

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

Leyla Kräulich

**ANÁLISE ESTOCÁSTICA DE TRANSITÓRIOS EM
SUBESTAÇÕES UTILIZANDO O MÉTODO DE MONTE
CARLO INTEGRADO AO ATP**

**Santa Maria, RS
2016**

Leyla Kräulich

**ANÁLISE ESTOCÁSTICA DE TRANSITÓRIOS EM
SUBESTAÇÕES UTILIZANDO O MÉTODO DE MONTE
CARLO INTEGRADO AO ATP**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Engenharia Elétrica**.

Orientador: Ghendy Cardoso Junior
Coorientador: Lenois Mariotto

Santa Maria, RS
2016

Kräulich, Leyla

Análise Estocástica de Transitórios em Subestações
Utilizando o Método de Monte Carlo Integrado ao ATP /
Leyla Kräulich. - Santa Maria, RS, 2016.

150 p.; 30 cm

Orientador: Ghendy Cardoso Junior

Coorientador: Lenois Mariotto

Dissertação (Mestrado) - Universidade Federal
de Santa Maria, Centro de Tecnologia, Programa de
Pós-Graduação em Engenharia Elétrica, RS, 2016.

1. Transitórios eletromagnéticos. 2. Subestações. 3.
ATP-EMTP. 4. Método de Monte Carlo. I. Cardoso Junior,
Ghendy II. Mariotto, Lenois III. Título

Leyla Kräulich

**ANÁLISE ESTOCÁSTICA DE TRANSITÓRIOS EM
SUBESTAÇÕES UTILIZANDO O MÉTODO DE MONTE
CARLO INTEGRADO AO ATP**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Mestre em Engenharia Elétrica**.

Aprovado em 26 de abril de 2016:

Ghendy Cardoso Junior (UFSM)
(Presidente / Orientador)

Lenois Mariotto (UFSM)
(Coorientador)

Arnaldo Gakiya Kanashiro (USP)

Olinto César Bassi de Araújo (UFSM)

Santa Maria, RS
2016

RESUMO

ANÁLISE ESTOCÁSTICA DE TRANSITÓRIOS EM SUBESTAÇÕES UTILIZANDO O MÉTODO DE MONTE CARLO INTEGRADO AO ATP

AUTOR: LEYLA KRÄULICH
ORIENTADOR: GHENDY CARDOSO JUNIOR
COORIENTADOR: LENOIS MARIOTTO

Este trabalho apresenta o desenvolvimento de uma metodologia que utiliza o recurso *foreign model* oferecido pelo *ElectroMagnetic Transient Program* (ATP-EMTP), que visa realizar simulações estocásticas de transitórios eletromagnéticos, com enfoque para aplicações em subestações de potência. Devido estas simulações contemplarem processos estocásticos, emprega-se o método de Monte Carlo para reproduzir aleatoriedades e incertezas inerentes aos eventos precursores de transitórios eletromagnéticos. Mais especificamente, este método é aplicado para modelar de estudos dos transitórios mais significativos e frequentes em subestações. Estes eventos compreendem: ocorrência e eliminação de curtos-circuitos, energização de transformadores, manobras de abertura e fechamento de disjuntores e descargas atmosféricas. A metodologia sugerida, permite que estes recursos sejam acessados através do pré-processador gráfico ATPDraw, de modo que não seja necessário a utilização de ferramentas externas para a realização destas simulações. Enfim, para demonstrar suas potencialidades, abordam-se exemplos aplicados em um sistema elétrico real. A partir dos estudos realizados, espera-se fornecer subsídios para simulações de eventos estocásticos de transitórios eletromagnéticos, agregando novos recursos ao ATP-EMTP. O presente trabalho é parte integrante de um projeto de Pesquisa e Desenvolvimento (P&D), desenvolvido pelo Laboratório de Análise e Proteção de Sistemas Elétricos do Centro de Estudos em Energia e Sistemas de Potência (LAPES-CEESP) da UFSM, em parceria com a Companhia Estadual de Geração e Transmissão de Energia Elétrica (CEEE-GT) do Rio Grande do Sul.

Palavras-chave: Transitórios eletromagnéticos. Subestações. ATP-EMPT. Método de Monte Carlo.

ABSTRACT

DEVELOPMENT OF A GRAPHICS PRE-PROCESSOR SOFTWARE FOR TRANSIENT ANALYSIS IN SUBSTATIONS

AUTHOR: LEYLA KRÄULICH
ADVISOR: GHENDY CARDOSO JUNIOR
COADVISOR: LENOIS MARIOTTO

This work presents the development of a methodology that uses the resource “*foreign model*”, offered by the Electromagnetic Transient Program (ATP-EMTP), in order to realize electromagnetic transient stochastic simulations with emphasis in power substation applications. Due to their stochastic nature, the simulations are performed by employing the Monte Carlo Method to reproduce the randomness and uncertainties that are inherent to the precursory events of some electromagnetic transients. The method is applied in the simulation in order to model the type of study to be analyzed, corresponding to the most frequent and significant transients that are found in substations. These events comprehend: occurrence and clearance of faults, transformer inrush, circuit-breaker switchings and lightning strokes. The suggested methodology allows the use of these resources through the graphical pre-processor software ATPDraw, and it does not need the use of any other external computational tool. Hence, in order to demonstrate the methodology potentialities, this work also presents some examples of applications through ATP-EMTP simulations, employed in a real power system. Throughout this study, it seeks to aid in simulations involving stochastic events in electromagnetic transients by adding new resources to the ATP-EMTP. In addition, this work is a part of a research and development project in a partnership between the Laboratório de Análise e Proteção de Sistemas Elétricos of the Centro de Estudos em Energia e Sistemas de Potência (LAPES-CEESP), located at UFSM, and the Companhia Estadual de Geração e Transmissão de Energia Elétrica (CEEE-GT) of the Rio Grande do Sul state.

Keywords: Electromagnetic transients. Substations. ATP-EMPT. Monte Carlo Method.

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Justificativa e motivação	14
1.2	Objetivos	15
1.2.1	Objetivo geral	15
1.2.2	Objetivos específicos	15
1.3	Estrutura do trabalho	15
2	TRANSITÓRIOS ELETROMAGNÉTICOS EM SUBESTAÇÕES . . .	17
2.1	Considerações Gerais	17
2.2	Curto-Circuitos	18
2.3	Energização de Transformadores	19
2.4	Manobras de Disjuntores	21
2.5	Descargas Atmosféricas	25
2.5.1	Fenômenos físico das descargas atmosféricas	26
2.5.2	Parâmetros típicos da onda de descarga	27
2.5.3	Representação das formas de onda da corrente de descarga	30
3	MÉTODO DE MONTE CARLO	33
3.1	Considerações Gerais	33
3.2	Processos estocásticos	34
3.2.1	Distribuições de probabilidade para variáveis aleatórias contínuas . . .	36
3.3	Estrutura do Método de Monte Carlo	37
3.4	Gerador de Números Aleatórios	38
3.5	Técnicas de Amostragem	42
3.5.1	Método da Transformação Inversa	42
3.5.2	Método de Box-Müller	43
3.6	Distribuições de Probabilidade Contínuas	43
3.6.1	Distribuição Uniforme	43
3.6.2	Distribuição Normal	46
3.6.3	Distribuição Log-Normal	48
4	O ALTERNATIVE TRANSIENTS PROGRAM (ATP-EMTP)	51
4.1	Breve histórico	51
4.2	Potencialidades do ATP-EMTP	53
4.3	Pré-processadores gráficos para o ATP-EMTP	54
4.4	Alternativas para Incorporação de Novos Modelos	56
4.5	Criação do <i>Foreign Models</i> no ATP-EMTP	59
4.5.1	Utilização de <i>foreign models</i>	63
5	METODOLOGIA	65
5.1	Considerações gerais	65
5.2	Concepção da metodologia	67
5.3	Simulação estocástica de curtos-circuitos	68
5.4	Simulação estocástica de energizações de transformador	70
5.5	Simulação estocástica de manobras	70

5.6	Simulação estocástica de descargas atmosféricas	71
5.7	Implementação da metodologia	71
6	APLICAÇÕES DA METODOLOGIA PROPOSTA	83
6.1	Sistema elétrico abordado	83
6.2	Estudo de Curto-circuito	85
6.2.1	Simulação de curto-circuito trifásico com abertura tripolar	86
6.2.2	Simulação de curto-circuito monofásico com abertura tripolar	91
6.2.3	Simulação de curto-circuito monofásico com abertura monopolar	96
6.3	Estudo de Energização de Transformadores	99
6.3.1	Simulação de Energização Individual	100
6.3.2	Simulação de Energização Solidária	102
6.4	Estudo de Manobras de Disjuntor	105
6.4.1	Manobra de Linha de Transmissão	106
6.4.2	Manobra de Carga	109
6.5	Estudo de Descargas Atmosféricas	111
6.5.1	Descargas Atmosféricas no Setor 230 kV	112
6.5.2	Descargas Atmosféricas no Setor 69 kV	114
7	CONCLUSÃO	117
7.1	Propostas de Continuidade	118
7.2	Publicações	119
	REFERÊNCIAS	121
	APÊNDICE A – CÓDIGO FONTE EM LINGUAGEM C++	125
	APÊNDICE B – SUBROTINA FGNMOD.F	143
	APÊNDICE C – MAKEFILE	149

1 INTRODUÇÃO

Os Sistemas Elétricos de Potências (SEPs) operam em regime permanente na maior parte do tempo, no entanto, estão sujeitos a uma série de eventos de origem interna e externa capazes de provocar alterações súbitas em seu funcionamento. De acordo com a intensidade destes eventos, podem acarretar uma redistribuição da energia eletromagnética e também eletromecânica no sistema. O tempo necessário à acomodação das grandezas do sistema em decorrência de alterações como estas é caracterizado como período transitório. O fenômeno que envolve apenas trocas de energia de natureza eletromagnética é denominado transitório eletromagnético.

Dentre os eventos precursores de transitórios eletromagnéticos em SEPs, destacam-se a ocorrência e eliminação de curtos-circuitos, energização de transformadores, energização e religamento de linhas de transmissão e incidência de descargas atmosféricas. Em virtude de sua severidade, alguns destes eventos podem acarretar também transitórios eletromecânicos, como verifica-se em alguns casos de curtos-circuitos. No entanto, a natureza mecânica do sistema possui uma resposta mais lenta, o que permite analisar separadamente estes dois tipos de transitórios na maioria dos estudos.

As subestações, parte integrante dos SEPs, também são afetadas pelos transitórios eletromagnéticos gerados pelos eventos relacionados. Em vista disso, o conhecimento das solicitações elétricas e mecânicas impostas durante sua ocorrência é fundamental para o projeto destas instalações, bem como para o planejamento de sua operação e manutenção.

Para a análise de transitórios de natureza eletromagnética em subestações e nos sistemas elétricos em geral, pode-se contar com medições realizadas por equipamentos denominados oscilógrafos. Estes equipamentos possuem um ajuste bastante sensível, conseguindo registrar alterações das grandezas elétricas do sistema, mesmo que estas não sejam suficientes para provocar a atuação de suas proteções.

Embora estas medições forneçam informações valiosas para a realização de diversas análises, como o diagnóstico de falhas, requerem que o sistema elétrico já esteja em operação. Além disso, não é possível avaliar os transitórios gerados no sistema antes de sua ocorrência, a menos que já se disponha de uma base histórica de eventos semelhantes.

Outra metodologia para a análise de transitórios eletromagnéticos consiste na simulação do fenômeno através de modelos. Estes são desenvolvidos a partir de informações obtidas dos sistemas físicos reais e validados geralmente por meio de comparações entre suas respostas. A primeira ferramenta criada com esta finalidade foi o Analisador de Transitórios em Redes (*Transient Network Analyzer – TNA*), através do qual o sistema é representado por componentes elétricos analógicos.

Com a evolução dos computadores, programas digitais ganharam destaque para a simulação do respectivo fenômeno. Através destes, os componentes do sistema são modelados por elementos de circuitos e o comportamento das variáveis de interesse é obtido com a solução de um conjunto de equações que regem o fenômeno.

Atualmente, o estado da arte para a análise de transitórios eletromagnéticos para fins de proteção e controle é a utilização de plataformas de simulação digital em tempo real (*real-time digital simulators*), que compreendem uma combinação de software e hardware. Com estas, pode-se modelar o sistema utilizando um software adequado e realizar a simulação em hardware. O fato da simulação transcorrer em tempo real permite que equipamentos externos ao hardware, como relés de proteção, sejam conectados e considerados no fluxo de informações da simulação. Através do software, pode-se acompanhar em tempo real o comportamento de grandezas do sistema em estudo.

No entanto, apesar das vantagens oferecidas por estas plataformas, seu elevado custo de aquisição limita seu acesso a um grupo seleto de usuários. Além disso, com a tecnologia empregada atualmente, apresentam limitação quanto ao menor passo de integração possível, o que pode inviabilizar a simulação de transitórios de frente rápida e ultra rápida.

Desta forma, os softwares computacionais ainda apresentam vantagens substanciais para a análise de transitórios eletromagnéticos. Dentre estes, destaca-se o ATP-EMTP (*Alternative Transients Program*, versão alternativa do *Electromagnetic Transients Program*), que ainda pode ser considerado o software mais empregado para a análise de transitórios desta natureza em sistemas de energia. Além disso, também é comumente utilizado na análise dinâmica de eventos eletromecânicos e transitórios envolvendo sistemas de controle de SEPs.

Para utilizar o ATP-EMTP, o usuário precisa fornecer um arquivo de entrada com formato padrão, contendo os parâmetros da simulação e a configuração do sistema elétrico analisado. No entanto, além do conhecimento especializado a respeito dos modelos disponíveis no programa, esta tarefa exige uma compreensão da sintaxe e da formatação desse arquivo.

Com intuito de auxiliar a utilização do ATP-EMTP, alguns softwares pré-processadores gráficos, como o ATPDraw (HOIDALEN, 2016) e o ATPDesigner (Power Engs, 2016), lhe proporcionam uma interface gráfica. Através destes, o usuário pode selecionar modelos de componentes em uma “paleta” e construir um diagrama representativo do sistema, a partir do qual é criado automaticamente o arquivo de entrada do ATP-EMTP. Estes softwares também gerenciam a execução da simulação e disponibilizam alguns recursos para visualização e análise dos resultados.

Além dos modelos incorporados pelo ATP-EMTP, os pré-processadores gráficos citados disponibilizam uma série de novos componentes, tais como relés de proteção, filtros, medidores de frequência, harmônicos e valor eficaz. Estes são configurados pelo usuário através da interface gráfica dos softwares e representados no arquivo de entrada do ATP-EMTP a partir de seus elementos básicos.

Para a simulação de transitórios eletromagnéticos no ATP-EMTP, além da parametrização da simulação e da representação do sistema elétrico, deve-se modelar também o tipo de estudo analisado, como curto-circuito, manobra, descarga atmosférica, entre outros. Tradicionalmente, adota-se o procedimento determinístico, em que o próprio usuário define os parâmetros do evento e seu instante de ocorrência na simulação. Por exemplo, em uma simulação determinística de descarga atmosférica, o usuário pode especificar os valores dos parâmetros que descrevem o fenômeno e determinar o instante exato de sua incidência no sistema.

No entanto, tem-se o entendimento de que a ocorrência de eventos como estes em situações reais é bastante complexa, e que suas aleatoriedades podem acarretar situações imprevistas nas simulações determinísticas. Uma alternativa a esta condição é a realização em massa de simulações estocásticas, modelando as aleatoriedades inerentes a estes eventos, e a posterior análise estatística dos resultados. Porém, o ATP-EMTP, bem como seus pré-processadores gráficos, oferecem recursos limitados para a realização de simulações desta natureza.

Neste contexto, o presente trabalho busca incorporar ao ATP-EMTP recursos específicos para a simulação estocástica de eventos precursores de transitórios eletromagnéticos, com enfoque de aplicação em subestações, de modo que não seja necessário a utilização de ferramentas externas. Este tema compõe um projeto de P&D desenvolvido pelo Laboratório de Análise e Proteção de Sistemas Elétricos do Centro de Estudos em Energia e Sistemas de Potência (LAPES-CEESP) da UFSM em parceria com a Companhia Estadual de Geração e Transmissão de Energia Elétrica (CEEE-GT) do Rio Grande do Sul. O referido projeto está registrado na CEEE-GT sob o número 9947883 e na Fundação de Apoio à Tecnologia e Ciência (FATEC) sob o número 3.07.0060.

1.1 JUSTIFICATIVA E MOTIVAÇÃO

Para simulações estocásticas, o ATP-EMTP disponibiliza apenas uma chave estatística (*Statistic Switch – Type 76*), na qual o instante de operação é determinado por uma distribuição de probabilidade uniforme ou normal. O acesso a esta chave é contemplado nos recursos oferecidos pelo software ATPDraw, já na versão gratuita do software ATPDesigner, ela não está disponível.

No entanto, para utilizar a chave *Type 76* em estudos de manobra, por exemplo, buscando analisar a influência do ângulo elétrico da tensão do sistema sobre os transitórios gerados em decorrência destes eventos, é preciso associá-la a outros elementos de controle, como MODELS. Este processo requer grande intervenção do usuário, além de aumentar a complexidade da simulação em virtude do maior número de elementos envolvidos.

Com relação a outros tipos e estudo, o programa e seus pré-processadores gráficos não oferecem nenhum tipo de recurso integrado para simulação estocástica, necessitando que o usuário crie suas próprias ferramentas. Exemplo disso são os trabalhos publicados por Martinez e Castro-Aranda (2005) e Gatta et al. (2016), que relatam a implementação de plataformas dedicadas à análise de transitórios originados por descargas atmosféricas, as quais realizam simulações estocásticas deste fenômeno gerenciando a execução do ATP-EMTP.

Embora seja possível a criação de ambientes computacionais dedicados como estes, sua implementação geralmente requer um processo investigativo e laborioso. Além disso, a complexidade envolvida na utilização de ferramentas externa ao ATP-EMTP para o

gerenciamento de simulações e pós-processamento dos resultados, muitas vezes inviabiliza sua utilização, principalmente em aplicações fora do âmbito acadêmico.

Em vista destas limitações, o presente trabalho visa facilitar a realização de simulações estocásticas de transitórios eletromagnéticos diretamente no ATP-EMTP, mediante a incorporação de recursos com esta finalidade específica. Estes poderão ser acessados através do ATPDraw, o pré-processador gráfico mais utilizado na atualidade, sem a necessidade de ferramentas externas. Os eventos contemplados neste trabalho correspondem aos mais significativos e frequentes em subestações.

1.2 OBJETIVOS

1.2.1 Objetivo geral

Desenvolver e agregar ao ATP-EMTP recursos para a simulação estocástica de eventos precursores de transitórios eletromagnéticos em subestações.

1.2.2 Objetivos específicos

- Investigar metodologias para a incorporação de novas funcionalidades ao ATP-EMTP;
- Desenvolver ferramentas para a simulação estocástica dos eventos precursores de transitórios eletromagnéticos mais significativos e frequentes em subestações;
- Elaborar uma sistemática para monitoração e processamento das grandezas de interesse na simulação;
- Implementar e agregar os novos recursos ao ATP-EMTP;
- Estabelecer um procedimento para acesso e utilização dos novos recursos através do software ATPDraw.

1.3 ESTRUTURA DO TRABALHO

Para melhor organização e relato das atividades realizadas no desenvolvimento do presente trabalho, este documento foi dividido em seis capítulos:

- O capítulo 1 apresenta a introdução do trabalho, a justificativa e motivação para sua realização e os objetivos almejados.

- No capítulo 2, realiza-se uma sucinta revisão da literatura a respeito de transitórios eletromagnéticos, buscando identificar os eventos mais relevantes em subestações;
- O capítulo 3 contempla o método de Monte Carlo, enfatizando a estrutura necessária para a sua utilização em simulações estocásticas;
- O capítulo 4 apresenta um breve histórico sobre o desenvolvimento do ATP-EMTP, suas principais potencialidades e alternativas para a incorporação de novos recursos;
- O capítulo 5 relata o desenvolvimento e implementação da metodologia proposta neste trabalho;
- No capítulo 6, aborda-se alguns exemplos de aplicação da metodologia desenvolvida, buscando evidenciar suas potencialidades para simulação estocástica de transitórios eletromagnéticos em subestações.
- No capítulo 7 é apresentada a conclusão, propostas de continuidade do trabalho e as considerações finais do trabalho, bem como as publicações realizadas.

2 TRANSITÓRIOS ELETROMAGNÉTICOS EM SUBESTAÇÕES

Este capítulo apresenta uma breve revisão da literatura a respeito de fenômenos transitórios eletromagnéticos, buscando identificar os eventos mais frequentes e relevantes em subestações, considerando seus aspectos aleatórios. Além disso, busca-se referências sobre quais parâmetros são relevantes para a modelagem destes fenômenos e a definição dos valores a serem utilizados no presente trabalho.

2.1 CONSIDERAÇÕES GERAIS

A análise de transitórios eletromagnéticos é um dos estudos necessários para o planejamento de um sistema elétrico, cujos efeitos gerados podem ser, principalmente, as sobretensões e as sobrecorrentes. Por esse motivo, na operação e análise de sistemas elétricos de potência é de grande importância a determinação de magnitude, forma, frequência e duração de sobretensões. As sobrecorrentes também são cruciais em fenômenos transitórios, por estarem ligadas às sobretensões e seus efeitos serem igualmente danosos.

Segundo a NBR 6939 (2000), sobretensões temporárias são caracterizadas por frequência fundamental de duração relativamente longa, enquanto que sobretensões transitórias correspondem às sobretensões de curta duração, em geral fortemente amortecidas. As sobretensões transitórias podem ser classificadas em frente lenta, com tempo de crista entre $20 \mu s < T_{cr} \leq 5000 \mu s$ e tempo de decaimento $T \leq 20 ms$; em frente rápida com tempo de crista $0,1 \mu s < T_{cr} \leq 20 \mu s$ e tempo de decaimento $T \leq 300 \mu s$. A Tabela 2.1 apresenta tal classificação para alguns dos eventos causadores de sobretensões.

Tabela 2.1 – Relação entre eventos e sobretensões produzidas

Evento	Sobretensões Temporárias	Sobretensões Transitórias	
		Frente Lenta	Frente Rápida
Rejeição de carga	X		
<i>Inrush</i> de transformador	X	X	
Energização de linhas	X	X	
Ocorrência de curto-circuito	X	X	
Eliminação de curto-circuito	X	X	
Descarga atmosférica direta			X

Fonte: (IEC TR 60071-4, 2004).

2.2 CURTOS-CIRCUITOS

Os curtos-circuitos são eventos de ocorrência frequente em sistemas elétricos, sendo importantes nos estudos de sobretensões transitórias. Neste caso, há dois momentos distintos que podem ser analisados. Um compreende o instante da ocorrência do curto-circuito e o outro ao instante de abertura dos disjuntores para isolar eletricamente o ponto de falta do sistema.

No primeiro instante, porém, não é comum a ocorrência de sobretensões elevadas. Já no segundo, verifica-se sobretensões elevadas, podendo exceder o valor de 1,7 p.u (D'AJUZ et al., 1987). Estas sobretensões podem se manifestar tanto nas fases sob falta como nas fases “sãs” do sistema. Nestes casos, tais informações são de grande relevância para a especificação de disjuntores e na seleção de para-raios, uma vez que estes são escolhidos de modo a não atuarem dinamicamente para esta manobra para não exceder sua capacidade de absorção de energia.

Na análise de eliminação de curtos-circuitos, é necessário variar a sequência de abertura dos polos de disjuntores, procurando-se obter os casos mais críticos de sobretensões resultantes. Outros fatores que influenciam a eliminação das faltas, é sua natureza e local, o comprimento da linha de transmissão, o grau de aterramento do sistema, bem como o grau de compensação da linha de transmissão.

Além das sobretensões, outro aspecto importante a ser analisado em estudos de curto-circuito são as sobrecorrentes geradas em decorrência destes eventos. Tais valores são considerados no projeto e análise de superação de equipamentos, principalmente os disjuntores

Para esse fenômeno, a maioria das ocorrências caracterizam-se por serem curtos-circuitos monofásicos. Dessa maneira, a probabilidade das maiores sobretensões está associada não somente aos fatores estatísticos que caracterizam a própria ocorrência, mas também à probabilidade de acontecer o defeito específico (NUNES, 2006).

De acordo com Westinghouse (1950), considerando-se os setores do SEP - geração, subestação e transmissão - cada um possui características e funções diferenciadas, as quais podem contribuir positivamente ou negativamente na ocorrência do curto-circuito. As contribuições de cada setor do sistema de energia elétrica são obtidas através de averiguações estatísticas no histórico de defeitos das empresas de energia, dispostas na tabela Tabela 2.2.

Tabela 2.2 – Estatística de defeitos nos setores de SEPs.

Setor do Sistema Elétrico	Ocorrência de Curto-circuito
Geração	6 %
Subestação	5%
Linhas de Transmissão	89%

Fonte: (WESTINGHOUSE, 1950).

Percebe-se que em um SEP as linhas de transmissão são as mais vulneráveis ao acometimento de curtos-circuitos devido sua grande dimensão física, pois ficam expostas a climas distintos, a descargas atmosféricas e posicionam-se em terrenos de condições variadas. Considerando-se a ocorrência dos tipos de curtos-circuitos em SEPs, de acordo com a Tabela 2.3, nota-se que o fase-terra é o mais frequente e, em contrapartida, o trifásico é o mais raro (WESTINGHOUSE, 1950).

Tabela 2.3 – Estatística dos tipos de curto-circuito no SEP.

Tipo de Sistema Elétrico	Percentual de Curto-circuito
Trifásico	5%
Bifásico	15%
Bifásico-Terra	10%
Monofásico	70%

Fonte: (WESTINGHOUSE, 1950).

De acordo com Tabela 2.3, cerca de 70% dos curtos-circuitos são do tipo fase-terra. Nas linhas de transmissão mais importantes, em especial aquelas com elevado fluxo de potência, recomenda-se a utilização de sistemas de proteção com abertura e religamento monopolar automático dos disjuntores. Além de ajudar na manutenção do sincronismo, através das duas fases remanescentes, aproximadamente 2/3 do fluxo de potência é mantido. Em Sistemas Elétricos de Alta e Extra Alta Tensão, por razões técnicas, aplica-se uma única tentativa de religamento. Como a maioria dos curtos-circuitos são de natureza transitória (fugitiva), dados estatísticos registram sucesso em cerca de 90% dos religamentos (KIMBARK, 1948).

2.3 ENERGIZAÇÃO DE TRANSFORMADORES

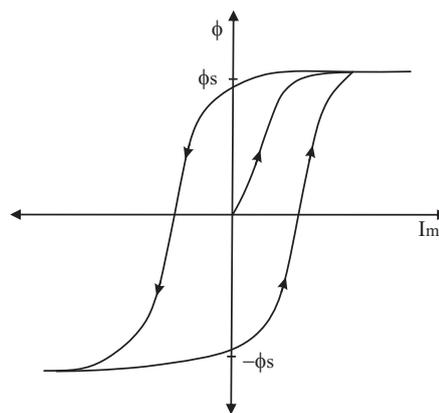
A energização de transformadores de potência podem provocar forte conteúdo de harmônicos, baixo amortecimento e altas correntes transitórias de magnetização, devido às

características de saturação do seu núcleo. Uma das principais preocupações acerca da corrente de magnetização está na possibilidade de sensibilizar os dispositivos de proteção sendo confundida com uma corrente de falta.

O núcleo do transformador opera, em condições normais, conforme a sua curva de histerese semelhante a da Figura 2.1. Tal curva pode ser analisada em duas regiões distintas. A primeira, é quando o enlace do fluxo atingir o valor ϕ_s , que corresponde à operação em regime permanente, fazendo com que o transformador se comporte como elemento essencialmente linear.

A segunda, após o ponto ϕ_s , cuja corrente de excitação aumenta muito rapidamente podendo ultrapassar a corrente nominal. Neste caso ocorre a saturação do seu núcleo, usualmente durante os transitórios do sistema. Na região saturada, as variações na tensão provocam grandes variações de corrente de magnetização. Sendo assim, corrente presente nos enrolamento primários é a corrente de excitação transitória, também conhecida como corrente de *inrush* (ARAÚJO; NEVES, 2005).

Figura 2.1 – Curva característica de histerese.



Fonte: Adaptado de (D'AJUZ et al., 1987).

Alguns fatores contribuem para caracterizar a amplitude e a duração da corrente de magnetização. Entre os principais fatores destacam-se o tamanho do banco transformador, a robustez do sistema de energia, as características construtivas do transformador, como o tipo e qualidade do material ferromagnético, e a existência ou quantidade de fluxo residual (ANDERSON; J.ERIKSSON, 1980).

Segundo D'ajuz et al. (1987), a situação que apresenta o fluxo de transitório máximo, é aquela em que o transformador é energizado no instante em que a tensão passa por zero e o fluxo magnético pelo seu valor máximo negativo. Esta condição extrema, está ligada ao maior risco de saturação do transformador.

Na corrente de *inrush* verificam-se todas as ordens de harmônicas, mas o destaque está nas segundas e terceiras componentes, cujas amplitudes são superiores que as demais. A segunda harmônica está presente em todas as formas de onda da corrente de magnetização, e sua amplitude varia de acordo com o grau de saturação do transformador.

Em sobretensões oriundas pela energização de transformador, a saturação do núcleo geralmente atua como um limitador, reduzindo a magnitude e distorcendo a forma de onda, devido à forte injeção de harmônicos na rede. Entretanto, caso a impedância harmônica seja muito elevada, é possível que as sobretensões também sejam elevadas. Via de regra, o amortecimento é baixo, pois a corrente de *inrush* permanece por muito tempo, considerando que as perdas no ferro costumam ser pequenas.

2.4 MANOBRAS DE DISJUNTORES

Sobretensões de manobra resultam da modificação brusca na configuração do sistema, abertura ou fechamento de circuitos, por isso são classificados como de origem interna. Através dos estudos de sobretensões pode-se definir as amplitudes, as frequências esperadas e as características das sobretensões temporárias e transitórias.

Os efeitos das sobretensões transitórias de manobra, atingem valores mais elevados em relação às sobretensões temporárias, por essa razão é um fator determinante no projeto econômico dos sistemas elétricos, tanto para equipamento de subestações, quanto para linhas de transmissão. Os resultados de tais estudos definem os requisitos de isolamento e de dispositivos necessários para minimizar as sobretensões excessivas, além das especificações dos equipamentos de proteção.

Os transitórios por manobras são associados por ocorrência e eliminação de curtos-circuitos, energização de transformadores, energização e religamento de linhas de transmissão. As particularidades destes eventos serão tratados nas devidas seções do capítulo 2.

A amplitude e a duração das sobretensões de manobra dependem dos parâmetros do sistema, da sua configuração e das condições em que o sistema se encontra no momento da manobra. Para um mesmo sistema, a mesma operação, efetuada em momentos diferentes, pode resultar em valores diferentes de sobretensão e também pode causar efeitos diferentes.

Por esses motivos é de grande interesse para sobretensões de manobras o aspecto probabilístico. As sobretensões dependerão não só da dispersão dos instantes de fechamento do disjuntor, como também no instante de ocorrência.

Conforme pode-se observar na Tabela 2.4, a análise e medidas das sobretensões estabeleceram faixas gerais de amplitude esperadas. Ainda assim, D'ajuz et al. (1987) afirma que os valores podem diferir consideravelmente destes listados devido ao valor das impedâncias da rede, comprimento das linhas de transmissão, das características dos disjuntores e das condições da manobra.

Tabela 2.4 – Valor máximo fase-terra para diferentes tipos de manobra.

Tipo de Manobra	Desvio Padrão	Valor Máximo em p.u.	
		Início Linha	Fim Linha
Energização de linhas	15 a 20	1,5 a 2,5	2,0 a 3,0
Religamento tripolar	10 a 25	2,0 a 3,0	3,0 a 5,0
Religamento monopolar		< 2,0	1,8 a 2,5
Eliminação de curto		< 2,0	< 2,5
Rejeição de carga		1,25 a 1,75	1,5 a 2,0
Ocorrência de curtos		1,3 a 1,6	1,6 a 2,2
Abertura de cargas indutivas		2,5	2,5
Abertura de linhas em vazio		1,3 a 1,5	1,3 a 1,5

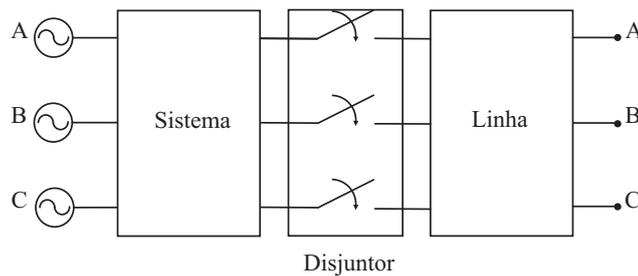
Fonte: Adaptado de (D'AJUZ et al., 1987).

O fenômeno de manobras no sistema elétrico pode ser representado sob o ponto de vista trifásico conforme a Figura 2.2. A partir desta configuração, é importante ressaltar que os três polos do disjuntor não se fecham simultaneamente, gerando uma dispersão dos contatos.

Para reproduzir esta dispersão presente na operação de abertura e fechamento de disjuntores, há necessidade de incluir a aleatoriedade na modelagem do evento. As diferenças mecânicas dos polos, o momento no ciclo em que o comando da operação é feito e o arco elétrico que se estabelece entre os contatos determinam os instantes de abertura ou fechamento efetivo.

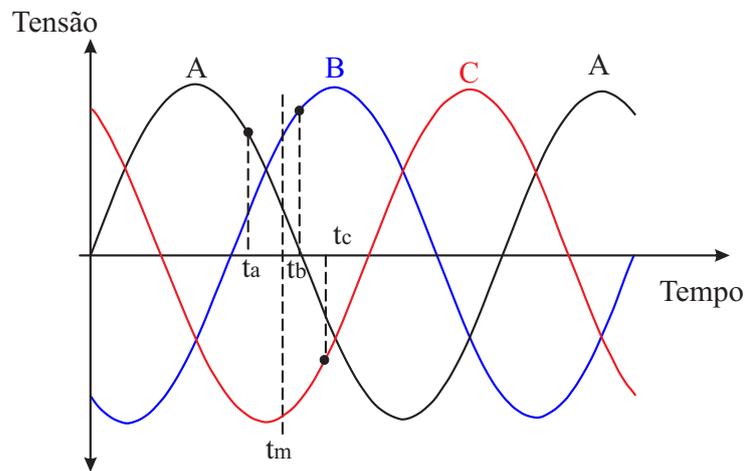
A Figura 2.3, ilustra o processo de manobra tripolar de um disjuntor em um sistema trifásico. Nesta figura, t_m representa o instante em que o comando da operação é enviado. No entanto, devido aos fenômenos aleatórios citados, a operação poderá não ocorrer efetivamente nesse instante em todas as fases. Nesta figura t_a , t_b e t_c representam os instantes efetivos de operação das fases “a”, “b” e “c”, respectivamente.

Figura 2.2 – Ilustração de um circuito trifásico em linha de transmissão.



Fonte: (ZANETTA, 2003).

Figura 2.3 – Operação de um disjuntor com dispersão de polos no circuito trifásico.



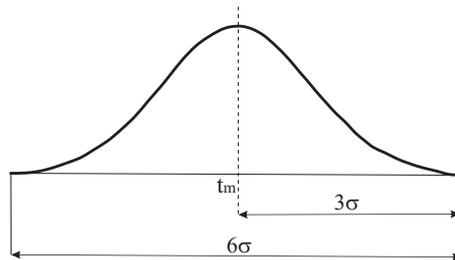
Fonte: Adaptado de Zanetta (2003).

Para representar o efeito da dispersão de polos em uma simulação de manobra de disjuntor, pode-se empregar uma distribuição de probabilidade Normal. O parâmetro a ser considerado nesta representação é o ângulo relacionado ao máximo tempo de dispersão entre os polos, representado por θ_{max} . O valor deste parâmetro é normalmente fornecido nas especificações de fabricantes de disjuntores, através dele pode-se determinar o desvio padrão σ .

Embora a curva de distribuição Normal varie entre $-\infty$ e $+\infty$, Taschini e Clerici (1970), Camm, Natarajan e Martinez (2000) sugerem truncá-la no valor referente a três desvios padrões em ambos os lados da média, conforme ilustrado na Figura 2.4. Desta forma, a partir do valor de θ_{max} , pode-se determinar o desvio padrão σ da distribuição normal através da Equação 2.1.

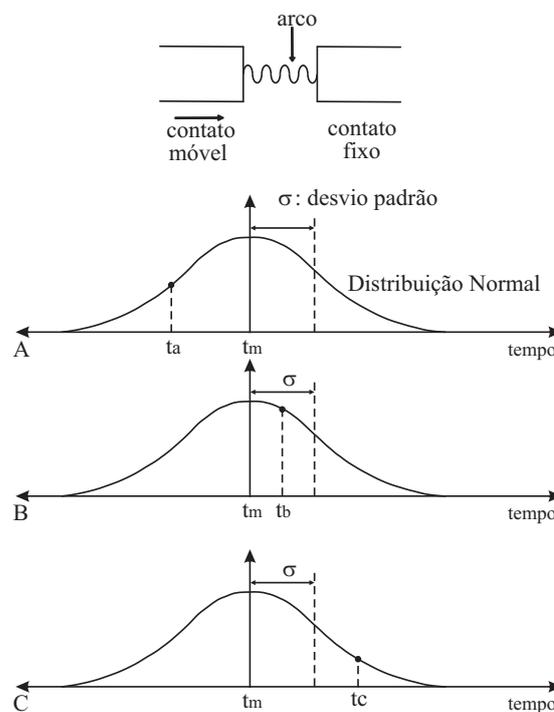
$$\sigma = \frac{\theta_{max}}{6} \quad (2.1)$$

Figura 2.4 – Distribuição Normal utilizada para modelar a dispersão entre polos de disjuntores.



Fonte: Adaptado de Taschini e Clerici (1970).

Figura 2.5 – Distribuição Normal considerando os tempos de abertura e fechamento dos polos do disjuntor.



Fonte: (ZANETTA, 2003).

Na Figura 2.5, tem-se a representação do comportamento de um disjuntor tripolar através de uma curva de distribuição Normal. Na figura, os tempos t_a , t_b e t_c , indicam um possível instante de tempo sorteado para operação do disjuntor, com um tempo médio (t_m) e um desvio padrão (σ). Zanetta (2003) sugere que a dispersão máxima entre o menor tempo e o maior, esteja dentro do limite de 5 ms ($t_c - t_a < 5$ ms).

2.5 DESCARGAS ATMOSFÉRICAS

As descargas atmosféricas são fenômenos importantes a serem analisados em um diagnóstico de sobretensões no sistema elétrico brasileiro, devido à configuração predominantemente aérea e a grande extensão das linhas de transmissão, bem como o grande número de subestações desabrigadas (ZANETTA, 2003).

Os transitórios eletromagnéticos originados por descargas atmosféricas são classificadas como de frente de onda rápida, por serem de curta duração e possuírem tempos de decaimento inferiores a $300 \mu s$ (IEC TR 60071-4, 2004). Em subestações, assim como nos demais componentes dos sistemas elétricos em geral, as descargas podem produzir surtos essencialmente de duas maneiras: por incidência direta nos condutores ou equipamentos; e por indução, através do acoplamento de campos eletromagnéticos com os condutores.

Os transitórios provocados por descargas indiretas são fenômenos mais frequentes, no entanto, provocam surtos com amplitudes relativamente reduzidas, sendo mais relevantes em sistemas de distribuição de energia elétrica. Já a incidência direta sobre os condutores fase ou equipamentos, apesar de menos frequentes, acarretam transitórios mais severos, podendo ocasionar a atuação de relés de proteção.

Um conhecimento mais exato dos parâmetros das descargas atmosféricas é essencial para a predição da severidade dos surtos que poderão atingir o sistema elétrico, causados tanto por descargas diretas como por sobretensões induzidas. Para a análise estocástica de transitórios, a avaliação das características da onda de corrente de descarga é de importância fundamental, pois refletem a mesma relevância para projetos de coordenação, de isolamento de linhas, de equipamentos de transmissão e de distribuição de energia.

2.5.1 Fenômenos físico das descargas atmosféricas

O fenômeno da descarga atmosférica é constituído por várias etapas. Para o caso da descarga entre nuvem-solo negativa, a partir de uma nuvem carregada com cargas elétricas negativas em sua base, esta induz no solo abaixo de sua sombra um acúmulo de cargas de sinal contrário, estabelecendo uma grande diferença de potencial. Devido ao rompimento da rigidez dielétrica do local, forma-se a “descarga piloto” (*stepped leader*) próxima à base da nuvem.

Devido a influência da descarga piloto, há formação de um intenso campo elétrico entre ela e o solo, com indução e conseqüente formação de descargas positivas ascendentes (*upward leaders*). Com o encontro entre descarga piloto e descargas ascendentes, há a formação da descarga principal denominada de descarga de retorno (*return stroke*) (SHIGIHARA, 2005).

Algumas vezes, após o fluxo de corrente de retorno, cessam os processos de transferência de carga negativa para o solo, correspondentes à descarga. Entretanto, pode haver seqüência nos processos elétricos, com a ocorrência de novas descargas através do mesmo canal, caracterizando as descargas subseqüentes. A primeira descarga normalmente tem maior amplitude de corrente do que as descargas subseqüentes. Embora não se conheça a correlação entre essas amplitudes, na média as descargas subseqüentes atingem cerca de 40% da primeira descarga (VISACRO, 2005).

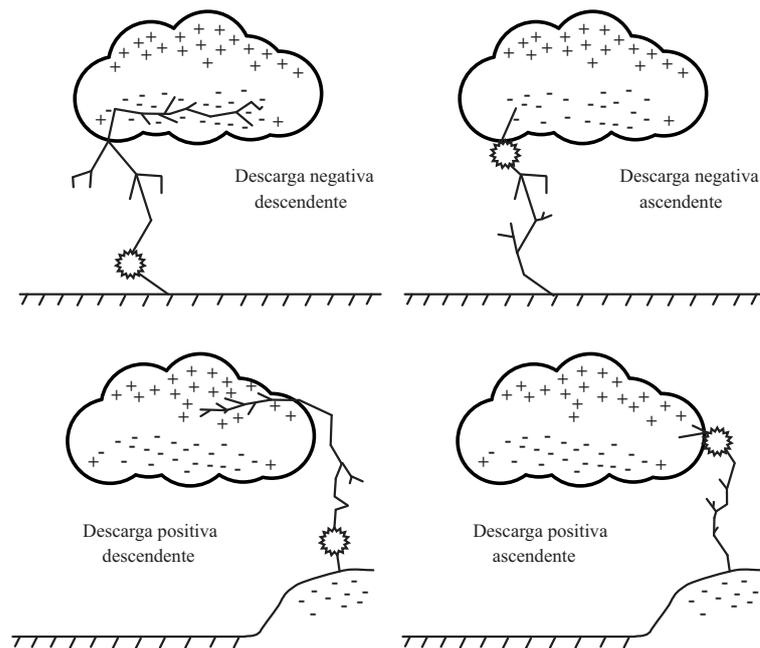
A corrente de retorno representa o ponto alto da descarga atmosférica, quando uma grande quantidade de cargas se move através do canal ionizado. Tal corrente é caracterizada por uma ascensão rápida ao pico, dentro de alguns microssegundos, e um decaimento relativamente mais lento, da ordem de algumas dezenas de microssegundos. A tensão gerada ou induzida pela descarga, é tão maior quanto menor for o tempo de frente da descarga de retorno. Assim, a velocidade da descarga de retorno afeta diretamente a componente da tensão induzida pelo campo eletromagnético provocado pela descarga (SARAN, 2009).

As descargas atmosféricas entre nuvem-solo podem ser classificadas, com relação à direção de propagação do canal precursor da descarga, em descendentes e ascendentes. Nas descargas descendentes, o canal precursor se origina na nuvem e evolui descendentemente até induzir o surgimento do canal de descarga ascendente, sendo que o fechamento do canal se dá próximo ao solo. Já nas descargas ascendentes, a origem do canal precursor ocorre a partir do

solo evoluindo ascendentemente, e o fechamento do canal ocorre próximo à nuvem.

Tanto as descargas descendentes quanto as ascendentes, podem possuir a polaridade negativa e positiva. Em descargas positivas, a parte da nuvem conectada ao solo está carregada positivamente. Em descargas negativas, o canal de descarga conecta a base negativa da nuvem com o solo. De acordo com a Figura 2.6, pode-se visualizar tais classificações.

Figura 2.6 – Classificação das descargas nuvem-solo.



Fonte: Adaptado de Visacro (2005)

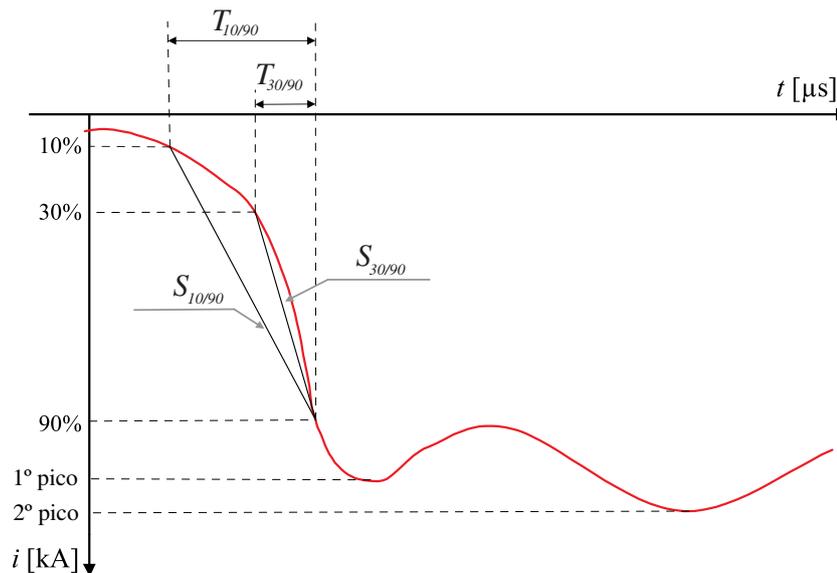
Com relação aos dados das descargas no Brasil, é oportuno ressaltar que o índice médio de densidade de descargas nuvem-solo é bastante elevado, bem como as descargas descendentes negativas possuem amplitude significativamente maior, quando se compara os dados das medidas apresentados por (BERGER et al., 1975), em Monte San Salvatore - Suíça; e (VISACRO, 2005) na Estação de Pesquisas Morro do Cachimbo - Minas Gerais. Além disso, as descargas com polaridade negativa têm uma maior probabilidade de ocorrência, aproximadamente 90% dos casos, em relação aos com polaridade positiva, apresentando esses últimos intensidades maiores.

2.5.2 Parâmetros típicos da onda de descarga

As primeiras caracterizações dos parâmetros das ondas de corrente foram realizadas por Berger et al. (1975), sendo posteriormente complementados por Anderson e J.Eriksson

(1980). Através do resultado de tais pesquisas, foi derivada a Figura 2.7, válida para descargas descendentes negativas. Dentre este conjunto de parâmetros, selecionou-se alguns essenciais para o propósito do presente trabalho, pois são potencialmente capazes de influenciar no valor dos surtos gerados:

Figura 2.7 – Forma de onda típica de uma corrente de retorno nuvem-solo negativa.



Fonte: Adaptado de CIGRE (1991)

Corrente de Pico (I_p): corresponde ao valor de pico ou amplitude da onda de corrente, o seu valor máximo. Este é o parâmetro físico da corrente de descarga considerado de maior interesse, pois determina os níveis máximos de solicitação imposta aos componentes do sistema. Logo é o parâmetro crítico para o dimensionamento do isolamento e definição de práticas de proteção. As ondas da primeira corrente de descarga negativa apresentam dois picos, geralmente o segundo pico superior ao primeiro, sendo o primeiro pico determinante para a definição dos outros parâmetros. A corrente de pico está ilustrada na Figura 2.7 e na Figura 2.8.

Tempo de Frente (T_f): corresponde ao intervalo de tempo decorrido entre o início da onda impulsiva até o alcance do primeiro pico a onda. Usualmente, alguns procedimentos definem o tempo de frente, assumindo como referência os instantes em que são atingidos os valores fixos percentuais em relação ao primeiro pico da onda de corrente.

O intervalo de tempo entre 30 % e 90 %, indicado na Figura 2.7 pelo parâmetro $T_{30/90}$, se deriva o tempo de frente simplesmente por dividi-lo por 0,6. Considera-se que o intervalo de

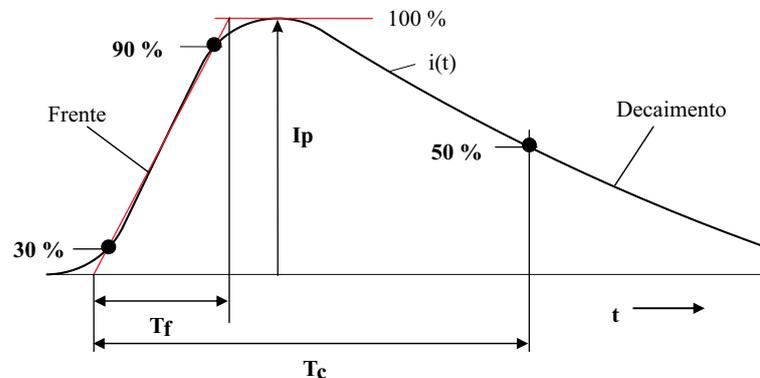
tempo decorrido entre os valores 0,3 e 0,9 do valor do pico da onda corresponde a 60% do tempo de frente.

Na Figura 2.8 também evidencia-se o parâmetro $T_{30/90}$, porém caracteriza-se uma forma de onda de descarga positiva. Em alguns trabalhos de medição direta, considera-se o T_f como o intervalo decorrido entre o instante em que a onda ultrapassa um valor limite, por exemplo 2 kA. O importante, ao adotar um desses parâmetros, é observar que, em eventuais comparações, seja sempre empregada a mesma referência de cálculo.

Sobre a influência deste parâmetro na perspectiva de aplicações, relaciona-se ao comportamento dos isolamentos dos sistemas elétricos e eletrônicos no que se refere à suportabilidade das sobretensões. Também influencia sensivelmente a amplitude de tensões induzidas em redes e sistemas, devido incidência de descargas próximas.

Tempo de Cauda (T_c): é caracterizado como tempo decorrido entre o início da onda impulsiva e o instante em que a corrente, após ter ultrapassado o valor de pico, tem sua intensidade reduzida a 50% deste valor. Este parâmetro está evidenciado na Figura 2.8.

Figura 2.8 – Identificação dos parâmetros típicos para modelagem de descargas atmosféricas.



Fonte: Adaptado de Heidler et al. (2008)

Os parâmetros característicos da corrente de descarga atmosférica acima definidos têm grande dispersão estatística, a qual deve ser considerada nas simulações de transitórios eletromagnéticos através de uma distribuição de probabilidade Log-Normal. Na Tabela 2.5 são exibidos alguns dados estatísticos apresentados pelo *Conseil International des Grands Réseaux Électriques* (CIGRE, 1991) e revisados pelo *Lightning and Insulator Subcommittee of the T&D Committee* (2005), provenientes de uma compilação de estudos realizadas por diversos autores envolvendo torres instrumentadas de captação e medição das descargas atmosféricas.

Tabela 2.5 – Parâmetros da distribuição Log-Normal para descargas descendentes negativas.

Parâmetro	Média	Desvio Padrão
Tempo de Cauda T_c	77,5 μs	0,577 μs
Tempo de Frente T_f	3,83 μs	0,553 μs
Corrente de Pico I_p	31,1 kA	0,484 kA

Fonte: (CIGRE, 1991), (Lightning and Insulator Subcommittee of the T&D Committee, 2005).

2.5.3 Representação das formas de onda da corrente de descarga

Para atender as simulações do presente estudo, optou-se pelas representações de forma de onda Dupla Rampa e Heilder.

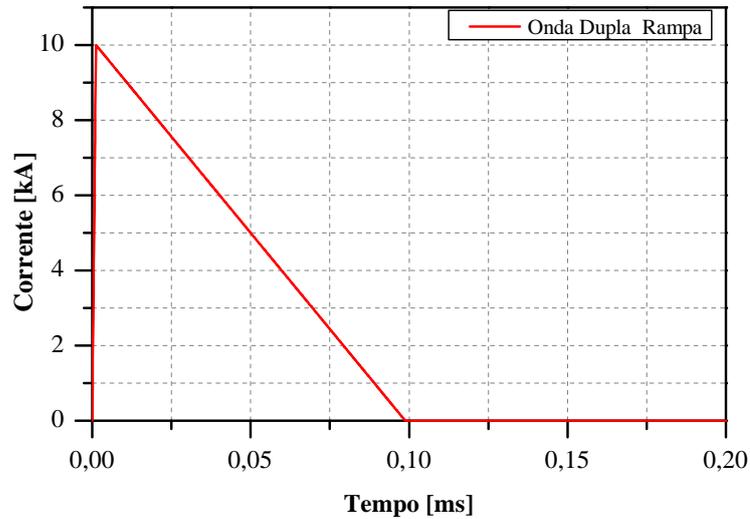
Forma de Onda Dupla Rampa: para fins de simulação em ambientes laboratoriais e em simuladores digitais, a forma de onda da corrente de descarga pode ser representada de maneira simplificada por uma fonte ideal modelada através da função Dupla Rampa, também conhecida como rampa triangular. Embora esta forma de onda não reflita certas características de curva original, como a frente de onda côncava, possibilita a simulação de sobretensões suficientemente representativas (VISACRO, 2005).

Outra vantagem desta simplificação, reside no fato de que não há necessidade de ajuste preliminar dos parâmetros, já que é completamente especificada pela corrente de pico I_p , tempo de frente T_f , e tempo de cauda T_c . Esta curva é obtida através da Equação 2.2 (rampa de subida) e da Equação 2.3 (rampa de descida), representada através da Figura 2.9.

$$i_{(t)} = \frac{I_p}{T_f} \quad (2.2)$$

$$i_{(t)} = -\frac{I_p}{2(T_c - T_f)}(T_c - T_f) + I_p \quad (2.3)$$

Figura 2.9 – Forma de onda Dupla Rampa com parâmetros $T_f = 1,2 \mu s$, $T_c = 50 \mu s$ e $I_p = 10 \text{ kA}$.



Fonte: Elaborado pelo autor.

Forma de Onda Heidler: a forma de onda Heidler constitui uma expressão analítica, concebida através de dados obtidos na observação das correntes de retorno em torres monitoradas com instrumentação específica.

Em cálculos que envolvem corrente de retorno, os parâmetros: corrente, taxa de inclinação da corrente, e carga transferida, são necessários em cada ponto do curso do canal ionizado. Portanto, a função deve ser capaz de ser diferenciada pelo menos duas vezes, especialmente na primeira derivada não pode haver descontinuidade no instante de tempo $t=0$ e $di/dt = 0$. Apesar de haver outros modelos propostos para forma de onda de corrente de descargas, Cvetic, Stanic e Heidler (1999) satisfazem em sua proposta o objetivo citado.

A curva é obtida através da Equação 2.4, e permite o ajuste, de forma independente, da amplitude da corrente, da derivada máxima da corrente e carga transferida, através das constantes I_p , τ_1 e τ_2 .

$$i(t) = \frac{I_p}{\eta} \frac{\left(\frac{1}{\tau_1}\right)^n}{1 + \left(\frac{1}{\tau_1}\right)^n} e^{\left(-\frac{1}{\tau_2}\right)} \quad (2.4)$$

Onde:

n = parâmetro de forma (2 a 10);

η = fator de correção de amplitude;

τ_1 = coeficiente relacionado ao T_f ;

τ_2 = coeficiente relacionado ao T_c ;

O fator de correção de amplitude (η) pode ser encontrado quando o instante t_m é conhecido, o qual corresponde ao instante em que a corrente atinge seu valor de pico. A partir da equação $\frac{di(t_m)}{dt} = 0$, o instante de tempo pode ser deduzido pela Equação 2.5.

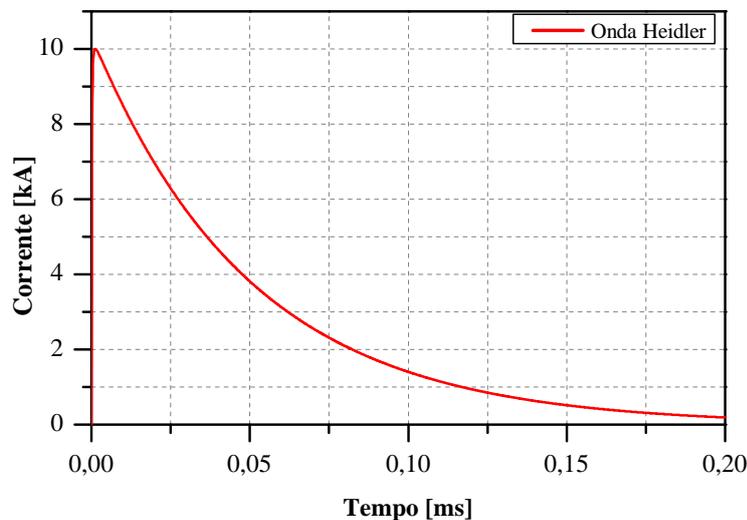
$$t_m = \tau_1 \left(n \frac{\tau_2}{\tau_1} - \frac{t_m}{\tau_1} \right)^{\frac{1}{n+1}} \quad (2.5)$$

Porém, a solução analítica da Equação 2.5 não pode ser determinada para um caso geral. No entanto, assumindo que $\left(\frac{t_m}{\tau_1} \right)^n \gg 1$, o tempo t_m pode ser deduzido pela Equação 2.6 em um processo iterativo (CVETIC; HEIDLER, 2002).

$$t_{m(k+1)} = \tau_1 \left(\xi - \frac{t_{m(k)}}{\tau_1} \right)^{\frac{1}{n+1}}, \text{ com } \xi = \frac{n\tau_2}{\tau_1} \quad (2.6)$$

Vale ressaltar que o parâmetro de forma altera significativamente o formato da onda Heidler, tipicamente se emprega $n = 2$. A forma de onda Heidler está ilustrada na Figura 2.10.

Figura 2.10 – Forma de onda Heidler com parâmetros $T_f = 1,2 \mu s$, $T_c = 50 \mu s$ e $I_p = 10 \text{ kA}$.



Fonte: Elaborado pelo autor.

3 MÉTODO DE MONTE CARLO

Neste capítulo, apresenta-se algumas considerações a respeito do Método de Monte Carlo (MMC), com ênfase para sua estrutura. Aborda-se também, uma breve caracterização dos processos estocásticos, particularizando os eventos contínuos. Outro tópico tratado no capítulo, necessário para aplicação do MMC, são os geradores de números aleatórios e as técnicas de amostragem. Além disso, são apresentados os procedimentos para amostragem das distribuições de probabilidade contínuas Uniforme, Normal e Log-Normal.

3.1 CONSIDERAÇÕES GERAIS

O Método de Monte Carlo (MMC) surgiu formalmente na década de 40 a partir da publicação do artigo “*The Monte Carlo Method*” da autoria dos matemáticos John Von Neumann e Stanislaw Ulam. Sua aplicação inicial como ferramenta de pesquisa foi em um projeto de armas nucleares no laboratório Nacional de Los Alamos, trabalho que envolveu uma simulação direta dos problemas probabilísticos acerca da difusão aleatória de nêutrons (SÓBOL, 1983)

O nome dado ao método deve-se à capital de principado de Mônaco, mundialmente conhecida pelo seu cassino, onde são comuns os jogos de azar, cujos resultados são essencialmente aleatórios. Na década de seu surgimento, foi pouco utilizado por se tratar de um método numérico cuja abordagem necessitava de experimentos repetitivos, o que o torna impraticável manualmente. Entretanto, sua eficácia foi diretamente favorecida pelo emprego dos computadores.

Desde então, o MMC se tornou uma ferramenta matemática utilizada em diversos segmentos da ciência para simular problemas representados tanto por modelos estocásticos, como por modelos determinísticos. No entanto, é na primeira aplicação citada que o método encontra sua verdadeira eficácia, sendo algumas vezes, a única solução viável (LAZO, 2009).

O MMC tem se tornado, ao longo dos anos, uma ferramenta matemática utilizada em diversos segmentos da ciência. A simplicidade de seus algoritmos e eficiência para geração de uma enorme quantidade de resultados numéricos, justifica sua utilização em áreas como economia, física, química, medicina e engenharias.

3.2 PROCESSOS ESTOCÁSTICOS

Para representação de eventos no ramo das ciências exatas e das engenharias, constroem-se modelos matemáticos, que podem ser determinísticos ou probabilísticos. Por evento determinístico, entende-se aquele em que se tem absoluta certeza do resultado de sua ocorrência, são totalmente caracterizados a priori, pois são fenômenos cujo resultado é sabido antes mesmo em que ele ocorra. Para isso, na grande maioria das vezes, empregam-se ferramentas matemáticas conhecidas como equações integro-diferenciais (BORTOLOSSI, 2012).

Entretanto, para representar com fidelidade as incertezas inerentes aos eventos que dão origem às sobretensões, optou-se pelo modelo probabilístico, que consiste em uma descrição matemática de um fenômeno aleatório, ou seja, um fenômeno em que os resultados individuais são incertos, embora haja uma distribuição regular de resultados em grande número de repetições. Para a representação deste fenômeno aleatório, atribuiu-se um resultado numérico a uma variável denominada variável aleatória.

Yates e Goodman (2005) trazem a seguinte definição para variável aleatória:

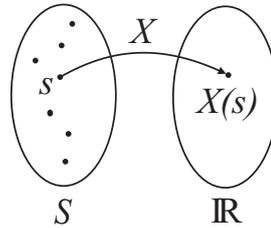
Para um dado espaço amostral S de um experimento, uma variável aleatória é qualquer regra que associe um valor a cada resultado de S . Em termos matemáticos, uma variável aleatória é uma função cujo domínio é o espaço amostral e o contradomínio é um conjunto de números reais.

Dentre os tipos de processos possíveis de serem analisados em termos de probabilidade, têm-se aqueles em que o instante de tempo ou a evolução temporal são necessários para explicar o evento aleatório em estudo. A estes, dá-se o nome de processos estocásticos. Portanto, quando estudamos processos estocásticos, analisamos funções aleatórias de tempo. Quase todas as aplicações práticas da probabilidade envolvem múltiplas observações tomadas ao longo de um período de tempo.

Assim, seja S o espaço amostral associado a um experimento, uma variável aleatória X associa a cada elemento $s \in S$ um número real $X(s)$ que pode ser representado conforme a Equação 3.1 e Figura 3.1:

$$X : S \rightarrow \mathbb{R}, s \mapsto X(s) \quad (3.1)$$

Figura 3.1 – Representação da definição de variável aleatória.



Fonte: Elaborado pelo autor.

Desta forma, pode-se escrever os resultados de um experimento aleatório através de números, possibilitando um tratamento matemático facilitado. Observa-se que, embora X seja a variável do modelo probabilístico, é na verdade uma função que traduz o resultado do experimento em números reais.

De acordo com Devore (2006), uma variável aleatória pode ser:

- Contínua: quando pode assumir qualquer valor dentro de determinado intervalo, sendo medida numa escala contínua;
- Discreta: quando o número de resultados possíveis é finito ou pode ser contado.

Uma vez definida a variável aleatória, tem-se interesse no cálculo dos valores das probabilidades correspondentes. Ao serem atribuídas probabilidades aos elementos s do espaço amostral S , estas determinam as probabilidades de qualquer variável aleatória associada. O conjunto das variáveis (x_i) e das probabilidades $f(x_i)$ correspondentes, expresso pela Equação 3.2, é denominado distribuição de probabilidades.

$$\{(x_i, f(x_i)); i = 1, 2, 3, \dots, n\} \quad (3.2)$$

A distribuição de probabilidade associa uma probabilidade a cada resultado numérico de um experimento, ou seja, dá a probabilidade de cada valor de uma variável aleatória. De acordo com o tipo de variável aleatória, uma distribuição de probabilidade pode ser contínua ou discreta.

A probabilidade acumulada de todos os valores menores ou iguais a um determinado valor x da variável aleatória X , recebe o nome de Função Distribuição Acumulada (*Cumulative Distribution Function – CDF*), também conhecida como Função Distribuição de Probabilidade (*Probability Distribution Function – PDF*), ou simplesmente Função Distribuição. Desta forma, a *CDF* $F(x)$ da variável aleatória X , seja ela discreta ou contínua, é definida pela Equação 3.3.

$$F(x) = P(X \leq x) \quad (3.3)$$

3.2.1 Distribuições de probabilidade para variáveis aleatórias contínuas

Quando se trabalha com variáveis aleatórias contínuas, não é possível enumerar todos os seus possíveis valores. Neste caso, recorre-se ao emprego de funções que se ajustem à distribuição de probabilidade, as quais são denominadas de função densidade de probabilidade (*probability density function – pdf*) (MOORE; NOTZ; FLIGNER, 2014).

É importante salientar que, no presente trabalho, *pdf* referenciada em letras minúsculas designa função densidade de probabilidade, em contraste com a *PDF* definida anteriormente. Mesmo assim, será utilizado o emprego do termo *CDF* em lugar de *PDF* para denotar a função distribuição acumulada de modo a evitar ambiguidades.

A *pdf* de uma variável aleatória contínua X é definida como sendo a função $f(x)$ expressa pela Equação 3.4.

$$P(a \leq X \leq b) = \int_a^b f(x)dx \quad (3.4)$$

A Equação 3.4 equivale dizer que “a probabilidade de X ter um determinado valor no intervalo $[a,b]$ é a área contida entre o intervalo e abaixo da curva da função de densidade” (Devore, 2006, p. 129).

Para que $f(x)$ seja uma *pdf* legítima, deve satisfazer as seguintes condições:

1. $f(x) \geq 0$ para todos os valores de x ;
2. $\int_{-\infty}^{\infty} f(x)dx = 1$ (área abaixo da curva de $f(x)$).

A partir da definição de *pdf*, pode-se inferir que a *CDF* para a mesma variável aleatória contínua X é definida pela Equação 3.5.

$$F(x) = P(X \leq x) = \int_{-\infty}^x f(\tau)d\tau \quad (3.5)$$

onde τ é uma variável auxiliar de integração.

Das propriedades da operação integral, tem-se a relação adicional, expressa pela Equação 3.6, entre uma *pdf* e a sua correspondente *CDF*.

$$\int_a^b f(x)dx = F(b) - F(a) \quad (3.6)$$

Na sequência, são apresentadas algumas distribuições contínuas de probabilidades clássicas, acompanhadas de sua *pdf* e técnicas de amostragem.

3.3 ESTRUTURA DO MÉTODO DE MONTE CARLO

O MMC consiste essencialmente em realizar uma simulação matemática de um processo em estudo para obter uma estimativa de solução, contrastando com métodos determinísticos que buscam solucionar a equação ou conjunto de equações que regem o problema. Como se trata de uma estimativa, este método é incapaz de indicar a solução exata do problema. No entanto, a medida em que um número suficientemente grande de amostras é processado a precisão da estimativa é melhorada (RUBINSTEIN, 1981).

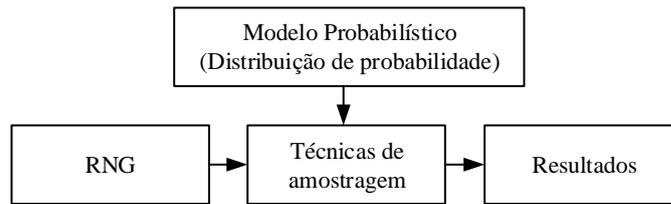
Para a aplicação do método, deve-se primeiramente verificar se o processo de interesse pode ser representado por um modelo probabilístico. Caso essa hipótese seja confirmada, deve-se identificar o modelo correspondente.

Na maioria dos casos, o modelo probabilístico é expresso através de uma função densidade de probabilidade (*pdf*). Quando se dispõe de uma base histórica de dados, pode-se aplicar testes estatísticos para identificar qual *pdf* melhor representa o processo (DEVORE, 2006). No entanto, o enfoque do presente trabalho não é a determinação do modelo, e sim a implementação das ferramentas necessárias para a realização da simulação do MMC. Para tanto, considera-se que o processo possa ser representado por uma *pdf* e que esta já é conhecido.

A simulação de MC é realizada através de sucessivas amostragens aleatórias da *pdf* utilizando um gerador de números aleatórios (*Random Number Generator — RNG*). Este procedimento imita a realização de um experimento do processo em estudo para a obtenção de resultados. Estes são então analisados estatisticamente para identificação do resultado mais provável.

O sucesso do MMC depende, da adequação do modelo estocástico subjacente, mas também, em grande parte, sobre a forma como os números aleatórios utilizados no método geram as variáveis aleatórias. Nota-se que a estrutura do método é bastante simplificada, requerendo basicamente um *RNG* e uma técnica de amostragem da *pdf* (Figura 3.2).

Figura 3.2 – Estrutura simplificada da simulação de MC.



Fonte: Adaptado de Yoriyaz (2009, p. 24)

3.4 GERADOR DE NÚMEROS ALEATÓRIOS

A simulação do MMC é realizada por meio de amostragens das *pdf*, as quais são realizadas através de números aleatórios. Portanto, qualquer programa computacional que utilize o MMC necessita de um *RNG*.

Era usual que livros texto de estatística incluíssem em apêndices tabelas de números aleatórios. No entanto, como é comum hoje em dia a utilização de computadores para a realização de simulações como a de MC, é mais conveniente recorrer a algoritmos computacionais para a geração de números aleatórios.

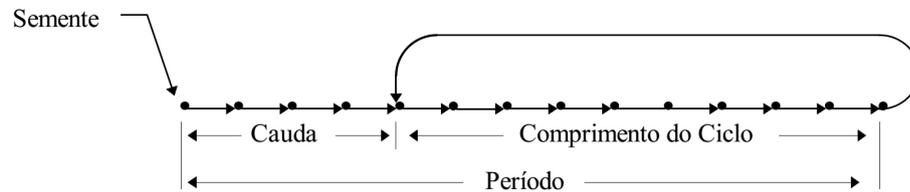
Segundo Gutmann (1997), uma excelente técnica para construir um *RNG* é utilizar medidas de propriedades físicas, tais como temperatura ambiente, umidade do ar, nível de intensidade sonora, entre outras. A companhia *Randomness and Integrity Services Ltd* (RANDOM.ORG, 2012), por exemplo, utiliza o ruído eletromagnético atmosférico para gerar números verdadeiramente aleatórios, os quais são disponibilizados através de seu site na internet.

No entanto, poucos computadores (ou usuários) tem acesso a hardwares especializados para capturar e analisar sinais de fontes aleatórias naturais. Em vista disso, recorre-se ao emprego de algoritmos matemáticos passíveis de implementação em computadores para a geração de números aleatórios.

Os números gerados por um computador, no entanto, são na verdade pseudoaleatórios, pois são obtidos por processos determinísticos. Estes números constituem uma sequência finita de tamanho usualmente denominado período ou comprimento de ciclo, sendo que o valor inicial da sequência é conhecido como semente (KNUTH, 1997). Assim, uma vez atingida o final da sequência, inicia-se uma outra idêntica, formando uma repetição cíclica dos valores. Alguns

geradores não repetem uma parte inicial do ciclo, chamada de cauda. Neste caso, o comprimento de seu período é dado pela somado comprimento da cauda mais o comprimento do ciclo. A Figura 3.3 ilustra as características deste ciclo.

Figura 3.3 – Comprimento do ciclo, cauda e período de gerador de números aleatórios.



Fonte: Viali (2014, p. 3)

Desta forma, para uma dada semente, pode-se afirmar com absoluta certeza quais serão os números na sequência. Embora isto seja verdadeiro, o objetivo primordial de qualquer algoritmo *RNG* é produzir uma sequência de números com propriedades semelhantes aquelas dos verdadeiros números aleatórios.

Uma sequência de números aleatórios $r_1, r_2, r_3, \dots, r_n$ deve apresentar as propriedades de uniformidade e independência (PRESS et al., 2007). Isto significa dizer que todo número r_i da sequência é uma amostra independente de uma distribuição uniforme e contínua no intervalo fechado entre 0 e 1.

Para verificar estas características, o produto do algoritmo *RNG* deve ser submetido a testes estatísticos de aleatoriedade. Existem muitos testes que podem ser utilizados para verificar a fidelidade desses números, dentre os mais conhecidos na literatura estatística estão o teste Qui-quadrado e de Cramer-Von Mises, os mesmos empregados em testes de aderência (KNUTH, 1997). Embora uma sequência de números pseudoaleatórios seja completamente determinística, uma vez aprovados em testes estatísticos, podem ser utilizados na simulação de MC e proporcionar resultados idênticos aos verdadeiramente aleatórios.

Além das características de uniformidade e independência acima citadas, Press et al. (2007) recomenda que um bom *RNG* contenha:

- Período Longo: um período curto pode fazer com que haja a reciclagem da sequência de números aleatórios, resultando em uma repetição da sequência de eventos. Idealmente, o gerador não deve repetir valores. Na prática, a repetição deve ocorrer somente depois da geração de um grande conjunto de números aleatórios.

- Facilidade de implementação e eficiência: devem ser fáceis de serem implementados em uma linguagem de alto nível. Além disso, as simulações geralmente necessitam da geração de milhares de números aleatórios, logo o tempo para processar cada geração deve ser mínimo;
- Repetição: os geradores devem possuir habilidade de repetir exatamente a mesma sequência, pois isto é importante em procedimentos de teste e desenvolvimento de programas.
- Portabilidade: os geradores devem ser portáteis, isto é, gerar exatamente a mesma sequência de números aleatórios em duas máquinas diferentes, possivelmente com tamanhos diferentes de palavras, sendo que uma palavra é igual a 2 bytes ou 16 bits.
- Deve-se evitar geradores disponibilizados em bibliotecas *default* de linguagens de programação, como C e C ++, especialmente o denotado por “rand”. Estes não tem implementação padrão e muitas vezes são falhos, além de possuírem período relativamente curto quando comparados a outros algoritmos disponíveis na literatura;
- Utilizar pelo menos dois métodos de geração, desde que eles não sejam relacionados. Os métodos combinados (híbridos) devem evoluir de forma independente, e a combinação deve ser dada por operações simples que não produzam resultados menos aleatórios que seus operandos.

Na sequência, são apresentados alguns dos tipos de geradores de números pseudoaleatórios mais conhecidos e utilizados apontados por Knuth (1997) e Viali (2014).

a) Geradores Congruentes Lineares: são um dos mais antigos e conhecidos algoritmos para a criação de números aleatórios, sendo o número gerado pela Equação 3.7.

$$x_{n+1} = (ax_n + c) \bmod m \quad (3.7)$$

onde:

x_{n+1} = número gerado;

x_n = número gerado no passo anterior, sendo x_0 a semente;

a, c e m = constantes;

\bmod = função que retorna o resto da divisão inteira de $(ax_n + c)$ por m .

b) Geradores Congruentes Lineares Multiplicativos: é uma derivação do método citado anteriormente. Nestes geradores o valor do incremento c é nulo, sendo representado pela Equação 3.8.

$$x_{n+1} = (ax_n) \bmod m \quad (3.8)$$

Os geradores baseados neste método são mais eficientes do que o anterior devido a ausência de adições, uma vez que o tempo de processamento necessário se reduz. Além disso, pode-se aumentar sua eficiência quando m assume uma potência de 2, fazendo com que a operação mod seja trivial.

b) Geradores de Atraso de Fibonacci: têm esse nome por causa da similaridade com a sequência de Fibonacci, têm a vantagem de serem rápidos. A Equação 3.9 expressa a forma de obtenção da sequência.

$$x_n = (x_{n-l} \text{ op } x_{n-k}) \bmod m, \quad 0 < k < l \quad (3.9)$$

onde:

- x_n = número gerado;
- x_{n-l} e x_{n-k} = penúltimo e o antepenúltimo números gerados, respectivamente;
- op = uma operação de adição, subtração, multiplicação ou módulo;
- mod = função que retorna o resto da divisão inteira de $(x_{n-l} \text{ op } x_{n-k})$ por m .

Sendo que l , k e m são constantes, e que a escolha destas constantes é crucial na produção de números aleatórios de boa qualidade. Segundo Gentle (2003), apesar destes geradores não terem uma base teórica tão sólida quanto a dos geradores congruenciais lineares, produzem bons resultados.

c) Geradores de Registradores de Deslocamento (*Shift Registers Generators*): consideram os bits de uma palavra de computador como os elementos de um vetor binário. Através de transformações lineares, geram sequências de vetores binários que são interpretadas como números inteiros aleatórios uniformes.

d) Geradores Híbridos: são uma combinação de dois ou mais geradores com o objetivo de obter um gerador com melhor desempenho.

Neste contexto, Press et al. (2007) recomenda a utilização o *RNG* denominado “Ran”, que é a combinação de quatro métodos de geradores diferentes, portanto um gerador híbrido, representado pela Equação 3.10.

$$Ran = [A1_L C3 + A3_R] \wedge B1 \quad (3.10)$$

Em que $A1$, $A3$, $B1$ e $C3$ representam os diferentes métodos formadores de *RNG*, e os subscritos L ou R indicam se a operação de deslocamento é feita primeiro para esquerda ou para a direita, respectivamente. O período de Ran é o mínimo múltiplo comum dos períodos de $A3$, $B1$ e $C3$. Segundo o autor, este *RNG* é capaz de suprir satisfatoriamente a maioria das demandas de geração de números aleatórios.

3.5 TÉCNICAS DE AMOSTRAGEM

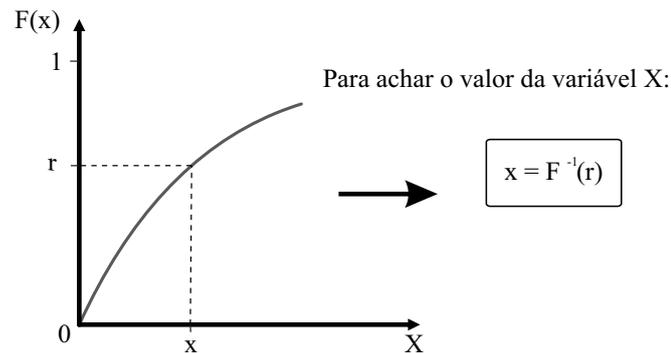
De posse de um algoritmo de geração de números aleatórios, é necessário fazer uso de uma técnica de amostragem para a reprodução de uma determinada *pdf*.

3.5.1 Método da Transformação Inversa

A técnica mais utilizada para a obtenção de uma variável aleatória contínua é a Transformação Inversa. Este método consiste na obtenção de uma expressão analítica para a variável aleatória a partir da função de distribuição acumulada. No entanto, isto só é possível se a *pdf* for integrável (RUBINSTEIN, 1981).

O método da Transformada Inversa é aplicado supondo uma distribuição probabilística com função de densidade contínua $f(x)$ com função de distribuição acumulada $F(x)$. Sendo que $F(x)$ esteja no mesmo intervalo que um número aleatório r gerado por um *RNG* no intervalo entre 0 e 1. Para gerar uma variável aleatória que siga essa distribuição probabilística, considera-se que r é um valor de $F(x)$ e determina-se sua inversa, conforme a Figura 3.4.

Figura 3.4 – Método da Transformação Inversa.



Fonte: Adaptado de Devore (2006)

3.5.2 Método de Box-Müller

Embora o método da Transformação Inversa seja a técnica de amostragem mais empregada, em alguns casos, a obtenção da integral da *pdf* não é trivial. Exemplo disso é o que ocorre com a distribuição de probabilidade Normal.

Em casos como este, recorre-se a procedimentos alternativos, como o método desenvolvido por George Edward Petham Box e Mervin Edgar Müller. Este consiste na geração de duas amostras aleatórias independentes da distribuição Normal com média nula e desvio padrão unitário a partir de um *RNG* (MOREIRA; SOUZA, 2012).

3.6 DISTRIBUIÇÕES DE PROBABILIDADE CONTÍNUAS

As distribuições de probabilidade contínuas representam quantidades aleatórias contínuas que podem tomar um número infinito de valores. Elas podem ser classificadas de acordo com as amostragens Uniforme, Normal e Log-Normal, as quais são utilizadas para modelar probabilisticamente uma grande quantidade de processos e fenômenos.

3.6.1 Distribuição Uniforme

A distribuição Uniforme é considerada a mais simples dentre as distribuições contínuas, entretanto uma das mais importantes e utilizadas dentro da teoria de probabilidade. Através dela, a probabilidade de se obter um valor em um determinado intervalo contido no espaço amostral é

proporcional ao tamanho do próprio intervalo, ou seja, cada valor tem a mesma probabilidade de ser obtido.

A função densidade de probabilidade da distribuição Uniforme no intervalo fechado $[a, b]$ é dada pela Equação 3.11.

$$f(x) = \begin{cases} \frac{1}{b-a} & , \quad a \leq x \leq b \\ 0 & , \quad \text{caso contrário} \end{cases} \quad (3.11)$$

onde:

a = valor mínimo assumido pela função;

b = valor máximo assumido pela função.

A equação para amostragem de uma variável que segue a distribuição Uniforme pode-se empregar a técnica da Transformada Inversa, pois a *pdf* representada na Equação 3.11 é integrável, conforme a Equação 3.12 e a Equação 3.13.

$$F(x) = \int_a^b f(x)dx = \int_a^b \frac{1}{(b-a)}dx = \frac{1}{(b-a)}x \Big|_a^x = \frac{1}{(b-a)}(x-a) \quad (3.12)$$

Fazendo $F(x) = r$, na busca da variável aleatória x , chega-se a equação Equação 3.13:

$$x = r(b-a) + a \quad (3.13)$$

A Equação 3.13 é utilizada no fluxograma da Figura 3.5. Este, sugere a demonstração de um algoritmo para obtenção de números aleatórios de acordo com a distribuição Uniforme. Neste fluxograma, tem-se que:

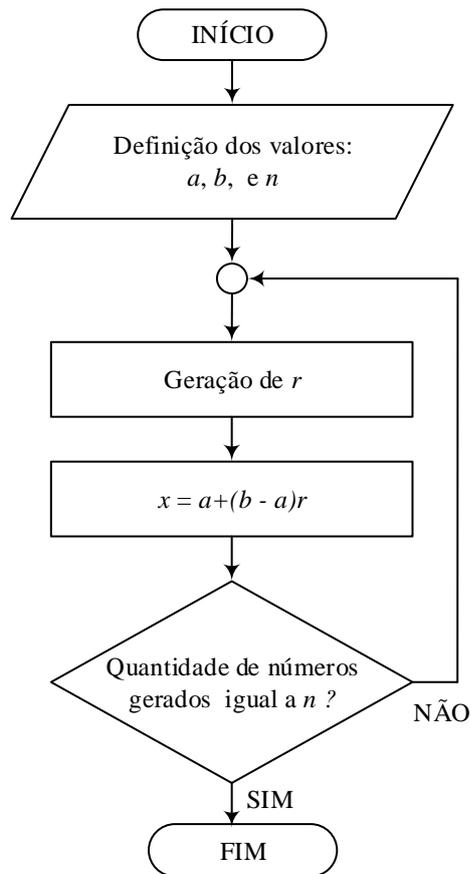
r = número aleatório;

x = variável aleatória segundo a distribuição Uniforme;

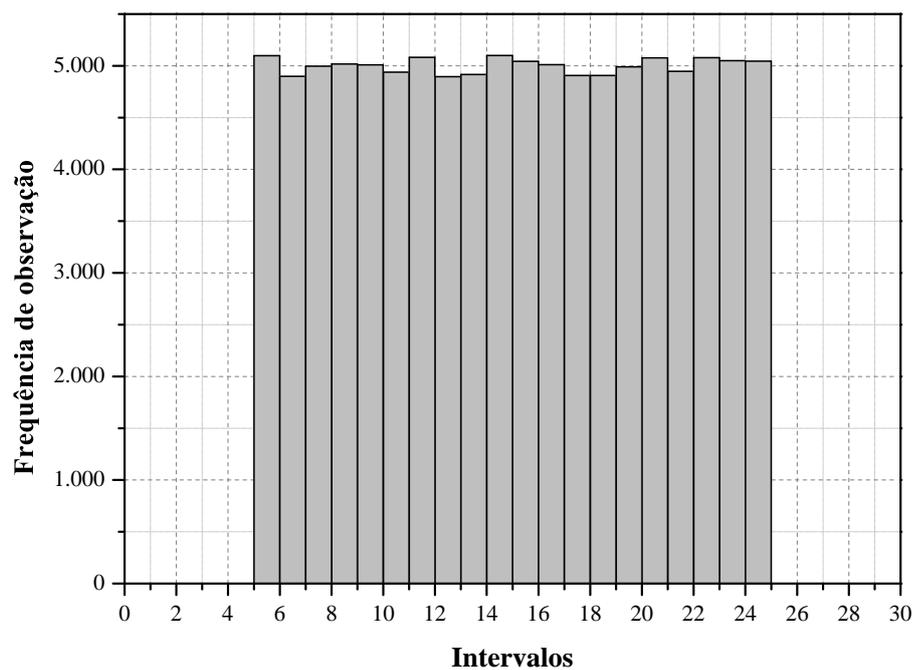
n = quantidade de números a serem gerados.

O gráfico da Figura 3.6 apresenta a divisão em vinte intervalos, considerando cem mil números aleatórios gerados segundo uma distribuição Uniforme. Seu valor mínimo é cinco e valor máximo vinte e cinco.

Figura 3.5 – Fluxograma de amostragem da distribuição Uniforme.



Fonte: Elaborado pelo autor.

Figura 3.6 – Distribuição Uniforme com $a = 5$ e $b = 25$.

Fonte: Elaborado pelo autor.

3.6.2 Distribuição Normal

As distribuições Normais, conhecidas como distribuições Gaussianas, são as mais importantes dentre as distribuições contínuas. As razões para tal consideração, está no fato de que são boas descrições de algumas distribuições de dados reais e, sobretudo, são boas aproximações para muitos tipos de resultados aleatórios.

Uma distribuição Normal é utilizada para descrever o comportamento de uma variável aleatória que flutua de forma simétrica em torno de um valor central, descrita por uma curva de densidade Normal. As curvas Normais são todas simétricas, em forma de sino, com um único pico, e são completamente caracterizadas pela média (μ) e desvio padrão (σ).

A média está localizada no centro da curva e coincide com a mediana. Enquanto que, o desvio-padrão da curva controla a dispersão da curva Normal, curvas com maior desvio padrão são mais dispersas (MOORE; NOTZ; FLIGNER, 2014).

Sua função densidade de probabilidade da distribuição Normal é representada pela Equação 3.14.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty \leq x \leq \infty \quad (3.14)$$

Considerando que a *pdf* da distribuição Normal não é integrável, utiliza-se a técnica de amostragem de Box-Müller. Nesta técnica tem-se que r_1 e r_2 são dois números aleatórios produzidos por um RNG, as variáveis aleatórias gaussianas X e Y de média μ e desvio padrão σ podem ser obtidas pela Equação 3.15 e Equação 3.16, respectivamente.

$$X = \cos(2\pi r_1) \sqrt{-2 \ln(r_2) \sigma_X^2} + \mu_X \quad (3.15)$$

$$Y = \sin(2\pi r_1) \sqrt{-2 \ln(r_2) \sigma_Y^2} + \mu_Y \quad (3.16)$$

onde:

μ = parâmetro de escala da distribuição (média);

σ = parâmetro de forma da distribuição (desvio padrão);

e = número neperiano (2,71828...).

Importante ressaltar que μ e σ representam parâmetros de escala e parâmetros de forma da distribuição, respectivamente. Entretanto, para a distribuição Normal estes parâmetros coincidem com a média (μ) e desvio padrão (σ).

O fluxograma da Figura 3.7 demonstra um algoritmo para amostragem da distribuição Normal, com base apenas na Equação 3.15:

r_1 = número aleatório 1;

r_2 = número aleatório 2;

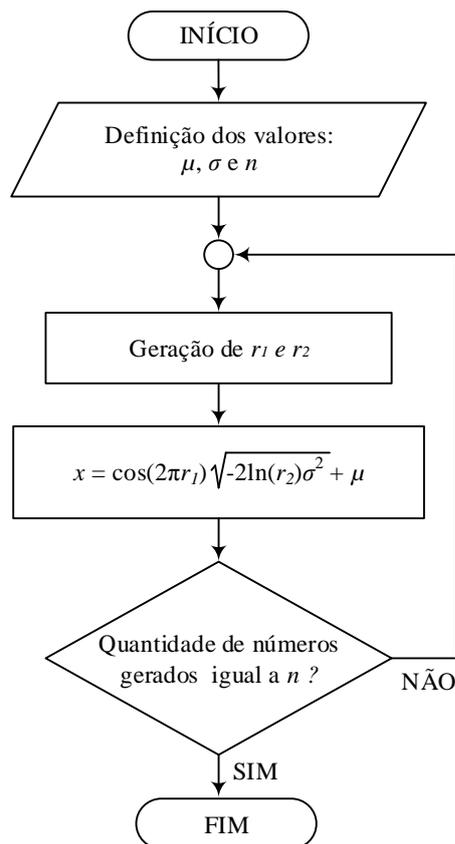
x = variável aleatória segundo a distribuição Normal;

n = quantidade de números a serem gerados;

μ = parâmetro de escala da distribuição (média);

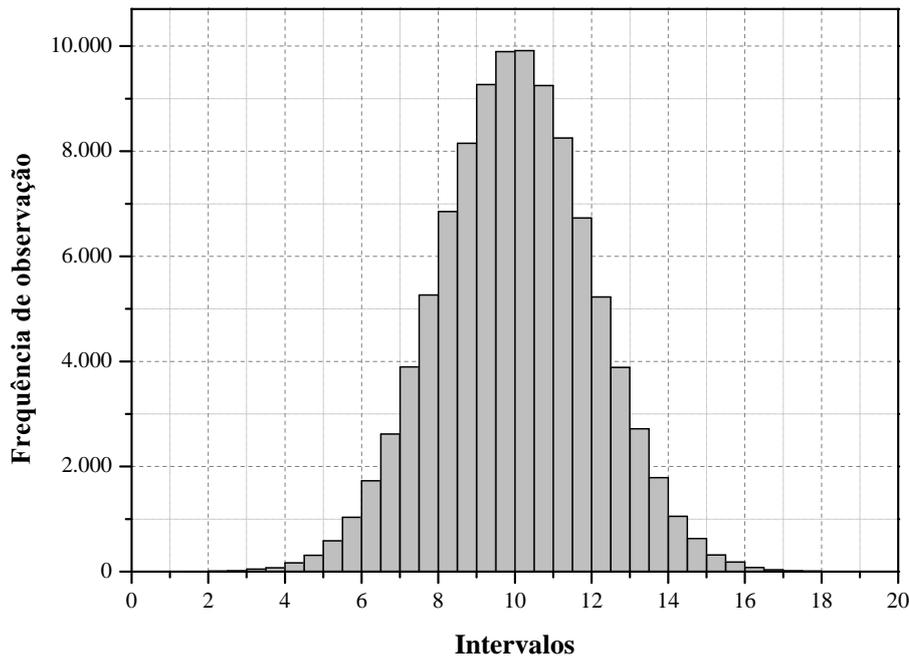
σ = parâmetro de forma da distribuição (desvio padrão).

Figura 3.7 – Fluxograma de amostragem da distribuição Normal.



Fonte: Produzido pelo autor.

O gráfico da Figura 3.8 apresenta a divisão em quarenta intervalos considerando cem mil números aleatórios gerados segundo a distribuição Normal. Para tal distribuição os parâmetros considerados são valor médio dez e desvio padrão dois.

Figura 3.8 – Distribuição Normal com $\mu = 10$ e $\sigma = 2$.

Fonte: Elaborado pelo autor.

3.6.3 Distribuição Log-Normal

A distribuição Log-Normal é uma distribuição de probabilidade contínua usada na representação de grandezas positivas, é frequentemente usada em modelagem estatística de fenômenos metrológicos. Essa distribuição é sempre positivamente assimétrica, significando que a extremidade longa é sempre para a direita. Isso força o valor mais provável ir para o lado esquerdo, assim a média está a sua direita.

Ao contrário do que o nome indica, não significa a densidade do logaritmo de uma variável normal, pois uma variável normal admite valores negativos, valores em que a função logaritmo não é definida. Uma variável aleatória com distribuição Log-Normal é encontrada tomando-se a exponencial de uma variável normal. A função densidade de probabilidade da distribuição Log-Normal é dada pela Equação 3.17.

$$f(x) = \begin{cases} \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}} & , x > 0 \\ 0 & , x \leq 0 \end{cases} \quad (3.17)$$

onde:

μ = parâmetro de escala da distribuição;

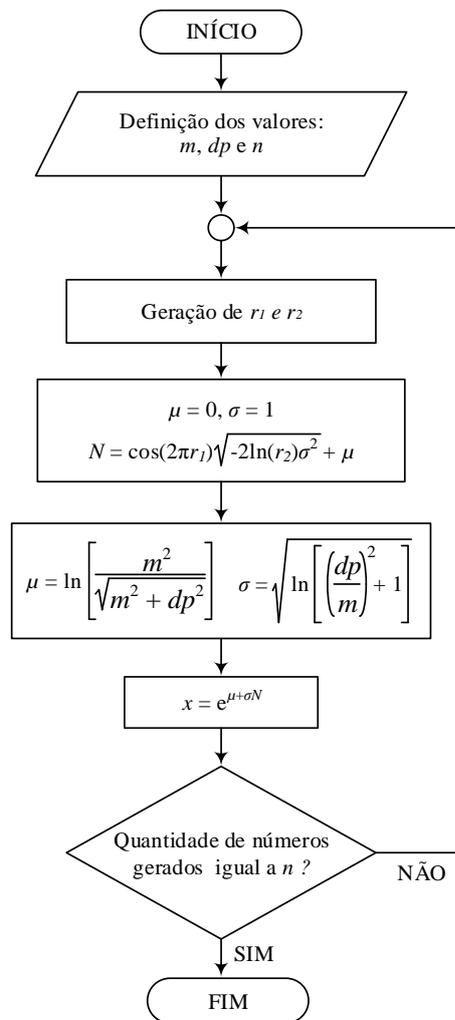
σ = parâmetro de forma da distribuição.

O logaritmo de uma variável com distribuição Log-Normal com parâmetros μ e σ é caracterizada por uma distribuição Normal com média m e desvio padrão dp . Os parâmetros μ e σ da distribuição Log-Normal são dados respectivamente pela Equação 3.18 e Equação 3.19.

$$\mu = \ln \left[\frac{m^2}{\sqrt{m^2 + dp^2}} \right] \quad (3.18)$$

$$\sigma = \sqrt{\ln \left[\left(\frac{dp}{m} \right)^2 + 1 \right]} \quad (3.19)$$

Figura 3.9 – Fluxograma de amostragem da distribuição Log-Normal.



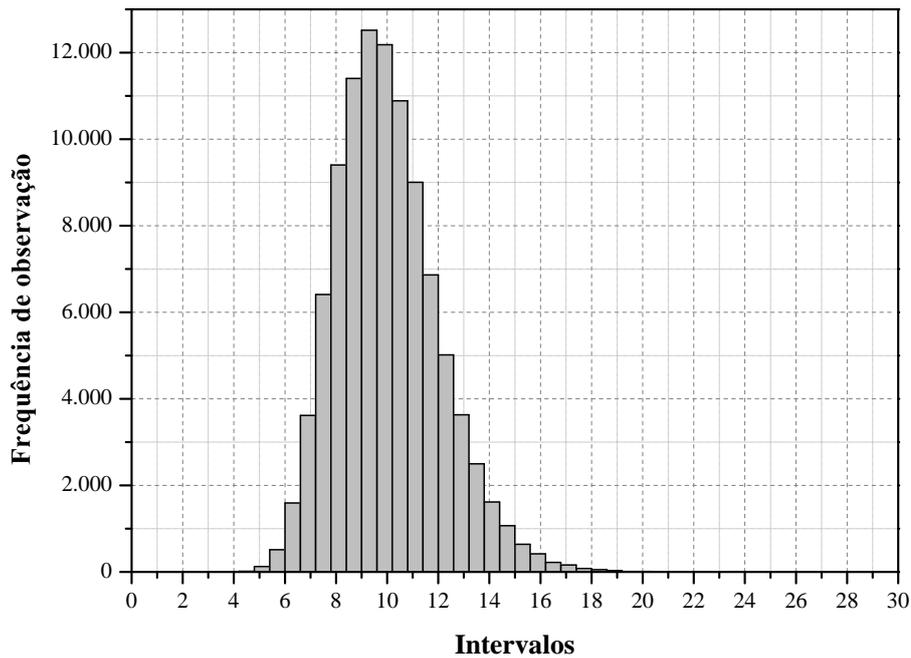
Fonte: Elaborado pelo autor.

O algoritmo para obtenção de números aleatórios segundo a distribuição Log-Normal é demonstrado no fluxograma da Figura 3.9, em que definiu-se:

- r_1 = número aleatório 1 gerado a partir da distribuição Normal com média zero e desvio padrão unitário;
- r_2 = número aleatório 2 gerado a partir da distribuição Normal com média zero e desvio padrão unitário;
- x = variável aleatória segundo a distribuição Log-Normal.

O gráfico da Figura 3.10 apresenta a divisão em cinquenta intervalos considerando cem mil números aleatórios gerados segundo a distribuição Log-Normal. Os parâmetros utilizados para essa distribuição são valor médio dez e desvio padrão dois.

Figura 3.10 – Distribuição Log-Normal com $m = 10$ e $dp = 2$.



Fonte: Elaborado pelo autor.

4 O ALTERNATIVE TRANSIENTS PROGRAM (ATP-EMTP)

Neste capítulo, apresenta-se um breve histórico sobre o desenvolvimento do ATP-EMTP, listando suas principais potencialidades para análise de transitórios em sistemas elétricos. Além disso, são apresentados dois pré-processadores gráficos existentes, que visam facilitar o processo de criação do arquivo de entrada do ATP-EMTP, com destaque especial para o software ATPDraw. Também são abordadas duas alternativas para incorporação de novas funcionalidades ao ATP-EMTP, que correspondem à TACS (*Transient Analysis of Control Systems*) e MODELS. Ao final, considerando o recurso oferecido em MODELS, são apontados as etapas relevantes a criação da interface de *foreign model*, e os procedimentos necessários para sua utilização.

4.1 BREVE HISTÓRICO

O desenvolvimento do *ElectroMagnetic Transients Program* (EMTP) teve início na *Technical University Munich*, Alemanha, na década de 1960, com o trabalho de doutorado (*Doktor-Ingenieur*) do professor Hermann W. Dommel. Nesta fase, o programa realizava apenas simulações de circuitos monofásicos e comportava um número reduzido de modelos. Em continuidade, seu desenvolvimento passou a ser custeado pela *Bonneville Power Administration* (BPA) que buscava um substituto digital para o analógico *Transient Network Analyzer* (TNA) (EPRI, 1986).

Em 1973, o professor Dommel foi para a *University of British Columbia* em Vancouver, Canadá. Com sua saída da BPA, duas versões do EMTP passaram a ser desenvolvidas separadamente, uma por Dommel, denominada EMTP-UBC (devido ao nome da universidade) e outra pela BPA, denominada EMTP-BPA. A versão EMTP-UBC foi aprimorada nos anos seguintes e em 1987 passou a ser denominada MicroTran[®]. Na BPA, o desenvolvimento do programa passou a ser coordenado pelo Dr. Scott Meyer, com colaboração da Dra. Tsu-Huei Liu.

Dr. Meyer estabeleceu um processo articulado de trabalho, reunindo contribuições de diversas universidades, fabricantes e concessionárias de energia elétrica, tornando o EMTP da BPA uma ferramenta poderosa para estudos de transitórios em sistemas elétricos. No entanto, a incorporação de ferramentas e modelos desenvolvidos por usuários acarretou a ocorrência de erros e carência de documentação formal.

Com intuito de estabelecer diretrizes para o desenvolvimento do programa e angariar fundos de outras organizações, fundou-se o *EMTP Development Coordination Group* (DCG) em 1982. O DCG originalmente era formado por três agências governamentais americanas: BPA, *U.S. Bureau of Reclamation* (USBR) e *Western Area Power Administration* (WAPA); e três organizações canadenses: *Hydro-Québec/IREQ*, *Ontario Hydro*, e *Canadian Electrical Association* (CEA).

Os membros do DCG inicialmente concordaram em manter o EMTP como domínio público, no entanto, com a associação do *Electric Power Research Institute* (EPRI) em 1984, o comitê iniciou uma tentativa de comercialização do programa. No entanto, esta ideia não era de consentimento dos Drs. Meyer e Liu e, tampouco agradou os grupos de usuários estrangeiros que haviam colaborado com o desenvolvimento do EMTP durante anos. Quando a proposta de comercialização foi oficializada, Dra. Liu em protesto, renunciou ao cargo de presidente do DCG, e Dr. Meyer começou a dedicar seu tempo extra para analisar alternativas. A “alternativa” encontrada foi o desenvolvimento de um novo programa, intitulado *Alternative Transients Program* (ATP) (MEYER; LIU, 1988).

Dr. Meyer iniciou o trabalho no ATP-EMTP formalmente em agosto de 1984, utilizando seu computador pessoal e contando com a ajuda do visitante chinês Ma Ren-ming, do *Wuhan High Voltage Institute*. Em menos de um ano, o ATP-EMTP já havia se tornado um novo programa distinto, com vantagens substanciais. Em novembro de 1985 ele foi apresentado na Europa, ocasião em que foi proposto o primeiro encontro anual do *Leuven EMTP Center* (LEC) com base no desenvolvimento cooperativo do EMTP não comercial. A adesão ao novo programa se consolidou 1987 com a realização do primeiro minicurso do ATP-EMTP, oferecido pelo LEC em Leuven, Bélgica, e a disponibilização da extensa documentação do programa denominada *ATP Rule Book* (1987-1998).

No encontro anual do LEC de 1987, firmou-se um acordo de cooperação admitindo grupos de usuários nas mesmas condições que organizações industriais. Este acordo fixou os termos pelos quais o ATP-EMTP seria disponibilizado gratuitamente, que incluem honestidade em todas as relações e não participação no desenvolvimento de versões comerciais do EMTP. Além disso, a habitual taxa de adesão foi extinta em troca de contribuições para o desenvolvimento do programa (MEYER; LIU, 1988). A política de distribuição do ATP-EMTP contribuiu para que o

programa se tornasse mundialmente utilizado para a análise de transitórios em sistemas elétricos.

O EMTP do DCG/EPRI continuou sendo desenvolvido até sua versão 3.0 de 1996 (EMTP96). Em 2003, foi lançada uma nova versão reestruturada, denominada EMTP-RV[®], desenvolvida sob a liderança técnica da Hydro-Québec. Atualmente, o EMTP-RV[®] não é considerado uma atualização, mas sim uma substituição completa do EMTP96.

4.2 POTENCIALIDADES DO ATP-EMTP

O ATP-EMTP conta atualmente com uma série de ferramentas e modelos de componentes para simulação de transitórios eletromagnéticos, eletromecânicos, e de controle em sistemas elétricos. Isto é resultado do processo de desenvolvimento colaborativo de vários anos e do empenho contínuo dos Drs. Meyer e Liu.

De acordo com o ATP Rule Book (1987-1998), dentre os modelos de equipamentos e componentes incorporados pelo ATP-EMTP, pode-se citar:

- Resistências, indutâncias e capacitâncias concentradas, desacopladas ou linearmente acopladas;
- Resistências e indutâncias concentradas variáveis;
- Cabos e linhas de transmissão com parâmetros distribuídos, dependentes ou não da frequência;
- Componentes não-lineares, como transformadores e para-raios;
- Chaves controladas por tempo, controladas por tensão e chaves com controle estatístico através do método de Monte Carlo;
- Válvulas, diodos e triacs;
- Fontes analíticas de tensão e corrente do tipo degrau, rampa, cossenoidal, surto e empírica;
- Máquinas síncronas trifásicas e máquina universal.

Sistemas de controle e componentes com características não-lineares podem ser modelados através de TACS ou MODELS, uma linguagem de programação própria do ATP-EMTP. Estas ferramentas podem ser utilizadas para a criação de novos componentes, sendo descritas brevemente na seção 4.4.

Para auxiliar a modelagem de alguns componentes complexos, o ATP-EMTP

disponibiliza rotinas de apoio que auxiliam a determinação de parâmetros do modelo a partir de dados de projeto ou de ensaio. Como exemplo, pode-se citar:

- Rotinas para determinação de parâmetros elétricos de linhas aéreas e cabos: LINE CONSTANTS, CABLE CONSTANTS e CABLE PARAMETERS;
- Rotinas para modelagem de linhas com parâmetros dependentes de frequência: Semlyen, J.Marti e Noda;
- Rotinas para modelagem de transformadores: XFORMER e BCTRAN;
- Rotinas para conversão de dados da curva de saturação e histerese.

Na realização da simulação no domínio do tempo, o ATP-EMTP emprega a regra de integração trapezoidal para solucionar numericamente o sistema de equações diferenciais. Condições iniciais não nulas podem ser determinadas através de uma solução em regime permanente ou inseridos diretamente pelo usuário. O programa também incorpora recursos para análise no domínio da frequência e de harmônicos através das opções FREQUENCY SCAN e HARMONIC FREQUENCY SCAN respectivamente.

4.3 PRÉ-PROCESSADORES GRÁFICOS PARA O ATP-EMTP

Devido a sua concepção, o arquivo de entrada do ATP-EMTP possui um formato rigidamente preestabelecido de alocação de dados. A montagem deste arquivo requer um bom conhecimento de sua estrutura, pois uma vez que a mesma não seja seguida, erros poderão ocorrer na leitura dos dados ou no processamento da simulação. Para minimizar erros como estes e facilitar a utilização do ATP-EMTP, dispõe-se atualmente de softwares pré-processadores gráficos compatíveis com o padrão “WYSIWYG” (“*What You See Is What You Get*”) dos modernos sistemas operacionais.

Softwares como estes fornecem uma interface gráfica para o ATP-EMTP, a partir da qual o usuário pode construir um circuito elétrico usando o “*mouse*”, selecionando componentes pré-definidos em uma “paleta”. A partir do desenho gráfico do circuito, o programa gera automaticamente o arquivo padrão de entrada e executa a simulação no ATP-EMTP.

O primeiro software pré-processador gráfico desenvolvido para o ATP-EMTP é denominado ATPDraw. De acordo com Prikler e Hoidalén (1998), uma versão bastante

rudimentar do software foi desenvolvida para fins educacionais no *Norwegian Institute of Technology* em 1991. Posteriormente, duas versões demonstrativas foram apresentadas nas reuniões anuais do LEC/EEUG (*European EMTP Users Group*) em 1991 e 1992. Nesta oportunidade, a BPA entrou em contato com os desenvolvedores com a intenção de financiar um aprimoramento no programa.

A primeira fase do projeto de desenvolvimento financiado pela BPA foi concluída em maio de 1994 e o programa foi disponibilizado para usuários através da internet, juntamente com seu manual. Na segunda etapa do projeto, concluída em dezembro de 1995, algumas limitações importantes da versão anterior foram solucionadas. Em seguida, o ATPDraw foi totalmente reescrito em linguagem de programação Delphi para ser executado nas versões de 16 bits dos ambientes operacionais Windows 3.x (HOIDALEN, 1997). Com apoio da BPA, o programa continua sendo aprimorado e hoje se encontra na versão 6.0, sendo compatível com os atuais sistemas operacionais Windows.

Atualmente o ATPDraw pode ser considerado o pré-processador gráfico mais utilizado por usuários do ATP-EMTP em ambientes Windows, sendo disponibilizado gratuitamente através da internet (HOIDALEN, 2016). Em sua versão 6.0, o software conta com uma vasta biblioteca de componentes, que inclui elementos lineares e não lineares, cabos, linhas de transmissão, transformadores, chaves, fontes, máquinas síncronas, máquinas de indução e diversos outros componentes.

O ATPDraw também conta com uma biblioteca especializada para sistemas elétricos de potência (*Power System Toolbox*), contendo modelos de cargas, filtros, relés de proteção, entre outros. Para a visualização de gráficos e pós-processamento dos resultados, o ATPDraw incorpora outros programas como o PlotXY.

O ATPDesigner[®], desenvolvido a partir de 2007, é outro software utilizado como pré-processador gráfico para o ATP-EMTP. No entanto, apenas uma versão “light” do programa, com algumas limitações em relação às funções e ao número de elementos, pode ser utilizada sem licença. Diferentemente do ATPDraw que possui uma aplicação generalista, esse software é mais voltado à simulação de sistemas elétricos de potência, integrando recursos para a representação de barramentos, disjuntores, relés de proteção, fusíveis, entre outros.

Além disso, o ATPDesigner[®] é capaz de ler e gravar arquivos no formato COMTRADE de acordo com a norma internacional IEC 60255-24 (Power Engs, 2016). Outra diferenciação deste pré-processador gráfico, diz respeito a disponibilidade de forma integrada, de recursos para a representação de gráficos de corrente, tensão, potência ativa, potência reativa, fator de deslocamento e métodos para análise de sinais.

4.4 ALTERNATIVAS PARA INCORPORAÇÃO DE NOVOS MODELOS

Considerando as formas de incorporação de novos modelos ao ATP-EMTP, o usuário dispõe dos recursos TACS e MODELS. A primeira delas se refere à transcrição de modelos utilizando a representação de funções de controle disponíveis no ATP-EMTP, denominadas TACS (*Transient Analysis of Control System*).

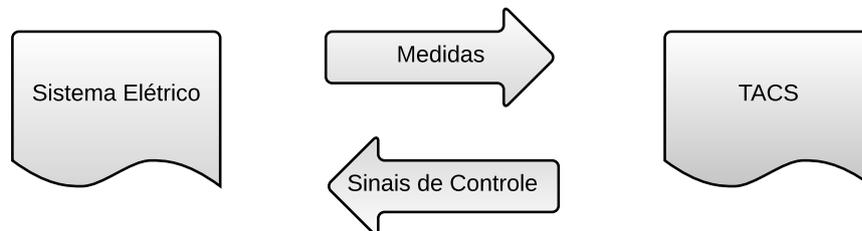
Originalmente, o recurso das TACS foi proposto para modelar conversores de corrente contínua em alta tensão (HVDC), e para modelar interações entre transitórios de sistemas e sinais de controle. Contudo, sua utilização pode ser mais ampla que o referido, podendo-se listar outras funcionalidades atribuídas às TACS:

- Realizar acionamento de relés;
- Inicializar circuitos da área de Eletrônica de Potência;
- Controlar loops para conversores elétricos de potência;
- Monitorar e pós-processar variáveis com aplicações diversas nas áreas de filtros digitais e analógicos;
- Cálculo de potências ativa e reativa;
- Cálculo do torque e do fluxo de motores;
- Aplicações com fontes de frequência variável;
- Projetos com respostas não lineares;
- Desenvolvimento de novos modelos(resistência de um arco elétrico, por exemplo);
- Simulação de sistemas mecânicos e eletromecânicos;

O circuito elétrico e os componentes TACS funcionam como sistemas separados, porém trabalham em conjunto trocando informações de variáveis. A representação desta interação está ilustrado na Figura 4.1. Basicamente, o ATP-EMTP analisa o sistema desenvolvido e processa

as TACS para cada intervalo de tempo, entretanto é importante considerar, que esse processo introduz atrasos no processamento.

Figura 4.1 – Relação entre modelos de TACS e o circuito projetado no ATP-EMPT.



Fonte: (FARIAS; GUTIERRES, 2012).

A segunda opção para inclusão de novos modelos, diz respeito à transcrição do modelo matemático para a linguagem especializada do ATP-EMTP com maior flexibilidade de sintaxe em relação às TACS, denominada MODELS. Esta corresponde a uma linguagem de programação própria do ATP-EMTP, por meio da qual novos componentes podem ser criados, além dos componentes previamente definidos. MODELS aceita também, a modelagem de componentes em que o usuário não é limitado a um conjunto pré-definido de componentes, podendo criar bibliotecas em função das suas necessidades.

Segundo Gabor Furst (1988), alguns objetivos que nortearam a elaboração de MODELS foram os seguintes:

- Possibilitar o desenvolvimento de modelos dos componentes e algoritmos de controle não disponíveis nas TACS;
- Proporcionar ao usuário a flexibilidade de utilização de uma linguagem de programação de alto nível, sem haver necessidade de interação em nível de programação do código fonte;
- Permitir que o usuário não só desenvolva o modelo do componente, mas também determine as suas condições iniciais;
- Possibilitar a utilização de uma interface com o ATP-EMTP, ao nível de modelagem, permitindo ao usuário a conexão de rotinas externas ao programa;

A estrutura da linguagem MODELS exige que o código do algoritmo seja preparado através do uso de algumas palavras-chaves. Sendo que a sequência apresentada nas instruções,

também é um requisito fundamental para o processamento das informações. De acordo com Gabor Furst (1988), tem-se uma breve descrição dos principais termos utilizados na linguagem MODELS:

- **MODEL - ENDMODEL:** entre esses dois termos será escrito o código em linguagem MODELS, sendo que MODEL indica o início do algoritmo e ENDMODEL o seu final;
- **USE - ENDUSE:** estes termos realizam a tarefa de ligação entre MODEL do ATP-EMTP e o MODEL que contém o algoritmo criado pelo usuário. A partir da identificação dada pelo usuário, o MODEL será referenciado. Esta é atribuída ao MODEL na sua execução em “USE...AS...”, dentro do comando EXEC - ENDEXEC.
- **COMMENT - ENDCOMMENT:** entre tais declarações é permitido incluir comentários, a outra maneira seria incluí-los entre duplos traços (- -);
- **DATA:** são os dados de entrada especificados pelo usuário;
- **INPUT:** permite que o usuário inclua uma grandeza externa que será repassada ao código a cada passo de integração.
- **OUTPUT:** disponibiliza o valor de uma variável do algoritmo para o circuito;
- **CONST:** indica a utilização de parâmetros constantes presentes no algoritmo;
- **VAR:** são todas as variáveis utilizadas no código, porém não devem constar as variáveis incluídas em INPUT, DATA e CONST;
- **TIMESTEP:** estabelece o passo de integração no qual a MODELS irá executar seus cálculos, pois o passo de integração do código pode ser diferente do passo executado no ATP-EMTP. Esta declaração não é obrigatória, e quando não for informada, o passo de execução do algoritmo será o mesmo do ATP-EMTP;
- **DELAY CELLS:** tamanho da memória que será utilizada para armazenar algumas variáveis;
- **FUNCTIONS:** permite a criação de funções no algoritmo;
- **HISTORY:** compreende no histórico das variáveis. Importante para informar o valor das variáveis antes de começar a execução da simulação. Esta declaração é utilizada entre as instruções EXEC e ENDEXEC;
- **INIT - ENDINIT:** permite que condições iniciais das variáveis possam ser inicializadas, além de ser possível informar o histórico das variáveis;

- **EXEC - ENDEXEC:** entre estas duas declarações, serão realizados os cálculos do algoritmo que dependem do tempo de simulação. Elas determinam o início e término da execução do algoritmo, respectivamente;
- **RECORD:** toda variável escrita em RECORD será visualizada no arquivo .PL4.

Mesmo possuindo tais ferramentas substanciais para a implementação de novos componentes, MODELS possui algumas desvantagens por apresentar certas limitações, na memória das suas dimensões para matrizes, no número de variáveis e nas linhas de programação. Em vista disso, a linguagem MODELS fornece o mecanismo da *foreign model* para expandir a sua capacidade e sua flexibilidade na programação (LUO; KEZUNOVIC, 2006).

4.5 CRIAÇÃO DO *FOREIGN MODELS* NO ATP-EMTP

Um *foreign model* se trata de uma interface pré-definida de procedimento de integração de modelos ou funções desenvolvidas em outras linguagens de programação, a exemplo de FORTRAN, C e C++, ou qualquer outra linguagem onde o compilador seja capaz de gerar arquivos-objetos compatíveis com o ATP-EMTP. As referências às novas funcionalidades desenvolvidas pelo usuário devem ser informadas dentro de um arquivo que compõe o código fonte do próprio ATP-EMTP.

Além disso, possui a capacidade de interagir com sinais de tensão, corrente, potência e energia gerados durante a simulação de qualquer caso no ATP-EMTP (NAKAGOMI et al., 2005). No entanto, para criar um *foreign model* é imprescindível adaptar o programa ATP-EMTP, ou seja, criar um novo arquivo executável personalizado, o qual deve ser composto do conteúdo original do ATP-EMTP acrescido das informações referentes às novas funcionalidades desenvolvidas.

O ATP-EMTP faculta a integração de modelos externos desenvolvidos em linguagem de alto nível, no entanto, é preciso salientar que somente usuários autorizados podem obter os arquivos-objetos para a reconstrução do núcleo do ATP-EMTP, uma vez que estes são mantidos em domínios privados na internet, pertencentes a grupos de usuários licenciados.

Segundo Nakagomi et al. (2005), pode-se identificar as seguintes indicações e restrições para a utilização de um *foreign model* no ATP-EMTP:

1. Na intenção de se utilizar o pré-processador gráfico ATPDraw, é necessário que a função tenha no máximo 12 portas de entradas e saídas, a fim de respeitar a dimensão limitada do elemento MODEL deste software;
2. Deve-se evitar dividir o programa na linguagem utilizada em pequenos programas e bibliotecas. É indicado manter todas as funções e definições dentro do mesmo arquivo;
3. O programa a ser desenvolvido em uma linguagem de alto nível não deve conter uma função principal;
4. É importante salientar que cada *foreign model* deve conter as seguintes estruturas obrigatórias: uma função de inicialização, executada somente uma vez no começo da simulação e outra função de execução, que será executada a cada passo de integração da simulação previamente configurada no ATP-EMTP.
5. A fim de permitir o transporte de parâmetros de sinais entre a simulação e as novas funções, deve-se declarar os seguintes vetores do tipo *double*:
 - **Vetor de dados:** representa o vetor de variáveis que guardam valores fixos, ou seja, que não mudam durante a simulação. Atuam como parâmetros de configuração/inicialização;
 - **Vetor de entrada:** representa o vetor de variáveis de entrada da função implementada. Estas variáveis guardam os valores dos sinais fornecidos pela simulação;
 - **Vetor de saída:** representa o vetor de variáveis da saída da função implementada. Estas variáveis possuem os valores resultantes dos cálculos realizados dentro da nova funcionalidade que serão passados da simulação;
 - **Vetor de variáveis:** representa o vetor de variáveis locais da nova função a ser implementada. Verificou-se que o valor mínimo a ser declarado deve ser o mesmo quantidade de portas de saída.

Outro importante requisito a ser atendido para a utilização de um *foreign model* é que o compilador FORTRAN, a ser empregado no processo de compilação e vinculação do código-fonte, seja o mesmo utilizado originalmente para a criação do ATP-EMTP (compilador

GNU FORTRAN versão 2.95.2). Este compilador também é disponibilizado nos sites de domínios privados de grupos de usuários licenciados do ATP-EMTP.

Após desenvolvido o algoritmo na linguagem de alto nível escolhida, é necessário alterar o arquivo que permite reconstruir o núcleo do ATP-EMTP, denominado *fgnmod.f*, que corresponde a um fragmento do código-fonte original do ATP-EMTP na linguagem FORTRAN. As alterações necessárias compreendem a adição de chamadas às funções de inicialização e execução, e uma identificação do *foreign model* que será utilizada para acessá-las durante a simulação.

Após preparado o arquivo *fgnmod.f*, deve-se compilá-lo juntamente com os demais arquivos do código-fonte do ATP-EMTP utilizando o compilador GNU FORTRAN versão 2.95.2. Deve-se compilar também o arquivo contendo a nova funcionalidade a ser incorporada. Após todos os arquivos convertidos para arquivos-objetos, deve-se finalmente realizar o processo de vinculação dos mesmos para a criação de um novo arquivo do ATP-EMTP, originalmente denominado de *TPBIG.exe*.

No caso em que for utilizado compiladores do pacote MinGW (*Minimalistic GNU for Windows*) para a compilação do código fonte das novas funções, a GNU disponibiliza uma ferramenta apropriada para executar as tarefas de compilação e vinculação de código chamada *make.exe*. Esta executa um conjunto de instruções contidas em um arquivo denominado *Makefile*, no qual é possível listar os arquivos-objeto obrigatórios que fazem parte do código-fonte original, especificar os nomes de todos os arquivos-objetos a serem adicionados para gerar o novo executável e vinculá-los automaticamente com bibliotecas do núcleo funcional do ATP-EMTP.

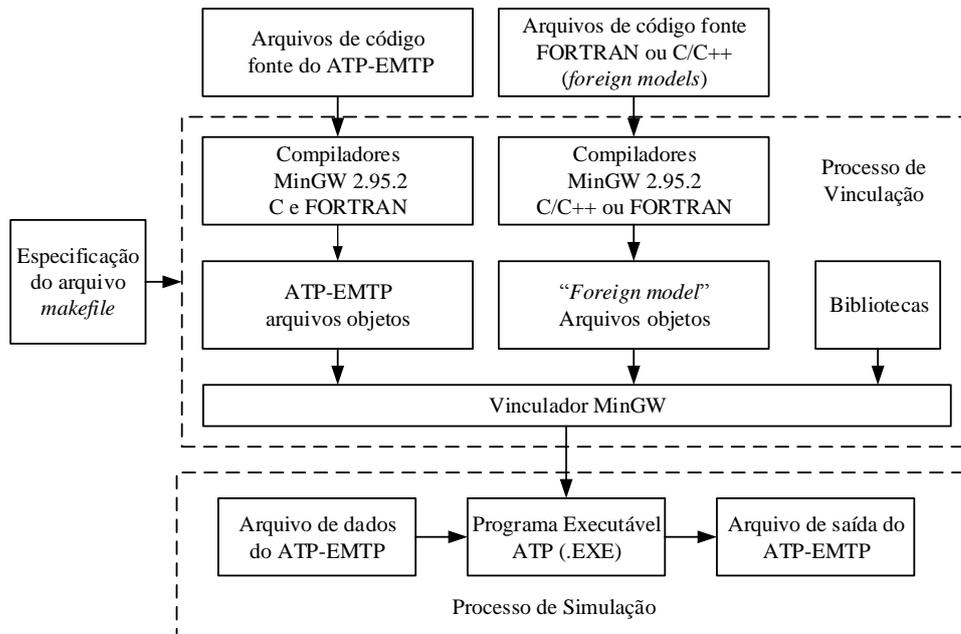
As extensões dos arquivos são identificadas através de *flags* e os devidos compiladores são executados diretamente pelos comandos do *Makefile*. O processo de compilação, vinculação e também de simulação utilizando o ATP-EMTP personalizado é mostrado na Figura 4.2.

A ferramenta *make.exe* pode ser executada diretamente através do prompt de comando do *Windows*, informando-se o diretório de origem do *Makefile*. Outra alternativa é a utilização do recurso oferecido pelo gerenciador ATP Launcher, através do comando “Make Tpbig.exe”, conforme ilustra a Figura 4.3.

Neste caso, ressalva-se que o nome dado ao novo arquivo executável do ATP-EMTP deve ser *mytpbig.exe*, uma vez que o ATP Launcher executa uma rotina de verificação ao final do processo de vinculação, conferindo se um executável com esse nome que foi criado. Uma vez

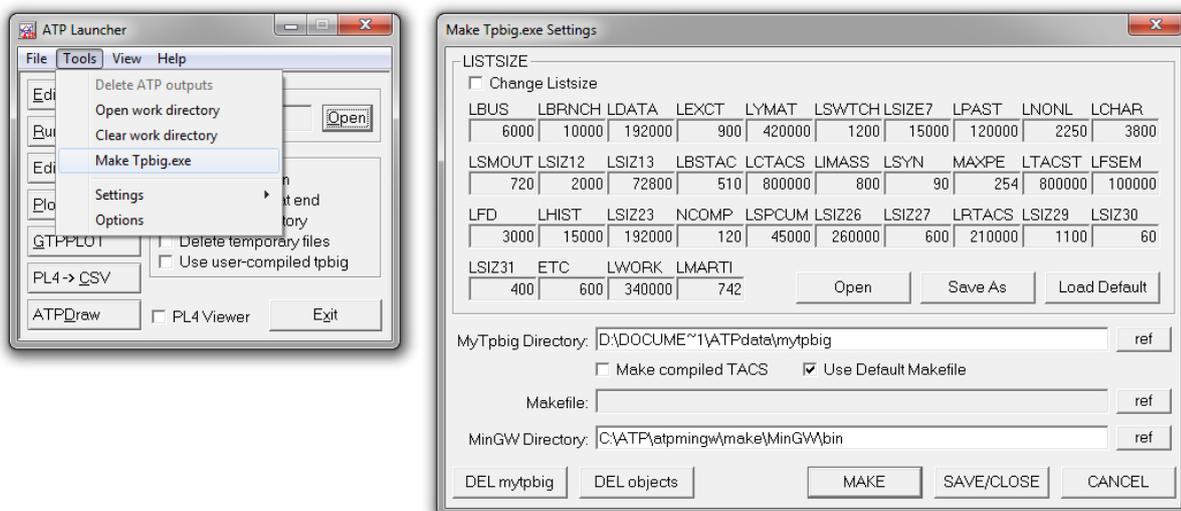
criada a versão personalizada do ATP-EMTP com as novas funcionalidades incorporadas como *foreign models*, pode-se utilizá-las em uma simulação.

Figura 4.2 – Processo de compilação, vinculação e simulação com o ATP-EMTP personalizado.



Fonte: Adaptado de (LUO; KEZUNOVIC, 2006).

Figura 4.3 – Recurso Make TPBIG.exe do ATP Launcher.



Fonte: Elaborado pelo autor.

4.5.1 Utilização de *foreign models*

Conforme descrito anteriormente, a possibilidade de criação de *foreign models* é um recurso oferecido pela linguagem MODELS para expandir sua capacidade. Desta forma, o acesso as novas funções implementadas deve ser realizado através desta linguagem.

Para facilitar esta tarefa, pode-se utilizar o ATPDraw, um dos pré-processadores gráficos para o ATP-EMTP citados no seção 4.3. Este software oferece um recurso que auxilia a criação de códigos em linguagem MODELS, bem como o intercâmbio de grandezas de entrada e saída com a rede elétrica simulada. O Quadro 4.1 apresenta a estrutura de código utilizada para cessar um *foreign model* em uma simulação via ATPDraw, onde:

- M1 – Identificação do model gerenciado pela interface gráfica do ATPDraw;
- M2 – Identificação do model criado para executar o *foreign model* M3;
- M3 – Identificação do *foreign model* utilizado, atribuída no arquivo *fgnmod.f*;
- D1, D2, D3, ..., Dn – Dados informados através da interface gráfica do model M1 e repassados ao model M2;
- I1, I2, I3, ..., In – Grandezas de entrada do model M1, obtidas da rede elétrica simulada, e repassadas ao model M2;
- O1, O2, O3, ..., On – Grandezas de saída do model M1, atualizadas pelo model M2, e repassadas a rede elétrica simulada;
- V1, V2, V3, ..., Vn – Variáveis declaradas no model M1 e utilizadas no *foreign model* M3. O número de variáveis de ser, no mínimo, igual ao número de saídas. Estas variáveis podem ser inicializadas no escopo do comando INIT / ENDINIT;

Quadro 4.1 – Código em linguagem MODELS para a utilização de *foreign models* no ATPDraw.

```

MODEL M1
DATA
  D1, D2, D3, ..., Dn
INPUT
  I1, I2, I3, ..., In
OUTPUT
  O1, O2, O3, ..., On
VAR
  V1, V2, V3, ..., Vn
INIT
  V1:=X1
  V2:=X2
  V3:=X3
  Vn:=Xn
ENDINIT
MODEL M2 FOREIGN M3{ixdata:nD, ixin:nI, ixout:nO, ixvar:nV}
EXEC
  USE M2 AS M2
    DATA
      xdata[1..nD]:=[D1, D2, D, ..., Dn]
    INPUT
      xin[1..nI]:=[I1, I2, I3, ..., In]
    HISTORY
      xvar[1..nO]:=[Y1, Y2, Y3, ..., Yn]
    OUTPUT
      O1:=xout[1]
      O2:=xout[2]
      O3:=xout[3]
      On:=xout[n]
    ENDUSE
  ENDEXEC
ENDMODEL

```

Fonte: Elaborado pelo autor.

5 METODOLOGIA

Este capítulo apresenta a metodologia utilizada para simulação estocástica de transitórios eletromagnéticos no ATP-EMTP. Inicialmente, destaca-se a seleção dos eventos precursores dos fenômenos a serem contemplados no presente trabalho. Além disso, aborda-se o processo de desenvolvimento de implementação da metodologia no ATP-EMTP através de *foreign models*, demonstrando como a mesma pode ser utilizada através do pré-processador gráfico ATPDraw.

5.1 CONSIDERAÇÕES GERAIS

Conforme exposto no capítulo 1, o objetivo principal deste trabalho é facilitar a realização de simulações estocásticas de eventos precursores de transitórios eletromagnéticos em subestações diretamente no ATP-EMTP, sem a necessidade de ferramentas externas para gerenciamento de simulações ou pós-processamento dos resultados. Desta forma, a primeira ação realizada foi a definição de quais eventos seriam contemplados. A partir da revisão da literatura apresentada no capítulo 2, onde buscou-se identificar as fontes de transitórios eletromagnéticos mais frequentes e relevantes em subestações, estabeleceu-se as seguintes prioridades de estudo:

- Ocorrência e eliminação de curtos-circuitos;
- Energização de transformadores de potência;
- Manobras de fechamento e abertura de disjuntores;
- Descargas atmosféricas.

Nos estudos de curtos-circuitos, a metodologia desenvolvida suporta a simulação estocástica de curtos-circuitos trifásicos, bifásicos e monofásicos, envolvendo ou não a terra, com e sem impedância de falta. Além disso, possibilita a abertura tripolar ou monopolar de disjuntores com dispersão entre polos para isolar a falta. Para análise de resultados deste estudo, a ferramenta efetua o cálculo da corrente eficaz verdadeira em decorrência do curto-circuito e detecta o valor máximo da tensão fase-terra gerada pela abertura de disjuntores em qualquer ponto do sistema. Os pontos (“nós”) do sistema em que estas grandezas são monitoradas podem ser definidos a critério do usuário. O mesmo pode ser realizado nos demais tipos de estudo.

Com relação a energização de transformadores de potência, a ferramenta implementada permite a simulação estocástica de energização individual e solidária destes equipamentos. Estes procedimentos são simulados mediante o fechamento tripolar de disjuntores, no qual também pode ser considerada a dispersão entre polos. Para a análise de resultados, a ferramenta proporciona a medida do valor eficaz verdadeiro da corrente de energização do transformador.

Embora os dois estudos descritos requeiram a operação de disjuntores, devido a sua importância, foram tratados separadamente dos demais eventos de manobra, tais como: energização e desenergização de linhas de transmissão; conexão e desconexão de cargas, reatores e capacitores *shunt* para compensação de reativos; entre outras. Para eventos como estes, a metodologia comporta a simulação estocástica de operações de fechamento e abertura tripolar, bipolar e monopolar de disjuntores, com dispersão entre polos. As máximas tensões fase-terra produzidas tanto no fechamento como na abertura dos disjuntores são detectadas e disponibilizadas para análise.

Por fim, mas não menos importante, a metodologia possibilita a simulação estocástica de transitórios eletromagnéticos oriundos de descargas atmosféricas. Para este tipo de estudo, permite-se a representação da aleatoriedade dos parâmetros de modelagem do fenômeno e do instante de sua incidência no sistema. Como recursos para análise de resultados, são medidas as máximas tensões fase-terra provocadas após cada evento simulado. A Tabela 5.1 apresenta um resumo com as grandezas analisadas nas simulações estocásticas de cada tipo de estudo contemplado neste trabalho.

Tabela 5.1 – Relação das grandezas analisadas em cada tipo de estudo.

Tipo de estudo	Grandezas analisadas
Curto-circuito	Corrente e tensão
Energização de TR	Corrente
Manobra de disjuntor	Tensão
Descarga atmosférica	Tensão

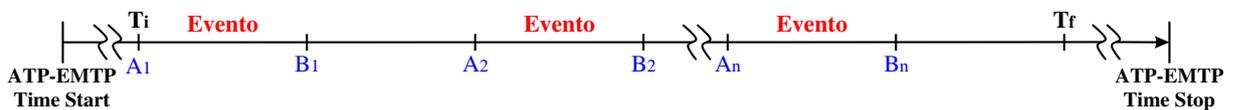
Fonte: Elaborado pelo autor.

5.2 CONCEPÇÃO DA METODOLOGIA

Para cada tipo de estudo destacado na seção 5.1, a metodologia desenvolvida permite a simulação em massa de eventos e a análise estatística de resultados das grandezas monitoradas. No entanto, a realização de diversas simulações careceria da utilização de ferramentas externas e da intervenção do usuário no gerenciamento dos arquivos de entrada e saída do ATP-EMTP.

Em vista disso, optou-se por reproduzir todos os eventos de determinado tipo de estudo em uma mesma simulação. Para facilitar a compreensão desta metodologia, pode-se acompanhar sua descrição através da Figura 5.1.

Figura 5.1 – Geração de eventos na simulação.



Fonte: Elaborado pelo autor.

Os instantes de tempo “ T_i ” e “ T_f ”, destacados na Figura 5.1, marcam respectivamente o início e o término da geração de eventos na simulação. Estes instantes são definidos pelo usuário, podendo ou não coincidir com o início (*Time Start*) e com o término (*Time Stop*) da simulação no ATP-EMTP.

Nos subintervalos compreendidos entre “ A_i ” e “ B_i ”, o evento é reproduzido em um instante aleatório, ou seja, cada instante discreto de tempo dentro deste subintervalo tem a mesma probabilidade para a ocorrência do evento. Isto é obtido graças a utilização do MMC com distribuição de probabilidade Uniforme entre os limites “ A_i ” e “ B_i ”. Já nos subintervalos compreendidos entre “ B_i ” e “ A_{i+1} ”, a metodologia prevê um tempo de espera para que as respostas transitórias do sistema aos eventos não se sobreponham, inviabilizando a análise dos resultados. Este procedimento é adotado na simulação de eventos em todos os estudos contemplados.

Embora a metodologia desenvolvida preveja um tempo para acomodação dos transitórios gerados após cada evento, sua performance é dependente do intervalo de simulação e do número de eventos definidos pelo usuário. Em vista disso, a Tabela 5.2 sugere valores para a parametrização da simulação, assumindo que os instantes “ T_i ” e “ T_f ” coincidam com o início e término da simulação no ATP-EMTP, respectivamente. Estes valores foram determinados com base em várias simulações e análises realizadas durante o desenvolvimento do presente trabalho.

Tabela 5.2 – Parâmetros de simulação recomendados.

Tipo de estudo	dT	$Tmax$
Curto-circuito	1×10^{-4} s	0,5 s para cada evento simulado
Energização de TR	1×10^{-5} s	1,0 s para cada evento simulado
Manobra de disjuntor	1×10^{-5} s	0,2 s para cada evento simulado
Descarga atmosférica	1×10^{-7} s	0,03 s para cada evento simulado

Fonte: Elaborado pelo autor.

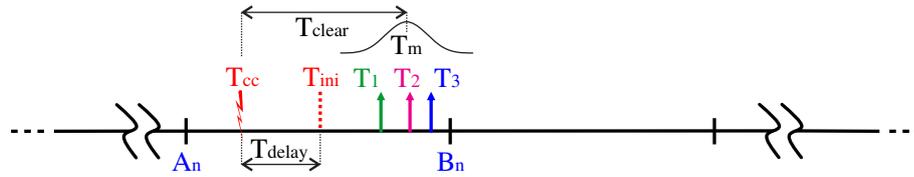
Os valores definidos para o passo de integração (dT) de cada estudo visam reduzir o tempo de simulação, porém, mantendo representatividade dos resultados. Já os valores de tempo total de simulação ($Tmax$) buscam garantir a extinção (ou atenuação significativa) da resposta transitória de um evento antes da geração do subsequente.

Após a simulação de cada evento, a ferramenta desenvolvida monitorando, medindo e armazenando o valor das grandezas analisadas de acordo com o tipo de estudo abordado. Na última iteração da simulação, efetua-se o cálculo destas grandezas em valor por unidade (p.u.) e a divisão em intervalos de frequência, exportando o resultado através de um arquivo em formato texto (codificação ASCII). Desta forma, nenhuma ferramenta de pós-processamento precisa ser utilizada para análise dos resultados. Os procedimentos específicos da metodologia para cada tipo de estudo são abordados nas próximas seções deste capítulo.

5.3 SIMULAÇÃO ESTOCÁSTICA DE CURTOS-CIRCUITOS

Neste tipo de estudo, o curto-circuito simulado ocorre em um instante aleatório definido pelo MMC com distribuição Uniforme em cada subintervalo limitado por “ A_i ” e “ B_i ”. Este instante é representado por “ T_{cc} ” na Figura 5.2.

Figura 5.2 – Geração de eventos de curto-circuito e abertura de disjuntores na simulação.



Fonte: Elaborado pelo autor.

A corrente eficaz de uma fase em qualquer ponto do sistema definido pelo usuário, em virtude do curto-circuito, é determinada por intermédio da Equação 5.1. O número de pontos por ciclo N é obtido em função do passo de integração empregado na simulação e da frequência nominal do sistema informada pelo usuário. A integral discreta é calculada através do método de integração trapezoidal.

$$I_{ef} = \sqrt{\frac{1}{N} \sum_{n=1}^N I_n^2} \quad (5.1)$$

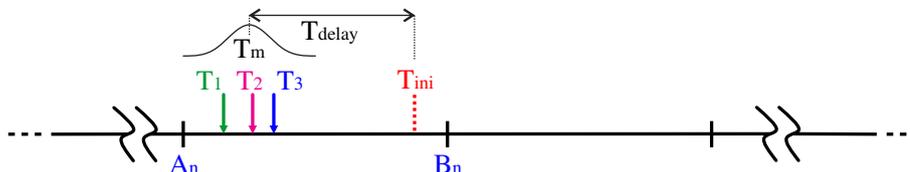
A Equação 5.1 é aplicada em um ciclo da frequência nominal do sistema, medido a partir do instante “ T_{ini} ” representado na Figura 5.2. Este é determinado a partir do tempo “ T_{delay} ”, informado pelo usuário, contado a partir o instante de ocorrência do curto-circuito “ T_{cc} ”. Esta opção foi adicionada caso o usuário queira mensurar a corrente eficaz após a atenuação da componente contínua de curto-circuito.

Já a abertura do disjuntor ocorre no instante “ T_m ”, representado na Figura 5.2. Este é determinado pelo tempo “ T_{clear} ”, também definido pelo usuário, contado a partir o instante de ocorrência do curto-circuito. O tempo “ T_{clear} ” deve atender a condição $T_{clear} \geq T_{delay} + 1/\text{freq}$ (onde freq corresponde a frequência nominal do sistema), para que a corrente eficaz possa ser calculada antes da abertura do disjuntor. Caso o usuário opte por incluir a dispersão entre polos na abertura do disjuntor, a ferramenta empregará o MMC com distribuição Normal para determinar os instantes de operação de cada polo, tomando “ T_m ” como tempo médio.

5.4 SIMULAÇÃO ESTOCÁSTICA DE ENERGIZAÇÕES DE TRANSFORMADOR

Neste tipo de estudo, o instante de fechamento do disjuntor para energização do transformador, representado por “ T_m ” na Figura 5.3, ocorre em um instante aleatório definido pelo MMC com distribuição Uniforme em cada subintervalo entre “ A_i ” e “ B_i ”. Novamente, caso o usuário opte por considerar a dispersão entre polos no fechamento do disjuntor, a ferramenta empregará o MMC com distribuição Normal para determinar os instantes de operação de cada polo, tomando “ T_m ” como tempo médio.

Figura 5.3 – Geração de eventos de energização de transformador na simulação.



Fonte: Elaborado pelo autor.

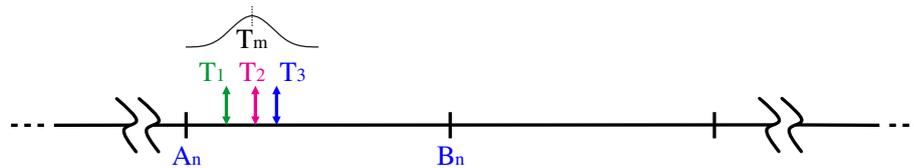
O cálculo da corrente eficaz de energização do transformador é realizado em um ciclo da frequência nominal do sistema, de forma semelhante ao adotado no estudo de curto-circuito. O instante de início deste cálculo é representado por “ T_{ini} ” na Figura 5.3, determinado a partir do tempo “ T_{delay} ” informado pelo usuário. A fase e o ponto do sistema onde esta grandeza será monitorada também podem ser definidos pelo usuário.

5.5 SIMULAÇÃO ESTOCÁSTICA DE MANOBRAS

Nos estudos de manobra, as operações de fechamento e abertura do disjuntor ocorrem intercaladamente na simulação, de acordo com o número total de manobras definido pelo usuário. Neste estudo, o disjuntor inicia a simulação na posição fechada, de modo que a primeira operação corresponde a uma abertura. Desta forma, caso um número ímpar de eventos seja definido, uma simulação de abertura a mais que o de fechamentos será realizada.

Em ambos os casos, a operação do disjuntor ocorre em um instante aleatório definido pelo MMC com distribuição Uniforme em cada subintervalo entre “ A_i ” e “ B_i ”. Este instante é representado por “ T_m ” na Figura 5.4.

Figura 5.4 – Geração de eventos de manobras de disjuntor na simulação.



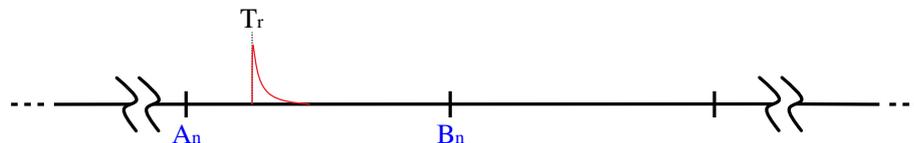
Fonte: Elaborado pelo autor.

Assim como nos demais estudos que envolvem operação de disjuntor, o usuário pode incluir a dispersão entre polos. Neste caso, a ferramenta empregará o MMC com distribuição Normal para determinar os instantes de fechamento e abertura de cada polo, tomando “ T_m ” como tempo médio.

5.6 SIMULAÇÃO ESTOCÁSTICA DE DESCARGAS ATMOSFÉRICAS

Nos estudos de descargas atmosféricas, o instante de ocorrência deste evento é definido pelo MMC com distribuição Uniforme em cada subintervalo entre “ A_i ” e “ B_i ”. Este instante é representado por “ T_r ” na Figura 5.5.

Figura 5.5 – Geração de eventos de descargas atmosféricas na simulação.



Fonte: Elaborado pelo autor.

Neste tipo de estudo, além do instante de incidência da descarga atmosférica no sistema, as aleatoriedades inerentes aos parâmetros de modelagem do fenômeno também são consideradas. Os parâmetros utilizados para representar uma descarga atmosférica na simulação correspondem a corrente de pico, ao tempo de frente e ao tempo de meia cauda. Para reproduzir a variação estatística destes parâmetros, emprega-se o MMC com distribuição de probabilidade Log-normal.

5.7 IMPLEMENTAÇÃO DA METODOLOGIA

Dentre as alternativas para criação de novas funcionalidades no ATP-EMTP listadas no capítulo 4, selecionou-se o recurso de *foreign models* para a incorporação da metodologia desenvolvida. Esta escolha se deve a maior flexibilidade obtida através dos *foreign models*, que

podem ser implementados em linguagem de programação de alto nível, facilitando ações como a armazenagem de resultados em arquivo.

Além disso, uma linguagem de alto nível apresenta melhor desempenho computacional quando comparada a outros recursos comportados pelo ATP-EMTP para a criação de novos modelos, como a linguagem MODELS, o que contribui para a redução do tempo total de simulação. Esta é uma característica relevante para a aplicação da metodologia, visto que todos os eventos de cada tipo de estudo serão reproduzidos em uma mesma simulação.

Para a implementação propriamente dita, empregou-se a linguagem de programação C++. Esta é uma das linguagens compatíveis com a criação de *foreign models* no ATP-EMTP, pois seus compiladores possibilitam a criação de arquivos-objetos a partir do código fonte.

Outra questão importante que levou a escolha da linguagem C++ diz respeito ao algoritmo RNG utilizado no presente trabalho. Este algoritmo, empregado na geração de números aleatórios para o MMC, é disponibilizado no livro *Numerical Recipes* (PRESS et al., 2007) diretamente nesta linguagem de programação.

Uma pequena modificação foi realizada no código deste RNG, que originalmente foi concebido na forma de uma estrutura de dados, para que o valor inicial da sequência aleatória (“semente”) pudesse ser definido através de uma função. Este é apresentado no Apêndice A juntamente com o código fonte das funções de inicialização e execução dos *foreign models* implementados.

As referências para as funções incorporadas como *foreign models* foram adicionadas na subrotina *fgnmod.f* disponibilizada no Apêndice B. Estas referências, apresentadas na Tabela 5.3, devem ser utilizadas na simulação no ATP-EMTP para acessar as novas funções de acordo com o tipo de estudo realizado.

Tabela 5.3 – Referências dos *foreign models* incorporados ao ATP-EMTP.

Tipo de estudo	<i>foreign models</i>
Curto-circuito	“SHORTCIRCUIT”
Energização de TR	“INRUSH”
Manobra de disjuntor	“SWITCHING”
Descarga atmosférica	“LIGHTNING”

Fonte: Elaborado pelo autor.

No processo de compilação e vinculação do código fonte das novas funções aos demais arquivos e bibliotecas do ATP-EMTP, empregou-se os compiladores do pacote GNU MingW, versão 2.95.2. Para facilitar a realização desta tarefa, utilizou-se o recurso Make TPBIG.exe do ATP Launcher, descrito no capítulo 4. O arquivo *Makefile* empregado é disponibilizado no Apêndice C. Na sequência, descreve-se o procedimento para utilização dos *foreign models* implementados em simulações estocásticas no ATP-EMTP via interface gráfica do ATPDraw.

a) *Foreign model* “SHORTCIRCUIT”

Para a realização de simulações estocásticas de curtos-circuitos através da metodologia desenvolvida, deve-se empregar o *foreign model* denominado “SHORTCIRCUIT”. No ATPDraw, isto pode ser realizado inserindo o elemento “*Default model*” e importando o código em linguagem MODELS apresentado no Quadro 5.1.

Os parâmetros do tipo **DATA** apresentados no Quadro 5.1 correspondem a:

- Seed: valor inicial da sequência aleatória do MMC (valor predefinido: 97);
- NumSC: número de curtos-circuitos simulados (valor predefinido: 100);
- PhaseSC: fases envolvidas no curto-circuito (valor predefinido: 123):
 - 1 = Fase A;
 - 2 = Fase B;
 - 3 = Fase C;
- PhaseOP: fases de abertura do disjuntor (valor predefinido: 123):
 - 1 = Fase A;
 - 2 = Fase B;
 - 3 = Fase C;
- Tdelay: tempo para o início do cálculo do valor eficaz da corrente após o curto-circuito (valor predefinido: 0,0 s);
- Tclear: tempo para abertura do disjuntor após a ocorrência do curto-circuito (valor predefinido: 0,1 s);
- Td: tempo de dispersão entre polos na abertura do disjuntor (valor predefinido: 5,0E-3 s);
- Vb: tensão base do sistema no ponto de análise (valor predefinido: 230E3 V);
- Sb: potência base do sistema (valor predefinido: 100E6 VA);

Quadro 5.1 – MODEL para acessar o *foreign model* “SHORTCIRCUIT” através do ATPDraw.

```

MODEL SHORT
DATA
  Seed      {dflt: 97}
  NumSC     {dflt: 100}
  PhaseSC   {dflt: 123}
  PhaseOP   {dflt: 123}
  Tdelay    {dflt: 0.0}
  Tclear    {dflt: 0.1}
  Td        {dflt: 5.0E-3}
  Vb        {dflt: 230E3}
  Sb        {dflt: 100E6}
  freq      {dflt: 60}
  N         {dflt: 10}
  Tstart    {dflt: -1.0}
  Tstop     {dflt: 1000.0}
INPUT
  Vin
  Iin
OUTPUT
  comSC[1..3]
  comCB[1..3]
VAR
  comSC[1..3]
  comCB[1..3]
INIT
  comSC[1..3]:=0
  comCB[1..3]:=1
ENDINIT
MODEL sc FOREIGN SHORTCIRCUIT {ixdata:13, ixin:6, ixout:6, ixvar:6}
EXEC
  USE sc AS sc
  DATA
    xdata[1..13]:=[Seed, NumSC, PhaseSC, PhaseOP, Tdelay, Tclear, Td, Vb, Sb, freq,
                  N, Tstart, Tstop]
  INPUT
    xin[1..6]:=[t, starttime, stoptime, timestep, Vin, Iin]
  HISTORY
    xvar[1..6]:=0
  OUTPUT
    comSC[1..3]:=xout[1..3]
    comCB[1..3]:=xout[4..6]
ENDUSE
ENDEXEC
ENDMODEL

```

Fonte: Elaborado pelo autor.

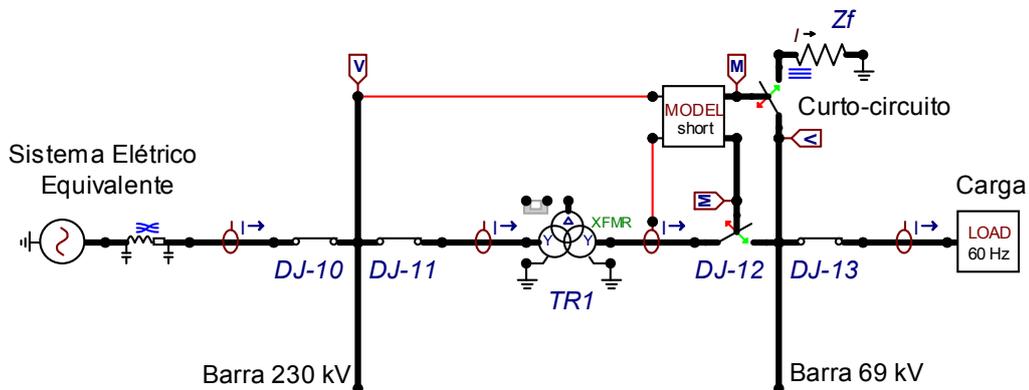
- freq: frequência nominal do sistema (valor predefinido: 60 Hz);
- N: número de intervalos para análise dos resultados (valor predefinido: 10);
- Tstart: instante de início da simulação de eventos (valor predefinido: -1,0 s);
- Tstop: instante de término da simulação de eventos (valor predefinido: 1000,0 s).

A entrada (**INPUT**) Iin é utilizada para monitorar a corrente em qualquer ponto do sistema em virtude do curto-circuito simulado. Já a entrada Vin é utilizada para monitorar a tensão após cada evento de abertura do disjuntor.

A saída (**OUTPUT**) comSC[1..3] é utilizada para comandar a ocorrência dos

curtos-circuitos na simulação e a saída `comCB [1 . . 3]` para comandar a abertura do disjuntor. Tanto o ponto em que ocorrerá o curto-circuito como o disjuntor devem ser representados através de chaves controladas por TACS (*TACS-controlled Switches – Type 13*) para receberem estes comandos. A impedância de curto-circuito pode ser modelada através de um elemento linear RLC trifásico (*Branch Linear – RLC 3-phase*). A Figura 5.6 apresenta um pequeno exemplo de utilização do *foreign model* “SHORTCIRCUIT” em uma simulação no ATPDraw.

Figura 5.6 – Utilização do *foreign model* “SHORTCIRCUIT” em uma simulação no ATPDraw.



Fonte: Elaborado pelo autor.

b) *Foreign model* “INRUSH”

Para a realização de simulações estocásticas de energização de transformador, deve-se empregar o *foreign model* denominado “INRUSH”. No ATPDraw, pode-se importar o código em linguagem MODELS apresentado no Quadro 5.2 a partir do elemento “*Default model*”.

Os parâmetros do tipo **DATA** apresentados no Quadro 5.2 correspondem a:

- Seed: valor inicial da sequência aleatória do MMC (valor predefinido: 97);
- NumINR: número de energizações simuladas (valor predefinido: 100);
- Phase: fase em que será analisada a corrente eficaz de energização do transformador (valor predefinido: 1):
 - 1 = Fase A;
 - 2 = Fase B;
 - 3 = Fase C;
- Tdelay: tempo para o início do cálculo do valor eficaz da corrente após a energização (valor predefinido: 0,0 s);

Quadro 5.2 – MODEL para acessar o *foreign model* “INRUSH” através do ATPDraw.

```

MODEL PTINR
DATA
  Seed      {dflt: 97}
  NumINR    {dflt: 100}
  Phase     {dflt: 1}
  Tdelay    {dflt: 0.0}
  Td        {dflt: 5.0E-3}
  Vb        {dflt: 230E3}
  Sb        {dflt: 100E6}
  freq      {dflt: 60}
  N         {dflt: 10}
  Tstart    {dflt: -1.0}
  Tstop     {dflt: 1000.0}
INPUT
  Current
OUTPUT
  com[1..3]
VAR
  com[1..3]
INIT
  com[1..3]:=0
ENDINIT
MODEL inru FOREIGN INRUSH {ixdata:11, ixin:5, ixout:3, ixvar:3}
EXEC
  USE inru AS inru
  DATA
    xdata[1..11]:=[Seed, NumINR, Phase, Tdelay, Td, Vb, Sb, freq, N, Tstart, Tstop]
  INPUT
    xin[1..5]:=[t, starttime, stoptime, timestep, Current]
  HISTORY
    xvar[1..3]:=0
  OUTPUT
    com[1..3]:=xout[1..3]
ENDUSE
ENDEXEC
ENDMODEL

```

Fonte: Elaborado pelo autor.

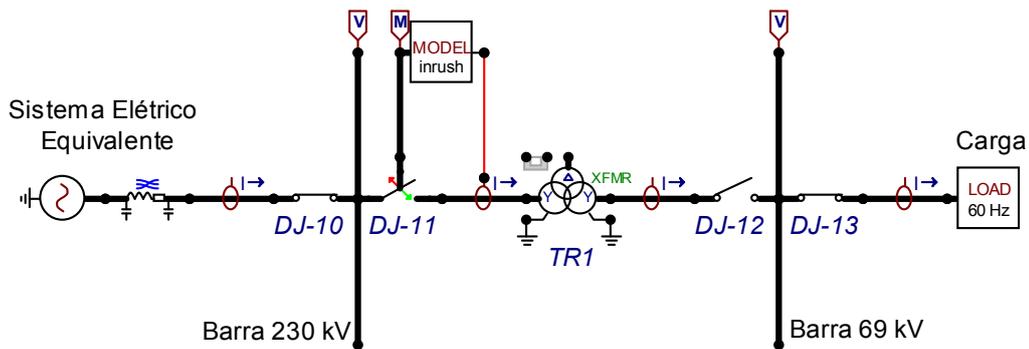
- Td: tempo de dispersão entre polos do disjuntor (valor predefinido: 5,0E-3 s);
- Vb: tensão base do sistema no ponto de análise (valor predefinido: 230E3 V);
- Sb: potência base do sistema (valor predefinido: 100E6 VA);
- freq: frequência nominal do sistema (valor predefinido: 60 Hz);
- N: número de intervalos para análise dos resultados (valor predefinido: 10);
- Tstart: instante de início da simulação de eventos (valor predefinido: -1,0 s);
- Tstop: instante de término da simulação de eventos (valor predefinido: 1000,0 s).

A entrada (**INPUT**) *Current* é utilizada para monitorar a corrente de energização do transformador na fase e no ponto (“no”) definidos pelo usuário. Já a saída (**OUTPUT**) *com[1..3]* é utilizada para comandar o fechamento do disjuntor.

Assim como na simulação de curto-circuito, o disjuntor deve ser representado através da

chave controlada por TACS (*TACS-controlled Switches – Type 13*) para receber os comandos do *foreign model*. A Figura 5.7 apresenta um exemplo de utilização desta ferramenta em uma simulação no ATPDraw.

Figura 5.7 – Utilização do *foreign model* “INRUSH” em uma simulação no ATPDraw.



Fonte: Elaborado pelo autor.

c) *Foreign model* “SWITCHING”

Com exceção das manobras de disjuntores devido a curto-circuito e energização de transformador, as demais simulações estocásticas de manobra podem ser realizadas com o *foreign model* denominado “SWITCHING”. Novamente, esta ferramenta pode ser acessada no ATPDraw através da importação do código em linguagem MODELS apresentado no Quadro 5.3 e utilização do elemento “*Default model*”.

Os parâmetros do tipo **DATA** apresentados no Quadro 5.3 correspondem a:

- Seed: valor inicial da sequência aleatória do MMC (valor predefinido: 97);
- NumSW: número total de manobras simuladas (valor predefinido: 100);
- Phase: fases em que ocorrerão as manobras (valor predefinido: 123):
 - 1 = Fase A;
 - 2 = Fase B;
 - 3 = Fase C;
- Td: tempo de dispersão entre polos do disjuntor (valor predefinido: 5,0E-3 s);
- Vb: tensão base do sistema no ponto de análise (valor predefinido: 230E3 V);
- Sb: potência base do sistema (valor predefinido: 100E6 VA);
- N: número de intervalos para análise dos resultados (valor predefinido: 10);

- Tstart: instante de início da simulação de eventos (valor predefinido: -1,0 s);
- Tstop: instante de término da simulação de eventos (valor predefinido: 1000,0 s).

Quadro 5.3 – MODEL para acessar o *foreign model* “SWITCHING” através do ATPDraw.

```

MODEL SWITCH
DATA
  Seed      {dflt: 97}
  NumSW     {dflt: 100}
  PhaseSW   {dflt: 123}
  Td        {dflt: 5.0E-3}
  Vb        {dflt: 230.0E3}
  Sb        {dflt: 100.0E6}
  N         {dflt: 10}
  Tstart    {dflt: -1.0}
  Tstop     {dflt: 1000.0}
INPUT
  Vin
OUTPUT
  com[1..3]
VAR
  com[1..3]
INIT
  com[1..3]:=0
ENDINIT
MODEL sw FOREIGN SWITCHING {ixdata:9, ixin:5, ixout:3, ixvar:3}
EXEC
  USE sw AS sw
    DATA
      xdata[1..9]:=[Seed, NumSW, PhaseSW, Td, Vb, Sb, N, Tstart, Tstop]
    INPUT
      xin[1..5]:=[t, starttime, stoptime, timestep, Vin]
    HISTORY
      xvar[1..3]:=0
    OUTPUT
      com[1..3]:=xout[1..3]
  ENDUSE
ENDEXEC
ENDMODEL

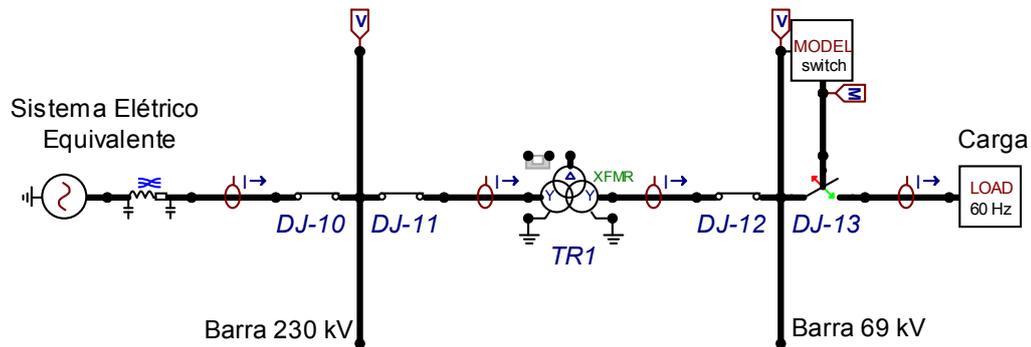
```

Fonte: Elaborado pelo autor.

A entrada (**INPUT**) *Vin* é utilizada para monitorar a tensão fase-ferra em qualquer ponto (“no”) do sistema em decorrência das manobras. Já a saída (**OUTPUT**) *com[1..3]* é utilizada para comandar as operações de fechamento e abertura do disjuntor.

Assim como nas outras simulações que envolvem operação de disjuntor, este equipamento deve ser representado através da chave (*Type 13*) para receber os comandos do *foreign model*. A Figura 5.8 apresenta um exemplo de utilização desta ferramenta em uma simulação no ATPDraw.

Figura 5.8 – Utilização do *foreign model* “SWITCHING” em uma simulação no ATPDraw.



Fonte: Elaborado pelo autor.

d) *Foreign model* “LIGHTNING”

para a realização de simulações estocásticas e descargas atmosféricas no ATP-EMTP empregando a metodologia desenvolvida, deve-se utilizar o *foreign model* denominado “LIGHTNING”. No ATPDraw isto pode ser realizado mediante a importação do código em linguagem MODELS apresentado no Quadro 5.4 e utilização do elemento “*Default model*”.

Os parâmetros do tipo **DATA** apresentados no Quadro 5.4 correspondem a:

- Seed: valor inicial da sequência aleatória do MMC (valor predefinido: 97);
- NumLG: número total de descargas atmosféricas simuladas (valor predefinido: 100);
- Waveform: forma de onda da corrente de descarga (valor predefinido: 0):
 - 0 = Heidler;
 - 1 = Dupla Rampa;
- n: parâmetro de forma da onda Heidler (valor predefinido: 2);
- IPm: corrente máxima (mediana) (valor predefinido: 31,10E3 A);
- IPsd: corrente máxima (desvio padrão) (valor predefinido: 0,484E3 A);
- TFm: tempo de frente da onda (mediana) (valor predefinido: 3,83E-6 s);
- TFsd: tempo de frente da onda (desvio padrão) (valor predefinido: 0,553E-6 s);
- THm: tempo de meia cauda da onda (mediana) (valor predefinido: 77,5E-6 s);
- THsd: tempo de meia cauda da onda (desvio padrão) (valor predefinido: 0,577E-6 s);
- Vb: tensão base do sistema no ponto de análise (valor predefinido: 230E3 V);
- Sb: potência base do sistema (valor predefinido: 100E6 VA);

- N: número de intervalos para análise dos resultados (valor predefinido: 10);
- Tstart: instante de início da simulação de eventos (valor predefinido: -1,0 s);
- Tstop: instante de término da simulação de eventos (valor predefinido: 1000,0 s).

Quadro 5.4 – MODEL para acessar o *foreign model* “LIGHTNING” através do ATPDraw.

```

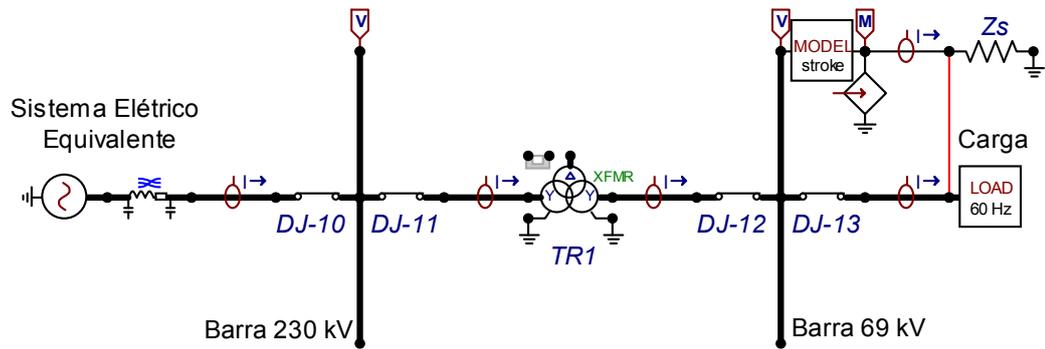
MODEL STROKE
DATA
  Seed      {dflt: 97}
  NumLG     {dflt: 100}
  Waveform  {dflt: 0}
  n         {dflt: 2}
  IPm       {dflt: 31.1E3}
  IPsd      {dflt: 0.484E3}
  TFm       {dflt: 3.83E-6}
  TFsd      {dflt: 0.553E-6}
  THm       {dflt: 77.5E-6}
  THsd      {dflt: 0.577E-6}
  Vb        {dflt: 230E3}
  Sb        {dflt: 100E6}
  N         {dflt: 10}
  Tstart    {dflt: -1.0}
  Tstop     {dflt: 1000.0}
INPUT
  Vin
OUTPUT
  wave
VAR
  wave
INIT
  wave:=0
ENDINIT
MODEL str FOREIGN LIGHTNING {ixdata:15, ixin:5 ixout:1, ixvar:1}
EXEC
  USE str AS str
  DATA
    xdata[1..15]:=[Seed, NumLG, Waveform, n, IPm, IPsd, TFm, TFsd, THm, THsd,
                  Vb, Sb,N, Tstart, Tstop]
  INPUT
    xin[1..5]:=[t, starttime, stoptime, timestep, Vin]
  HISTORY
    histdef(xvar[1]):=0
  OUTPUT
    wave:=xout[1]
ENDUSE
ENDEXEC
ENDMODEL

```

Fonte: Elaborado pelo autor.

A entrada (**INPUT**) *Vin* é utilizada para monitorar a tensão fase-ferra em qualquer ponto (“no”) do sistema em decorrência das descargas atmosféricas. Já a saída (**OUTPUT**) *wave* é utilizada para gerar a corrente de descarga no sistema.

Como a variável de saída do MODEL é adimensional, esta deve ser inserida em uma fonte de corrente controlada por TACS (*TACS Controlled Source – Type60*). A impedância de surto do sistema pode ser modelada através de um elemento linear RLC monofásico (*Branch Linear – RLC*). A Figura 5.9 apresenta um exemplo de utilização desta ferramenta através do ATPDraw.

Figura 5.9 – Utilização do *foreign model* “LIGHTNING” em uma simulação no ATPDraw.

Fonte: Elaborado pelo autor.

6 APLICAÇÕES DA METODOLOGIA PROPOSTA

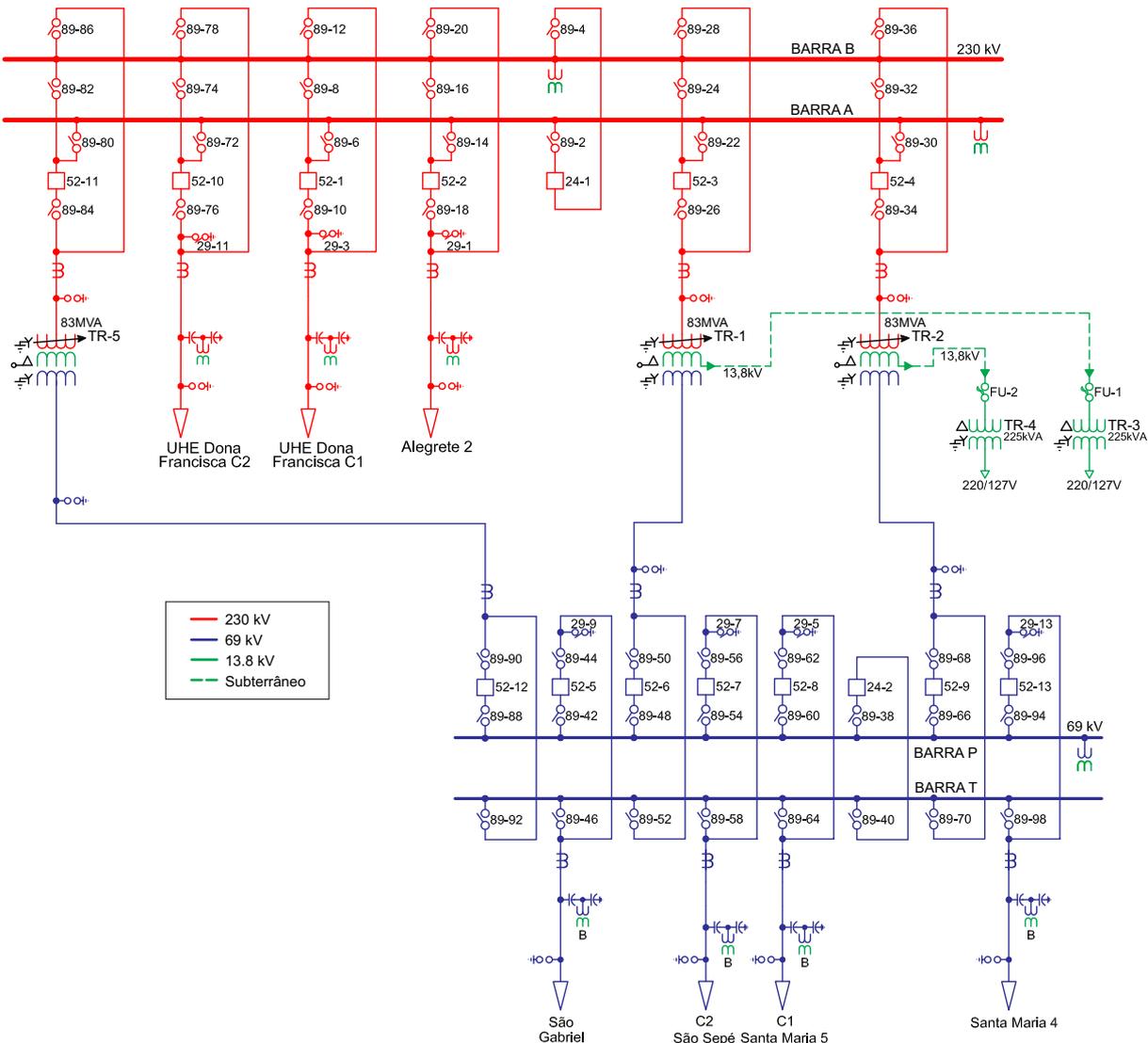
Neste capítulo alguns exemplos de aplicação são propostos, visando demonstrar as principais funcionalidades da metodologia. Apresenta-se também, os parâmetros utilizados nas simulações com abordagem aos seguintes eventos: curtos-circuitos, energizações de transformador de potência, manobras de disjuntor e descargas atmosféricas. O sistema elétrico utilizado nestes exemplos compreende a subestação denominada “Santa Maria 3” (SMA3), localizada no município de Santa Maria – RS.

6.1 SISTEMA ELÉTRICO ABORDADO

A subestação SMA3 é integrante da Rede Básica do Sistema Interligado Nacional (SIN), sendo gerenciada pela CEEE-GT, com exceção da linha de transmissão UHE Dona Francisca C2, que tem operação e manutenção sob responsabilidade da Omega Energia Renovável S.A. Esta subestação (Figura 6.1) é composta de:

- Um setor de 230 kV com:
 - Arranjo de barramento duplo;
 - Um módulo de acoplamento inter-barras;
 - Três módulos de transformador de potência;
 - Três módulos de linha de transmissão (LT): Alegrete 2; Usina Hidrelétrica (UHE) Dona Francisca C1 e Usina Hidrelétrica (UHE) Dona Francisca C2.
- Um setor de 69 kV com:
 - Arranjo de barramento do tipo principal-transferência;
 - Um módulo de acoplamento inter-barras;
 - Três módulos de transformador de potência;
 - Quatro módulos de linhas de transmissão: Santa Maria 4; Santa Maria 5 C1; São Sepé C2 e São Gabriel.
- Três transformadores de potência de 230 kV / 69 kV / 13,8 kV – 50 MVA (ONAN) / 66,5 MVA (ONAFI) / 83 MVA (ONAFII);
- Dois transformadores para serviços auxiliares de 13,8 kV / 220-127 V – 225 kVA.

Figura 6.1 – Diagrama representativo da subestação Santa Maria 3.



Fonte: Adaptado de CEEE-GT (2014).

Ressalta-se que a metodologia de modelagem do sistema e a representação deste arquivo não são contempladas nos objetivos deste trabalho. No caso da subestação SMA3, tais procedimentos estão descritos no Relatório Técnico Final do Projeto de P&D CEEE-GT 9947883 (LAPES-CEESP, 2016). Neste relatório, aborda-se também o procedimento para obtenção de equivalentes elétricos do SIN através do software de Análise de Falhas Simultâneas (ANAFAS[®]) do Centro de Pesquisas de Energia Elétrica (CEPEL).

Conforme descrito no referido relatório, a subestação SMA3 foi representada através de dois modelos distintos: um para simulação de eventos precursoros de transitórios de baixas e médias frequências e outro para transitórios de altas frequências. Para facilitar o processo de

criação do arquivo padrão de entrada do ATP-EMTP, utilizou-se o software ATPDraw.

No Quadro 6.1 apresenta-se um resumo dos principais elementos utilizados na modelagem da subestação. Maiores detalhes estão disponíveis no Relatório Final LAPES-CEESP (2016).

Quadro 6.1 – Resumo da modelagem de componentes da subestação.

Elementos	Modelo para baixas e médias frequências	Modelo para altas frequências
Para-raios	Resistor dependente de corrente (<i>Type 92</i>)	Modelo IEEE
Disjuntores e seccionadoras	Chave controlada por tempo (<i>Time-controlled Switch</i>)	Chave controlada por tempo, capacitâncias para terra e entre contatos
Linhas de transmissão e cabos de interligação	PI-equivalente concentrado (componentes simétricas)	Modelo dependente da frequência de JMarti
Transformadores de corrente (TCs)	–	Capacitâncias para terra
Transformadores de potencial capacitivos (TPCs) e indutivos (TPIs)	Capacitâncias para terra	
Transformadores de potência	Modelo Híbrido (<i>Hybrid Transformer Model</i>)	
Equivalentes elétricos	Fonte de tensão cossenoidal (<i>Type 14</i>) e elemento PI-equivalente concentrado (componentes simétricas)	
Cargas	Impedância constante concentrada	

Fonte: Elaborado pelo autor.

De posse dos modelos da subestação a ser analisada, descreve-se na sequência deste capítulo os procedimentos para simulação e análise de transitórios através do pré-processador gráfico ATPDraw.

6.2 ESTUDO DE CURTO-CIRCUITO

Nestes exemplos de aplicação, efetuou-se simulações estocásticas de curtos-circuitos com impedância de falta nula no início da linha de transmissão Santa Maria 4, com posterior abertura do disjuntor 52-13. Para os três casos de curtos-circuitos, os “nós” de medição de tensão

foram selecionados junto ao ponto de instalação do primário do TPI da Barra Principal de 69 kV. Tanto nas medições das tensões como nas de corrente, somente a Fase A foi selecionada.

Para a medição de corrente, selecionou-se o ponto de instalação no primário do TC da linha de transmissão Santa Maria 4 (LT - SM4). A configuração e os resultados específicos de cada simulação são apresentados na sequência.

6.2.1 Simulação de curto-circuito trifásico com abertura tripolar

Para que a simulação de curto-circuito trifásico com abertura tripolar seja realizada, é necessário a inclusão dos parâmetros apresentados na Tabela 6.1. Nesta tabela, o parâmetro “Tdelay” foi configurado em 0 (zero) segundos, indicando que o cálculo da corrente eficaz no ponto monitorado é efetuado em um ciclo da frequência nominal imediatamente após a ocorrência de cada um dos 100 curtos-circuitos simulados.

Tabela 6.1 – Parâmetros utilizados no estudo de curto-circuito trifásico com abertura tripolar do disjuntor.

Parâmetro	Valor	Comentário
Seed	97	Valor inicial da sequência aleatória do MMC
NumSC	100	Número de curtos-circuitos simulados
PhaseSC	123	Curto-circuito trifásico
PhaseOP	123	Abertura tripolar do disjuntor
Tdelay	0 s	Tempo para o início do cálculo do valor eficaz da corrente
Tclear	0,1 s	Tempo para abertura do disjuntor após o curto-circuito
Td	0,005 s	Tempo de dispersão entre polos do disjuntor
Vb	69×10^3 V	Tensão base do sistema no ponto de análise
Sb	100×10^6 VA	Potência base do sistema
freq	60 Hz	Frequência nominal do sistema
N	10	Número de intervalos
Tstart	-1 s	Instante de início da geração de eventos na simulação
Tstop	1000 s	Instante de encerramento da geração de eventos na simulação

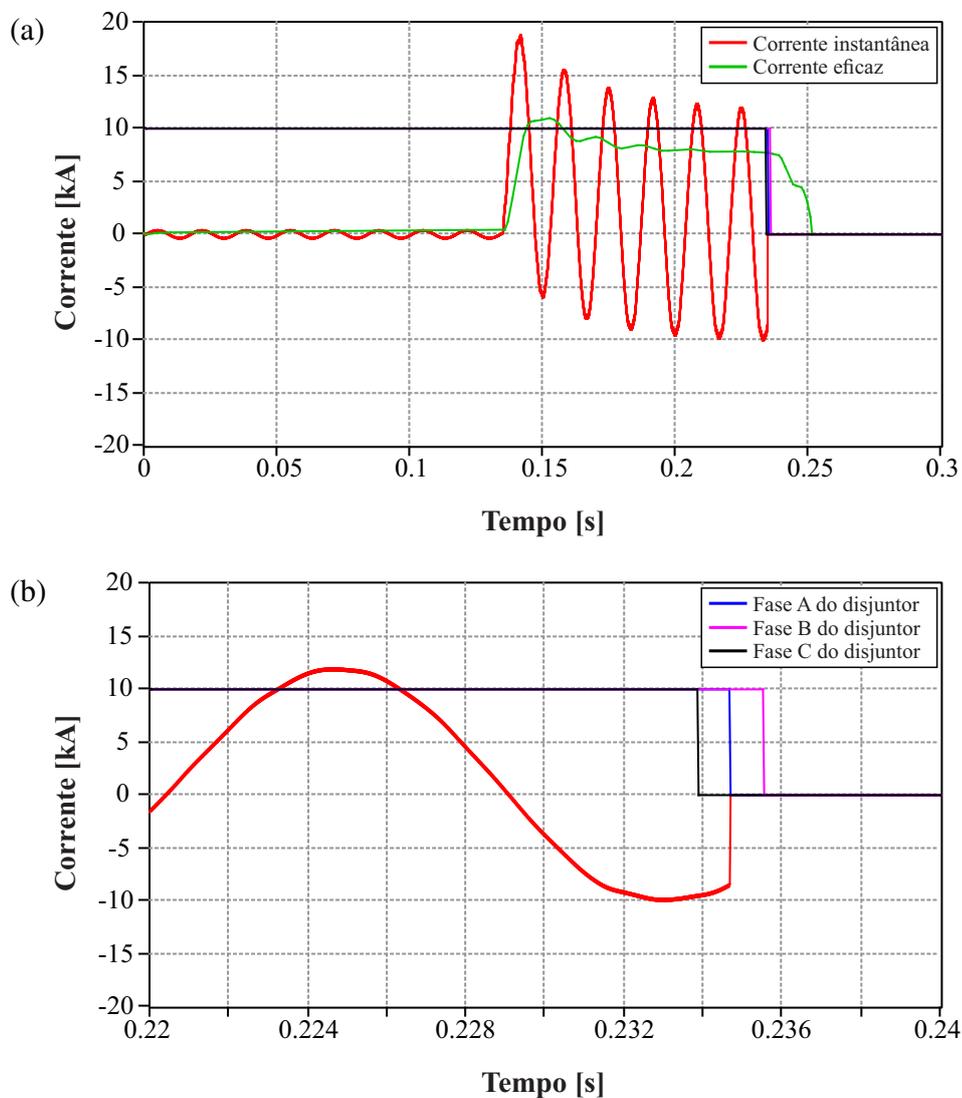
Fonte: Elaborado pelo autor.

A Figura 6.2 (a) exhibe o gráfico da corrente medida no TC da LT - SM4 no domínio do tempo. Para efetuar tal medida, considerou-se a Fase A do primário do TC. Importante ressaltar, que neste gráfico selecionou-se apenas o primeiro evento de curto-circuito trifásico dentre os

100 simulados. Através dele, pode-se visualizar detalhes com relação ao instante de ocorrência do curto-circuito, determinado através do MMC. Além disso, pode-se visualizar o tempo de abertura do disjuntor, que está ocorrendo 100 milissegundos após o início do evento.

Já na Figura 6.2 (b), pode-se notar a mesma medida de corrente da figura anterior, mas com detalhes para a abertura tripolar do disjuntor e a respectiva dispersão de polos ocasionada pela utilização do MMC. O método define tanto a ordem, quanto o tempo de abertura das fases do disjuntor.

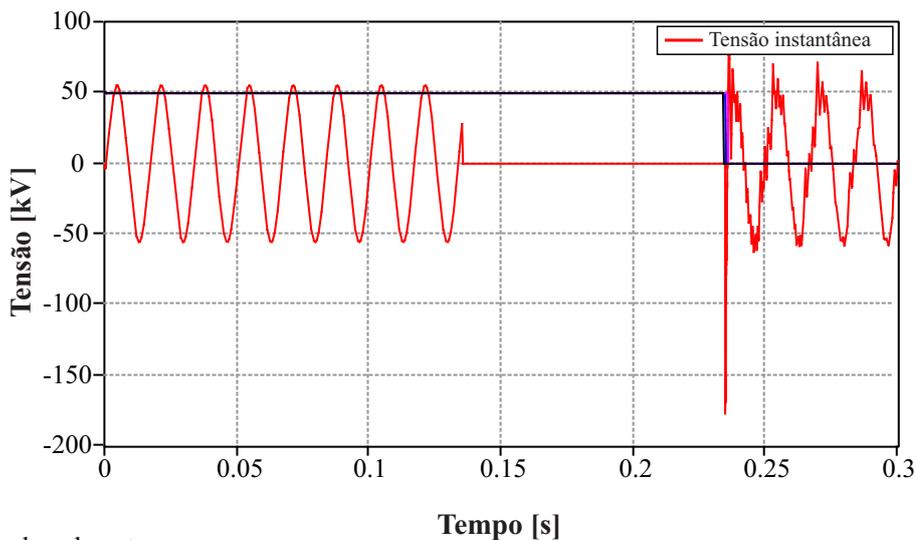
Figura 6.2 – Corrente medida no primário do TC da LT - SM4 (Fase A), resultante do curto-circuito trifásico com abertura tripolar do disjuntor.



Fonte: Elaborado pelo autor.

Os valores representados na Figura 6.3 correspondem ao transitório da tensão fase-terra medida no ponto de instalação do primário do TPI da Barra Principal de 69 kV. Os valores são relativos à tensão fase-terra instantânea da Fase A. Da mesma forma que nos gráficos anteriores, este refere-se ao primeiro evento dentre os 100 simulados.

Figura 6.3 – Tensão fase-terra medida no primário do TPI da Barra Principal de 69 kV (Fase A), resultante do curto-circuito trifásico com abertura tripolar do disjuntor



Fonte: Elaborado pelo autor.

A Tabela 6.2 apresenta a divisão em 10 intervalos dos valores de corrente eficaz medidos junto a Fase A no primário do TC da LT - SM4, exportados pelo *foreign model* “SHORTCIRCUIT”. A Figura 6.4 exibe a representação de um histograma elaborado a partir dos dados desta tabela. Analisando-se a frequência dos intervalos do gráfico, verifica-se que 17% dos eventos simulados geram correntes entre 6,50 p.u. e 7,05 p.u.. No entanto, percebe-se também que há uma distribuição relativamente proporcional de ocorrências entre 6,50 p.u. e 10,35 p.u..

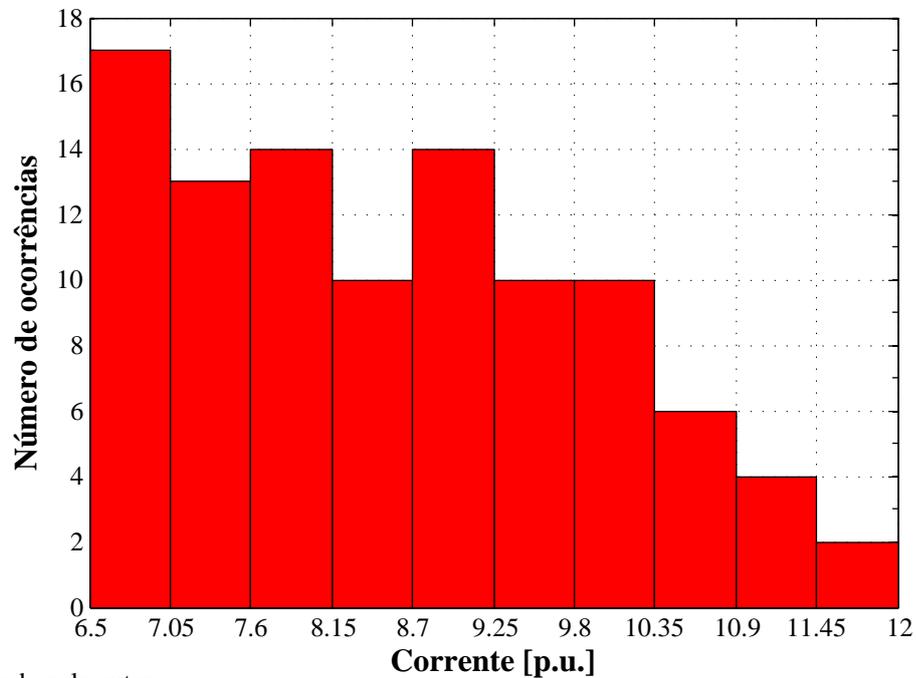
Já a Tabela 6.3 apresenta a divisão em 10 intervalos dos valores de tensão fase-terra medidos junto ao ponto de instalação do primário do TPI da Barra Principal de 69 kV (Fase A), exportados pelo *foreign model* “SHORTCIRCUIT”. Na Figura 6.5, tem-se o histograma de elaborado a partir dos dados desta tabela. Através deste histograma, verifica-se que 65% das aberturas tripolares do disjuntor 52-13, em decorrência de curtos-circuitos trifásicos, geram tensões entre 3,00 p.u. e 3,25 pu..

Tabela 6.2 – Divisão em intervalos dos valores de corrente eficaz para curto-circuito trifásico com abertura tripolar.

Intervalo [p.u.]	Ocorrências
$6,50 \leq I \leq 7,05$	17
$7,05 < I \leq 7,60$	13
$7,60 < I \leq 8,15$	14
$8,15 < I \leq 8,70$	10
$8,70 < I \leq 9,25$	14
$9,25 < I \leq 9,80$	10
$9,80 < I \leq 10,35$	10
$10,35 < I \leq 10,90$	6
$10,90 < I \leq 11,45$	4
$11,45 < I \leq 12,00$	2

Fonte: Elaborado pelo autor.

Figura 6.4 – Histograma de frequências: corrente no primário do TC da LT - SM4, decorrente do curto-circuito trifásico com abertura tripolar do disjuntor.



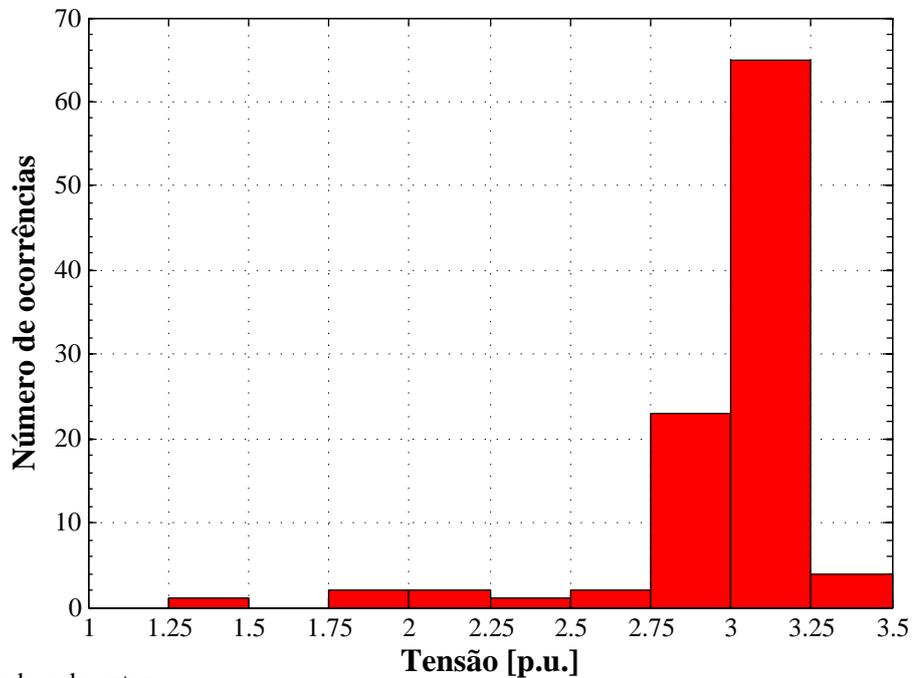
Fonte: Elaborado pelo autor.

Tabela 6.3 – Divisão em intervalos dos valores de tensão fase-terra para curto-circuito trifásico com abertura tripolar.

Intervalo [p.u.]	Ocorrências
$1,00 \leq V \leq 1,25$	0
$1,25 < V \leq 1,50$	1
$1,50 < V \leq 1,75$	0
$1,75 < V \leq 2,00$	2
$2,00 < V \leq 2,25$	2
$2,25 < V \leq 2,50$	1
$2,50 < V \leq 2,75$	2
$2,75 < V \leq 3,00$	23
$3,00 < V \leq 3,25$	65
$3,25 < V \leq 3,50$	4

Fonte: Elaborado pelo autor.

Figura 6.5 – Histograma de frequências: tensão fase-terra no TPI da Barra Principal de 69 kV, resultante do curto-circuito trifásico com abertura tripolar do disjuntor.



Fonte: Elaborado pelo autor.

6.2.2 Simulação de curto-circuito monofásico com abertura tripolar

Os parâmetros para a realização da simulação de curto-circuito monofásico com abertura tripolar estão expostos na Tabela 6.4. Nota-se que dentre estes, o parâmetro “PhaseSC” foi alterado para o valor 1, indicando que os curtos-circuitos simulados serão monofásicos.

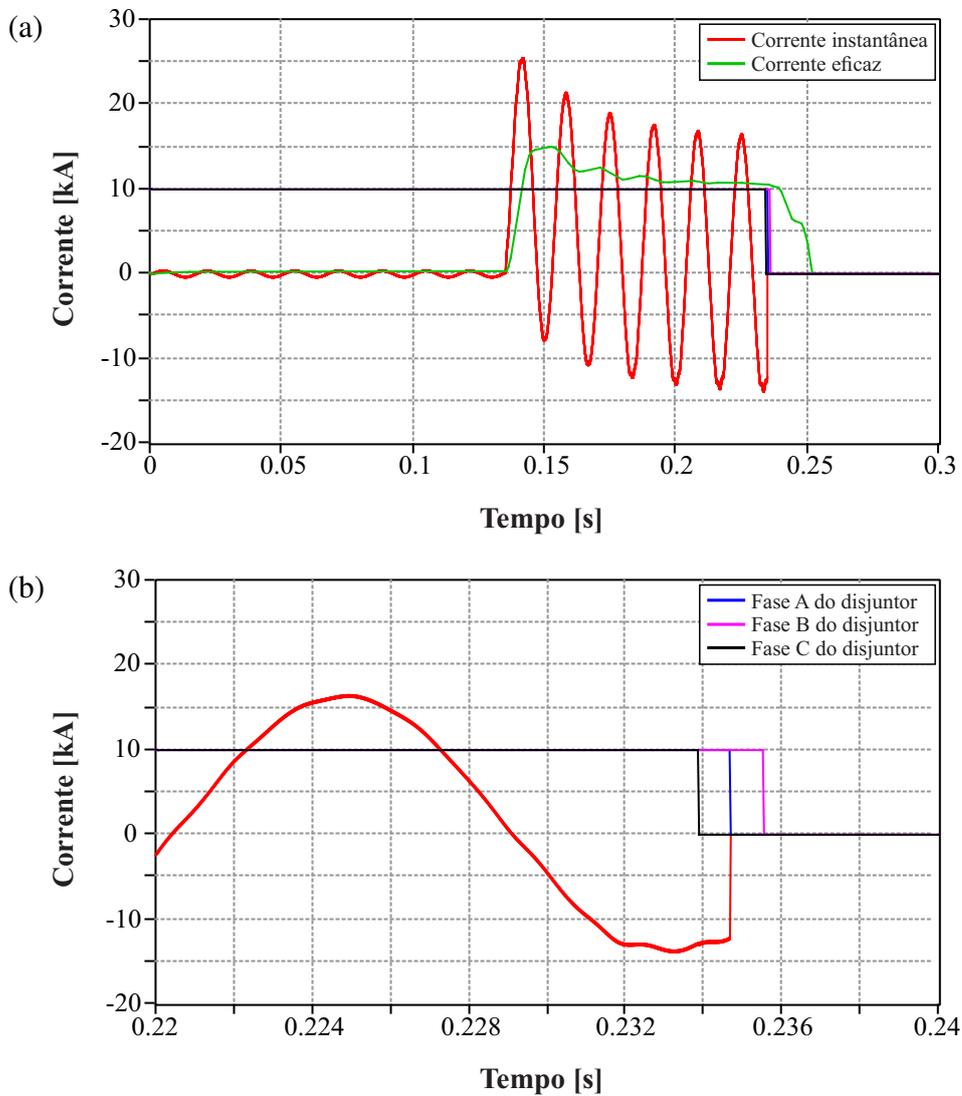
Tabela 6.4 – Parâmetros utilizados no estudo de curto-circuito monofásico com abertura tripolar do disjuntor.

Parâmetro	Valor	Comentário
Seed	97	Valor inicial da sequência aleatória do MMC
NumSC	100	Número de curtos-circuitos simulados
PhaseSC	1	Curto-circuito monofásico
PhaseOP	123	Abertura tripolar do disjuntor
Tdelay	0 s	Tempo para o início do cálculo do valor eficaz da corrente
Tclear	0,1 s	Tempo para abertura do disjuntor após o curto-circuito
Td	0,005 s	Tempo de dispersão entre polos do disjuntor
Vb	69×10^3 V	Tensão base do sistema no ponto de análise
Sb	100×10^6 VA	Potência base do sistema
freq	60 Hz	Frequência nominal do sistema
N	10	Número de intervalos
Tstart	-1 s	Instante de início da geração de eventos na simulação
Tstop	1000 s	Instante de encerramento da geração de eventos na simulação

Fonte: Elaborado pelo autor.

A Figura 6.6 (a) exhibe o gráfico da corrente medida no primário do TC da LT - SM4 (Fase A), no domínio do tempo. Para a Figura 6.6 (a) e (b), selecionou-se apenas o primeiro evento de curto-circuito trifásico dentre os 100 simulados. Quando o sistema simulado está submetido a um curto-circuito monofásico na Fase A, observa-se que as sobrecorrentes são maiores em relação o curto-circuito trifásico. A ocorrência do curto-circuito é aleatória, determinado através do MMC. Na Figura 6.6 (b), pode-se notar a mesma medida de corrente da figura anterior, mas com detalhes para a abertura tripolar do disjuntor e a respectiva dispersão de polos ocasionada pela utilização do MMC.

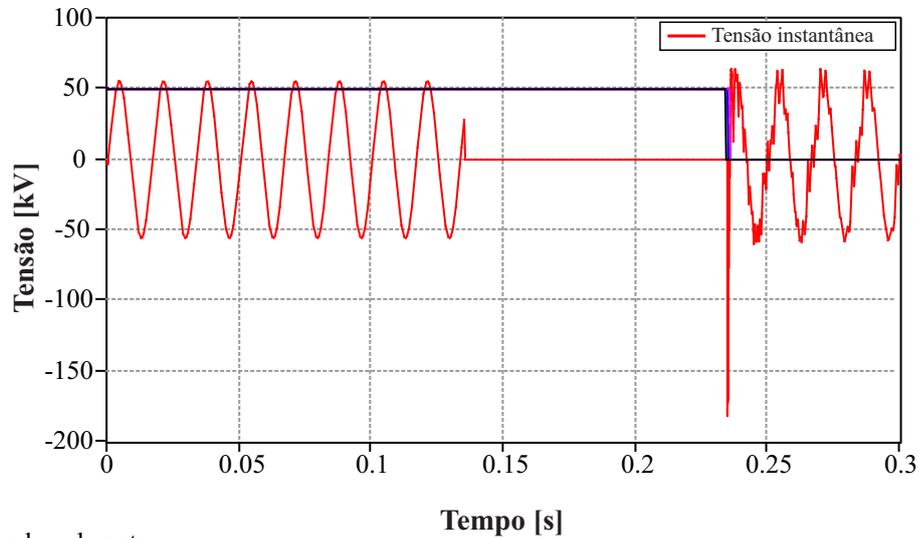
Figura 6.6 – Corrente medida no primário do TC da LT Santa Maria 4, decorrente do curto-circuito monofásico com abertura tripolar do disjuntor.



Fonte: Elaborado pelo autor.

A Figura 6.7 exibe o gráfico correspondente ao transitório da tensão medida no primário do TPI da Barra Principal de 69 kV. Os valores são relativos à tensão fase-terra instantânea da Fase A. Da mesma forma que nos gráficos anteriores, este refere-se ao primeiro evento dentre os 100 simulados. Observa-se neste gráfico pouca diferença nos valores de sobretensões, em comparação ao curto-circuito trifásico apresentado na Figura 6.3.

Figura 6.7 – Tensão fase-terra medida no primário do TPI da Barra Principal de 69 kV (Fase A), decorrente do curto-circuito monofásico com abertura tripolar do disjuntor



Fonte: Elaborado pelo autor.

A Tabela 6.5 exibe a divisão em 10 intervalos dos valores de corrente eficaz medidos junto a Fase A no primário do TC da LT - SM4 (Fase A), exportados pelo *foreign model* “SHORTCIRCUIT”. A Figura 6.8 apresenta o histograma elaborado a partir destes dados.

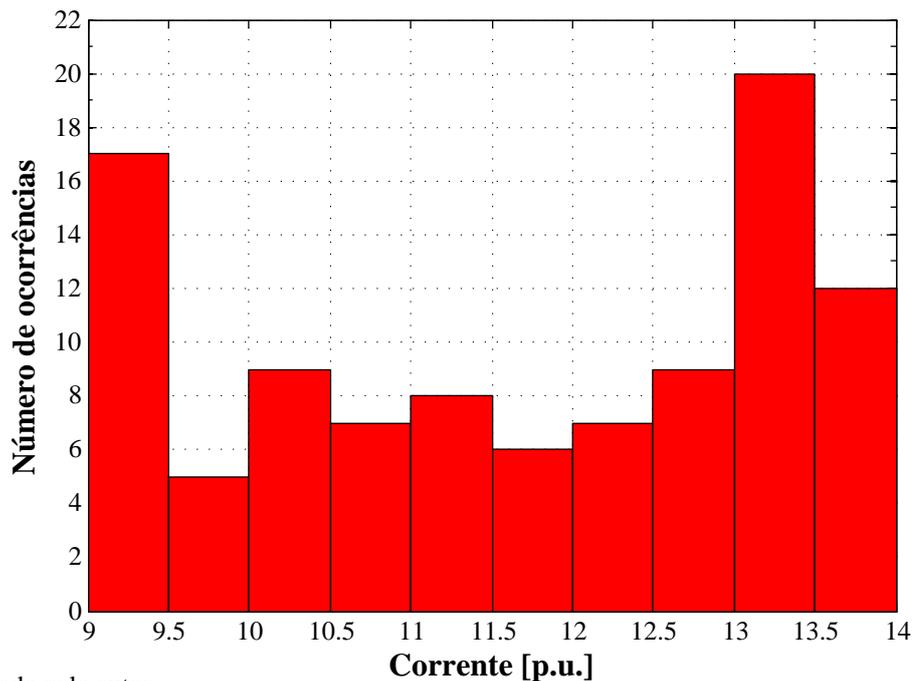
Tabela 6.5 – Divisão em intervalos dos valores de corrente eficaz para curto-circuito monofásico com abertura tripolar.

Intervalo [p.u.]	Ocorrências
$9,00 \leq I \leq 9,50$	17
$9,50 < I \leq 10,00$	5
$10,00 < I \leq 10,50$	9
$10,50 < I \leq 11,50$	7
$11,00 < I \leq 11,50$	8
$11,50 < I \leq 12,00$	6
$12,00 < I \leq 12,50$	7
$12,50 < I \leq 13,00$	9
$13,00 < I \leq 13,50$	20
$13,50 < I \leq 14,00$	12

Fonte: Elaborado pelo autor.

Analisando-se a frequência dos intervalos do gráfico da Figura 6.8, verifica-se que 20% dos eventos simulados geram correntes entre 13 p.u. e 13,05 p.u., e cerca de 17% geram correntes entre 9 p.u. e 9,50 p.u.. Além disso, percebe-se também que há uma distribuição relativamente proporcional de ocorrências entre 9,50 p.u. e 13 p.u..

Figura 6.8 – Histograma de frequências: corrente no primário do TC da LT Santa Maria 4 (Fase A), resultante do curto-circuito monofásico com abertura tripolar do disjuntor.



Fonte: Elaborado pelo autor.

Os valores de tensão fase-terra, medidos junto ao ponto de instalação do TPI da Barra Principal de 69 kV (Fase A), estão amostrados de acordo com a divisão de 10 intervalos na Tabela 6.6. Tais dados foram exportados pelo *foreign model* “SHORTCIRCUIT”.

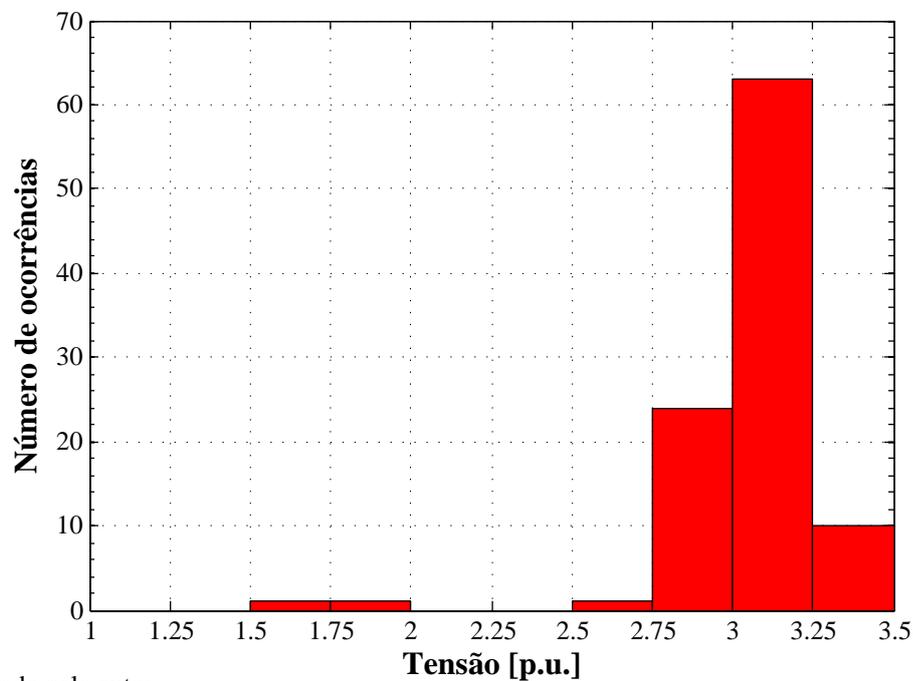
A partir de tais dados referidos na tabela, plotou-se o histograma da figura Figura 6.9. Através deste histograma, verifica-se que de 65% das aberturas tripolares do disjuntor 52-13, em decorrência de curtos-circuitos monofásicos, geram tensões entre 3,00 p.u. e 3,25 pu.. Cerca de 25% das aberturas, geram tensões entre 2,75 e 3,00 p.u..

Tabela 6.6 – Divisão em intervalos dos valores de tensão-fase terra para curto-circuito monofásico com abertura tripolar.

Intervalo [p.u.]	Ocorrências
$1,00 \leq V \leq 1,25$	0
$1,25 < V \leq 1,50$	0
$1,50 < V \leq 1,75$	1
$1,75 < V \leq 2,00$	1
$2,00 < V \leq 2,25$	0
$2,25 < V \leq 2,50$	0
$2,50 < V \leq 2,75$	1
$2,75 < V \leq 3,00$	24
$3,00 < V \leq 3,25$	63
$3,25 < V \leq 3,50$	10

Fonte: Elaborado pelo autor.

Figura 6.9 – Histograma de frequências: tensão fase-terra no TPI da Barra Principal de 69 kV (Fase A), resultante do curto-circuito monofásico com abertura tripolar do disjuntor.



Fonte: Elaborado pelo autor.

6.2.3 Simulação de curto-circuito monofásico com abertura monopolar

Os parâmetros para a realização da simulação de curto-circuito monofásico com abertura monopolar estão expostos na Tabela 6.7. Com relação à simulação anterior, o parâmetro “PhaseOP” foi alterado para o valor 1, indicando a abertura do disjuntor será monopolar.

Tabela 6.7 – Parâmetros utilizados no estudo de curto-circuito monofásico abertura monopolar.

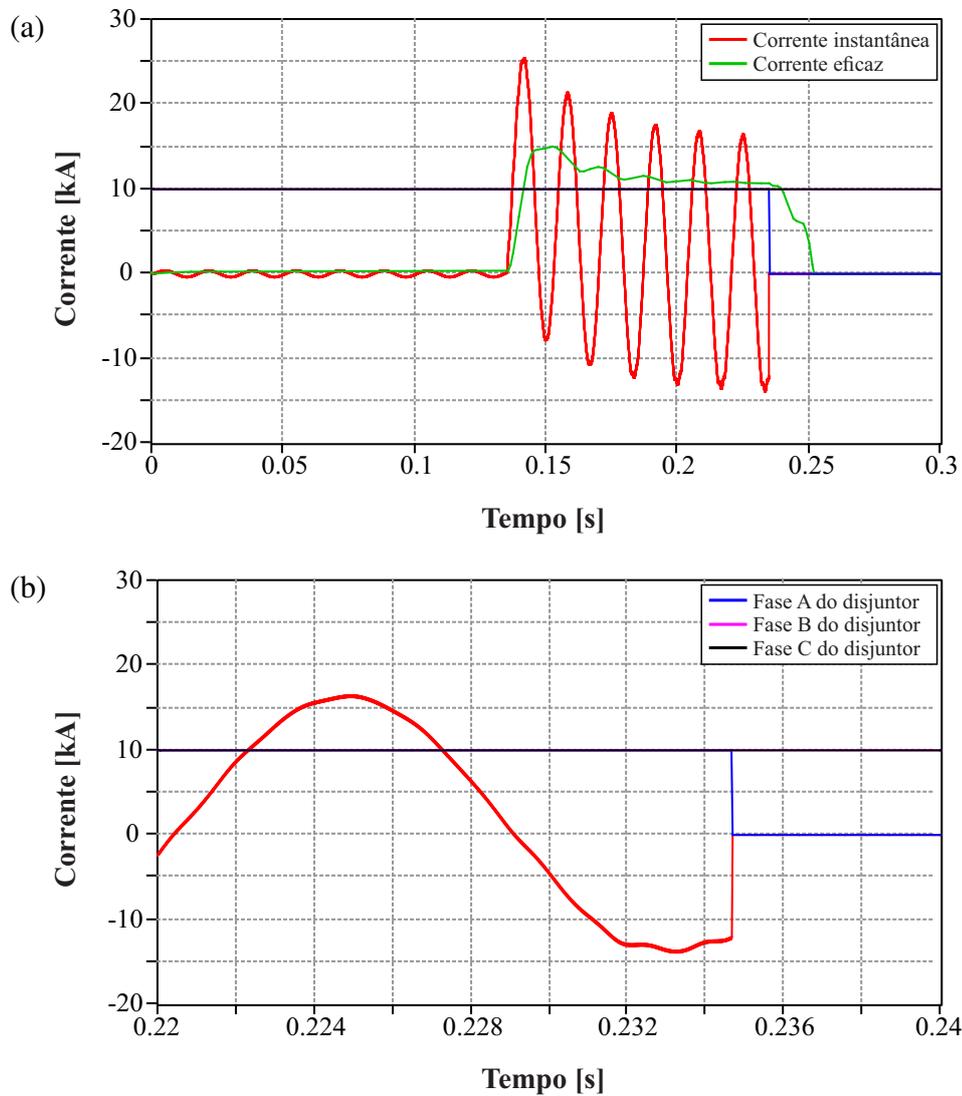
Parâmetro	Valor	Comentário
Seed	97	Valor inicial da sequência aleatória do MMC
NumSC	100	Número de curtos-circuitos simulados
PhaseSC	1	Curto-circuito monofásico
PhaseOP	1	Abertura monopolar do disjuntor
Tdelay	0 s	Tempo para o início do cálculo do valor eficaz da corrente
Tclear	0,1 s	Tempo para abertura do disjuntor após o curto-circuito
Td	0,005 s	Tempo de dispersão entre polos do disjuntor
Vb	69×10^3 V	Tensão base do sistema no ponto de análise
Sb	100×10^6 VA	Potência base do sistema
freq	60 Hz	Frequência nominal do sistema
N	10	Número de intervalos
Tstart	-1 s	Instante de início da geração de eventos na simulação
Tstop	1000 s	Instante de encerramento da geração de eventos na simulação

Fonte: Elaborado pelo autor.

Como citado anteriormente na seção 2.2, considerando curtos-circuitos monofásicos, recomenda-se a utilização de sistemas de proteção com abertura e religamento monopolar automático dos disjuntores. Pois além de ajudar na manutenção do sincronismo, através das duas fases remanescentes, aproximadamente 2/3 do fluxo de potência é mantido.

A Figura 6.10 (a) apresenta o resultado da simulação no domínio do tempo, com relação à corrente de curto-circuito monofásico medida no primário do TC da LT-SM4. Embora, a corrente seja a mesma da simulação anterior, na Figura 6.10 (b) é evidenciada a abertura monopolar.

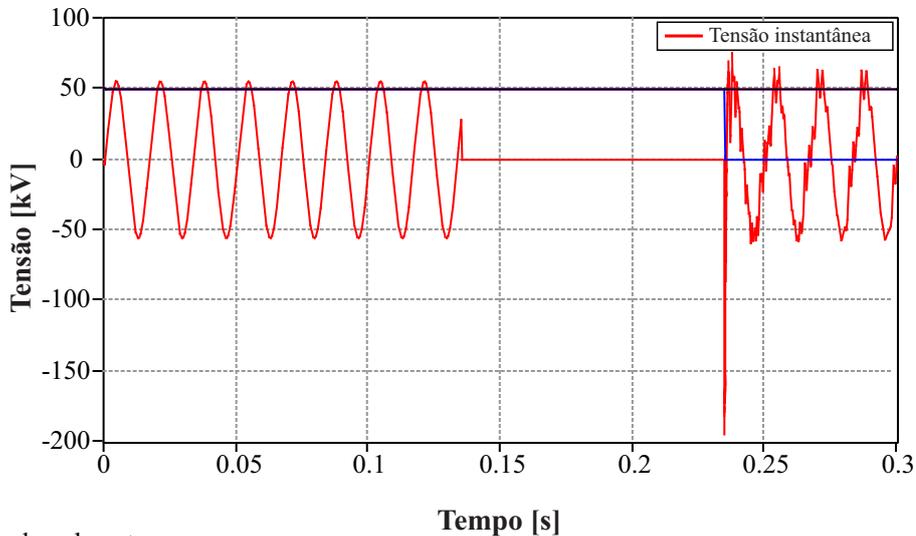
Figura 6.10 – Corrente medida no primário do TC da LT Santa Maria 4, decorrente do curto-circuito monofásico com abertura monopolar do disjuntor.



Fonte: Elaborado pelo autor.

Já na Figura 6.11, constata-se que para o curto-circuito monofásico com abertura monopolar, as tensões são ligeiramente superiores com relação à abertura tripolar. Tal comportamento pode ser justificado devido ao acoplamento mútuo da fase em curto-circuito com as demais fases energizadas.

Figura 6.11 – Tensão fase-terra medida no primário do TPI da Barra Principal de 69 kV (Fase A), resultante do curto-circuito monofásico com abertura monopolar do disjuntor



Fonte: Elaborado pelo autor.

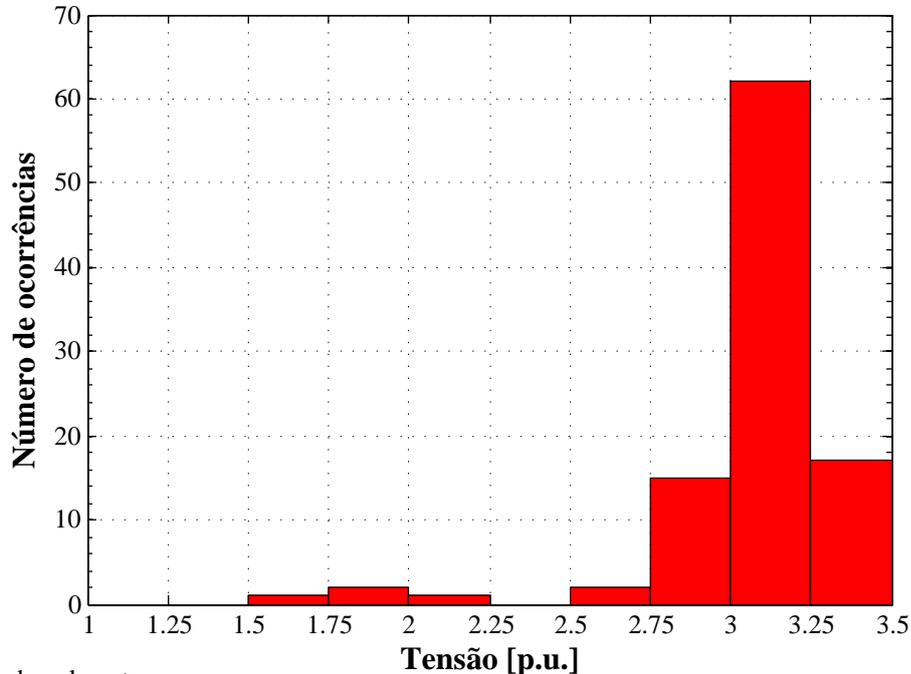
Os dados apresentados na Tabela 6.8 referem-se aos valores de tensão fase-terra, medidos no primário do TPI da Barra Principal de 69 kV (Fase A). Com base nos valores da tabela, obteve-se o histograma da Figura 6.12. Quando comparada à Figura 6.9, percebe-se que de mais de 60% dos casos, em ambos os gráficos, corresponderem às tensões entre 3,00 p.u. e 3,25 p.u.. Entretanto, para o curto-circuito monofásico houve um aumento das tensões entre 3,25 p.u. a 3,50 p.u.

Tabela 6.8 – Divisão em intervalos dos valores de tensão-fase terra para curto-circuito monofásico com abertura tripolar.

Intervalo [p.u.]	Ocorrências
$1,00 \leq V \leq 1,25$	0
$1,25 < V \leq 1,50$	0
$1,50 < V \leq 1,75$	1
$1,75 < V \leq 2,00$	2
$2,00 < V \leq 2,25$	1
$2,25 < V \leq 2,50$	0
$2,50 < V \leq 2,75$	2
$2,75 < V \leq 3,00$	15
$3,00 < V \leq 3,25$	62
$3,25 < V \leq 3,50$	17

Fonte: Elaborado pelo autor.

Figura 6.12 – Histograma de frequências: tensão fase-terra no TPI da Barra Principal de 69 kV (Fase A), resultante do curto-circuito monofásico com abertura monopolar do disjuntor.



Fonte: Elaborado pelo autor.

6.3 ESTUDO DE ENERGIZAÇÃO DE TRANSFORMADORES

Para o presente estudo realizou-se a energização individual e solidária do transformador de potência TR-2, a partir do disjuntor 52-4 (lado de 230 kV). Para a energização individual, os disjuntores dos transformadores TR-1 e TR-5 foram configurados com o estado aberto. Para a medição de corrente, selecionou-se o ponto de instalação no primário do TC do módulo de 230 kV deste transformador.

No caso da energização solidária do TR-2, os transformadores TR-1 e TR-5 apresentam-se em estado previamente energizados na condição à vazio, ou seja, sem carga conectada ao secundário. Os pontos de medições de corrente considerados são os primários dos TCs do módulo de 230 kV para os transformadores TR-1, TR-2 e TR-5. Como *foreign model* para Energização apresenta somente uma entrada de medição de corrente, realizou-se três simulações a fim de medir as correntes nos respectivos transformadores.

6.3.1 Simulação de Energização Individual

Para o presente estudo analisado, realizou-se 100 energizações no transformador TR-2. Os dados específicos para a configuração do estudo para energização individual são apresentados na Tabela 6.9.

Tabela 6.9 – Parâmetros utilizados no estudo de energização individual no TR-2.

Parâmetro	Valor	Comentário
Seed	97	Valor inicial da sequência aleatória do MMC
NumINR	100	Número de energizações simuladas
Phase	1	Cálculo do valor eficaz da corrente da Fase A
Tdelay	0 s	Tempo para início do cálculo do valor eficaz da corrente
Td	0,005 s	Tempo de dispersão entre polos do disjuntor
Vb	230×10^3 V	Tensão base do sistema no ponto de análise
Sb	50×10^6 VA	Potência base do sistema
freq	60 Hz	Frequência nominal do sistema
N	10	Número de intervalos
Tstart	-1 s	Instante de início da geração de eventos na simulação
Tstop	1000 s	Instante de encerramento da geração de eventos na simulação

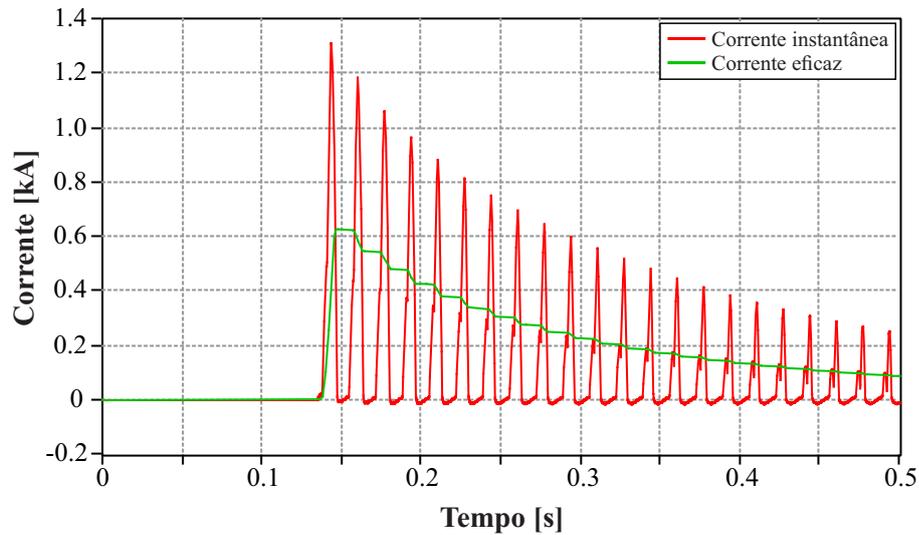
Fonte: Elaborado pelo autor.

Assim como no estudo de curto-circuito, o parâmetro “Tdelay” apresentado na Tabela 6.9 foi configurado em 0 (zero) segundos, para que o cálculo da corrente eficaz fosse efetuado imediatamente após cada evento de energização do transformador.

A Figura 6.13 exibe o gráfico da corrente eficaz com ponto de medição junto a Fase A do TC no módulo de 230 kV do transformador 2. Neste gráfico, pode-se verificar o transitório da corrente ocasionado pela energização do transformador em que o instante de fechamento é determinado pelo MMC.

Através do *foreign model* “INRUSH”, exportou-se os dados de corrente eficaz medida junto a Fase A do TC no módulo de 230 kV do transformador 2. Tais valores estão organizados em 10 intervalos apresentados na Tabela 6.10, condizentes a 100 energizações simuladas.

Figura 6.13 – Corrente medida no TC no módulo de 230 kV do transformador 2 (Fase A).



Fonte: Elaborado pelo autor.

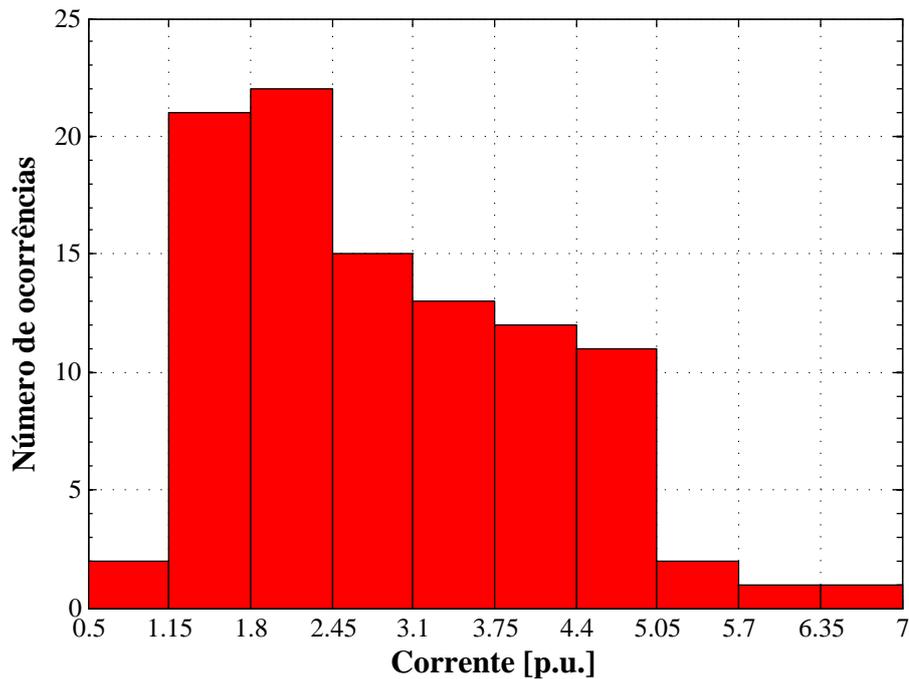
A partir destes dados da tabela plotou-se o histograma exibido na Figura 6.14. Através deste gráfico, verifica-se que os valores simuladores entre 0,50 p.u. e 7,00 p.u. na base nominal do transformador, e que 43% dos eventos geraram correntes entre 1,15 p.u. e 2,45 p.u.. Além disso, percebe-se também que há uma distribuição relativamente proporcional de ocorrências entre 2,45 p.u. e 5,05 p.u..

Tabela 6.10 – Divisão em intervalos dos valores de corrente medidas no primário do TC do TR-2 (230 kV) para energização individual.

Intervalo [p.u.]	Ocorrências
$0,50 \leq I \leq 1,15$	2
$1,15 < I \leq 1,80$	21
$1,80 < I \leq 2,45$	22
$2,45 < I \leq 3,10$	15
$3,10 < I \leq 3,75$	13
$3,75 < I \leq 4,40$	12
$4,40 < I \leq 5,05$	11
$5,05 < I \leq 5,70$	2
$5,70 < I \leq 6,35$	1
$6,35 < I \leq 7,00$	1

Fonte: Elaborado pelo autor.

Figura 6.14 – Histograma de frequências: corrente no TC do TR-2, lado de 230 kV (Fase A).



Fonte: Elaborado pelo autor.

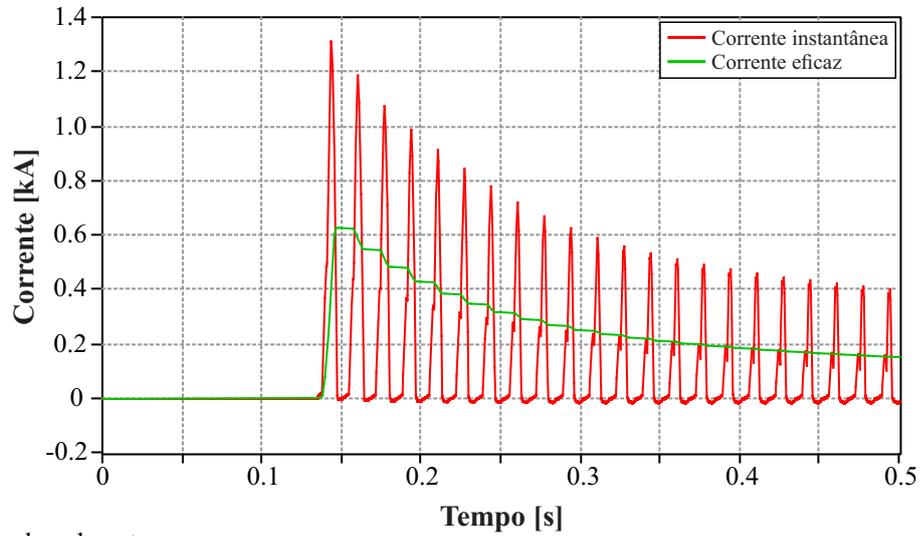
6.3.2 Simulação de Energização Solidária

Para simulação de energização solidária, os parâmetros inseridos correspondem aos mesmos valores apresentados na Tabela 6.9. A diferença entre os casos de energização individual e solidária, caracterizam-se por manter transformadores TR-1 e TR-5 previamente energizados. Da mesma maneira que para a energização individual, realizou-se 100 energizações no TR-2.

No gráfico da Figura 6.15, tem-se a medida de corrente na Fase A do primário do TC no módulo de 230 kV do transformador TR-2 para energização solidária. Comparando-se tal figura, com o gráfico da Figura 6.13, percebe-se que para o caso da energização individual ocorre um amortecimento maior nos picos de corrente.

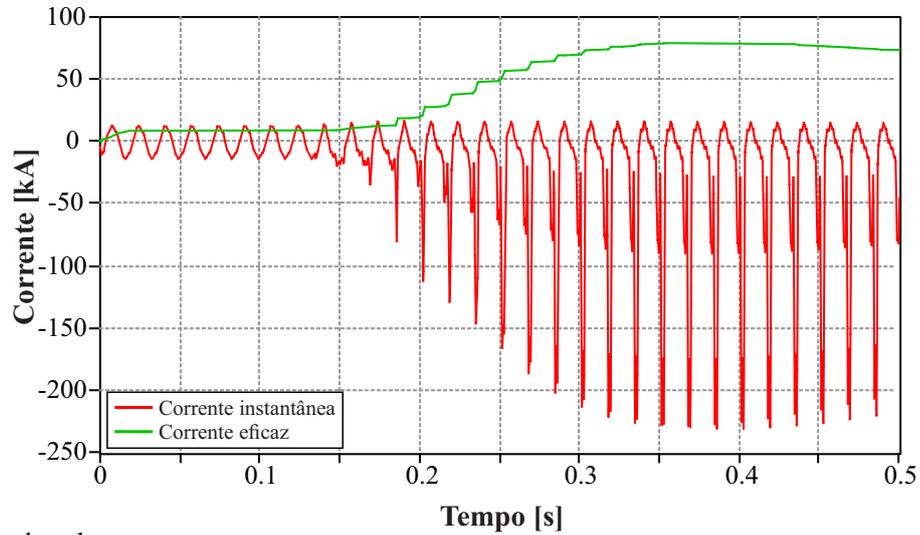
Com relação a Figura 6.16 e Figura 6.17, tem-se as medidas das correntes na Fase A do primário do TC no módulo de 230 kV para os TR-1 e TR-5, respectivamente. Através dos gráficos, é notável as alterações com relação às amplitudes nas correntes de magnetização em ambos os transformadores, causados pela energização do TR-2.

Figura 6.15 – Corrente medida no TC no módulo de 230 kV do transformador 2 (Fase A) para energização solidária.



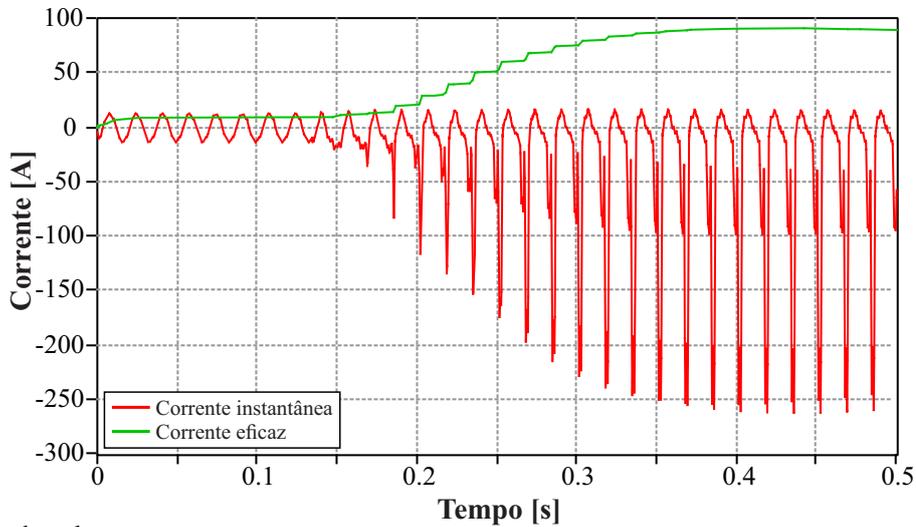
Fonte: Elaborado pelo autor.

Figura 6.16 – Corrente medida no TC no módulo de 230 kV do transformador 1 (Fase A) para energização solidária.



Fonte: Elaborado pelo autor.

Figura 6.17 – Corrente medida no TC no módulo de 230 kV do transformador 5 (Fase A) para energização solidária.



Fonte: Elaborado pelo autor.

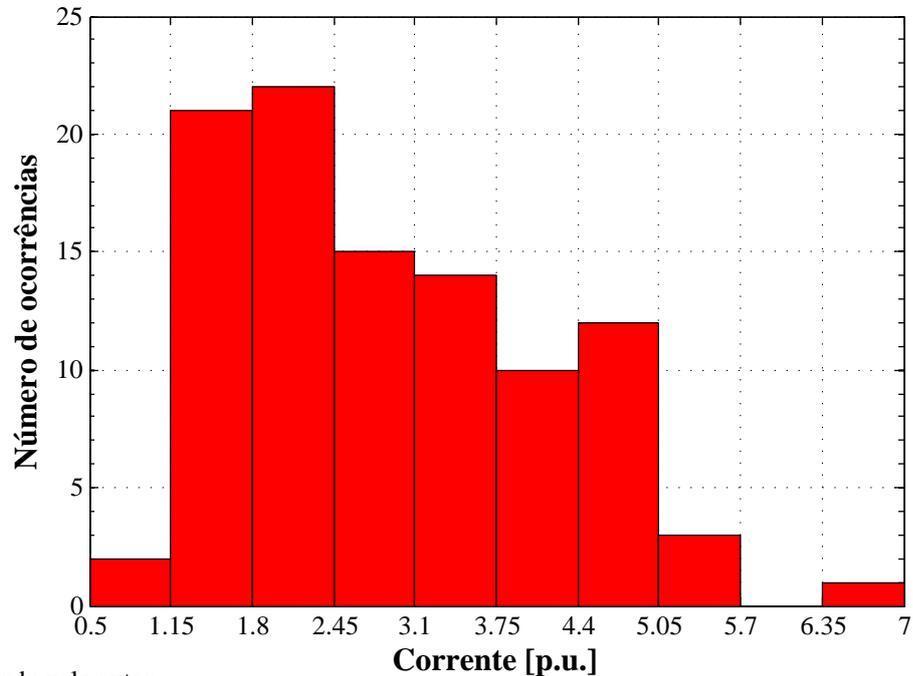
A Tabela 6.11 apresenta a divisão em 10 intervalos dos valores de corrente eficaz para 100 energizações simuladas. Estes dados foram exportados pelo *foreign model* “INRUSH”, no ponto de medição do primário do TC no módulo de 230 kV para o TR-2. A partir dos dados da tabela, elaborou-se o histograma exposto na Figura 6.18. As alterações mais significativas no histograma, quando comparado ao da Figura 6.14 são conferidas entre 3,10 p.u. a 6,35 p.u..

Tabela 6.11 – Divisão em intervalos dos valores de corrente medidas no primário do TC do TR-2 (230 kV) para energização solidária.

Intervalo [p.u.]	Ocorrências
$0,50 \leq I \leq 1,15$	2
$1,15 < I \leq 1,80$	21
$1,80 < I \leq 2,45$	22
$2,45 < I \leq 3,10$	15
$3,10 < I \leq 3,75$	14
$3,75 < I \leq 4,40$	10
$4,40 < I \leq 5,05$	12
$5,05 < I \leq 5,70$	3
$5,70 < I \leq 6,35$	0
$6,35 < I \leq 7,00$	1

Fonte: Elaborado pelo autor.

Figura 6.18 – Histograma de frequências: corrente no primário do TC do TR-2, lado de 230 kV (Fase A), para energização solidária.



Fonte: Elaborado pelo autor.

6.4 ESTUDO DE MANOBRAS DE DISJUNTOR

Tendo em vista a análise de outros eventos que requeiram a operação de disjuntores, diferentes dos já contemplados anteriormente, abordou-se nessa seção, os eventos de manobras de linha de transmissão e manobras de carga. Na subestação adotada como sistema base neste capítulo, selecionou-se para manobras de energização e desenergização da LT-Alegrete 2, a partir da subestação SMA3, por meio do disjuntor 52-2 desta subestação. O ponto de medição selecionado foi o primário do TPC desta linha (Fase A), e todas as manobras tripolares.

Para as manobras de carga, realizou-se a conexão e desconexão da carga C1 Santa Maria 5 (SMA5). Esta subestação foi modelada como uma carga com impedância constante concentrada, desprezando o pequeno trecho da linha de transmissão que interliga as subestações SM-3 e SM-5. As manobras de carga são realizadas através do disjuntor 52-8, todas tripolares. A tensão monitorada junto ao ponto de instalação do primário do TPC da carga C1 no setor de 69 kV (Fase A). Os parâmetros de configuração e os resultados das simulações destes exemplos de aplicação são apresentados a seguir.

6.4.1 Manobra de Linha de Transmissão

Para o estudo de energização e desenergização da LT-Alegrete 2, realizou-se 100 manobras, dentre as quais metade são aberturas e metade são fechamentos. Os parâmetros específicos para tal simulação está indicado na tabela Tabela 6.12.

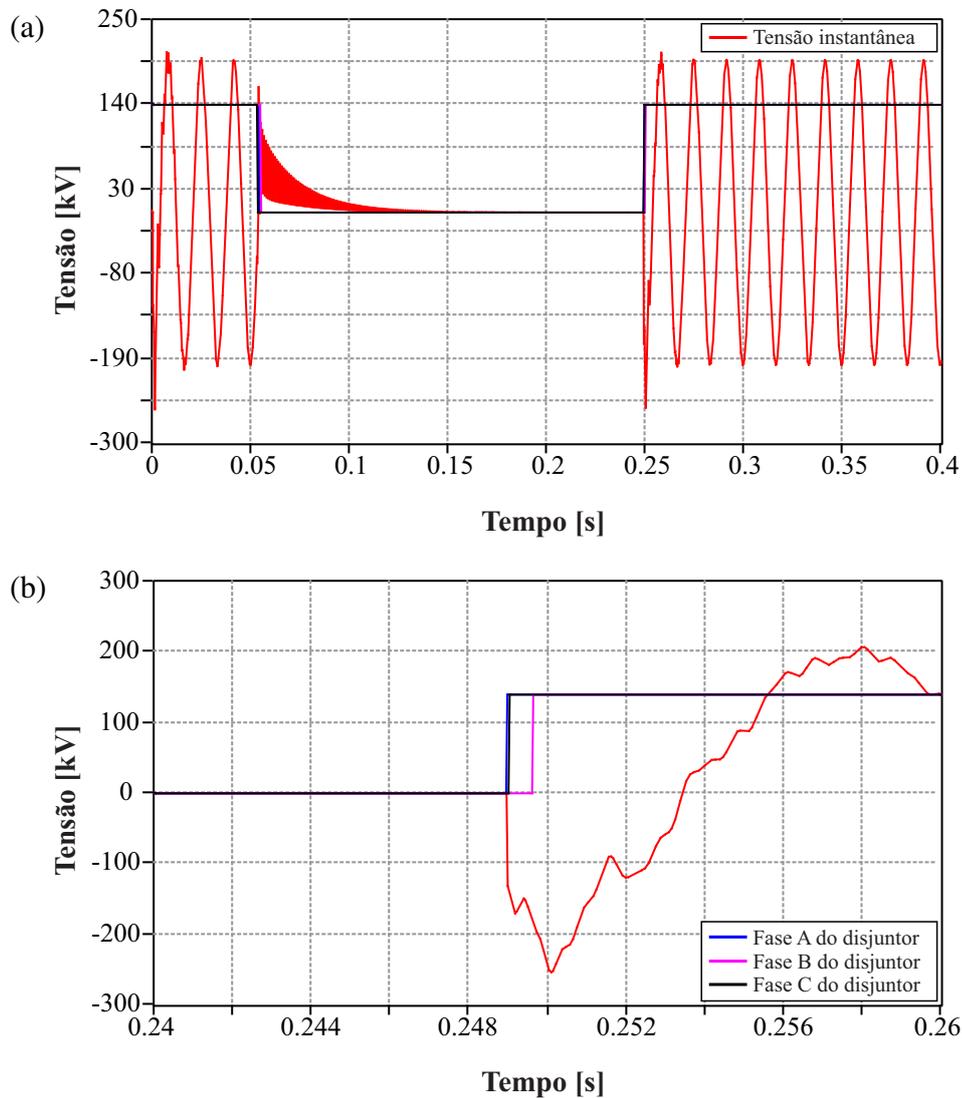
Tabela 6.12 – Parâmetros utilizados no estudo de manobra da linhas de transmissão Alegrete 2.

Parâmetro	Valor	Comentário
Seed	97	Valor inicial da sequência aleatória do MMC
NumSW	100	Número de manobras
PhaseSW	123	Abertura tripolar do disjuntor
Td	0,005 s	Tempo de dispersão entre polos do disjuntor
Vb	230×10^3 V	Tensão base do sistema no ponto de análise
Sb	100×10^6 VA	Potência base do sistema
N	10	Número de intervalos
Tstart	-1 s	Instante de início da geração de eventos na simulação
Tstop	1000 s	Instante de encerramento da geração de eventos na simulação

Fonte: Elaborado pelo autor.

A Figura 6.19 (a) apresenta o resultado da simulação no domínio do tempo, com relação às máximas tensões verificadas após o evento simulado, ou seja, tensões instantâneas fase-terra. Para este gráfico, selecionou-se apenas o primeiro evento de abertura seguido de fechamento do disjuntor, cujas sobretensões são verificadas no fechamento. Já a Figura 6.19 (b) é evidenciado a o fechamento tripolar do disjuntor com detalhes, observa-se a aleatoriedade atribuído pelo MMC.

Figura 6.19 – Tensão medida no primário do TPC da LT-Alegrete 2 na Fase A (230 kV).



Fonte: Elaborado pelo autor.

Através do *foreign model* “SWITCHING”, exportou-se dados das máximas tensões fase-terra instantâneas, que estão divididos em 10 intervalos e evidenciados na Tabela 6.13. Dentre as 100 manobras, 50 são de abertura e 50 são de fechamento do disjuntor. As tensões foram monitoradas junto ao primário do TPC da linha de transmissão Alegrete 2.

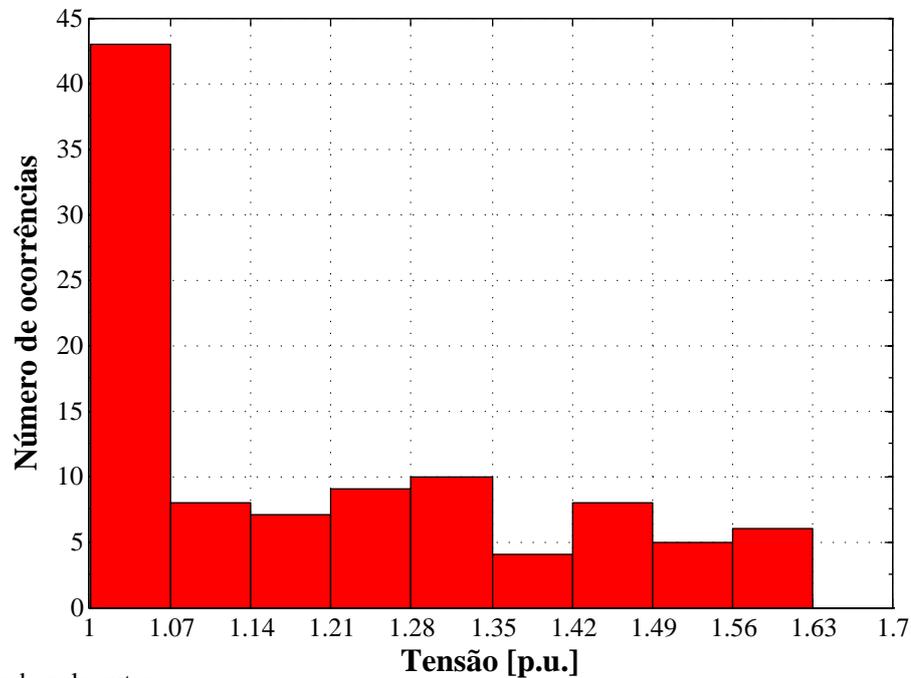
Com base nos dados apresentados na tabela, plotou-se o histograma da Figura 6.20. Este demonstra que 43% das manobras estão no intervalo de 1,00 p.u. a 1,07 p.u., indicando que para estes casos não há sobretensões, isso ocorre devido ao fato está sendo medida máxima tensão verificada após o evento. Para outros intervalos há regularidade de ocorrências.

Tabela 6.13 – Divisão de intervalos de tensão fase-terra medidos no primário do TPC para manobras LT-Alegrete 2.

Intervalo	Ocorrências
$1,00 \leq V \leq 1,07$	43
$1,07 < V \leq 1,14$	8
$1,14 < V \leq 1,21$	7
$1,21 < V \leq 1,28$	9
$1,28 < V \leq 1,35$	10
$1,35 < V \leq 1,42$	4
$1,42 < V \leq 1,49$	8
$1,49 < V \leq 1,56$	5
$1,56 < V \leq 1,63$	6
$1,63 < V \leq 1,70$	0

Fonte: Elaborado pelo autor.

Figura 6.20 – Histograma de frequências: tensão fase-terra medidos no primário do TPC para manobras LT-Alegrete 2.



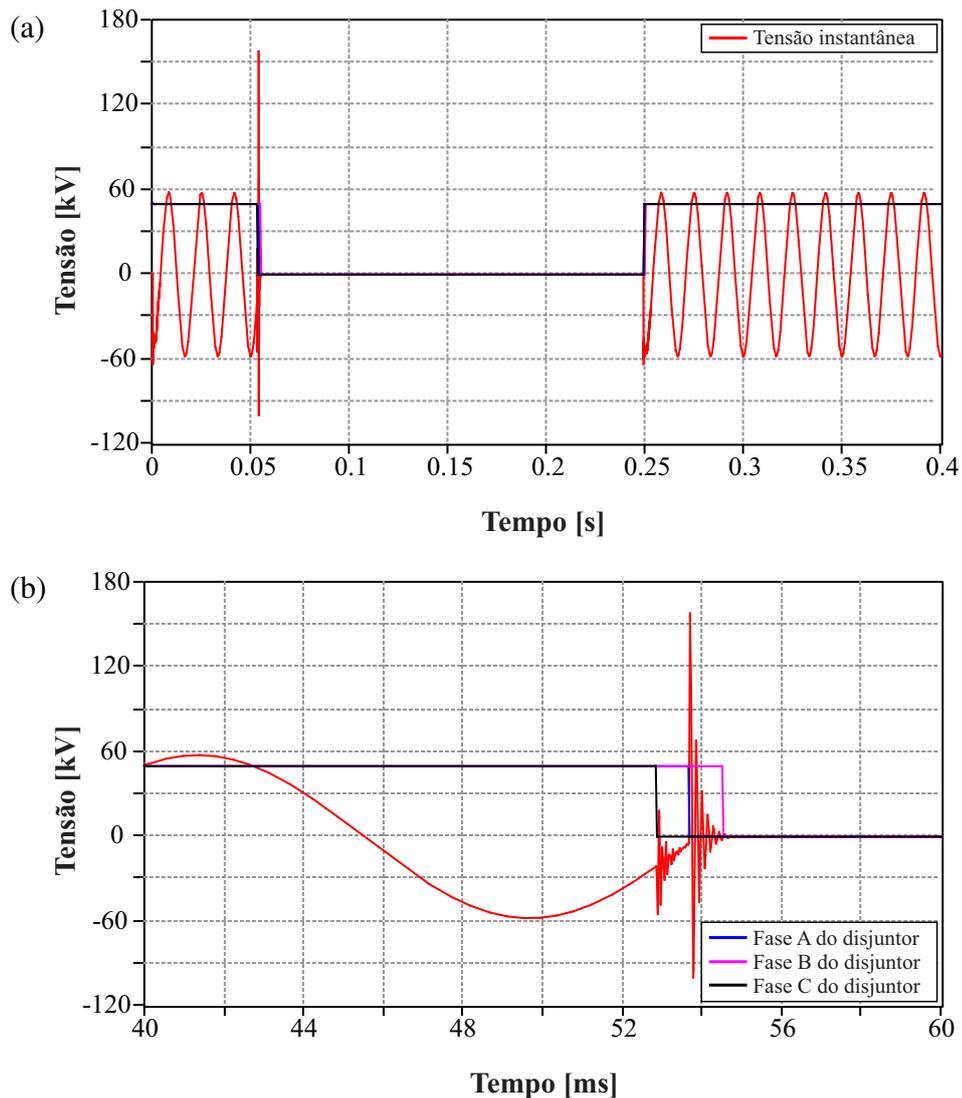
Fonte: Elaborado pelo autor.

6.4.2 Manobra de Carga

Os parâmetros específicos envolvendo simulações de conexão e desconexão da carga C1 SMA5, são idênticos aos valores apresentados na Tabela 6.12. À exceção do parâmetro que indica a tensão base do sistema no ponto de análise, sendo para este caso 69 kV.

A Figura 6.21 (a) exibe a tensão monitorada junto ao ponto de instalação do primário do TPC da linha de 69 kV (Fase A). No presente gráfico, visualiza-se uma abertura e um fechamento de disjuntor. Nota-se que há presença de sobretensões com grande amplitude na abertura do disjuntor, enquanto que no fechamento, estas não são evidentes.

Figura 6.21 – Tensão medida no primário do TPC para manobra da carga C1 SMA5 na Fase A (69 kV).



Fonte: Elaborado pelo autor.

Já a Figura 6.21 (b) mostra com detalhes o fechamento do disjuntor com sua dispersão de polos e as respectivas sobretensões. E através do gráfico, nota-se que no fechamento da fase A a sobretensão é maior em relação ao fechamento das outras fases.

A Tabela 6.14 contém os valores das máximas tensões fase-terra instantâneas, assim como o caso anterior, porém para o modelo de tensão de 69 kV. Tais dados estão divididos em 10 intervalos, sendo que foram realizadas 100 manobras, metade aberturas e metade fechamentos do disjuntor. As tensões foram monitoradas no primário do TPC da respectiva carga C1.

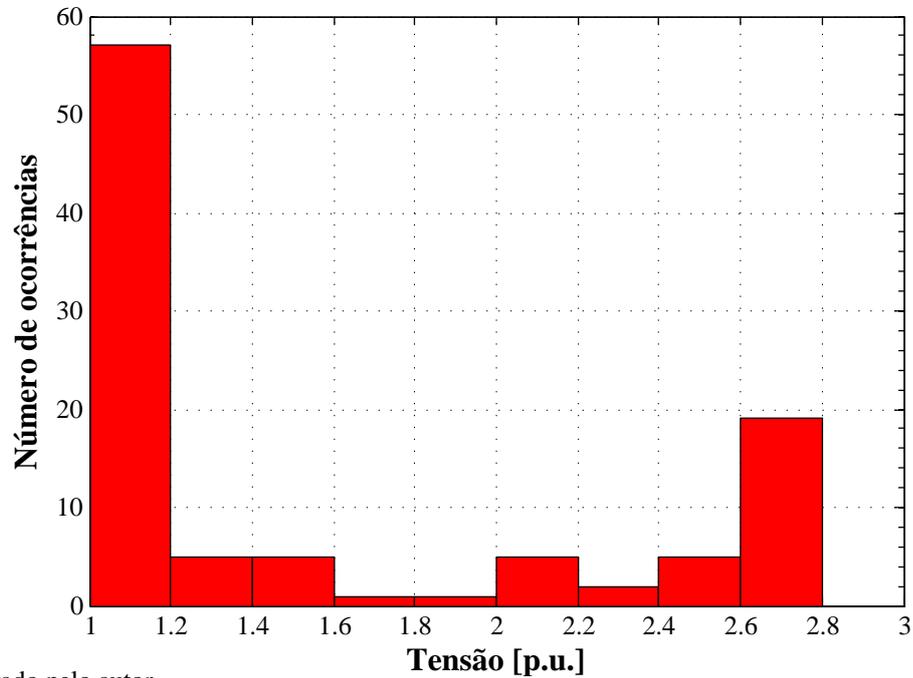
Tabela 6.14 – Valor máximo fase-terra para diferentes tipos de manobra.

Intervalo	Ocorrências
$1,00 \leq V \leq 1,20$	57
$1,20 < V \leq 1,40$	5
$1,40 < V \leq 1,60$	5
$1,60 < V \leq 1,80$	1
$1,80 < V \leq 2,00$	1
$2,00 < V \leq 2,20$	5
$2,20 < V \leq 2,40$	2
$2,40 < V \leq 2,60$	5
$2,60 < V \leq 2,80$	19
$2,80 < V \leq 3,00$	0

Fonte: Elaborado pelo autor.

A partir dos valores da Tabela 6.14, elaborou-se o histograma da Figura 6.22. Observa-se que 57% das ocorrência encontram-se na faixa de 1,00 p.u. a 1,2 p.u., assim como no caso anterior, pois são medidas máximas de tensão verificadas após o evento. Há 19% dos casos no intervalo de 2,6 p.u. e 2,8 p.u., com certa regularidade no restante dos intervalos.

Figura 6.22 – Histograma de frequências: tensão fase-terra medidos no primário do TPC para manobra da carga C1 SM-5 na Fase A (69 kV).



Fonte: Elaborado pelo autor.

6.5 ESTUDO DE DESCARGAS ATMOSFÉRICAS

Eventos como descargas atmosféricas geram transitórios em altas frequências. Em vista disso, o modelo do sistema a ser simulado necessita ser compatível a esta faixa de frequências. Nos exemplos de aplicações da sequência, utilizou-se o modelo para altas frequências da subestação SMA3.

Cabe ressaltar que para modelos de subestação, não foram representados os cabos guarda. No entanto, sabe-se que o raio da esfera de proteção destes cabos é proporcional a amplitude da corrente de descarga atmosférica, de modo que correntes de amplitude maiores tem menor probabilidade de atingirem diretamente os condutores das fases do sistema. Desta forma, mesmo que os cabos de guarda não sejam modelados, pode-se realizar simulações representativas de descargas diretas reduzindo o valor de sua corrente máxima.

Considerando os eventos de descargas atmosféricas, optou-se por utilizar dois pontos para simulações. Um dos pontos, corresponde à descarga direta atingindo ao trecho de interligação do transformador TR-1 com o setor de 69 kV. O outro ponto, considera descarga direta na chegada da linha de transmissão Dona Francisca 2 no setor de 230 kV. A forma de onda escolhida para as simulações foi a de Heidler, outros parâmetros específicos e os resultados das respectivas simulações são apresentados a seguir.

6.5.1 Descargas Atmosféricas no Setor 230 kV

Nesta análise, foram realizadas 100 de descargas atmosféricas diretas na Fase A. Os parâmetros específicos para tal simulação está indicado na Tabela 6.15.

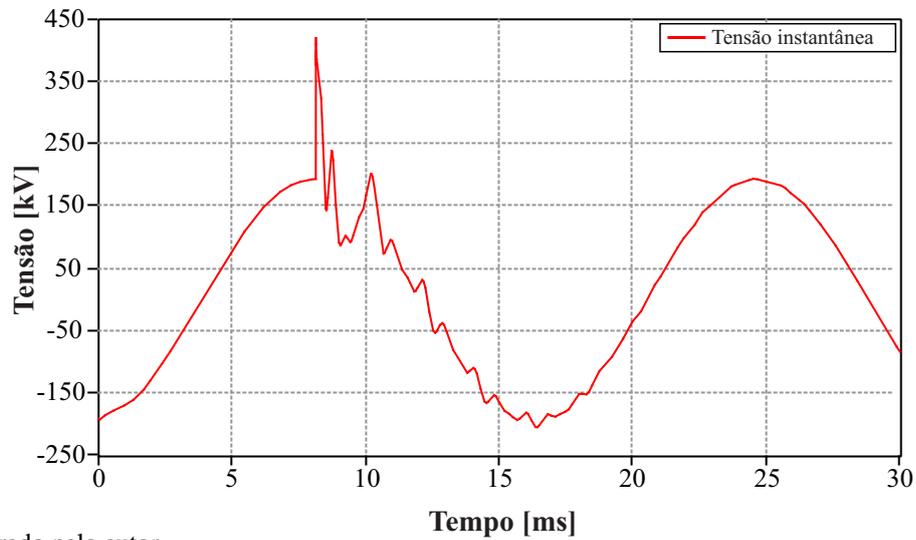
Tabela 6.15 – Parâmetros utilizados no estudo de descargas atmosféricas no Setor 230 kV.

Parâmetro	Valor	Comentário
Seed	97	Valor inicial da sequência aleatória do MMC
NumLG	100	Número de descargas atmosféricas
Waveform	0	Forma de onda da corrente (Heidler)
n	2	Parâmetro de forma
IPm	31,10 kA	Corrente máxima (mediana)
IPsd	0,484 kA	Corrente máxima (desvio padrão)
TFm	$3,83 \times 10^{-6}$	Tempo de frente da onda (mediana)
TFsd	$0,553 \times 10^{-6}$	Tempo de frente da onda (desvio padrão)
THm	$77,5 \times 10^{-6}$	Tempo de meia cauda da onda (mediana)
THsd	$0,577 \times 10^{-6}$	Tempo de meia cauda da onda (desvio padrão)
Vb	230×10^3 V	Tensão base do sistema no ponto de análise
Sb	100×10^6 VA	Potência base do sistema
N	10	Número de intervalos
Tstart	-1 s	Instante de início da geração de eventos na simulação
Tstop	1000 s	Instante de encerramento da geração de eventos na simulação

Fonte: Elaborado pelo autor.

A Figura 6.23 apresenta tensão fase-terra medida junto a Fase A da Barra A de 230 kV. Através deste gráfico, pode-se verificar o transitório ocasionado na tensão devido a incidência de uma descarga atmosférica direta, cujo tempo foi definido pelo MMC. Importante ressaltar que nesta figura, tem-se a representação de um caso dentre os 100 simulados.

Figura 6.23 – Tensão fase-terra medida na Barra A do Setor 230 kV.



Fonte: Elaborado pelo autor.

Para os 100 eventos simulados, exportou-se através do *foreign model* “LIGHTNING” os valores de tensão fase-terra medidos na Barra A de 230 kV. Estes valores foram divididos em 10 intervalos, conforme é apresentado na Tabela 6.16.

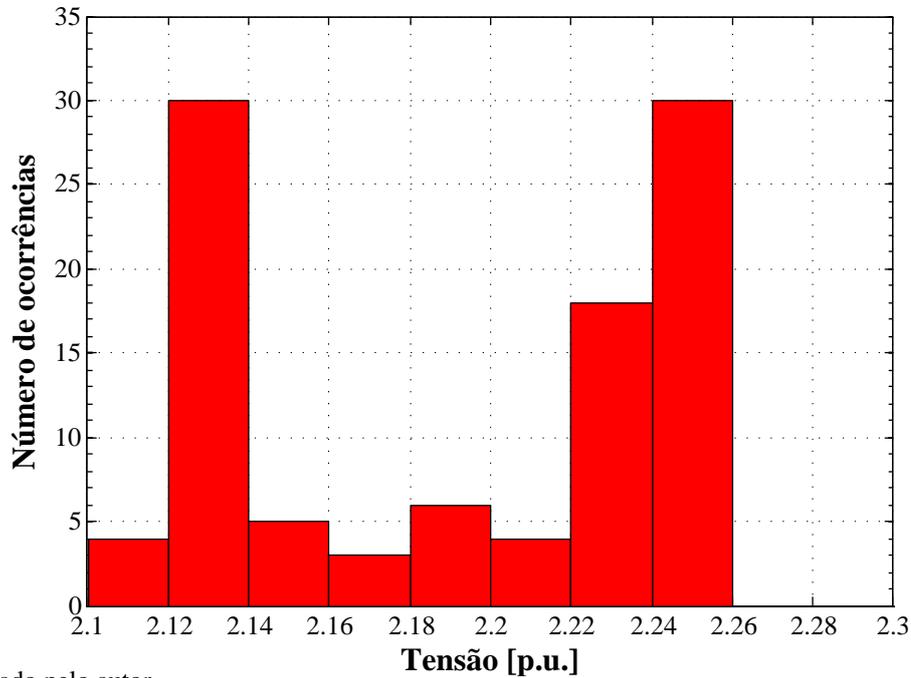
A partir dos dados fornecidos na tabela, plotou-se o histograma de frequências da Tabela 6.16. Através deste histograma, verifica-se que 30% dos eventos geraram sobretensões no intervalo de 2,12 p.u. a 2,14 p.u. e 48% geraram sobretensões entre 2,2 p.u. a 2,26 p.u..

Tabela 6.16 – Valores de tensão fase-terra na Fase A da Barra A de 230 kV em decorrência de descargas atmosféricas.

Intervalo	Ocorrências
$2,10 \leq V \leq 2,12$	4
$2,12 < V \leq 2,14$	30
$2,14 < V \leq 2,16$	5
$2,16 < V \leq 2,18$	3
$2,18 < V \leq 2,20$	6
$2,20 < V \leq 2,22$	4
$2,22 < V \leq 2,24$	18
$2,24 < V \leq 2,26$	30
$2,26 < V \leq 2,28$	0
$2,28 < V \leq 2,30$	0

Fonte: Elaborado pelo autor.

Figura 6.24 – Histograma de frequências:tensão fase-terra na Fase A da Barra A de 230 kV em decorrência de descargas atmosféricas.

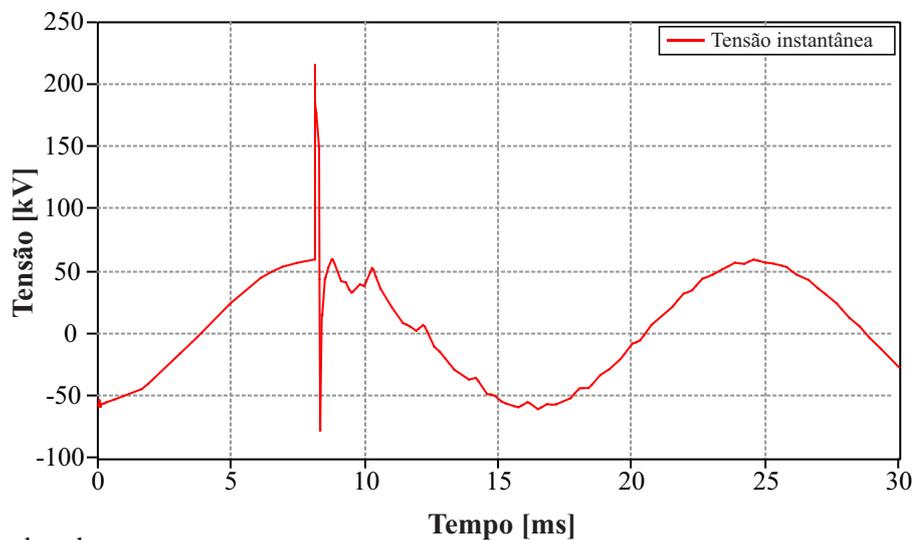


Fonte: Elaborado pelo autor.

6.5.2 Descargas Atmosféricas no Setor 69 kV

Os dados utilizados para configuração de simulações envolvendo descargas atmosféricas no Setor de 69 kV, são idênticos aos valores apresentados na Tabela 6.15. À exceção do parâmetro que indica a tensão base do sistema no ponto de análise, sendo para este caso 69 kV.

Figura 6.25 – Tensão fase-terra medida no TPI da Barra Principal do Setor 69 kV.



Fonte: Elaborado pelo autor.

A Figura 6.25 apresenta a tensão fase-terra instantânea medida junto a Fase A do TPI da Barra Principal de 69 kV. Pela análise do gráfico, percebe-se que há presença transitório de tensão devido a incidência de uma descarga atmosférica direta.

Para esta simulação, assim com a anterior, foram realizados um total de 100 descargas atmosféricas, cujos dados exportados resultaram nos dados da Tabela 6.17. Estes dados foram organizados em 10 intervalos.

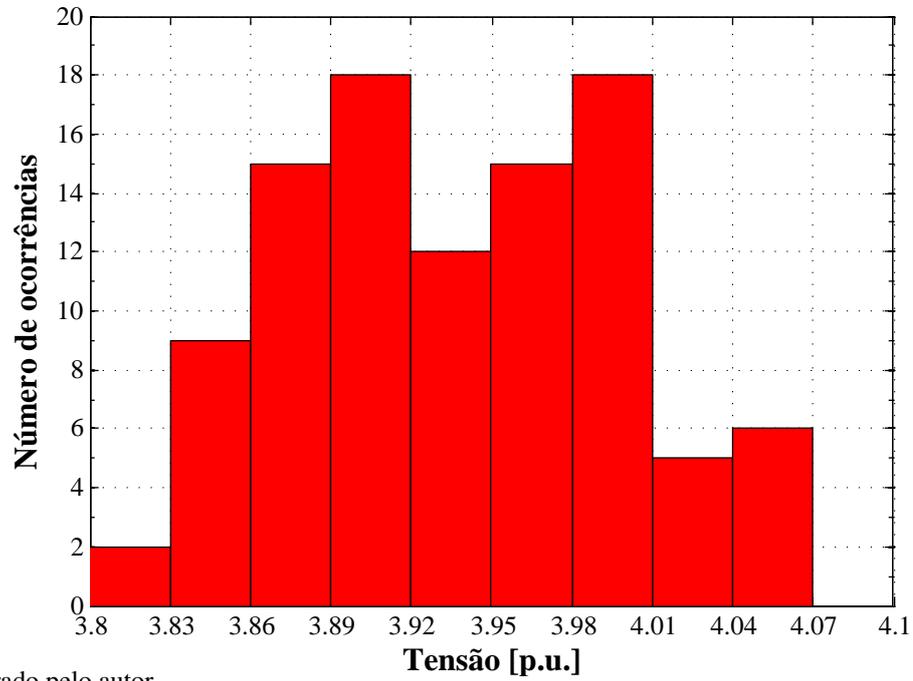
Tabela 6.17 – Valores de tensão fase-terra medida na Fase A do TPI da Barra Principal do Setor 69 kV em decorrência de descargas atmosféricas.

Intervalo	Ocorrências
$3,80 \leq V \leq 3,83$	2
$3,83 < V \leq 3,86$	9
$3,86 < V \leq 3,89$	15
$3,89 < V \leq 3,92$	18
$3,92 < V \leq 3,95$	12
$3,95 < V \leq 3,98$	15
$3,98 < V \leq 4,01$	18
$4,01 < V \leq 4,04$	5
$4,04 < V \leq 4,07$	6
$4,07 < V \leq 4,10$	0

Fonte: Elaborado pelo autor.

Com base nos valores expostos na tabela, tem-se o histograma da Figura 6.26. Através deste histograma, verifica-se que as descargas atmosféricas simuladas geraram tensões entre 3,80 p.u. e 4,07 p.u.. No entanto, a maior parcela dos eventos, cerca de 78%, ocasionam tensões entre 3,86 p.u. e 4,01 p.u..

Figura 6.26 – Histograma de frequências: tensão fase-terra medida na Fase A do TPI da Barra Principal do Setor 69 kV em decorrência de descargas atmosféricas.



Fonte: Elaborado pelo autor.

7 CONCLUSÃO

O trabalho utilizou o clássico método de Monte Carlo para incluir em simulações no ATP-EMTP aleatoriedades e incertezas peculiares aos eventos causadores de transitórios eletromagnéticos em subestações. Dentre os eventos contemplados no estudo estão presentes ocorrência de curtos-circuitos, energização de transformadores, manobras de disjuntores e descargas atmosféricas, cada um representado pela respectiva distribuição de probabilidade. A metodologia foi desenvolvida e implementada no ATP-EMTP através do recurso de *foreign model* e, ao final, pode ser acessada por meio do pré-processador gráfico ATPDraw.

Tem-se conhecimento de que a capacidade de personalização oferecida pelo ATPDraw permite que o usuário possa implementar alguns dos processos estocásticos utilizando a chave estatística (*Statistic Switch*). Entretanto, necessitaria da associação desta chave com outras ferramentas de controle, tornando o processo de simulação mais difícil e complexo, demandando mais tempo para realização destas implementações. Para alguns grupos de usuários do ATP-EMTP, principalmente fora do âmbito acadêmico, essa morosidade muitas vezes inviabiliza a realização destas simulações.

Em vista disso, a metodologia implementada ofereceu uma versão personalizada do ATP-EMTP, a qual é composta do conteúdo original do software, acrescido com as informações referentes às novas funcionalidades desenvolvidas. Dessa forma, o usuário pode utilizá-las diretamente no ATPDraw, sem necessidade de ferramentas externas, simplificando os processos de simulações em análises estocásticas de transitórios eletromagnéticos.

Considerando que em análises estocásticas utilizando o MMC o número de simulações influi diretamente na qualidade dos resultados, faz-se necessário a simulação em massa dos eventos e posterior análise estatística. Para isso, outra vantagem proporcionada pela metodologia, consiste em fornecer os valores da variável de interesse já divididos em intervalos de frequência.

Através dos exemplos de aplicações, abordados no capítulo 6, pode-se constatar que a metodologia é eficiente para modelagem de eventos aleatórios, também para realizar simulações em massa e posterior análise estatística de tais dados. Neste contexto, o estudo desenvolvido agregou novas funcionalidades ao ATP-EMTP, bem como estabeleceu um procedimento de acesso e utilização destes recursos através do ATPDraw. Os recursos e ferramentas incorporadas

auxiliam a realização de simulações e análises específicas de transitórios eletromagnéticos em subestações.

7.1 PROPOSTAS DE CONTINUIDADE

Uma importante funcionalidade que não estava especificada na etapa de definição dos recursos oferecidos, consiste na implementação a análise harmônica da corrente de energização através da Transformada Discreta de Fourier (*Discrete Fourier Transform - DFT*) para estudos de energização de transformadores. Vale salientar ainda, que a metodologia desenvolvida para simulação de manobras pode ser utilizada em análises de energização e desenergização de Linhas de Transmissão, bastando que estas sejam representadas através de um dos modelos disponíveis no ATP-EMTP, como o modelo *J. Marti*.

7.2 PUBLICAÇÕES

Durante o desenvolvimento da metodologia apresentada, realizou-se a publicação de trabalhos em dois eventos da área de sistemas de potência. Estas publicações são descritas a seguir:

- 1. Título:** Stochastic analysis of transient overvoltages at substations by applying the Monte Carlo method integrated with ATP.

Autores: KRÄULICH, L.; MARCHESAN, A. C.; MARIOTTO, L.; LOOSE, F.; CAMARGO, M. C.; MARCHESAN, G.; CARDOSO JR., G.

Evento: International Conference on Power Systems Transients (IPST).

Local: Cavtat - HR.

Data: Junho, 2015.

Referência: Kräulich et al. (2015).

- 2. Título:** Desenvolvimento de um software pré-processador gráfico para análise estocástica de transitórios eletromagnéticos em subestações.

Autores: MARCHESAN, A. C.; KRAULICH, L., MARIOTTO, L.; CARDOSO JR., G.; LOOSE, F.; ALVES Fº, P. F.

Evento: XI Latin-American Congress on Electricity Generation and Transmission (CLAGTEE).

Local: São José dos Campos.

Data: Novembro, 2015.

Referência: Marchesan et al. (2015).

REFERÊNCIAS

- Alternative Transients Program – Rule Book. Canadian/American EMTP User Group, 1987–1998.
- ANDERSON, A.; J.ERIKSSON, A. A summary of lightning parameters for engineering application. **International Conference on Large High Voltage Electric Systems**, 1980.
- ARAÚJO, A. E. A.; NEVES, W. L. A. **Cálculo de transitórios eletromagnéticos em sistemas de energia**. Belo Horizonte, MG: Editora UFMG, 2005.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6939**: Coordenação de isolamento - procedimento. Rio de Janeiro, 2000. 15 p.
- BERGER, K. et al. Parameters of lightning flashes. **Electra**, p. 23–37, 1975.
- BORTOLOSSI, H. J. **Números (pseudo) aleatórios, probabilidade geométrica, métodos de Monte Carlo e estereologia**. Niterói: Universidade Federal Fluminense, 2012. 5 p. Projeto KLEIN de matemática - Artigos. Disponível em: <[http://klein.sbm.org.br/wp-content/uploads/2012/10/nu\\$klein.pdf](http://klein.sbm.org.br/wp-content/uploads/2012/10/nu$klein.pdf)>. Acesso em: 04 jan. 2014.
- CAMM, E.; NATARAJAN, R.; MARTINEZ, J. A. Comparison of statistical switching results using gaussian, uniform and systematic switching approaches. **IEEE Transactions on Power Delivery**, abr. 2000.
- Companhia Estadual de Energia Elétrica – Geração e Transmissão. **Diagrama de operação**: Subestação Santa Maria 3. [S.l.], 2014.
- CONSEIL INTERNATIONAL DES GRANDS RÉSEAUX ÉLECTRIQUES. **Guide to procedures for estimating the lightning performance of transmission lines**. Paris, FR, 1991. 64 p.
- CVETIC, J. M.; HEIDLER, F. A class of analytical functions to study the lightning effects associated with the current front. **European Transactions on Electrical Power Engineering (ETEP)**, v. 12, mar. 2002.
- CVETIC, J. M.; STANIC, B. V.; HEIDLER, F. Calculation of lightning current parameters. **IEEE Transactions on Power Delivery**, v. 14, abr. 1999.
- D’AJUZ, A. et al. **Transitórios elétricos e coordenação de isolamento**. 1. ed. Rio de Janeiro, RJ: Editora Universitária Universidade Federal Fluminense, 1987. 425 p.
- DEVORE, J. L. **Probabilidade e estatística: para engenharia e ciências**. 4. ed. São Paulo: Pioneira Thomson Learning, 2006. 706 p.
- Electric Power Research Institute, EMTP Development Coordination Group. **Electromagnetic Transients Program (EMTP) – Workbook**. Palo Alto – US, 1986.
- FARIAS, P.; GUTIERRES, L. **Curso – ATP**. Santa Maria, RS, BR, 2012.
- GATTA, F. M. et al. Monte carlo evaluation of the impact of subsequent strokes on backflashover rate. **Energies**, v. 9, n. 3, p. 139, 2016. ISSN 1996-1073. Disponível em: <<http://www.mdpi.com/1996-1073/9/3/139>>.

- GENTLE, J. E. **Random number generation and Monte Carlo Methods**. 2. ed. New York, NY, US: Springer, 2003.
- GUTMANN, P. Software generation of practically strong random numbers. In: CONGRESSO BRASILEIRO DE CIÊNCIA DO SOLO, 7., 1998. **Anais...** San Antonio, US: USENIX, 1997. Disponível em: <<https://www.usenix.org/legacy/publications/library/proceedings/sec98/gutmann.html>>. Acesso em: 06 jan. 2014.
- HEIDLER, F. et al. Parameters of lightning current given in IEC 62305: Background experience and outlook. **International conference on lightning protection**, Uppsala, SE, v. 29, jun. 2008.
- HOIDALEN, H. K. ATPDraw for Windows Ver. 1.1. **EEUG News**, v. 3, n. 2, p. 18–23, maio 1997.
- HOIDALEN, H. K. **ATPDraw – The graphical preprocessor to ATP**. Norway, 2016. Disponível em: <<http://www.atpdraw.net>>. Acesso em: 11 jan. 2016.
- INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC TR 60071-4, Insulation co-ordination – Part 4: computational guide to insulation co-ordination and modelling of electrical networks**. [S.l.], 2004. 122 p.
- KIMBARK, E. W. **Power system stability: Power circuit breakers and protective relays**. USA: John Willey and Sons, 1948.
- KNUTH, D. E. **The art of computer programming: seminumerical algorithms**. 3. ed. Reading, US: Addison-Wesley, 1997. v. 2. 762 p.
- KRÄULICH, L. et al. Stochastic analysis of transient overvoltages at substations by applying the monte carlo method integrated with atp. In: INTERNATIONAL CONFERENCE ON POWER SYSTEMS TRANSIENTS, 12th., 2015, Cavtat – HR. **Anais...** Cavtat – HR: IEEE, 2015. p. 1–7.
- Laboratório de Análise e Proteção de Sistemas Elétricos, Centro de Estudos em Energia e Sistemas de Potência. **Relatório final: projeto de P&D CEEE-GT 9947883**. Santa Maria, 2016.
- LAZO, P. P. D. Monte carlo para problemas determinísticos: sistemas lineares. **XIII Encontro Regional de Matemática Aplicada e Computacional**, Cascavel, PR, v. 1, 2009.
- Lightning and Insulator Subcommittee of the T&D Committee. Parameters of lightning strokes: a review. **IEEE Transactions on Power Delivery**, v. 20, n. 1, p. 346–358, jan. 2005.
- LUO, X.; KEZUNOVIC, M. Interactive protection system simulation using atp models and c++. **Transmission and distribution Conference and Exhibition IEEE PES**, Dallas, EUA, v. 1, n. 9087076, p. 874–879, mai. 2006.
- MARCHESAN, A. C. et al. Desenvolvimento de um software pré-processador gráfico para análise estocástica de transitórios eletromagnéticos em subestações. In: CONGRESSO LATINO-AMERICANO DE GERAÇÃO E TRANSMISSÃO DE ENERGIA ELÉTRICA, 11., 2015, São José dos Campos. **Anais...** São José dos Campos: UNESP, 2015.
- MARTINEZ, J. A.; CASTRO-ARANDA, F. Lightning performance analysis of overhead transmission lines using the emtp. **IEEE Transactions on Power Delivery**, v. 20, n. 3, p. 2200–2210, July 2005. ISSN 0885-8977.

MEYER, W. S.; LIU, T. huei. The birth and evolution of ATP and LEC. **Can/Am EMTP News**, v. 88-1, p. 4–5, set. 1988.

MODELS PRIMER: For first time models users. Version 1. Vancouver,CA, 1988.

MOORE, D. S.; NOTZ, W. I.; FLIGNER, M. A. **A estatística básica e sua prática**. 6. ed. Rio de Janeiro, RJ: LTC, 2014. 582 p.

MOREIRA, L. A.; SOUZA, R. de. Métodos computacionais para geração de sinais aleatórios aplicados a sistemas de transmissão digital. **Anais do Congresso de Iniciação Científica do INATEL**, Santa Rita do Sapucaí, MG, v. 1, p. 188–194, abr. 2012.

NAKAGOMI, R. M. et al. Desenvolvimento e implementação de dispositivos de proteção. **Seminário Técnico de Proteção no ATP**, Rio de Janeiro, BR, v. 1, p. 1–6, jul. 2005.

NUNES, R. R. **Coordenação de isolamento para transitórios de manobra considerando a forma de onda das sobretensões**. 122 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal de Minas Gerais, Belo Horizonte, MG, 2006.

Power Engs. **ATPDesigner – Design and simulation of electrical power networks**. 2016. Disponível em: <<http://www.atpdesigner.de>>. Acesso em: 11 jan. 2016.

PRESS, W. H. et al. **Numerical recipes: the art of scientific computing**. 3. ed. New York, US: Cambridge University Press, 2007. 1235 p.

PRIKLER, L.; HOIDALEN, H. K. ATPDraw for Windows – User’s Manual. **EEUG News**, v. 4, n. 3-4, p. 25–32, ago. 1998.

Randomness and Integrity Services Ltd. **Testimonials – random sampling**. Dublin, IE, 2012. Disponível em: <<http://www.random.org>>. Acesso em: 01 jan. 2014.

RUBINSTEIN, R. Y. **Simulation and the Monte Carlo Method**. CA: John Willey and Sons, 1981.

SARAN, M. A. M. **Sobretensões de origem atmosférica em linhas de média tensão**. 117 p. Dissertação (Mestrado em Ciências em Engenharia Elétrica) — Universidade Federal de Itajubá, Itajubá, 2009.

SHIGIHARA, M. **Avaliação de correntes de descargas atmosféricas através de medições diretas em estruturas altas**. 156 p. Dissertação (Mestrado em Energia) — Universidade de São Paulo, São Paulo, 2005.

SÓBOL, I. M. **Método de Monte Carlo**. New York, NY, US: Editorial MIR, 1983.

TASCHINI, A.; CLERICI, A. Influence of switching surges of the switched line zero sequence impedance. **Transmission and distribution Committee of the IEEE**, v. 70, abr. 1970.

VIALI, L. **Uma introdução aos métodos de geração de números e variáveis aleatórias para aplicações em simuladores**. Porto Alegre, RS: Pontifícia Universidade Católica do Rio Grande do Sul, 2014. Especialização em Energias Renováveis - Material Didático. Disponível em: <http://www.pucrs.br/famat/viali/especializa/mia_ima_fafis/material/ead/outros/Geracao_de_numeros_e_variaveis_aleatorias.pdf>. Acesso em: 27 jan. 2016.

VISACRO, S. F. **Descargas atmosféricas: uma abordagem de engenharia**. São Paulo, BR: Editora Altliber, 2005. 268 p.

WESTINGHOUSE, C. S. E. of the. **Electrical transmission and distribution reference book**. Pennsylvania, USA: Westinghouse Electric Corporation, 1950.

YATES, R. D.; GOODMAN, D. J. **Probability stochastic process**. 2. ed. US: John Wiley Sons, 2005. 519 p.

YORIYAZ, H. Método de monte carlo: princípios e aplicações em física médica. **Revista Brasileira de Física Médica**, Natal, v. 3, n. 1, p. 141–149, out. 2009.

ZANETTA, L. C. **Transitórios eletromagnéticos em sistemas de potência**. 1. ed. São Paulo, BR: Editora da Universidade de São Paulo, 2003. 712 p.

APÊNDICE A – CÓDIGO FONTE EM LINGUAGEM C++

```

1 //=====
2 // Bibliotecas
3 //=====
4 #include "math.h"
5 #include "string.h"
6 #include "stdio.h"
7 #include "stdlib.h"
8 //=====
9 // Definicao de constantes
10 //=====
11 #define pi 3.14159265359
12 //=====
13 // Prototipos de funcoes
14 //=====
15 extern "C"
16 {
17     void lightning_ini__(double[], double[], double[], double[]);
18     void lightning_exe__(double[], double[], double[], double[]);
19     void switching_ini__(double[], double[], double[], double[]);
20     void switching_exe__(double[], double[], double[], double[]);
21     void shortcircuit_ini__(double[], double[], double[], double[]);
22     void shortcircuit_exe__(double[], double[], double[], double[]);
23     void inrush_ini__(double[], double[], double[], double[]);
24     void inrush_exe__(double[], double[], double[], double[]);
25 }
26 //=====
27 // Declaracao de variaveis gobais
28 //=====
29 FILE *pFile;           // Ponteiro para criar arquivo e exportar os resultados
30 double Tstart;        // Tempo de inicio da simulacao de eventos
31 double Tstop;         // Tempo de encerramento da simulacao de eventos
32 double A;             // Limite inferior do sub-intervalo de simulacao de um evento
33 double B;             // Limite superior do sub-intervalo de simulacao de um evento
34 double timeRand = 0.0; // Instante de ocorrencia de um evento
35 double Ta = -1.0;     // Instante de atuacao do disjuntor da fase A
36 double Tb = -1.0;     // Instante de atuacao do disjuntor da fase B
37 double Tc = -1.0;     // Instante de atuacao do disjuntor da fase C
38 double Vmax = 0.0;    // Maxima sobretensao verificada apos um evento
39 double *pDataRMS;     // Ponteiro utilizado para o calculo do valor eficaz
40 double dataRMS[1000];
41 bool event = true;     // Gerar um evento
42 bool firstEvent = true; // Primeiro evento
43 bool lastEvent = true; // Ultimo evento
44 bool exportData = false; // Exportar dados
45 bool windowRMS = false; // Verifica se a janela possui um ciclo completo
46 int pointsRMS = 0;     // Numero de pontos na janela
47 int pointsCycle;      // Numero de pontos em um ciclo da forma de onda fundamental
48 //-----
49 // Variaveis globais do foreign model LIGHTNING
50 double t = 0.0;       // Variavel temporal para a geracao da forma de onda
51 double IP;           // Corrente maxima da forma de onda
52 double TF;           // Tempo de frente da forma de onda
53 double TC;           // Tempo de meia cauda da forma de onda
54 double factor;       // Fator de correcao de amplitude da forma de onda
55 //-----
56 // Variaveis globais do foreign model SWITCHING
57 bool switchingA = false; // Manobra a fase A do disjuntor
58 bool switchingB = false; // Manobra a fase B do disjuntor
59 bool switchingC = false; // Manobra a fase C do disjuntor
60 bool firstSwitchingA = true; // Primeira manobra da fase A do disjuntor
61 bool firstSwitchingB = true; // Primeira manobra da fase B do disjuntor
62 bool firstSwitchingC = true; // Primeira manobra da fase C do disjuntor
63 //-----
64 // Variaveis globais do foreign model SHORTCIRCUIT
65 bool shortCircuit = false; // Habilita a ocorrencia de um curto-circuito

```

```

66 int shortA = 0;           // Curto-circuito na fase A
67 int shortB = 0;           // Curto-circuito na fase B
68 int shortC = 0;           // Curto-circuito na fase C
69 int openA = 1;           // Abertura da fase A do disjuntor
70 int openB = 1;           // Abertura da fase B do disjuntor
71 int openC = 1;           // Abertura da fase C do disjuntor
72 //=====
73 // Geracao de numeros aleatorios
74 //=====
75 typedef unsigned long long Ullong;
76 Ullong u, v, w;
77 // Retorna um numero aleatorio inteiro de 64 bits
78 Ullong int64()
79 {
80     Ullong x;
81     u = u*2862933555777941757LL + 7046029254386353087LL;
82     v ^= v>>17;
83     v ^= v<<31;
84     v ^= v>>8;
85     w = 4294957665U*(w & 0xffffffff) + (w >> 32);
86     x = u^(u<<21);
87     x ^= x >> 35;
88     x ^= x << 4;
89     return (x + v) ^ w;
90 }
91 // Define o valor inicial (semente) para a geracao da sequencia aleatoria
92 void seed(Ullong s)
93 {
94     v = 4101842887655102017LL;
95     w = 1;
96
97     u=s^v;
98     int64();
99     v = u;
100    int64();
101    w = v;
102    int64();
103 }
104 // Retorna um numero aleatorio de dupla precisao entre 0 e 1
105 double random()
106 {
107     return 5.42101086242752217E-20 * int64();
108 }
109 //=====
110 // Gera uma variavel aleatoria com distribuicao uniforme
111 //=====
112 double uniformDist(double min, double max, double r)
113 {
114     return (min + (max - min)* r);
115 }
116 //=====
117 // Gera uma variavel aleatoria com distribuicao normal
118 //=====
119 double normalDist(double m, double sd, double r1, double r2)
120 {
121     return(m+sd*(sqrt(-2*log(r1))*cos(2*pi*r2)));
122 }
123 //=====
124 // Gera uma variavel aleatoria com distribuicao log-normal
125 //=====
126 double logNormalDist(double m, double sd, double r1, double r2)
127 {
128     return(exp(m+sd*normalDist(0, 1, r1, r2)));
129 }
130 //=====
131 // Valor eficaz calculado pelo metodo de integracao trapeizodal
132 //=====
133 double trueRMS(double *pv, double dt)
134 {
135     long int i;

```

```

136     double *v2, *vSum;
137     double RMS;
138
139     v2 = (double*) malloc(pointsCycle*sizeof(double));
140     vSum = (double*) malloc(pointsCycle*sizeof(double));
141
142     /*v2 = *pv;
143     *v2 = 0;
144     *vSum = *v2;
145
146     for(i = 1; i < pointsCycle; i++)
147     {
148         pv++;
149         v2++;
150         vSum++;
151
152         *v2 = (0.5)*dt*(pow(*pv, 2) + pow(*(pv-1), 2));
153         *vSum = *(vSum-1) + *v2;
154     }
155     RMS = sqrt(*vSum/(dt*(pointsCycle-1)));
156
157     v2 = v2 - pointsCycle + 1;
158     vSum = vSum - pointsCycle + 1;
159     free(vSum);
160     free(v2);
161     vSum = NULL;
162     v2 = NULL;
163
164     return RMS;
165 }
166 /*=====
167  * Mapeamento de dados, entradas, saidas e variaveis do foreign model LIGHTNING *
168 -----*/
169 -- DADOS
170 * xdata_ar[0]: valor inicial da sequencia aleatoria (semente)
171 * xdata_ar[1]: numero total de descargas atmosfericas simuladas
172 * xdata_ar[2]: forma de onda da corrente de descarga (Heidler ou Dupla Rampa)
173 * xdata_ar[3]: parametro da forma de onda Heidler
174 * xdata_ar[4]: corrente maxima da forma de onda (media)
175 * xdata_ar[5]: corrente maxima da forma de onda (desvio padrao)
176 * xdata_ar[6]: tempo de frente da forma de onda (media)
177 * xdata_ar[7]: tempo de frente da forma de onda (desvio padrao)
178 * xdata_ar[8]: tempo de meia cauda da forma de onda (media)
179 * xdata_ar[9]: tempo de meia cauda da forma de onda (desvio padrao)
180 * xdata_ar[10]: tensao base do sistema
181 * xdata_ar[11]: potencia base do sistema
182 * xdata_ar[12]: tempo de inicio da geracao de eventos
183 * xdata_ar[13]: tempo de encerramento da geracao de eventos
184 -- ENTRADAS
185 * xin_ar[0]: tempo corrente do ATP-EMTP
186 * xin_ar[1]: tempo de inicio da simulacao no ATP-EMTP
187 * xin_ar[2]: tempo final da simulacao no ATP-EMTP
188 * xin_ar[3]: passo da simulacao no ATP-EMTP
189 * xin_ar[4]: tensao monitorada em um ponto do sistema
190 -- SAIDAS E VARIAVEIS
191 * xout_ar[0] = xvar_ar[0]: corrente de descarga atmosferica
192 /*-----*/
193 void lightning_ini__(double xdata_ar[], double xin_ar[], double xout_ar[],
194                    double xvar_ar[])
195 {
196     printf("Starting Foreign Model LIGHTNING\n");
197     Ullong initial = (Ullong)xdata_ar[0];
198     double Vb = xdata_ar[10];
199     double Sb = xdata_ar[11];
200     Tstart = xdata_ar[12];
201     Tstop = xdata_ar[13];
202     double ATPtimeStart = xin_ar[1];
203     double ATPtimeStop = xin_ar[2];
204
205     // Define o valor inicial da sequencia aleatoria

```

```

206     seed(initial);
207
208     // Verifica se o tempo de inicio da simulacao de eventos e menor ou igual
209     // ao tempo de inicio da simulacao no ATP-EMTP
210     if(Tstart <= ATPtimeStart)
211     {
212         Tstart = ATPtimeStart;
213     }
214
215     // Verifica se o tempo de encerramento da simulacao de eventos e maior ou igual
216     // ao tempo de encerramento da simulacao no ATP-EMTP
217     if(Tstop >= ATPtimeStop)
218     {
219         Tstop = ATPtimeStop;
220     }
221
222     // Inicializa o limite superior do subintervalo para o primeiro evento
223     B = Tstart;
224
225     // Abre um novo arquivo para salvar os dados
226     pFile = fopen("dados_estocasticos.txt", "w+");
227     fprintf(pFile, "%s\n", "Tensao");
228     fprintf(pFile, "%.2f\n", Vb);
229     fprintf(pFile, "%.2f\n", Sb);
230
231     // Faz com que a corrente de descarga atmosferica inicie em zaro
232     xvar_ar[0] = 0;
233     // Atualizada a saida
234     xout_ar[0] = xvar_ar[0];
235
236     return;
237 }
238 //-----
239 void lightning_exe__(double xdata_ar[], double xin_ar[], double xout_ar[],
240                    double xvar_ar[])
241 {
242     int wave = (int)xdata_ar[2];
243     double n = xdata_ar[3];
244
245     double ATPtime = xin_ar[0];
246     double ATPtimeStop = xin_ar[2];
247     double ATPtimeStep = xin_ar[3];
248     double Vin = xin_ar[4];
249
250     if((ATPtime >= Tstart) && (ATPtime <= (Tstop + ATPtimeStep)))
251     {
252         if(ATPtime >= B)
253         {
254             double num = xdata_ar[1];
255             A = B;
256             B = B + (Tstop - Tstart)/(2*num);
257             if(event)
258             {
259                 event = false;
260                 // MMC para definir o instante de ocorrencia do evento
261                 timeRand = uniformDist(A, B , random());
262
263                 double IPm = xdata_ar[4];
264                 double IPsd = xdata_ar[5];
265                 double TFm = xdata_ar[6];
266                 double TFsd = xdata_ar[7];
267                 double TCm = xdata_ar[8];
268                 double TCsd = xdata_ar[9];
269
270                 // MMC para definir o valor de IP
271                 double m = log(pow(IPm, 2)/sqrt(pow(IPm, 2)+pow(IPsd, 2)));
272                 double sd = sqrt(log(pow(IPsd/IPm, 2)+1));
273                 IP = logNormalDist(m, sd, random(), random());
274
275                 // MMC para definir o valor de TF

```

```

276     m = log(pow(TFm,2)/sqrt(pow(TFm,2)+pow(TFsd,2)));
277     sd = sqrt(log(pow(TFsd/TFm,2)+1));
278     TF = logNormalDist(m, sd, random(), random());
279
280     // MMC para definir o valor de TC
281     m = log(pow(TCm,2)/sqrt(pow(TCm,2)+pow(TCsd,2)));
282     sd = sqrt(log(pow(TCsd/TCm,2)+1));
283     TC = logNormalDist(m, sd, random(), random());
284
285     exportData = true; // Habilita a exportacao de dados
286     t = 0; // Reinicia a variavel temporal
287
288     if(wave == 0) // Se a forma de onda escolhida for Heidler
289     {
290         double a = 0.0;
291         double b = TF;
292         double Tmax;
293         double TFn;
294         double error;
295         double root[11];
296         int i;
297
298         do
299         {
300             TFn = a+(b-a)/2;
301             error = 1.0;
302             i = 0;
303             root[i] = pow(n*TCm/TFn, 1/(n+1));
304             while((error > 1E-3) && (i < 10))
305             {
306                 i++;
307                 root[i] = pow((n*TCm/TFn)-root[i-1], 1/(n+1));
308                 error = fabs(root[i] - root[i-1]);
309             }
310             Tmax = TFn*root[i];
311
312             if(Tmax > TF)
313             {
314                 b = TFn;
315             }
316             else
317             {
318                 a = TFn;
319             }
320         }
321         while(fabs(Tmax - TF) > 1E-10);
322
323         TF = TFn;
324         factor = (pow(Tmax/TF,n)/(1+pow(Tmax/TF,n)))*exp(-Tmax/TCm);
325     }
326 }
327 else
328 {
329     event = true;
330 }
331 }
332
333 if(fabs(Vin) > Vmax) // Verifica o valor da maxima sobretensao
334 {
335     Vmax = fabs(Vin);
336 }
337
338 if(ATPtime >= timeRand-ATPtimeStep)
339 {
340     if(exportData)
341     {
342         if(firstEvent) // Nao exportar dados antes do primeiro evento
343         {
344             firstEvent = false;
345         }

```

```

346         else
347         {
348             fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da tensao
349         }
350         Vmax = 0.0;
351         exportData = false;
352     }
353
354     if(wave == 0) // Se a forma de onda escolhida for Heidler
355     {
356         xvar_ar[0] = (IP/factor)*(pow(t/TF,n)/(1 + pow(t/TF,n)))*exp(-t/TC);
357     }
358     else // Se a forma de onda escolhida for Dupla Rampa
359     {
360         if(t <= TF) // Para a rampa ascendente
361         {
362             xvar_ar[0] = (IP/TF) * t;
363         }
364         else // Para a rampa descendente
365         {
366             double slope2 = ((-IP/(2*(TC-TF)))*(t-TF)) + IP;
367             if(slope2 <= 0)
368             {
369                 xvar_ar[0] = 0.0;
370             }
371             else
372             {
373                 xvar_ar[0] = slope2;
374             }
375         }
376     }
377     t = t + ATPtimeStep; // Incrementa a variavel temporal
378 }
379 }
380
381 if(ATPtime >= ATPtimeStop-ATPtimeStep) // Na penultima iteracao
382 {
383     if(lastEvent) // Se for o ultimo evento
384     {
385         fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da sobretensao
386         lastEvent = false;
387         exportData = false;
388     }
389 }
390
391 // Atualizada a saida
392 xout_ar[0] = xvar_ar[0];
393
394 return;
395 }
396 /*=====
397 * Mapeamento de dados, entradas, saidas e variaveis do foreign model SWITCHING *
398 -----*/
399 -- DADOS
400 * xdata_ar[0]: valor inicial da sequencia aleatoria (semente)
401 * xdata_ar[1]: numero total de manobras simuladas
402 * xdata_ar[2]: fases manobradas
403 * xdata_ar[3]: tempo de dispercao entre os polos do disjuntor
404 * xdata_ar[4]: tensao base do sistema
405 * xdata_ar[5]: potencia base do sistema
406 * xdata_ar[6]: tempo de inicio da geracao de eventos
407 * xdata_ar[7]: tempo de encerramento da geracao de eventos
408 -- ENTRADAS
409 * xin_ar[0]: tempo corrente do ATP-EMTP
410 * xin_ar[1]: tempo de inicio da simulacao no ATP-EMTP
411 * xin_ar[2]: tempo final da simulacao no ATP-EMTP
412 * xin_ar[3]: passo da simulacao no ATP-EMTP
413 * xin_ar[4]: tensao monitorada em um ponto do sistema
414 -- SAIDAS E VARIAVEIS
415 * xout_ar[0] = xvar_ar[0]: comando da fase A do disjuntor

```

```

416 * xout_ar[1] = xvar_ar[1]: comando da fase B do disjuntor
417 * xout_ar[2] = xvar_ar[2]: comando da fase C do disjuntor
418 /*-----*/
419 void switching_ini__(double xdata_ar[], double xin_ar[], double xout_ar[],
420                    double xvar_ar[])
421 {
422     printf("Starting Foreign Model SWITCHING\n");
423     Ullong initial = (Ullong)xdata_ar[0];
424     double Vb = xdata_ar[4];
425     double Sb = xdata_ar[5];
426     Tstart = xdata_ar[6];
427     Tstop = xdata_ar[7];
428     double ATPtimeStart = xin_ar[1];
429     double ATPtimeStop = xin_ar[2];
430
431     // Define o valor inicial da sequencia aleatoria
432     seed(initial);
433
434     // Verifica se o tempo de inicio da simulacao de eventos e menor ou igual
435     // ao tempo de inicio da simulacao no ATP-EMTP
436     if(Tstart <= ATPtimeStart)
437     {
438         Tstart = ATPtimeStart;
439     }
440
441     // Verifica se o tempo de encerramento da simulacao de eventos e maior ou igual
442     // ao tempo de encerramento da simulacao no ATP-EMTP
443     if(Tstop >= ATPtimeStop)
444     {
445         Tstop = ATPtimeStop;
446     }
447
448     // Inicializa o limite superior do sub-intervalo para o primeiro evento
449     B = Tstart;
450
451     // Abre um novo arquivo para salvar os dados
452     pFile = fopen("dados_estocasticos.txt", "w+");
453     fprintf(pFile, "%s\n", "Tensao");
454     fprintf(pFile, "%.2f\n", Vb);
455     fprintf(pFile, "%.2f\n", Sb);
456
457     // Faz com que todas as fases do disjuntor iniciem fechadas
458     xvar_ar[0] = 1;
459     xvar_ar[1] = 1;
460     xvar_ar[2] = 1;
461     // Atualizada o estado do disjuntor em todas as fases
462     xout_ar[0] = xvar_ar[0];
463     xout_ar[1] = xvar_ar[1];
464     xout_ar[2] = xvar_ar[2];
465
466     return;
467 }
468 /*-----*/
469 void switching_exe__(double xdata_ar[], double xin_ar[], double xout_ar[],
470                    double xvar_ar[])
471 {
472     double ATPtime = xin_ar[0];
473     double ATPtimeStop = xin_ar[2];
474     double ATPtimeStep = xin_ar[3];
475     double Vin = xin_ar[4];
476
477     if((ATPtime >= Tstart) && (ATPtime <= (Tstop + ATPtimeStep)))
478     {
479         if(ATPtime >= B)
480         {
481             double num = xdata_ar[1];
482             A = B;
483             B = B + (Tstop - Tstart)/(2*num);
484             if(event)
485             {

```

```

486     event = false;
487     // MMC para definir o instante de ocorrencia do evento
488     timeRand = uniformDist(A, B , random());
489
490     int fase = (int)xdata_ar[2];
491     // Verifica quais fases serao operadas
492     int f;
493     for(f=1000; f>1; f=f/10)
494     {
495         switch((fase%f)/(f/10))
496         {
497             case 1:
498                 switchingA = true;
499                 break;
500             case 2:
501                 switchingB = true;
502                 break;
503             case 3:
504                 switchingC = true;
505                 break;
506             default:
507                 break;
508         }
509     }
510
511     double Tdisp = xdata_ar[3];
512
513     if(switchingA)
514     {
515         do
516         {
517             // Instane de operacao da fase A do disjuntor
518             Ta = normalDist(timeRand, Tdisp/6, random(), random());
519         }
520         while(Ta < ATPtime);
521     }
522
523     if(switchingB)
524     {
525         do
526         {
527             // Instane de operacao da fase B do disjuntor
528             Tb = normalDist(timeRand, Tdisp/6, random(), random());
529         }
530         while(Tb < ATPtime);
531     }
532
533     if(switchingC)
534     {
535         do
536         {
537             // Instane de operacao da fase C do disjuntor
538             Tc = normalDist(timeRand, Tdisp/6, random(), random());
539         }
540         while(Tc < ATPtime);
541     }
542
543     exportData = true; // Habilita a exportacao de dados
544 }
545 else
546 {
547     event = true;
548 }
549 }
550
551 if(firstSwitchingA && firstSwitchingB && firstSwitchingC)
552 {
553     Vmax = 0.0;
554 }
555 else if(fabs(Vin) > Vmax) // Veririca o valor da maxima sobretensao

```

```
556     {
557         Vmax = fabs(Vin);
558     }
559
560     if((ATPtime >= Ta) && switchingA)
561     {
562         switchingA = false;
563         if(firstSwitchingA) // Se for a primeira manobra da fase A
564         {
565             firstSwitchingA = false;
566         }
567         else
568         {
569             if(exportData)
570             {
571                 exportData = false;
572                 fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da tensao
573                 Vmax = 0.0;
574             }
575         }
576     }
577
578     if((int)xvar_ar[0] == 1) // Se a fase A do disjuntor estiver fechada
579     {
580         xvar_ar[0] = 0;
581     }
582     else // Se a fase A do disjuntor estiver aberta
583     {
584         xvar_ar[0] = 1;
585     }
586 }
587
588     if((ATPtime >= Tb) && switchingB)
589     {
590         switchingB = false;
591         if(firstSwitchingB) // Se for a primeira manobra da fase B
592         {
593             firstSwitchingB = false;
594         }
595         else
596         {
597             if(exportData)
598             {
599                 exportData = false;
600                 fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da tensao
601                 Vmax = 0.0;
602             }
603         }
604     }
605
606     if((int)xvar_ar[1] == 1) // Se a fase B do disjuntor estiver fechada
607     {
608         xvar_ar[1] = 0;
609     }
610     else // Se a fase B do disjuntor estiver aberta
611     {
612         xvar_ar[1] = 1;
613     }
614 }
615
616     if((ATPtime >= Tc) && switchingC)
617     {
618         switchingC = false;
619         if(firstSwitchingC) // Se for a primeira manobra da fase C
620         {
621             firstSwitchingC = false;
622         }
623         else
624         {
625             if(exportData)
626             {
```

```

626         exportData = false;
627         fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da tensao
628         Vmax = 0.0;
629     }
630 }
631
632     if((int)xvar_ar[2] == 1)// Se a fase C do disjuntor estiver fechada
633     {
634         xvar_ar[2] = 0;
635     }
636     else // Se a fase C do disjuntor estiver aberta
637     {
638         xvar_ar[2] = 1;
639     }
640 }
641 }
642
643 if(ATPtime >= ATPtimeStop-ATPtimeStep) // Na penultima iteracao
644 {
645     if(lastEvent) // Se for o ultimo evento
646     {
647         fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da sobretensao
648         lastEvent = false;
649         exportData = false;
650     }
651 }
652
653 // Atualizada o estado do disjuntor em todas as fases
654 xout_ar[0] = xvar_ar[0];
655 xout_ar[1] = xvar_ar[1];
656 xout_ar[2] = xvar_ar[2];
657
658 return;
659 }
660 /*=====
661 * Mapeamento de dados, entradas, saidas e variaveis do foreign model SHORTCIRCUIT *
662 -----*/
663 -- DADOS
664 * xdata_ar[0]: valor inicial da sequencia aleatoria (semente)
665 * xdata_ar[1]: numero total de curtos-circuitos simulados
666 * xdata_ar[2]: fases em curto-circuito
667 * xdata_ar[3]: fases em que ocorrerao abertura
668 * xdata_ar[4]: tempo para inicio do calculo da corrente eficaz de curto-circuito
669 * xdata_ar[5]: tempo para abertura do disjuntor apos o curto-circuito
670 * xdata_ar[6]: tempo de dispercao entre os polos do disjuntor
671 * xdata_ar[7]: tensao base do sistema
672 * xdata_ar[8]: potencia base do sistema
673 * xdata_ar[9]: frequencia nominal do sistema
674 * xdata_ar[10]: tempo de inicio da geracao de eventos
675 * xdata_ar[11]: tempo de encerramento da geracao de eventos
676 -- ENTRADAS
677 * xin_ar[0]: tempo corrente do ATP-EMTP
678 * xin_ar[1]: tempo de inicio da simulacao no ATP-EMTP
679 * xin_ar[2]: tempo final da simulacao no ATP-EMTP
680 * xin_ar[3]: passo da simulacao no ATP-EMTP
681 * xin_ar[4]: tensao monitorada em um ponto do sistema
682 -- SAIDAS E VARIAVEIS
683 * xout_ar[0] = xvar_ar[0]: curto-circuito na fase A
684 * xout_ar[1] = xvar_ar[1]: curto-circuito na fase B
685 * xout_ar[2] = xvar_ar[2]: curto-circuito na fase C
686 * xout_ar[3] = xvar_ar[3]: comando da fase A do disjuntor
687 * xout_ar[4] = xvar_ar[4]: comando da fase B do disjuntor
688 * xout_ar[5] = xvar_ar[5]: comando da fase C do disjuntor
689 /*-----*/
690 void shortcircuit_ini__(double xdata_ar[], double xin_ar[], double xout_ar[],
691                        double xvar_ar[])
692 {
693     printf("Starting Foreign Model SHORTCIRCUIT\n");
694     Ullong initial = (Ullong)xdata_ar[0];
695     int faseCC = (int)xdata_ar[2];

```

```

696 int faseOp = (int)xdata_ar[3];
697 double Tdelay = xdata_ar[4];
698 double Vb = xdata_ar[7];
699 double Sb = xdata_ar[8];
700 double freq = xdata_ar[9];
701 Tstart = xdata_ar[10];
702 Tstop = xdata_ar[11];
703 double ATPtimeStart = xin_ar[1];
704 double ATPtimeStop = xin_ar[2];
705 double ATPtimeStep = xin_ar[3];
706
707 // Define o valor inicial da sequencia aleatoria
708 seed(initial);
709
710 // Verifica se o tempo de inicio da simulacao de eventos e menor ou igual
711 // ao tempo de inicio da simulacao no ATP-EMTP
712 if(Tstart <= ATPtimeStart)
713 {
714     Tstart = ATPtimeStart;
715 }
716
717 // Verifica se o tempo de encerramento da simulacao de eventos e maior ou igual
718 // ao tempo de encerramento da simulacao no ATP-EMTP
719 if(Tstop >= ATPtimeStop)
720 {
721     Tstop = ATPtimeStop;
722 }
723
724 // Verifica quais fases serao curto-circuitadas
725 int f;
726 for(f=1000; f>1; f=f/10)
727 {
728     switch((faseCC%f)/(f/10))
729     {
730     case 1:
731         shortA = 1;
732         break;
733     case 2:
734         shortB = 1;
735         break;
736     case 3:
737         shortC = 1;
738         break;
739     default:
740         break;
741     }
742 }
743
744 // Verifica quais fases serao abertas
745 for(f=1000; f>1; f=f/10)
746 {
747     switch((faseOp%f)/(f/10))
748     {
749     case 1:
750         openA = 0;
751         break;
752     case 2:
753         openB = 0;
754         break;
755     case 3:
756         openC = 0;
757         break;
758     default:
759         break;
760     }
761 }
762
763 // Inicializa o limite superior do sub-intervalo para o primeiro evento
764 B = Tstart;
765

```

```

766 // Calculo do numero de pontos em um ciclo da forma de onda fundamental
767 pointsCycle = (int)(1/(freq*ATPtimeStep) + 1);
768 // Alocacao dinamica de memoria
769 pDataRMS = (double*) malloc(pointsCycle*sizeof(double));
770
771 // Abre um novo arquivo para salvar os dados
772 pFile = fopen("dados_estocasticos.txt", "w+");
773 fprintf(pFile, "%s\t%.2E\n", "Corrente Tensao", Tdelay);
774 fprintf(pFile, "%.2f\n", Vb);
775 fprintf(pFile, "%.2f\n", Sb);
776
777 // Inicializa todas as fases sem curto-circuito
778 xvar_ar[0] = 0;
779 xvar_ar[1] = 0;
780 xvar_ar[2] = 0;
781 // Faz com que todas as fases do disjuntor iniciem fechadas
782 xvar_ar[3] = 1;
783 xvar_ar[4] = 1;
784 xvar_ar[5] = 1;
785 // Atualizada as saidas
786 xout_ar[0] = xvar_ar[0];
787 xout_ar[1] = xvar_ar[1];
788 xout_ar[2] = xvar_ar[2];
789 xout_ar[3] = xvar_ar[3];
790 xout_ar[4] = xvar_ar[4];
791 xout_ar[5] = xvar_ar[5];
792
793 return;
794 }
795 //-----
796 void shortcircuit_exe__(double xdata_ar[], double xin_ar[], double xout_ar[],
797                        double xvar_ar[])
798 {
799     double Tdelay = xdata_ar[4];
800     double freq = xdata_ar[9];
801
802     double ATPtime = xin_ar[0];
803     double ATPtimeStop = xin_ar[2];
804     double ATPtimeStep = xin_ar[3];
805     double Vin = xin_ar[4];
806     double curr = xin_ar[5];
807
808     if((ATPtime >= Tstart) && (ATPtime <= (Tstop + ATPtimeStep)))
809     {
810         if(ATPtime >= B)
811         {
812             double num = xdata_ar[1];
813             A = B;
814             B = B + (Tstop - Tstart)/(2*num);
815             if(event)
816             {
817                 event = false;
818                 // MMC para definir o instante de ocorrencia do evento
819                 timeRand = uniformDist(A, B , random());
820
821                 double Tclear = xdata_ar[5];
822                 double Tdisp = xdata_ar[6];
823
824                 do
825                 {
826                     // Instane de abertura da fase A do disjuntor
827                     Ta = normalDist(timeRand+Tclear, Tdisp/6, random(), random());
828                 }
829                 while(Ta < ATPtime);
830
831                 do
832                 {
833                     // Instane de abertura da fase B do disjuntor
834                     Tb = normalDist(timeRand+Tclear, Tdisp/6, random(), random());
835

```

```

836         while(Tb < ATPtime);
837
838     do
839     {
840         // Instane de abertura da fase C do disjuntor
841         Tc = normalDist(timeRand+Tclear, Tdisp/6, random(), random());
842     }
843     while(Tc < ATPtime);
844
845     // Elimina o curto-circuito em todas as fases
846     xvar_ar[0] = 0;
847     xvar_ar[1] = 0;
848     xvar_ar[2] = 0;
849     // Fecha todas as fases do disjuntor, reconfigurando o sistema
850     xvar_ar[3] = 1;
851     xvar_ar[4] = 1;
852     xvar_ar[5] = 1;
853
854     if(exportData) // Nao exporta dados antes do primeiro evento
855     {
856         fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da tensao
857     }
858     exportData = true; // Habilita a exportacao de dados
859     shortCircuit = true; // Habilita a ocorrencia de um curto-circuito
860 }
861 else
862 {
863     event = true;
864 }
865 }
866
867 if(fabs(Vin) > Vmax) // Veririca o valor da maxima sobretensao
868 {
869     Vmax = fabs(Vin);
870 }
871
872 if((ATPtime >= timeRand) && shortCircuit)
873 {
874     shortCircuit = false;
875     Vmax = 0.0;
876     // Efetua o curto-circuito
877     xvar_ar[0] = shortA;
878     xvar_ar[1] = shortB;
879     xvar_ar[2] = shortC;
880 }
881
882 // Monta a janela com pontos de um ciclo para o calculo do valor eficaz
883 if(ATPtime >= (timeRand+Tdelay) && ATPtime < (timeRand+Tdelay)+1/freq)
884 {
885     *pDataRMS = curr;
886     pDataRMS++;
887     pointsRMS++;
888     windowRMS = true;
889 }
890 else if(windowRMS)
891 {
892     // Calcula o valor eficaz da corrente de curto-circuito
893     pDataRMS = pDataRMS - pointsRMS;
894     fprintf(pFile, "%.2f\t", trueRMS(pDataRMS, ATPtimeStep));
895     windowRMS = false;
896     pointsRMS = 0;
897 }
898
899 // Abertura dos polos do disjuntor
900 if(ATPtime >= Ta)
901 {
902     xvar_ar[3] = openA;
903 }
904
905 if(ATPtime >= Tb)

```

```

906     {
907         xvar_ar[4] = openB;
908     }
909
910     if(ATPtime >= Tc)
911     {
912         xvar_ar[5] = openC;
913     }
914 }
915
916 if(ATPtime >= ATPtimeStop-ATPtimeStep) // Na penultima iteracao
917 {
918     if(lastEvent) // Se for o ultimo evento
919     {
920         fprintf(pFile, "%.12E\n", Vmax); // Exporta o valor da sobretensao
921         lastEvent = false;
922         exportData = false;
923     }
924 }
925
926 // Atualizada as saidas
927 xout_ar[0] = xvar_ar[0];
928 xout_ar[1] = xvar_ar[1];
929 xout_ar[2] = xvar_ar[2];
930 xout_ar[3] = xvar_ar[3];
931 xout_ar[4] = xvar_ar[4];
932 xout_ar[5] = xvar_ar[5];
933
934 return;
935 }
936 /*=====
937 * Mapeamento de dados, entradas, saidas e variaveis do foreign model INRUSH *
938 -----*/
939 -- DADOS
940 * xdata_ar[0]: valor inicial da sequencia aleatoria (semente)
941 * xdata_ar[1]: numero total de energizacoes simuladas
942 * xdata_ar[2]: tempo para inicio da analise apos a energizacao
943 * xdata_ar[3]: tempo de dispercao entre os polos do disjuntor
944 * xdata_ar[4]: fase analisada
945 * xdata_ar[5]: tensao base do sistema
946 * xdata_ar[6]: potencia base do sistema
947 * xdata_ar[7]: frequencia nominal do sistema
948 * xdata_ar[8]: tempo de inicio da geracao de eventos
949 * xdata_ar[9]: tempo de encerramento da geracao de eventos
950 -- ENTRADAS
951 * xin_ar[0]: tempo corrente do ATP-EMTP
952 * xin_ar[1]: tempo de inicio da simulacao no ATP-EMTP
953 * xin_ar[2]: tempo final da simulacao no ATP-EMTP
954 * xin_ar[3]: passo da simulacao no ATP-EMTP
955 * xin_ar[4]: corrente de energizacao
956 -- SAIDAS E VARIAVEIS
957 * xout_ar[0] = xvar_ar[0]: comando da fase A do disjuntor
958 * xout_ar[1] = xvar_ar[1]: comando da fase B do disjuntor
959 * xout_ar[2] = xvar_ar[2]: comando da fase C do disjuntor
960 /*-----*/
961 void inrush_ini__(double xdata_ar[], double xin_ar[], double xout_ar[],
962                 double xvar_ar[])
963 {
964     printf("Starting Foreign Model INRUSH\n");
965     Ullong initial = (Ullong)xdata_ar[0];
966     double Tdelay = xdata_ar[2];
967     double Vb = xdata_ar[5];
968     double Sb = xdata_ar[6];
969     double freq = xdata_ar[7];
970     Tstart = xdata_ar[8];
971     Tstop = xdata_ar[9];
972     double ATPtimeStart = xin_ar[1];
973     double ATPtimeStop = xin_ar[2];
974     double ATPtimeStep = xin_ar[3];
975

```

```

976 // Define o valor inicial da sequencia aleatoria
977 seed(initial);
978
979 // Verifica se o tempo de inicio da simulacao de eventos e menor ou igual
980 // ao tempo de inicio da simulacao no ATP-EMTP
981 if(Tstart <= ATPtimeStart)
982 {
983     Tstart = ATPtimeStart;
984 }
985 // Verifica se o tempo de encerramento da simulacao de eventos e maior ou igual
986 // ao tempo de encerramento da simulacao no ATP-EMTP
987 if(Tstop >= ATPtimeStop)
988 {
989     Tstop = ATPtimeStop;
990 }
991 // Inicializa o limite superior do sub-intervalo para o primeiro evento
992 B = Tstart;
993
994 // Calculo do numero de pontos em um ciclo da forma de onda fundamental
995 pointsCycle = (int) (1/(freq*ATPtimeStep) + 1);
996 // Alocao dinamica de memoria
997 pDataRMS = (double*) malloc(pointsCycle*sizeof(double));
998
999 // Abre um novo arquivo para salvar os dados
1000 pFile = fopen("dados_estocasticos.txt", "w+");
1001 fprintf(pFile, "%s\n", "Corrente");
1002 fprintf(pFile, "%.2f\n", Vb);
1003 fprintf(pFile, "%.2f\n", Sb);
1004
1005 // Faz com que o disjuntor inicie aberto em todas as fases
1006 xvar_ar[0] = 0;
1007 xvar_ar[1] = 0;
1008 xvar_ar[2] = 0;
1009 // Atualizada o estado do disjuntor em todas as fases
1010 xout_ar[0] = xvar_ar[0];
1011 xout_ar[1] = xvar_ar[1];
1012 xout_ar[2] = xvar_ar[2];
1013
1014 return;
1015 }
1016 //-----
1017 void inrush_exe__(double xdata_ar[], double xin_ar[], double xout_ar[],
1018                 double xvar_ar[])
1019 {
1020     double num = xdata_ar[1];
1021     double Tdelay = xdata_ar[2];
1022     double Tdisp = xdata_ar[3];
1023     double freq = xdata_ar[7];
1024
1025     double ATPtime = xin_ar[0];
1026     double ATPtimeStep = xin_ar[3];
1027     double curr = xin_ar[4];
1028
1029     // Verifica o intervalo de geracao de eventos
1030     if((ATPtime >= Tstart) && (ATPtime <= (Tstop + ATPtimeStep)))
1031     {
1032         if(ATPtime >= B) // A cada sub-intervalo
1033         {
1034             // Atualiza os limites do sub-intervalo
1035             A = B;
1036             B = B + (Tstop - Tstart)/(2*num);
1037
1038             if(event) // Se for para gerar uma energizacao neste sub-intervalo
1039             {
1040                 event = false; // Nao gera no proximo
1041                 // MMC para definir o instante em que ocorrera a energizacao
1042                 timeRand = uniformDist(A, B , random());
1043
1044                 do
1045                 {

```

```

1046         // Instane de fechamento da fase A do disjuntor
1047         Ta = normalDist(timeRand, Tdisp/6, random(), random());
1048     }
1049     while(Ta < ATPtime);
1050
1051     do
1052     {
1053         // Instane de fechamento da fase B do disjuntor
1054         Tb = normalDist(timeRand, Tdisp/6, random(), random());
1055     }
1056     while(Tb < ATPtime);
1057
1058     do
1059     {
1060         // Instane de fechamento da fase C do disjuntor
1061         Tc = normalDist(timeRand, Tdisp/6, random(), random());
1062     }
1063     while(Tc < ATPtime);
1064
1065     // Identifica a fase analisada
1066     switch((int)xdata_ar[4])
1067     {
1068         case 1:
1069             Tc1 = Ta;
1070             break;
1071         case 2:
1072             Tc1 = Tb;
1073             break;
1074         case 3:
1075             Tc1 = Tc;
1076             break;
1077     }
1078     //
1079     pointsFFT = 0;
1080
1081     // Abre todas as fases do disjuntor
1082     xvar_ar[0] = 0;
1083     xvar_ar[1] = 0;
1084     xvar_ar[2] = 0;
1085 }
1086 else // Se nao for para gerar uma energizacao neste sub-intervalo
1087 {
1088     event = true; // Gera no proximo
1089 }
1090 }
1091 // Fecha a fase A do disjuntor
1092 if(ATPtime >= Ta)
1093 {
1094     xvar_ar[0] = 1;
1095 }
1096 // Fecha a fase B do disjuntor
1097 if(ATPtime >= Tb)
1098 {
1099     xvar_ar[1] = 1;
1100 }
1101 // Fecha a fase C do disjuntor
1102 if(ATPtime >= Tc)
1103 {
1104     xvar_ar[2] = 1;
1105 }
1106
1107 // Monta a janela com pontos de um ciclo para o calculo do valor evicaz
1108 if(ATPtime >= (Tc1+Tdelay) && ATPtime < (Tc1+Tdelay)+1/freq)
1109 {
1110     dataRMS[pointsRMS] = curr;
1111     pointsRMS++;
1112     windowRMS = true;
1113 }
1114 else if(windowRMS)
1115 {

```

```
1116         fprintf(pFile, "%.2f\n", trueRMS(dataRMS, ATPtimeStep));
1117         windowRMS = false;
1118         pointsRMS = 0;
1119     }
1120
1121     // Monta a janela da FFT com 512 pontos
1122     if(ATPtime >= (Tcl+Tdelay) && pointsFFT < 512)
1123     {
1124         dataFFT[pointsFFT] = curr;
1125         pointsFFT++;
1126         windowFFT = true;
1127     }
1128     else if(windowFFT)
1129     {
1130         // Calcula a FFT
1131         FFT(dataFFT, magFFT);
1132         int i = 0;
1133         for(i=0; i<256; i++)
1134         {
1135             fprintf(pFileFFT, "%.5f ", magFFT[i]);
1136         }
1137         fprintf(pFileFFT, "\n");
1138         windowFFT = false;
1139     }
1140 }
1141 // Atualizada o estado do disjuntor em todas as fases
1142 xout_ar[0] = xvar_ar[0];
1143 xout_ar[1] = xvar_ar[1];
1144 xout_ar[2] = xvar_ar[2];
1145
1146 return;
1147 }
```


APÊNDICE B – SUBROTINA FGNMOD.F

```

1  SUBROUTINE FGNMOD ( name, namlen, xdata, xin, xout, xvar,
2  1          iniflg, ierflg)
3  IMPLICIT REAL*8 (A-H, O-Z), INTEGER*4 (I-N)
4  DIMENSION xdata(*), xin(*), xout(*), xvar(*)
5  CHARACTER*1 name(*)
6  PARAMETER ( namcnt = 8 )
7  CHARACTER*32 refnam(namcnt)
8  CONTINUE ! -----
9  CONTINUE ! You may increase namcnt above to allow more names:
10 CONTINUE ! -----
11 CONTINUE ! In the following lines, register your foreign model
12 CONTINUE ! names as they are declared in your models:
13 CONTINUE ! - use only uppercase characters for the name here
14 CONTINUE ! - you can use any case for the name in the models
15 CONTINUE ! - make a copy of the modifications you make to this
16 CONTINUE ! file so that you don't lose them when installing
17 CONTINUE ! a newer version of ATP later
18 DATA refnam(1) / 'SAMPLE_MODEL' / ! Do not modify this line
19 DATA refnam(2) / 'LIGHTNING' /
20 DATA refnam(3) / 'SWITCHING' /
21 DATA refnam(4) / 'SHORTCIRCUIT' /
22 DATA refnam(5) / 'INRUSH' /
23 DATA refnam(6) / ' ' /
24 DATA refnam(7) / ' ' /
25 DATA refnam(8) / ' ' /
26 CONTINUE ! -----
27 CONTINUE ! Name identification loop
28 CONTINUE ! -- no need to change anything here
29  iname = 1
30  lpflg = 1
31  DO WHILE (iname.LE.namcnt .AND. lpflg.GT.0)
32    ICHAR = 1
33    DO WHILE (ichar.LE.namlen
34  1      .AND. name(ichar).EQ.refnam(iname)(ichar:ichar))
35      ichar = ichar + 1
36    ENDDO
37    IF (ichar.GT.namlen) THEN
38      lpflg = 0
39    ELSE
40      iname = iname + 1
41    ENDIF
42  ENDDO
43  IF (iname.GT.namcnt) THEN
44    ierflg = 1
45    RETURN
46  ENDIF
47  CONTINUE ! -----
48  CONTINUE ! In the following lines, this is where you call the
49  CONTINUE ! actual foreign subroutines/procedures:
50  CONTINUE ! - actual names may be different from the foreign
51  CONTINUE ! names used in the models
52  CONTINUE ! - notice how each one uses both an
53  CONTINUE ! initialization routine and an execution routine
54  IF ( iname.EQ.1 ) THEN
55    IF (iniflg.EQ.1) THEN
56      CALL sampli(xdata, xin, xout, xvar)
57    ELSE
58      CALL samplm(xdata, xin, xout, xvar)
59    ENDIF
60  CONTINUE ! -----
61  ELSE IF ( iname.EQ.2 ) THEN
62    IF (iniflg.EQ.1) THEN
63      CALL lightning_ini(xdata, xin, xout, xvar)
64    ELSE
65      CALL lightning_exe(xdata, xin, xout, xvar)

```

```

66     ENDIF
67     CONTINUE ! -----
68     ELSE IF ( iname.EQ.3 ) THEN
69         IF ( iniflg.EQ.1 ) THEN
70             CALL switching_ini(xdata, xin, xout, xvar)
71         ELSE
72             CALL switching_exe(xdata, xin, xout, xvar)
73         ENDIF
74     CONTINUE ! -----
75     ELSE IF ( iname.EQ.4 ) THEN
76         IF ( iniflg.EQ.1 ) THEN
77             CALL shortcircuit_ini(xdata, xin, xout, xvar)
78         ELSE
79             CALL shortcircuit_exe(xdata, xin, xout, xvar)
80         ENDIF
81     CONTINUE ! -----
82     ELSE IF ( iname.EQ.5 ) THEN
83         IF ( iniflg.EQ.1 ) THEN
84             CALL inrush_ini(xdata, xin, xout, xvar)
85         ELSE
86             CALL inrush_exe(xdata, xin, xout, xvar)
87         ENDIF
88     CONTINUE ! -----
89     ELSE IF ( iname.EQ.6 ) THEN
90         IF ( iniflg.EQ.1 ) THEN
91             ELSE
92         ENDIF
93     CONTINUE ! -----
94     ELSE IF ( iname.EQ.7 ) THEN
95         IF ( iniflg.EQ.1 ) THEN
96             ELSE
97         ENDIF
98     CONTINUE ! -----
99     ELSE IF ( iname.EQ.8 ) THEN
100        IF ( iniflg.EQ.1 ) THEN
101            ELSE
102        ENDIF
103    ENDIF
104    CONTINUE ! -----
105    RETURN      ! Pass control back to MODELS (the CALL is from FTNSUB)
106    END
107    SUBROUTINE FGNFUN ( name, namlen, xarg, nval, ierflg)
108    IMPLICIT REAL*8 (A-H, O-Z), INTEGER*4 (I-N)
109    DIMENSION xarg(*)
110    CHARACTER*1 name(*)
111    PARAMETER ( namcnt = 8 )
112    CHARACTER*32 refnam(namcnt)
113    CONTINUE ! -----
114    CONTINUE ! You may increase namcnt above to allow more names:
115    CONTINUE ! -----
116    CONTINUE ! In the following lines, register your foreign function
117    CONTINUE ! names as they are declared in your models:
118    CONTINUE ! - use only uppercase characters for the name here
119    CONTINUE ! - you can use any case for the name in the models
120    CONTINUE ! - make a copy of the modifications you make to this
121    CONTINUE ! file so that you don't lose them when installing
122    CONTINUE ! a newer version of ATP later
123    DATA refnam(1) / 'SAMPLE_FUNCTION' / ! Do not modify this line
124    DATA refnam(2) / 'SAMPLE_C_FUNCTION' /
125    DATA refnam(3) / 'C_SEED' /
126    DATA refnam(4) / 'C_RANDOM' /
127    DATA refnam(5) / ' ' /
128    DATA refnam(6) / ' ' /
129    DATA refnam(7) / ' ' /
130    DATA refnam(8) / ' ' /
131    CONTINUE ! -----
132    CONTINUE ! Name identification loop
133    CONTINUE ! -- no need to change anything here
134    iname = 1
135    lpflg = 1

```

```

136 DO WHILE (iname.LE.namcnt .AND. lpflg.GT.0)
137   ichar = 1
138   DO WHILE (ichar.LE.namlen
139     1 .AND. name(ichar).EQ.refnam(iname)(ichar:ichar))
140     ichar = ichar + 1
141   ENDDO
142   IF (ichar.GT.namlen) THEN
143     lpflg = 0
144   ELSE
145     iname = iname + 1
146   ENDIF
147 ENDDO
148 IF (iname.GT.namcnt) THEN
149   ierflg = 1
150   RETURN
151 ENDIF
152 CONTINUE ! -----
153 CONTINUE ! In the following lines, this is where you call the
154 CONTINUE ! actual foreign functions:
155 CONTINUE ! - actual names may be different from the foreign
156 CONTINUE ! names used in the models
157 IF ( iname.EQ.1 ) THEN
158   nval = samplf(xarg)
159 ELSE IF ( iname.EQ.2 ) THEN
160   nval = cfun(xarg)
161 ELSE IF ( iname.EQ.3 ) THEN
162   xarg(1) = C_seed()
163   nval = 1
164 ELSE IF ( iname.EQ.4 ) THEN
165   xarg(1) = C_random()
166   nval = 1
167 ELSE IF ( iname.EQ.5 ) THEN
168   nval = 0
169 ELSE IF ( iname.EQ.6 ) THEN
170   nval = 0
171 ELSE IF ( iname.EQ.7 ) THEN
172   nval = 0
173 ELSE IF ( iname.EQ.8 ) THEN
174   nval = 0
175 ENDIF
176 CONTINUE ! -----
177 RETURN
178 END
179 SUBROUTINE SAMPLM (xdata, xin, xout, xvar)
180 IMPLICIT REAL*8 (A-H, O-Z), INTEGER*4 (I-N)
181 DIMENSION xdata(*), xin(*), xout(*), xvar(*)
182 CHARACTER*80 text80 ! Buffer used to assemble messages for output
183 CONTINUE !-----
184 CONTINUE ! This is a sample user-defined foreign model !
185 CONTINUE !-----
186 CONTINUE ! For this particular model: !
187 CONTINUE ! - the dimension of xin,xout,xvar is given in xdata(1)!
188 CONTINUE ! - xvar is to be initialized with HISTDEF !
189 CONTINUE ! in the USE statement !
190 CONTINUE !-----
191 CONTINUE ! For any foreign model: !
192 CONTINUE ! - the user can assign history to xin and xvar !
193 CONTINUE ! using HISTDEF in the USE statement !
194 CONTINUE ! - MODELS always saves the values that are in xvar !
195 CONTINUE ! between uses, so that the memory of multiple !
196 CONTINUE ! instances of using the code can be managed !
197 CONTINUE ! automatically by MODELS !
198 CONTINUE !-----
199 CONTINUE ! To introduce a foreign model in a simulation: !
200 CONTINUE ! - create an initialization and an execution routine !
201 CONTINUE ! for that model !
202 CONTINUE ! - provide access to your routines by modifying !
203 CONTINUE ! the routine fgnmod at the beginning of this file !
204 CONTINUE ! - declare and use the foreign model in a model !
205 CONTINUE ! of the simulation !

```

```

206 CONTINUE !-----!
207 CONTINUE ! This sample foreign model could be used in a model !
208 CONTINUE ! as follows: !
209 CONTINUE ! !
210 CONTINUE ! MODEL anyname !
211 CONTINUE ! INPUT ... !
212 CONTINUE ! VAR a[1..3], ... !
213 CONTINUE ! ... !
214 CONTINUE ! MODEL mymodel FOREIGN sample_model !
215 CONTINUE ! followed by values for ixdata, ixin, ixout, !
216 CONTINUE ! and ixvar a pair of inside brackets !
217 CONTINUE ! -- the preceding line declares foreign model "sub" !
218 CONTINUE ! -- "mymodel" is the local name of the foreign model !
219 CONTINUE ! -- "sample_model" is the identifier used for that !
220 CONTINUE ! -- foreign model in the user-modifiable !
221 CONTINUE ! -- MODELS interface routine "fgnmod" !
222 CONTINUE ! ... !
223 CONTINUE ! EXEC !
224 CONTINUE ! ... !
225 CONTINUE ! USE mymodel AS mymodel !
226 CONTINUE ! DATA xdata[1]:=3 !
227 CONTINUE ! INPUT xin[1..3]:=[ list of 3 expressions... ] !
228 CONTINUE ! OUTPUT a[1..3]:=xout[1..3] !
229 CONTINUE ! HISTORY histdef(xvar[1..3]):=0 !
230 CONTINUE ! -- can be any expression !
231 CONTINUE ! -- note the fixed names xdata, xin, xout, xvar !
232 CONTINUE ! ENDUSE !
233 CONTINUE ! ... !
234 CONTINUE ! ENDEXEC !
235 CONTINUE ! ENDMODEL !
236 CONTINUE !-----!
237 text80 = 'Executing model "sample_model".' ! Assemble output mes
238 CALL OUTSIX( text80, 80 ) ! Send text80 (arg 1) to ATP listing
239 N8 = xdata(1)
240 DO 1000 i=1, N8
241 xvar(i)=(2*xvar(i)+xin(i))/2
242 xout(i)=xvar(i)+100.0
243 1000 CONTINUE
244 RETURN
245 ENTRY sampli(xdata, xin, xout, xvar)
246 text80 = 'Initializing model "sample_model".'
247 CALL OUTSIX( text80, 80 )
248 RETURN
249 END
250 FUNCTION SAMPLF (arg)
251 IMPLICIT REAL*8 (A-H, O-Z), INTEGER*4 (I-N)
252 DIMENSION arg(2)
253 CONTINUE !-----!
254 CONTINUE ! This is a sample user-defined foreign function !
255 CONTINUE !-----!
256 CONTINUE ! For any foreign function: !
257 CONTINUE ! - the function can use as many arguments as it !
258 CONTINUE ! expects to receive !
259 CONTINUE ! - the function receives its argument in an array !
260 CONTINUE ! - the function loads its output values in !
261 CONTINUE ! the same array !
262 CONTINUE ! - the function returns the number of output values !
263 CONTINUE ! as its value !
264 CONTINUE !-----!
265 CONTINUE ! To introduce a foreign function in a simulation: !
266 CONTINUE ! - create a function procedure !
267 CONTINUE ! - provide access to your functions by modifying !
268 CONTINUE ! the routine fgnfun at the beginning of this file !
269 CONTINUE ! - declare and use the foreign function in a model !
270 CONTINUE ! of the simulation !
271 CONTINUE !-----!
272 CONTINUE ! This function could be declared in a model as follows: !
273 CONTINUE ! FUNCTION myfunc FOREIGN sample_function !
274 CONTINUE ! followed by ixarg:2 inside a pair of brackets !
275 CONTINUE ! which defines the function "myfunc" to be a foreign !

```

```
276 CONTINUE ! function where: !
277 CONTINUE ! - "myfunc" is the local name of the foreign function !
278 CONTINUE ! - "sample_function" is the identifier used for that !
279 CONTINUE ! foreign function in the user-modifiable !
280 CONTINUE ! MODELS interface routine "fgnfun" !
281 CONTINUE ! - "ixarg" is a fixed keyword indicating !
282 CONTINUE ! the argument count !
283 CONTINUE !-----!
284 CONTINUE ! This function could be used in a model as follows: !
285 CONTINUE ! a[1..2] := 3*myfunc( 36.3, sqrt(c) ) !
286 CONTINUE !-----!
287 arg(1)=arg(1) +arg(2)
288 arg(2)=10*arg(1)
289 samplf=2
290 RETURN
291 END
```


APÊNDICE C – MAKEFILE

```

1 #-----#
2 # Arquivo: Makefile #
3 # Proposito: Compilar e adicionar novas funcoes ao ATP-EMTP #
4 # Desenvolvido por Adriano C. Marchesan e Leyla Kraulich #
5 #-----#
6
7 #----- Definicao de compiladores -----#
8
9 CC = gcc.exe
10 FOR = g77.exe
11
12 #----- Diretorios -----#
13
14 # MYDOCS contem os arquivos a serem adicionados e a subrotina
15 # "fgnmod.f" alterada para acessar as novas funcoes
16 MYDOCS = D:/DOCUME~1/ATPdata/mytpbig
17
18 # ATPMINGW contem as bibliotecas e arquivos do ATP
19 ATPMINGW = C:/ATP/atpmingw/make
20
21 # Diretorio de Destino do ATP personalizado: mytpbig.exe
22 IMAGE = $(MYDOCS)/mytpbig.exe
23
24 #----- Arquivos objetos do ATP -----#
25
26 OBJECTS = analyt.o \
27          comtac.o \
28          devt69.o \
29          dimdef.o \
30          fgnmod.o \
31          hopcod.o \
32          newmods.o \
33          user10.o \
34          usernl.o \
35          usrfun.o
36
37 #----- Arquivos objetos adicionados -----#
38
39 MYOBJECTS = foreign_models.o
40
41 #----- Arquivos do ATP -----#
42
43 INSDIR = blkcom.ins \
44         comta1.ins \
45         comta2.ins \
46         comta3.ins \
47         deksal.ins \
48         dekspy.ins \
49         sixty4.ins \
50         space2.ins \
51         tacsar.ins
52
53 #----- Definicoes requeridas para compilacao -----#
54
55 CFLAGS = -DUNDERScore -O2
56 FFLAGS = -O2
57
58 #----- Bibliotecas do ATP -----#
59
60 LIBRARY = $(ATPMINGW)/tpbig.a $(ATPMINGW)/dismg7.a -lgdi32 \
61          -L"$(ATPMINGW)/MinGW/lib"
62
63 #----- Compilacao e encadeamento de arquivos -----#
64
65 all: $(IMAGE)

```

```
66 .f.o:
67     $(FOR) -c $(FFLAGS) $<
68 .c.o:
69     $(CC) -c $(CFLAGS) $<
70 $(IMAGE) : $(OBJECTS) $(MYOBJECTS) $(INSFILE)
71     $(FOR) -s -o $(IMAGE) $(OBJECTS) $(MYOBJECTS) $(LIBRARY)
72 foreign_models.o: $(MYDOCS)\foreign_models.cpp
73     $(CC) -c $(CFLAGS) $(MYDOCS)\foreign_models.cpp
74 fgnmod.o: $(MYDOCS)\fgnmod.f
75     $(FOR) -c $(FFLAGS) $(MYDOCS)\fgnmod.f
76
77 #-----#
```