AENS
    }

    if isPrint:

        print(f'SAIFI = {SAIFI}')
        print(f'SAIDI = {SAIDI}')
        print(f'CAIDI = {CAIDI}')

```

```
    print(f'ASAI = {ASAI}')
    print(f'ASUI = {ASUI}')
    print(f'ENS = {ENS}')
    print(f'AENS = {AENS}')

    return indexes

## class MonteCarloSimulation:
class MonteCarloSimulation:

    COMPONENT_UP_STATE = 1
    COMPONENT_DOWN_STATE = 0

    def __init__(self, systemData, totalSimulationTime):
        self._systemData = systemData
        self._totalSimulationTime = totalSimulationTime

    @property
    def systemData(self):
        return self._systemData

    @property
    def totalSimulationTime(self):
        return self._totalSimulationTime

    def printIndexes(self, output):
        tab_text = [{"ndice", "valor"}]
        tab_text.append( ['SAIFI',
            "{:.2f}".format(output['indexes']['SAIFI'])] )
        tab_text.append( ['SAIDI',
            "{:.2f}".format(output['indexes']['SAIDI'])] )
        tab_text.append( ['CAIDI',
            "{:.2f}".format(output['indexes']['CAIDI'])] )
        tab_text.append( ['ASAI',
            "{:.6f}".format(output['indexes']['ASAI'])] )
        tab_text.append( ['ASUI',
            "{:.6f}".format(output['indexes']['ASUI'])] )
        tab_text.append( ['ENS',
            "{:.0f}".format(output['indexes']['ENS'])] )
        tab_text.append( ['AENS',
            "{:.2f}".format(output['indexes']['AENS'])] )
        print(tabulate(tab_text))

    def execute(self):
        # Passo 1: Iniciar todos os componentes no estado de sucesso
        simulationTime = 0
```

```

numberExperiments      = [0]
numberComponents      = self._systemData.getNumberComponents()
stateMatriz           = np.ones((numberComponents, 1))
timeDurationMatrix    = np.zeros((numberComponents, 1))
simulationTimeRecord  = [0]
stateMatrizRecord     = stateMatriz.copy()

# Inicializando variáveis
numberOfLoads = self._systemData.getNumberOfLoads()
NintConsTotal = 0
Num_int_cons   = 0
NumHoursInterrup = 0
Eint_cons     = 0
impacted_consumers = 0
loadConsumersOnes = np.ones((numberOfLoads, 1))
loadConsumers     = np.zeros((numberOfLoads, 1))
loadCarga         = np.zeros((numberOfLoads, 1))

for idxLoad in range( numberOfLoads ):
    loadConsumers[idxLoad] = self._systemData.getConsumersLoad(
        idxLoad )
    loadCarga[idxLoad] = self._systemData.getCargaLoad( idxLoad )

impacted_consumers = 0
impacted_consumers_fault = 0
shed_load = 0
while simulationTime < self._totalSimulationTime:
    numberExperiments.append( numberExperiments[-1] + 1)

    # Passo 2: Gerar uma dura o no estado de sucesso ou falha para
    # cada um dos componentes
    for indexComponent in range(numberComponents):
        componentState = stateMatriz[indexComponent]
        failureRate    =
            self._systemData.getFailureRateHours(indexComponent)
        repairRate     = 1 /
            self._systemData.getMeanTimeRepairHours(indexComponent)
        stateDurationTime =
            self.generateStateDurationTime(componentState, failureRate,
            repairRate)
        timeDurationMatrix[indexComponent][0] = stateDurationTime

    # Passo 3: Identificar componente com menor tempo
    minTime = np.min(timeDurationMatrix)
    idxComp = np.argmin(timeDurationMatrix)

```

```

# Passo 4: Se o componente foi para estado de falha, avaliar o
    impacto nos consumidores
previousState = stateMatriz[idxComp][-1]
if previousState == MonteCarloSimulation.COMPONENT_UP_STATE:
    # Falha
    stateMatriz[idxComp][0] =
        MonteCarloSimulation.COMPONENT_DOWN_STATE
    # N s: protetor, seccionadora e alternativa
    pLT = self._systemData.getProtector(idxComp)
    sLT = self._systemData.getSwitch(idxComp)
    aLT = self._systemData.getAlternativeFeeder(idxComp)
    # Varrendo os n s de carga (A, B, C e D)
    BDP = np.zeros((numberOfLoads, 1))
    BDS = np.zeros((numberOfLoads, 1))
    BDA = np.zeros((numberOfLoads, 1))
    for idxLoad in range( numberOfLoads ):
        nodeLoad = self._systemData.getNodeLoad( idxLoad )
        R = self.systemData.reachMatrix
        BDP[idxLoad][0] = R[pLT-1][nodeLoad-1]
        if sLT != 0: # implica em seccionadora
            BDS[idxLoad][0] = R[sLT-1][nodeLoad-1]
        else:
            BDS[idxLoad][0] = 1
        if aLT != 0: # implica em alternativa e seccionadora
            BDA[idxLoad][0] = R[aLT-1][nodeLoad-1]
        else:
            BDA[idxLoad][0] = 0
    #endFor
    BDP = BDP.astype(int)
    BDS = BDS.astype(int)
    BDA = BDA.astype(int)
    loadState = np.abs(BDP - BDS) | BDA

    impacted_consumers_fault = sum( BDP * loadConsumers)
    impacted_consumers = sum( (loadConsumersOnes - loadState ) *
        loadConsumers)
    shed_load = sum((loadConsumersOnes - loadState ) * loadCarga) #
        pot ncia cortada
else:
    # componente reparado
    stateMatriz[idxComp][0] =
        MonteCarloSimulation.COMPONENT_UP_STATE

# Passo 5: incrementar as fun oes teste dos ndices de
    confiabilidade

```

```

    NumHoursInterrup = NumHoursInterrup + minTime*impacted_consumers
    Num_int_cons = Num_int_cons + impacted_consumers_fault
    Eint_cons      = Eint_cons      + minTime*shed_load

# Passo 6: Atualizar o tempo
simulationTime = simulationTime + minTime
simulationTimeRecord.append(simulationTime)
stateMatrizRecord = np.append(stateMatrizRecord, stateMatriz,
                               axis=1)

# Outputs
output = {
    'simulationTimeRecord': simulationTimeRecord,
    'stateMatrizRecord': stateMatrizRecord,
    'Eint_cons': float(Eint_cons[0]) if isinstance(Eint_cons,
                                                    np.ndarray) else float(Eint_cons),
    'NumHoursInterrup': float(NumHoursInterrup[0]) if
        isinstance(NumHoursInterrup, np.ndarray) else
        float(NumHoursInterrup),
    'NumConsumersTotal': float(sum(loadConsumers)[0]) if
        isinstance(sum(loadConsumers), np.ndarray) else
        float(sum(loadConsumers)),
    'Num_int_cons': float(Num_int_cons[0]) if
        isinstance(Num_int_cons, np.ndarray) else float(Num_int_cons)
}

indexes = MonteCarloSimulation.getIndexes(output)
output['indexes'] = indexes

return output

def generateStateDurationTime(self, componentState, failureRate,
                              repairRate):
    U = random.random()
    durationTime = 0
    if componentState == 1:
        durationTime = - (1/failureRate)*np.log( U );
    if componentState == 0:
        durationTime = - (1/repairRate)*np.log( U );
    return durationTime;

def getIndexes(output):
    SAIFI = output['Num_int_cons'] / output['NumConsumersTotal']
    SAIDI = output['NumHoursInterrup'] / output['NumConsumersTotal']

```

```
if SAIFI > 0:
    CAIDI = SAIDI / SAIFI
else:
    CAIDI = 0
ASAI = 1 - (SAIDI / 8760)
ASUI = 1 - ASAI
ENS = output['Eint_cons']
AENS = output['Eint_cons'] / output['NumConsumersTotal']

indexes = {
    'SAIFI': SAIFI,
    'SAIDI': SAIDI,
    'CAIDI': CAIDI,
    'ASAI': ASAI,
    'ASUI': ASUI,
    'ENS': ENS,
    'AENS': AENS
}
return indexes

## class batchMonteCarlo:
class BatchMonteCarlo:

    def __init__(self, monteCarloSimulation, NumSimulations):
        self._monteCarloSimulation = monteCarloSimulation
        self._NumSimulations = NumSimulations

    def execute(self, mainCase=0):

        SAIFI = []
        SAIDI = []
        CAIDI = []
        ASAI = []
        ASUI = []
        ENS = []
        AENS = []

        for k in range(self._NumSimulations):
            output = self._monteCarloSimulation.execute()
            indexes = output['indexes']

            SAIFI.append( indexes['SAIFI'] )
            SAIDI.append( indexes['SAIDI'] )
            CAIDI.append( indexes['CAIDI'] )
            ASAI.append( indexes['ASAI'] )
            ASUI.append( indexes['ASUI'] )
            ENS.append( indexes['ENS'] )
```

```
AENS.append( indexes['AENS'] )

# E[G]      uma média do valor esperado do índice
E_SAIFI = sum(SAIFI) / self._NumSimulations
E_SAIDI = sum(SAIDI) / self._NumSimulations
E_CAIDI = sum(CAIDI) / self._NumSimulations
E_ASAI  = sum(ASAI)  / self._NumSimulations
E_ASUI  = sum(ASUI)  / self._NumSimulations
E_ENS   = sum(ENS)   / self._NumSimulations
E_AENS  = sum(AENS)  / self._NumSimulations

# Cálculo das variâncias usando pvariance (variância populacional)
Var_SAIFI = pvariance(SAIFI)
Var_SAIDI = pvariance(SAIDI)
Var_CAIDI = pvariance(CAIDI)
Var_ASAI  = pvariance(ASAI)
Var_ASUI  = pvariance(ASUI)
Var_ENS   = pvariance(ENS)
Var_AENS  = pvariance(AENS)

# Cálculo do erro relativo (beta)
Beta_SAIFI = (np.sqrt(Var_SAIFI / self._NumSimulations) / E_SAIFI)
Beta_SAIDI = (np.sqrt(Var_SAIDI / self._NumSimulations) / E_SAIDI)
Beta_CAIDI = (np.sqrt(Var_CAIDI / self._NumSimulations) / E_CAIDI)
Beta_ASAI  = (np.sqrt(Var_ASAI / self._NumSimulations) / E_ASAI)
Beta_ASUI  = (np.sqrt(Var_ASUI / self._NumSimulations) / E_ASUI)
Beta_ENS   = (np.sqrt(Var_ENS / self._NumSimulations) / E_ENS)
Beta_AENS  = (np.sqrt(Var_AENS / self._NumSimulations) / E_AENS)

case1Ref = [2.20, 6.00, 2.73, 0.999315, 0.000685, 84000, 28,0 ]
case2Ref = [1.15, 3.91, 3.39, 0.999554, 0.000446, 54800, 18.3 ]
case3Ref = [1.15, 2.58, 2.23, 0.999706, 0.000294, 35200, 11.7 ]
case4Ref = [1.15, 1.80, 1.56, 0.999795, 0.000205, 25100, 8.4 ]

if mainCase == 1:
    analitico = case1Ref
elif mainCase == 2:
    analitico = case2Ref
elif mainCase == 3:
    analitico = case3Ref
elif mainCase == 4:
    analitico = case4Ref
```

```

else:
    analitico = np.array(case1Ref)*0

tab_text = [ [" ndice ", "Anal tico", "E(G)", "Und", "Beta" ] ]
tab_text.append( [ 'SAIFI', "{:.4f}".format(analitico[0]),
    "{:.4f}".format(E_SAIFI), "int/ano",
    "{:.4f}".format(Beta_SAIFI) ] )
tab_text.append( [ 'SAIDI', "{:.4f}".format(analitico[1]),
    "{:.4f}".format(E_SAIDI), "h/ano",
    "{:.4f}".format(Beta_SAIDI) ] )
tab_text.append( [ 'CAIDI', "{:.4f}".format(analitico[2]),
    "{:.4f}".format(E_CAIDI), "h/int",
    "{:.4f}".format(Beta_CAIDI) ] )
tab_text.append( [ 'ASAI', "{:.4f}".format(analitico[3]),
    "{:.4f}".format(E_ASAI), "%", "{:.4f}".format(Beta_ASAI)
    ] )
tab_text.append( [ 'ASUI', "{:.6f}".format(analitico[4]),
    "{:.6f}".format(E_ASUI), "%", "{:.4f}".format(Beta_ASUI)
    ] )
tab_text.append( [ 'ENS', "{:.0f}".format(analitico[5]),
    "{:.0f}".format(E_ENS), "MWh/ano",
    "{:.4f}".format(Beta_ENS) ] )
tab_text.append( [ 'AENS', "{:.4f}".format(analitico[6]),
    "{:.4f}".format(E_AENS), "MWh/cons.ano",
    "{:.4f}".format(Beta_AENS) ] )

print(tabulate(tab_text))

# Gerar o histograma SAIFI
plt.hist(SAIFI, bins=5, edgecolor='black')
plt.title('Histograma dos Valores de SAIFI')
plt.xlabel('Interrup es por ano [int/ano]')
plt.ylabel('Frequ ncia')
plt.grid(True)
plt.show()

# Gerar o histograma SAIDI
plt.hist(SAIDI, bins=5, edgecolor='black')
plt.title('Histograma dos Valores de SAIDI')
plt.xlabel('Interrup es em horas ano [h/ano]')
plt.ylabel('Frequ ncia')
plt.grid(True)
plt.show()

# Gerar o histograma ENS
plt.hist(ENS, bins=5, edgecolor='black')
plt.title('Histograma dos Valores de ENS')

```

```
plt.xlabel('Energia n o fornecida por ano [MWh/ano]')
plt.ylabel('Frequ ncia')
plt.grid(True)
plt.show()

# Application
## Dados
# ["Component", "Length (km)", "Failure rate (failure/year)",
  "Mean Time to repair r (h)"]
reliabilityData = [
  ["Section 1", 2, 0.2, 4],
  ["Section 2", 1, 0.1, 4],
  ["Section 3", 3, 0.3, 4],
  ["Section 4", 2, 0.2, 4],
  ["Distrib. a", 1, 0.2, 2],
  ["Distrib. b", 3, 0.6, 2],
  ["Distrib. c", 2, 0.4, 2],
  ["Distrib. d", 1, 0.2, 2],
]

# ["Name", "Circuit node", "Consumers", "Load (kW)"]
loadData = [
  ["A", 6, 1000, 5000],
  ["B", 7, 800, 4000],
  ["C", 8, 700, 3000],
  ["D", 9, 500, 2000],
]

# Montagem da matriz
fromToData = np.array([
  [1, 2],
  [2, 3],
  [3, 4],
  [4, 5],
  [2, 6],
  [3, 7],
  [4, 8],
  [5, 9]
])

protectorData1 = [
  ["Section 1", 1],
  ["Section 2", 1],
  ["Section 3", 1],
  ["Section 4", 1],
]
```

```
    ["Distrib. a", 1],  
    ["Distrib. b", 1],  
    ["Distrib. c", 1],  
    ["Distrib. d", 1],  
]
```

```
protectorData2 = [  
    ["Section 1", 1],  
    ["Section 2", 1],  
    ["Section 3", 1],  
    ["Section 4", 1],  
    ["Distrib. a", 6],  
    ["Distrib. b", 7],  
    ["Distrib. c", 8],  
    ["Distrib. d", 9],  
]
```

```
protectorData3 = [  
    ["Section 1", 1],  
    ["Section 2", 1],  
    ["Section 3", 1],  
    ["Section 4", 1],  
    ["Distrib. a", 6],  
    ["Distrib. b", 7],  
    ["Distrib. c", 8],  
    ["Distrib. d", 9],  
]
```

```
protectorData4 = [  
    ["Section 1", 1],  
    ["Section 2", 1],  
    ["Section 3", 1],  
    ["Section 4", 1],  
    ["Distrib. a", 6],  
    ["Distrib. b", 7],  
    ["Distrib. c", 8],  
    ["Distrib. d", 9],  
]
```

```
switchData1 = [  
    ["Section 1", 0],  
    ["Section 2", 0],  
    ["Section 3", 0],  
    ["Section 4", 0],  
    ["Distrib. a", 0],  
    ["Distrib. b", 0],  
    ["Distrib. c", 0],  
]
```

```
    ["Distrib. d", 0],
]

switchData2 = [
    ["Section 1", 0],
    ["Section 2", 0],
    ["Section 3", 0],
    ["Section 4", 0],
    ["Distrib. a", 0],
    ["Distrib. b", 0],
    ["Distrib. c", 0],
    ["Distrib. d", 0],
]

switchData3 = [
    ["Section 1", 0],
    ["Section 2", 3],
    ["Section 3", 4],
    ["Section 4", 5],
    ["Distrib. a", 0],
    ["Distrib. b", 0],
    ["Distrib. c", 0],
    ["Distrib. d", 0],
]

switchData4 = [
    ["Section 1", 2],
    ["Section 2", 3],
    ["Section 3", 4],
    ["Section 4", 5],
    ["Distrib. a", 0],
    ["Distrib. b", 0],
    ["Distrib. c", 0],
    ["Distrib. d", 0],
]

alternativeFeederData1 = [
    ["Section 1", 0],
    ["Section 2", 0],
    ["Section 3", 0],
    ["Section 4", 0],
    ["Distrib. a", 0],
    ["Distrib. b", 0],
    ["Distrib. c", 0],
    ["Distrib. d", 0],
]

alternativeFeederData2 = [
```

```
["Section 1", 0],
["Section 2", 0],
["Section 3", 0],
["Section 4", 0],
["Distrib. a", 0],
["Distrib. b", 0],
["Distrib. c", 0],
["Distrib. d", 0],
]
alternativeFeederData3 = [
  ["Section 1", 0],
  ["Section 2", 0],
  ["Section 3", 0],
  ["Section 4", 0],
  ["Distrib. a", 0],
  ["Distrib. b", 0],
  ["Distrib. c", 0],
  ["Distrib. d", 0],
]

alternativeFeederData4 = [
  ["Section 1", 3],
  ["Section 2", 4],
  ["Section 3", 5],
  ["Section 4", 0],
  ["Distrib. a", 0],
  ["Distrib. b", 0],
  ["Distrib. c", 0],
  ["Distrib. d", 0],
]

dataCase1      = SystemData(reliabilityData, loadData, fromToData,
  protectorData1, switchData1, alternativeFeederData1)
monteCarlo1    = MonteCarloSimulation(dataCase1, 8760)
#output1 = monteCarlo1.execute()
#monteCarlo1.printIndexes(output1)
batchMonteCarlo1 = BatchMonteCarlo(monteCarlo1, 2000)
print('Caso 1')
batchMonteCarlo1.execute(1)

dataCase2      = SystemData(reliabilityData, loadData, fromToData,
  protectorData2, switchData2, alternativeFeederData2)
monteCarlo2    = MonteCarloSimulation(dataCase2, 8760)
#output2      = monteCarlo2.execute()
#monteCarlo2.printIndexes(output2)
batchMonteCarlo2 = BatchMonteCarlo(monteCarlo2, 2000)
print('Caso 2')
```

```
batchMonteCarlo2.execute(2)

dataCase3      = SystemData(reliabilityData, loadData, fromToData,
    protectorData3, switchData3, alternativeFeederData3)
monteCarlo3    = MonteCarloSimulation(dataCase3, 8760)
#output3      = monteCarlo3.execute()
#monteCarlo3.printIndexes(output3)
batchMonteCarlo3 = BatchMonteCarlo(monteCarlo3, 2000)
print('Caso 3')
batchMonteCarlo3.execute(3)

dataCase4      = SystemData(reliabilityData, loadData, fromToData,
    protectorData4, switchData4, alternativeFeederData4)
monteCarlo4    = MonteCarloSimulation(dataCase4, 8760)
#output4      = monteCarlo4.execute()
#monteCarlo4.printIndexes(output4)
batchMonteCarlo4 = BatchMonteCarlo(monteCarlo4, 2000)
print('Caso 4')
batchMonteCarlo4.execute(4)
```



ANEXO VIII - TERMO DE RESPONSABILIDADE

O texto do trabalho de conclusão de curso intitulado “Análise da Confiabilidade na Rede de Distribuição de Energia Elétrica” é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico ou qualquer outro material pertencente a terceiros sem a devida citação ou consentimento dos referidos autores.

João Monlevade, 18 de Outubro de 2024 .

Nome completo do(a) aluno(a)