

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

**MÉTODOS E ALGORITMOS PARA AJUSTE DOS
MEIOS BÁSICOS PARA CONTROLE DE TENSÃO EM
SISTEMAS DE DISTRIBUIÇÃO**

DISSERTAÇÃO DE MESTRADO

Rafael Gressler Milbradt

**Santa Maria, RS, Brasil
2007**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

2007

Mestre

MILBRADT, Rafael Gressler

PPGEE/UFSM, RS

MÉTODOS E ALGORITMOS PARA AJUSTE DOS MEIOS BÁSICOS PARA CONTROLE DE TENSÃO EM SISTEMAS DE DISTRIBUIÇÃO

por

Rafael Gressler Milbradt

Dissertação apresentada ao Curso de Mestrado do Programa de
Pós-Graduação em Engenharia Elétrica, Área de Concentração
em Processamento de Energia, Centro de Estudos em Energia e Meio
Ambiente (CEEMA) da
Universidade Federal de Santa Maria (UFSM, RS),
como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica.

Orientadora: Prof^a. Dr^a Luciane Neves Canha

**Santa Maria, RS, Brasil
2007**

**Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**MÉTODOS E ALGORITMOS PARA AJUSTE DOS MEIOS
BÁSICOS PARA CONTROLE DE TENSÃO EM SISTEMAS DE
DISTRIBUIÇÃO**

elaborada por
Rafael Gressler Milbradt

como requisito parcial para obtenção do grau de
Mestre em Engenharia Elétrica

COMISSÃO EXAMINADORA:

Luciane Neves Canha, Dr^a
(Presidente/Orientador)

Petr Ekel, PhD. (PUC-MG)

Alzenira da Rosa Abaide, Dr^a. (UFSM)

Santa Maria, 9 de novembro de 2007.

*Dedico esta dissertação a Deus que
torna tudo possível. Aos meus pais
Claudir e Liana e a minha irmã Bruna por
serem meus alicerces.*

AGRADECIMENTOS

Gostaria de agradecer a todos que de uma forma ou outra tornaram este trabalho possível, aos meus amigos da APFP, aos meus colegas da Mega que trabalharam por mim nos momentos em que eu não me fiz presente. À Karine por ser inspiração, pelo carinho e amor que amenizaram todas as tempestades.

Agradeço ao Professor Vladimir Andreevitch Popov pelo brilhantismo, inspiração e orientação que se fizeram presentes no início da elaboração deste trabalho. E por fim, meu agradecimento especial a minha orientadora Professora Luciane Neves Canha, pela paciência, dedicação, conhecimento que conseguiu me transmitir e principalmente pela oportunidade de realizar e estar concluindo este trabalho.

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Santa Maria

MÉTODOS E ALGORITMOS PARA AJUSTE DOS MEIOS BÁSICOS PARA CONTROLE DE TENSÃO EM SISTEMAS DE DISTRIBUIÇÃO

AUTOR: RAFAEL GRESSLER MILBRADT

ORIENTADORA: LUCIANE NEVES CANHA

Data e Local da Defesa: Santa Maria, 9 de novembro de 2007.

A qualidade da energia fornecida é uma característica cada vez mais desejada. Os órgãos que regulam a distribuição de energia no Brasil aplicam pesadas multas às concessionárias que transgridem os níveis aceitáveis de tensão. Em contrapartida, as concessionárias buscam reduzir os seus custos operacionais, ou seja, melhorar o produto fornecido sem fazer maiores investimentos.

A utilização dos meios básicos de controle de tensão, como os tapes dos transformadores de distribuição e comutador de tapes sob carga na Subestação, são meios que já existem na maior parte dos sistemas de distribuição e que, se ajustados corretamente, podem minimizar ou até mesmo sanar o problema das transgressões nos níveis de tensão.

Entretanto, realizar a estimação de estado onde as informações sobre as redes secundárias são pouco confiáveis, ou mesmo ausentes requer tratamento e metodologias adequadas. Este trabalho propõe o uso de modelos simplificados de redes de baixa tensão tornando possível a estimação de estado mesmo nos casos de ausência de informações.

O ajuste dos tapes dos transformadores de distribuição é realizado de forma a compensar as quedas existentes na rede primária, quedas de tensão do transformador de distribuição e na rede secundária até o centro de carga da mesma. A realização desta regulação de tensão para o centro de carga da rede de baixa tensão proporciona um regime adequado de tensão, senão para todos os consumidores do circuito, pelo menos para a maioria deles.

O ajuste do comutador de tapes sob carga da Subestação, através da Lei de Controle de Tensão, funciona de forma complementar ao ajuste anterior, visto que o

comutador de tapes sob carga possibilita uma regulação automática, onde se podem compensar as variações na queda de tensão que ocorrem ao longo do dia. A Lei de Controle de Tensão também considera as quedas de tensão até o centro de carga da rede secundária. Desta forma a lei de controle de tensão na subestação é calculada pela média ponderada pela potência das quedas de tensão até o centro de carga de cada transformador da Subestação.

O trabalho também propõe e implementa um indicador, nomeado Energia Fornecida Fora do Padrão – EFPF, para que seja possível a medição do desempenho das redes quanto às transgressões nos níveis de tensão. Este indicador também serve para medir o desempenho dos algoritmos de ajuste dos meios de controle de tensão utilizados.

No que diz respeito aos resultados do trabalho foi desenvolvido um Software, o ASD-CT, que implementa todos os algoritmos aqui propostos. O Software foi desenvolvido totalmente em linguagem Java, utilizando os mais modernos conceitos de Engenharia de Software. O Software ASD-CT obteve ótimos resultados em alimentadores reais da Companhia Estadual de Distribuição de Energia Elétrica (CEEE-D).

Palavras-chave: Sistemas de Distribuição, Modelos Representativos, Planejamento, Controle de Tensão.

ABSTRACT

Master's Dissertation
Post Graduation Program in Electric Engineering
Federal University of Santa Maria

METHODS AND ALGORITHMS FOR VOLTAGE CONTROL IN DISTRIBUTION SYSTEMS

AUTHOR: RAFAEL GRESSLER MILBRADT

TUTOR: LUCIANE NEVES CANHA

Santa Maria, november, 2007.

Nowadays, the quality of energy supplied to customers is an important aspect that must be taken into account in distribution systems analysis. The Brazilian national agency which regulates the distribution systems standards is applying punishments to the energy supplying companies which can not keep the voltage levels inside the allowed limits.

Besides, in most cases, energy supplying companies are trying to improve the quality of energy without make use of large investments.

The voltage control through the regulation of distribution transformers taps and transformer under load tap changer (ULTC) are being used in most of distribution systems. These types of voltage control may reduce or eliminate the condition which voltage levels exceeding the allowed limits.

It is important to observe that use of estimations with few or unfaithful information about secondary networks requires appropriate methodologies. Therefore, this study uses simplified models of secondary networks which may estimate the final results, even in cases without any information.

The adjustment of transformers taps is defined to reduce undervoltage in primary and secondary networks. As regards secondary networks, voltage drop are considered up to the center of load, which provides an appropriate voltage levels to the most of customers in analysis.

The adjustment of ULTC uses the voltage control law, in order to complement the adjustment earlier described. In this case, the ULTC allows an automatic regulation, which provides a balance of the voltage drop variation during a day.

The voltage control law considers voltage drop up to the center of load in secondary networks. Thus the substation voltage control law is calculated by the weighed mean for the power of the voltage drop until the center of load of each distribution transformer.

This work also implements an indicator, called “energy supplied outside the allowed limits” (EFPF). This indicator allows to measure the efficiency of networks, considering voltage levels violations. Thus, it is possible to evaluate the algorithms used in this analysis.

As regards the final results, the software called ASD-CT was developed to implement all the algorithms here proposed. This software uses the Java proceedings, according to modern concepts applied on software engineering.

Finally, it is important to emphasize that the ASD-CT software presented excellent outputs, considering radially operated distribution networks belong to the State Company of Electric Energy (CCEE-D)

LISTA DE FIGURAS

Figura 3.1 – Modelo 1 de rede de baixa tensão.....	28
Figura 3.2 – Modelo 2 de rede de baixa tensão.....	29
Figura 3.3 – Modelo 3 de rede de baixa tensão.....	29
Figura 4.4 – Modelo 4 de rede de baixa tensão.....	30
Figura 3.5 – Modelo 5A de rede de baixa tensão.....	30
Figura 3.6 – Modelo 5B de rede de baixa tensão.....	31
Figura 3.7 – Modelo 6 de rede de baixa tensão.....	31
Figura 3.8 – Modelo 7 de rede de baixa tensão.....	32
Figura 3.9 – Elemento A.....	33
Figura 3.10 – Elemento A com cargas.....	33
Figura 3.11 – Exemplo de cálculo do Elemento A com cargas.....	34
Figura 3.12 – Elemento B com cargas.....	35
Figura 3.13 – Exemplo de cálculo do Elemento B com cargas.....	35
Figura 3.14 – Modelo 5A – Esquemático.....	38
Figura 3.15 – Modelo 5B - Esquemático.....	39
Figura 3.16 – Modelo 6 - Esquemático.....	39
Figura 3.17 – Modelo 7 – Esquemático.....	41
Figura 3.18 – Representação do Elemento B.....	44
Figura 3.19 – Exemplo de um modelo composto.....	45
Figura 3.20 – Partes de um modelo composto.....	45
Figura 3.21 – Exemplo de Modelo 5A.....	47
Figura 3.21 – Exemplo de Modelo 5B.....	48
Figura 3.22 – Exemplo de Modelo 7.....	49
Figura 4.1 – Pal9-AL8 (Parte 1).....	57
Figura 4.2 – Pal9-AL8 (Parte 2).....	58
Figura 4.3 – Pal9-AL8 (Parte 3).....	58
Figura 4.4 – Apropriação de um Modelo Representativo.....	59
Figura 5.1: Pontos Críticos de Transgressão de Tensão.....	62
Figura 6.1 – Perfil simplificado de tensão em uma rede de distribuição.....	72
Figura 6.2 – Exemplo detalhado de perfil de tensão em redes de distribuição.....	74
Figura 6.3 – Alimentador convencional de uma rede de distribuição.....	79
Figura 6.4 – Elevação da Tensão em Função do Tempo.....	80
Figura 6.5 – Elevação da Tensão em Função da Corrente.....	81
Figura 6.6 – Equação da Elevação da Tensão em Função do Tempo.....	81
Figura 7.1 – ASD – Controle de Tensão.....	84
Figura 7.2 – Parâmetros EFP - 1.....	85
Figura 7.3 – Parâmetros EFP 2.....	86
Figura 7.4 – Curva de EFP.....	86
Figura 7.5 – Resultados - EFP.....	87
Figura 7.6 – Ajuste nos tapes.....	88
Figura 7.7 – Gráfico da Elevação da Tensão em Função do Tempo.....	88
Figura 7.8 – Resultados de EFP após ajuste.....	89
Figura A1.1 – Esboço genérico de uma rede de baixa tensão.....	99
Figura A1.2 – Representação de rede de baixa tensão.....	101
Figura A1.3 – Elemento estrutural simplificado “A”.....	102
Figura A1.4 – Situação 2 – Elemento A.....	103
Figura A1.5 – Situação 5 – Elemento A.....	106
Figura A1.6 – Situação 6 – Elemento A.....	108
Figura A1.7 – Situação 2 – Elemento B.....	110
Figura A1.8 – Situação 5 – Elemento B.....	112
Figura A1.9 – Situação 6 – Elemento B.....	113
Figura A1.10 – Modelo 5A - EFP.....	115
Figura A2.1 - Definição da lei de controle de tensão na SE.....	123

LISTA DE ABREVIATURAS E SIGLAS

AL: Alimentador;

ANEEL: Agência Nacional de Energia Elétrica;

ASD-CT: Análise de Sistemas de Distribuição – Controle de Tensão;

BT: Baixa Tensão;

CAD: Computer Aided Design – Desenho Assistido por Computador;

CC: Centro de Carga;

CEEE-D: Companhia Estadual de Distribuição de Energia Elétrica;

EFFP: Energia Fornecida Fora do Padrão;

PAL9: Subestação Porto Alegre 9;

P&D: Pesquisa e Desenvolvimento;

SE: Subestação;

TD: Transformador de Distribuição.

LISTA DE APÊNDICES

APÊNDICE I	99
<i>A1. Método Clássico para o Cálculo de EFPF</i>	<i>99</i>
<i>A1.1. Metodologia</i>	<i>99</i>
<i>A1.2. Modelo 1</i>	<i>100</i>
<i>A1.2.1. Situação 1</i>	<i>100</i>
<i>A1.2.2. Situação 2</i>	<i>100</i>
<i>A1.2.3. Situação 3</i>	<i>100</i>
<i>A1.3. Modelo 2</i>	<i>101</i>
<i>A1.3.1. Situação 1</i>	<i>101</i>
<i>A1.3.2. Situação 2</i>	<i>101</i>
<i>A1.3.3. Situação 3</i>	<i>102</i>
<i>A1.4. Cálculo de EFPF para as unidades elementares</i>	<i>102</i>
<i>A1.4.1. Cálculo de EFPF no Elemento A</i>	<i>102</i>
<i>A1.4.1.1 Situação 2</i>	<i>103</i>
<i>A1.4.1.2. Situação 5</i>	<i>106</i>
<i>A1.4.1.3. Situação 6</i>	<i>108</i>
<i>A1.3.2. Elemento B</i>	<i>109</i>
<i>A1.4.2.1. Situação 2</i>	<i>110</i>
<i>A1.4.2.1. Situação 5</i>	<i>112</i>
<i>A1.3.2.1. Situação</i>	<i>113</i>
<i>A1.4 – Cálculo de EFPF para o Modelos de Redes BT</i>	<i>113</i>
<i>A1.4.1 Modelo 3</i>	<i>113</i>
<i>A1.4.3. Modelo 5A</i>	<i>115</i>
APÊNDICE II	120
<i>A2. Determinação da Lei de Controle de Tensão na SE pela análise de Casos Extremos</i>	<i>120</i>
<i>A2.1. Caso 1</i>	<i>120</i>
<i>A2.2. Caso 2</i>	<i>121</i>
<i>A2.3. Caso 3</i>	<i>121</i>
<i>A2.4. Análise de um Exemplo</i>	<i>122</i>
APÊNDICE III	124
<i>A3. Análise de Sistemas de Distribuição – Controle de Tensão – Códigos Fonte</i>	<i>124</i>
<i>A3.1 – Algoritmo para Ajuste de Tapes (br.com.mega.ppgee.plugins.AjusteTaps)</i>	<i>124</i>
<i>A3.2 Algoritmo para ajuste da Lei de Controle de Tensão na SE (br.com.mega.ppgee.plugins.CentrodeCarga)</i>	<i>130</i>
<i>A3.3 – Algoritmo para cálculo da EFPF (br.com.mega.ppgee.plugins.EnergiaFornecidaForaDoPadrao)</i>	<i>137</i>
<i>A3.4 – Modelos Representativos</i>	<i>144</i>

SUMÁRIO

CAPÍTULO I	16
1. <i>INTRODUÇÃO</i>	16
1.1. Objetivos	18
1.2. Disposição dos capítulos	19
CAPÍTULO II	20
2. <i>REVISÃO BIBLIOGRÁFICA</i>	20
2.1. Análise do Setor Elétrico Recente	20
2.2. Histórico da Modelagem de Redes de Baixa Tensão	22
2.3. O Controle de Tensão nas Redes de Distribuição	23
2.4. Resumo do Capítulo	26
CAPÍTULO III	27
3. <i>MODELOS REPRESENTATIVOS DE REDES DE BAIXA TENSÃO</i>	27
3.1. Justificativa do Uso dos Modelos	27
3.2. Modelos Representativos Propostos	28
3.2.1. Modelo 1	28
3.2.2. Modelo 2	28
3.2.3. Modelo 3	29
3.2.4. Modelo 4	29
3.2.5. Modelo 5A	30
3.2.6. Modelo 5B	30
3.2.7. Modelo 6	31
3.2.8. Modelo 7	31
3.3. Metodologia Adotada para a Abordagem dos Modelos	32
3.3.1. Determinação da Queda de Tensão até o Final da Rede	32
3.3.1.1. Elemento A	33
3.3.1.2. Elemento B	35
3.3.1.3. Modelo 1	36
3.3.1.4. Modelo 2	37
3.3.1.5. Modelo 3	37
3.3.1.6. Modelo 4	37
3.3.1.7. Modelo 5A	38
3.3.1.8. Modelo 5B	39
3.3.1.9. Modelo 6	39
3.3.1.10. Modelo 7	41
3.4. Determinação da Queda de Tensão até o Centro de Carga	42
3.4.1. Elemento A	42
3.4.2. Elemento B	43
3.4.3. Modelos Compostos	44
3.4.3.1. Modelo 1	46
3.4.3.2. Modelo 2	46
3.4.3.3. Modelo 3	46
3.4.3.4. Modelo 4	46
3.4.3.5. Modelo 5A	47
3.4.3.6. Modelo 5B	48
3.4.3.7. Modelo 6	49
3.4.3.8. Modelo 7	49
3.5. Ausência e Disponibilidade de Parâmetros	51
3.6. Queda de Tensão até o centro de carga desconhecendo-se os valores de LR_0	53
3.7. Revisão do Capítulo	55

CAPÍTULO IV.....	56
4. REDES REAIS DE BAIXA TENSÃO	56
4.1. Conceitos e Características.....	56
4.2. Aplicação dos Modelos Representativos	56
4.3. Referências quanto à Localização	57
4.4. Topologias Reais.....	57
4.5. Apropriação dos Modelos Representativos.....	59
CAPÍTULO V.....	60
5. ENERGIA FORNECIDA FORA DO PADRÃO.....	60
5.1. Fundamentação e Significado	60
5.2. Cálculo de EFPF através algoritmo iterativo	61
5.2.1. Modelo 3.....	63
5.2.2. Modelo 4.....	64
5.2.3. Modelo 5A	64
5.2.4. Modelo 5B	65
5.2.5. Modelo 6.....	67
5.2.6. Modelo 7.....	68
5.3. Resumo do Capítulo	69
CAPÍTULO VI.....	71
6. MÉTODOS DE ADEQUAÇÃO DOS NÍVEIS DE TENSÃO	71
6.1. Contextualização	71
6.2. Ajuste dos Taps dos Transformadores de Distribuição.....	73
6.3. Escolha do regime de demanda para ajuste dos tapes	76
6.4. Escolha dos tapes baseado em EFPF	76
6.5. Lei de Controle de Tensão na Subestação	77
6.5.1. Análise Geral da Rede de Distribuição	77
6.6. Ajuste da Lei de Controle na SE	78
6.7. Resumo do Capítulo	82
CAPÍTULO VII.....	83
7. RESULTADOS OBTIDOS	83
7.1. Software Desenvolvido.....	83
7.3. Energia Fornecida Fora de Padrão	85
7.4. Ajuste dos tapes dos Transformadores de Distribuição.....	87
7.5. Lei de Controle de Tensão na SE	88
7.6. Melhora do Indicador EFPF após ajustes do Controle de Tensão	89
7.7. Resumo do Capítulo	90
CAPÍTULO VIII.....	91
8. CONCLUSÃO	91
8.1. Contribuições do trabalho.....	93
8.2. Perspectivas para Trabalhos Futuros.....	93
9. Referências Bibliográficas	94

CAPÍTULO I

1. INTRODUÇÃO

O fornecimento de energia com qualidade e dentro dos padrões recomendados para os níveis de tensão é um problema de extrema importância para todas as concessionárias de energia. Os níveis permitidos para as variações de tensão são estabelecidos e controlados por agentes reguladores e estão definidos através de várias resoluções (ANEEL, 2003).

A dificuldade para solução dos problemas de avaliação e otimização dos regimes de tensão está associada às necessidades de modelagem simultânea e adequada de toda a estrutura do sistema de distribuição: subestações, redes de média tensão, equipamentos, transformadores de distribuição, redes de baixa tensão e cargas dos consumidores.

Contudo, faz-se necessário às distribuidoras uma ferramenta que una estas metodologias de modelagem e torne possível a estimação de estado das redes de distribuição, incluindo a rede secundária. Tal ferramenta deve ter o papel de auxiliar a identificação de pontos críticos de transgressão de níveis de tensão, bem como auxiliar na alteração da configuração da rede para que estes problemas possam ser mais facilmente sanados.

Atualmente, algumas distribuidoras de energia dispõem de uma grande quantidade de medições de curvas de carga para vários grupos de consumidores e também informações bastante detalhadas sobre os elementos das redes de baixa tensão. Estas informações têm papel importante para a avaliação do regime de tensão e escolha dos meios corretos para o aumento da qualidade da energia.

Entretanto, o que mais se vê nas companhias distribuidoras de energia é a ausência de informações e/ou a pouca confiabilidade dos dados, principalmente no que diz respeito ao cadastro das redes secundárias. Estas redes possuem

características bastante dinâmicas, ou seja, sofrem constantes mudanças estruturais uma vez que são conectados e desconectados consumidores com grande frequência. Unindo-se estes fatores ao fato de que muitas companhias terceirizam este serviço, tem-se uma possível inviabilidade de manutenção dos cadastros atualizados nos sistemas de informações dessas companhias de distribuição. Nestes casos faz-se necessário o uso de uma abordagem alternativa que possibilite a estimação de estado, onde as informações sobre as redes secundárias são pobres ou ausentes.

Em geral, as redes de baixa tensão podem ser apresentadas através de modelos representativos. Neste trabalho utilizaram-se, como solução para o problema da ausência de informações topológicas das redes secundárias estes modelos representativos. Foram desenvolvidos oito modelos, baseados nas situações mais típicas encontradas nas redes secundárias.

Tendo em mãos um conjunto de informações satisfatórias, bem como uma metodologia capaz de tratar os dados adequadamente e tornar possível a estimação de estado nas redes primárias e secundárias, é possível, então, tratar o problema das transgressões nos níveis de tensão de forma mais adequada e abrangente.

Usualmente quando existe um problema nos níveis de tensão este é sanado através do controle da potência reativa na rede, ou até mesmo utilizando-se reguladores de tensão. Na primeira solução, caso os bancos de capacitores não possuam alguma forma de controle de atuação, isto pode acarretar em transgressões por sobretensão em horários de baixa demanda. Outro fator negativo seriam os investimentos necessários para a instalação destes equipamentos, como o caso dos reguladores de tensão que possuem grande eficiência, porém um elevado custo.

Os meios básicos de controle de tensão são os tapes dos transformadores de distribuição e o comutador de tapes sob carga na subestação. Estes recursos têm capacidade de solucionar transgressões nos níveis de tensão em boa parte dos casos, quando devidamente configurados. No entanto, muitas vezes são sub-utilizados. Além disto, o custo de sua utilização é reduzido, visto que estes já existem. Portanto, o custo de reconfiguração dos meios básicos de controle de tensão é apenas o operacional. Por estes motivos este trabalho dá ênfase aos meios básicos de controle de tensão como forma de minimizar as transgressões dos níveis de tensão nas redes de distribuição.

Outro ponto importante para a definição da configuração dos meios básicos de controle de tensão é a utilização das informações de queda de tensão também nas redes secundárias. Isto torna o controle de tensão mais eficiente para todos os consumidores, não somente para os ramais conectados mais próximos aos transformadores de distribuição.

Logo esteja realizada a reconfiguração destes meios básicos de controle de tensão, é necessário ter-se em mãos também uma ferramenta que auxilie a mensuração do desempenho obtido pela rede no que diz respeito às transgressões nos níveis de tensão. Esta ferramenta, desenvolvida nesta dissertação, representa um indicador, denominado E.F.F.P. – Energia Fornecida Fora do Padrão. Este foi definido para que fosse possível comparar os resultados de melhoria obtidos pelo algoritmo quando utilizado em situações reais.

E por fim, esta dissertação também apresenta um Software, denominado ASD – Controle de Tensão, que foi desenvolvido baseado em todas as metodologias e algoritmos aqui propostos. Neste Software foi possível verificar de maneira prática os resultados obtidos a partir das alterações nos tapes dos transformadores de distribuição em um alimentador real da Companhia Estadual de Distribuição de Energia Elétrica, em Porto Alegre.

1.1. Objetivos

A realização deste trabalho tem os seguintes objetivos:

- A criação de modelos representativos das redes de baixa tensão que tornem possível a estimação de estado nestas redes, mesmo com ausência de informações de cadastro sobre as mesmas;
- Tratamento das transgressões nos níveis de tensão nas redes de distribuição fazendo ajuste apenas dos meios básicos de controle de tensão: tapes dos transformadores de distribuição e a Lei de Controle de Tensão na SE (comutador de tapes sob carga);
- Utilizar uma metodologia que leve em consideração as quedas na rede secundária para o ajuste dos meios básicos de controle de tensão;
- Desenvolvimento de um indicador para medir o desempenho das redes quanto às transgressões nos níveis de tensão;

- Implementação de um Software que torne possível pôr em prática o ajuste dos meios básicos de controle de tensão, em redes reais de distribuição, utilizando as metodologias e algoritmos propostos nesta dissertação.

1.2. Disposição dos capítulos

No Capítulo 1 faz-se uma apresentação em termos gerais do trabalho, bem como seus objetivos e motivações principais.

No capítulo 2 é feita uma revisão bibliográfica de trabalhos de outros autores que abordaram temas relacionados a esta dissertação, bem como situa o leitor dentro do contexto histórico do setor energético.

O Capítulo 3 apresenta os Modelos Representativos de Redes de Baixa Tensão, bem como a metodologia de cálculo necessária para a estimação de estado nas redes secundárias fazendo uso destes.

O Capítulo 4 trata das redes reais de baixa tensão, e como os modelos representativos apresentados anteriormente podem ser utilizados para estimação de estado.

O Capítulo 5 apresenta o indicador EFP e a forma obtenção deste através da metodologia de modelos representativos.

O Capítulo 6 define como os meios básicos de controle de tensão são utilizados para minimizar as transgressões nos níveis de tensão.

No Capítulo 7 é demonstrado o Software desenvolvido e também os resultados obtidos com a aplicação desta em situações reais.

Finalmente no Capítulo 8, é feita a Conclusão do trabalho, assim como as contribuições feitas por este e também são abertas possibilidades de trabalhos futuros.

CAPÍTULO II

2. REVISÃO BIBLIOGRÁFICA

Neste capítulo será apresentada uma série de estudos realizados na área de modelos de redes de baixa tensão, estimação de estado e controle de tensão em sistemas de distribuição. Estes trabalhos serviram como suporte técnico e metodológico para esta dissertação, onde foram observados todos os avanços que os estudos nesta área tiveram ao longo dos anos.

2.1. Análise do Setor Elétrico Recente

A partir do final da década de 70, o setor elétrico internacional começou a passar por profundas mudanças, iniciadas pelo governo de Margaret Thatcher na Inglaterra, que defendia a redução da presença do Estado na economia (ROSA, 2008).

Começavam assim vários debates, em diversos países, sobre as conseqüências da privatização de monopólios naturais e os diferentes modelos para torná-la viável.

No Brasil, a transição para uma nova política de energia, iniciada na década de 90, vem sofrendo várias influências, como a grave crise financeira do Estado, que provoca a deterioração da administração pública e a impossibilidade de transferência de recursos do Tesouro para o setor de eletricidade.

No cenário internacional, com a globalização da economia, começaram a predominar as transações estritamente financeiras sobre o comércio de bens e serviços.

No campo político brasileiro, disputas estaduais e a desobediência civil das concessionárias regionais, em relação ao cumprimento das decisões do órgão regulador e à Constituição Federal de 1988 – que não atribuía para o Poder Público

estadual e municipal responsabilidades explícitas de investimentos na expansão dos serviços de energia elétrica – comprometeram o alcance das diretrizes energéticas nacionais.

Outros aspectos relevantes foram: o aumento da força política de grandes consumidores, pressionando a redução de tarifas; e a forte influência de empreiteiros, que passavam a interferir no plano de obras das concessionárias públicas, pois dependiam dessas obras estatais para suas ações empresariais.

A reunião desses ingredientes, somados a outros – como a falta de estímulo à eficiência – levou ao aumento do endividamento e à inviabilidade financeira do setor elétrico.

Visando corrigir os principais problemas discutidos, o governo Fernando Henrique Cardoso optou pela privatização das empresas estatais federais e exerceu forte influência sobre os governos estaduais, para que também trilhassem esse caminho.

Dentre as semelhanças observadas na reestruturação de diversos países que também implementaram esse tipo de participação privada, percebe-se forte estímulo à concorrência, com a tentativa de se manter um grau satisfatório de regulamentação, que não prejudique a liberdade de mercado e estabeleça regras para garantir os interesses das Nações.

Com a privatização das empresas concessionárias de energia elétrica, foram assinados contratos de concessão com o Poder Concedente (Agência Nacional de Energia Elétrica – ANEEL), que estabelecem um conjunto de exigências, obrigações, procedimentos, indicadores de natureza técnica e comercial, além de penalidades, com o objetivo de assegurar a qualidade dos serviços prestados por essas empresas aos consumidores.

A legislação passou a estabelecer limites diferenciados para cada uma das empresas concessionárias, baseando-se em seus históricos, visando preservar e/ou aumentar a qualidade do fornecimento existente antes das privatizações.

O Estado, através da ANEEL e de órgãos estaduais, passou a exercer as atividades de regulador e fiscalizador das concessionárias, quanto ao correto cumprimento das obrigações assumidas.

Algumas empresas já estão atuando dentro desse novo contexto e assinaram Contratos de Concessão, comprometendo-se em cumprir todas as regras

relacionadas à sistemática de controle da qualidade do fornecimento de energia elétrica, neles dispostas.

Esse controle está sendo executado em etapas sucessivas, considerando indicadores e padrões individuais e coletivos, sendo que, a violação dos padrões individuais definidos pode gerar penalidades para as concessionárias em favor dos consumidores, assim como a violação de padrões coletivos acarreta recolhimentos ao órgão regulador (ANEEL, 2003).

Os vários indicadores controlados contemplam três enfoques: a qualidade do atendimento comercial, a qualidade do produto e a qualidade do serviço.

2.2. Histórico da Modelagem de Redes de Baixa Tensão

A modelagem de redes de baixa tensão, com objetivos tanto de planejamento quanto de operação, é um assunto que começou a ser tratado a partir dos anos 60 com o auxílio de algoritmos computacionais nos trabalhos de BURANDT (1961) e mais tarde BACKLUND (1978). A motivação para este tratamento foi o aparecimento dos computadores digitais e, como consequência, o desenvolvimento de modelos computacionais que conseguiram representar as redes de baixa tensão existentes e tornar possíveis os cálculos para estimação de estado.

Uma das características das redes de baixa tensão que dificultava as análises era a enorme quantidade de dados envolvidos. Pode-se imaginar isto, observando-se que cada transformador de distribuição obrigatoriamente gera um circuito de baixa tensão. Se, associarem-se a estas redes, algumas características como bitola de cabo, tipo de rede, localização, carregamento máximo e médio, dentre outras, rapidamente obtém-se um volume de informações bastante grande.

KAGAN (1988) propôs um trabalho considerando que uma área elementar poderia ser atendida por alguns modelos representativos já estabelecidos e utilizou a técnica de programação dinâmica para seleção da alternativa com mínimo custo operacional.

Nos anos 90, o CODI (www.abradee.org.br) publicou diversos trabalhos de planejamento de redes de baixa tensão e representação destas redes, entre outros assuntos. Posteriormente ANTUNES (1999) complementou o trabalho de KAGAN apresentando técnicas de modelagem para os modelos representativos. MÉFFE

(2001) em um projeto de P&D na concessionária Gaúcha RGE, propôs a utilização de modelos representativos para o cálculo de perdas técnicas nas redes secundárias onde a base cadastral destas redes era insuficiente.

Em outro trabalho, SQUAIELLA (2004), também propõe a utilização de modelos de redes secundárias, entretanto com o objetivo de planejamento, e previsão de crescimento destes circuitos secundários.

Atualmente, a ANEEL está elaborando o PRODIST, que são documentos na forma de regulamentações, normatizações e padronizações que tem como objetivo disciplinar todos os aspectos técnicos relativos ao planejamento e operação das redes de distribuição. A Nota Técnica 00035/2007 que trata de metodologia e procedimentos para a apuração de perdas técnicas já propõe a utilização de modelos representativos para o cálculo de perdas nas redes BT.

2.3. O Controle de Tensão nas Redes de Distribuição

A qualidade é uma das mais importantes características da energia elétrica fornecida aos clientes. Considerando, que energia de má qualidade pode ser fonte de grandes prejuízos para os consumidores, cada país tem um padrão rígido que define os principais indicadores de qualidade da energia (ANEEL, 2003). O nível permissível de tensão nos pontos de entrega de energia para os consumidores é um destes indicadores. Os níveis de tensão em redes de distribuição sofrem alterações de acordo com as variações de carga que ocorrem ao longo do dia. Para compensar estas alterações, tornando a tensão o mais estável possível para os consumidores, utilizam-se os meios básicos de controle como os tapes dos transformadores de distribuição e transformadores nas subestações com o comutador de tap sob carga. No caso destas medidas não atingirem os efeitos desejados podem ser utilizados equipamentos adicionais, como por exemplo, reguladores de tensão e bancos de capacitores.

Especificações importantes do controle de tensão estão relacionadas com a necessidade de considerar, ao mesmo tempo, vários elementos do sistema: transformadores na subestação, conjunto dos alimentadores, transformadores de distribuição, redes de baixa tensão, consumidores. Isto exige o processamento de

grande quantidade de informação, que são: topologia e parâmetros das redes, cargas horárias dos consumidores, parâmetros dos transformadores, etc.

O controle ótimo de tensão inclui três problemas relativamente independentes: avaliação dos níveis de tensão em vários pontos do sistema, escolha de parâmetros nos equipamentos de controle de tensão, definição da necessidade de utilização de meios adicionais de controle de tensão e seus locais de instalação.

A estimação de estado é um dos problemas mais importantes para análise e controle dos sistemas de distribuição. A avaliação dos níveis de tensão é parte deste problema. Por isso, todas as dificuldades associadas com a modelagem dos elementos que compõem as redes de distribuição, utilização de curvas típicas de carga e definição dos níveis de tensão em sistemas de distribuição estão representadas por uma grande quantidade de bibliografia, por exemplo, o trabalho de CESPEDDES (1990) sugere a estimação de estado baseado em um equivalente elétrico, eliminando o cálculo do ângulo de fase nas equações. No artigo de CHEN (1990) é apresentada uma abordagem mais rígida para o cálculo de fluxo de potência, considerando desbalanceamento de fases, geração distribuída, e rede primária e secundária inclusive. Os trabalhos de BARAN (1995) e BERNARDON (2004) fazem a estimação de estado através de um método recursivo baseado na hierarquia dos nós da rede de distribuição. MIRANDA (1990) trata o problema do fluxo de potência utilizando lógica difusa, permitindo incorporar incertezas na geração e representação das cargas.

Um bom método para o cálculo de fluxo de potência deve ser empregado em conjunto com um método de previsão de carga que considere as variações de demanda ao longo das 24 horas, então JARDINI (1999) e KONIG (2004) em seus trabalhos propuseram a utilização de curvas típicas de carga e classificação de consumidores de acordo com esta de forma a prever a curva de carga diária de cada consumidor (ou conjunto destes) baseado no seu consumo mensal.

A falta de dados confiáveis é típica para os sistemas de distribuição. Então, foram desenvolvidos vários algoritmos para a modelagem dos modos de operação, considerando ausência das informações (medidas diretas) necessárias, como WANG (1992) que trata o problema da incerteza dos dados na estimação de estado em redes de distribuição utilizando intervalos aritméticos. Já LEOU (1996) tratou a incerteza dos dados utilizando as informações de telemetria colhidas na SE.

De uma maneira mais geral, o problema do controle de tensão nas redes de distribuição é apresentado nos trabalhos de CALOVIC (1984), CHANG (1990) e THORNLEY (2001). Porém, na maioria das modernas metodologias, o problema de controle de tensão é analisado juntamente com o controle de potência reativa, algumas vezes incluindo a redução das perdas de energia com o objetivo de otimização. No trabalho de BARAN (1999) o uso dos capacitores na SE é feito coordenado com o ajuste do comutador de tapes sob carga. LIANG (2001) propõe o controle de tensão na SE utilizando o comutador de tapes sob carga e o controle de reativos. É feita uma previsão de carga horária considerando um número máximo diário de chaveamentos.

Em 2003 LIANG propõe a coordenação na atuação dos capacitores na SE de forma a reduzir o número de chaveamentos diários do comutador e manter a tensão em nível adequado. Este algoritmo foi implementado utilizando lógica difusa. Nesta mesma linha de uso de lógica difusa para o Controle de Tensão na SE existem ainda os trabalhos de KASZTENNY (1998), LU (1997) e SU (2001).

Neste caso, os níveis de tensão e número permissível de operação dos tapes dos transformadores nas subestações são considerados como restrições.

Em alguns trabalhos, para aumento da eficiência das soluções, é proposta a utilização de novos equipamentos eletrônicos de controle dos tapes dos transformadores, como em JIANG (2000) que utiliza um novo controlador de tensão na SE que processa leituras das três fases e controla tensão em sistemas desbalanceados.

Nos últimos anos, muitas publicações apresentam os vários aspectos da integração de fontes de geração distribuída em sistemas de distribuição, particularmente, considerando sua influência para o controle de tensão. CANHA (2007) faz uma análise do problema de controle de tensão nos sistemas de distribuição com a utilização de fontes de armazenamento e geração distribuída. Outros trabalhos como WAGNER (1990), HAUBRICH (1997), FAN (1997), CHOI (2001) e BONHOMME (2001) e consideram as fontes de armazenamento e geração distribuídas para analisar a posição dos tapes no comutador de tapes sob carga na SE.

Em 1996, STANGARLIN (1996) propôs o controle de tensão em sistemas de distribuição utilizando os meios básicos de controle de tensão, na subestação e nos

transformadores de distribuição. Neste mesmo trabalho também foi proposto um método de estimação da qualidade de energia nas redes secundárias.

Por fim, é relevante salientar que os métodos utilizados para a avaliação dos níveis de tensão devem ser adaptados para a quantidade e a qualidade das informações realmente disponíveis nas concessionárias.

2.4. Resumo do Capítulo

Neste capítulo foram apresentadas várias bibliografias acerca do tema ao qual esta dissertação refere-se, como o controle de tensão nas redes de distribuição e a utilização de modelos representativos de redes secundárias. Na seção 1 foi feito um pequeno balanço histórico do setor energético no Brasil e no mundo.

Na segunda seção foram apresentados estudos realizados sobre a utilização de modelos simplificados para a operação ou planejamento das redes de baixa tensão. O uso destes modelos simplificados está sendo proposto pela ANEEL no PRODIST, o que tornará o seu uso bastante difundido nas distribuidoras de energia elétrica no Brasil.

Na seção 3 foram apresentados trabalhos anteriormente realizados por outros autores na área de controle de tensão em redes de distribuição. Os trabalhos apontados estão bastante focados na realização do controle da tensão apenas na SE, não levando em consideração os circuitos de baixa tensão como relevantes para os cálculos de compensação das quedas de tensão, ao contrário do que está sendo proposto na presente dissertação.

CAPÍTULO III

3. MODELOS REPRESENTATIVOS DE REDES DE BAIXA TENSÃO

3.1. Justificativa do Uso dos Modelos

As concessionárias de energia geralmente não possuem um cadastro confiável e completo acerca das suas redes de baixa tensão. Há diversas dificuldades associadas à sua determinação, tais como o grande número de circuitos, a falta de dados atualizados, a diversidade de topologias e o seu comportamento dinâmico.

Por isso, o trabalho propõe a utilização de modelos representativos de determinados grupos de redes de baixa tensão.

No desenvolvimento destes modelos, foram feitas as seguintes hipóteses:

- Todos os consumidores são trifásicos com distribuição de carga uniforme entre as fases.
- A carga ao longo da rede de baixa tensão tem distribuição uniforme.
- Em regime de demanda máxima, a queda de tensão na rede de baixa tensão corresponde à queda de tensão permissível pelas normas de projeto.

Estas considerações são necessárias para contornar os vários problemas decorrentes da falta de dados disponíveis.

Têm-se, como exemplos destes problemas, as seguintes constatações:

- A ausência do conhecimento da fase em que cada consumidor está ligado.
- A indeterminação do ponto da rede em que o consumidor está ligado.
- O desconhecimento da topologia da rede analisada.

- A falta de determinação do condutor utilizado.

3.2. Modelos Representativos Propostos

Foram desenvolvidos oito tipos de modelos representativos de redes de baixa tensão. A seguir, encontram-se estes oito tipos, seguidos de suas metodologias construtivas.

Para cada um dos modelos tem-se:

- N: número de consumidores do circuito de baixa tensão.

3.2.1. Modelo 1

No modelo 1, na Figura 3.1, verifica-se o mais simples de todos os modelos.

Nele, o consumidor está ligado diretamente ao transformador de distribuição. Neste caso, as perdas na rede de baixa tensão são desconsideradas.

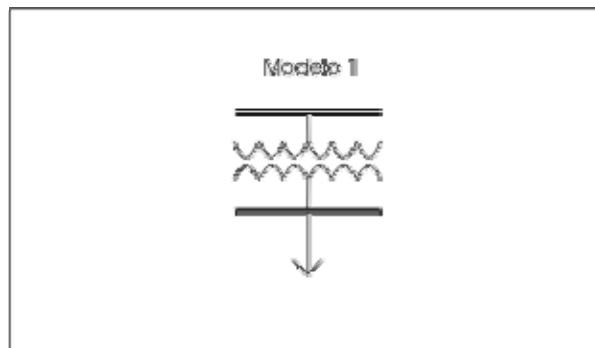


Figura 3.1 – Modelo 1 de rede de baixa tensão.

3.2.2. Modelo 2

O modelo 2, presente na Figura 3.2, é definido por apenas um consumidor conectado ao final da rede de baixa tensão.

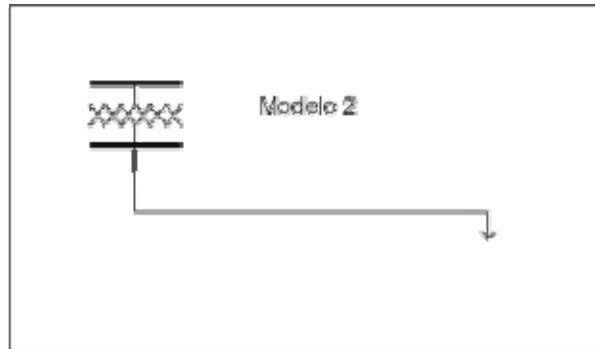


Figura 3.2 – Modelo 2 de rede de baixa tensão.

3.2.3. Modelo 3

O modelo 3, na Figura 3.3, representa uma rede de baixa tensão onde os 'n' consumidores de mesma potência estão uniformemente distribuídos ao longo desta.

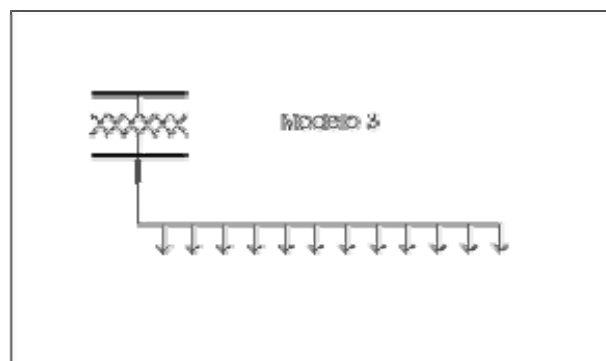


Figura 3.3 – Modelo 3 de rede de baixa tensão.

3.2.4. Modelo 4

O modelo 4, verificado na Figura 3.4, representa uma rede de baixa tensão onde os consumidores de igual potência estão homogeneamente distribuídos.

O transformador de distribuição, no entanto, está localizado numa posição central da rede.

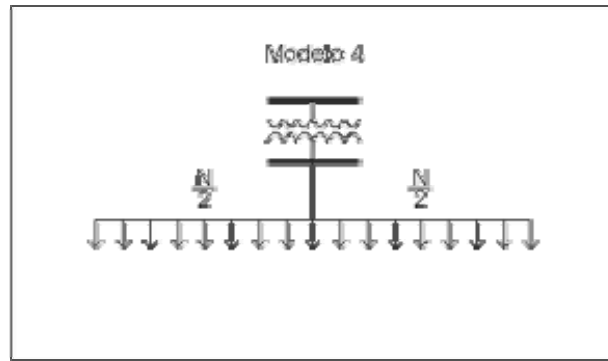


Figura 3.4 – Modelo 4 de rede de baixa tensão.

3.2.5. Modelo 5A

O Modelo 5A, presente na Figura 3.5, possui dois trechos de rede com consumidores de igual potência uniformemente distribuídos ao longo de cada trecho. Um destes trechos está ligado centralmente à extremidade do outro.

No modelo 5A, os dois trechos possuem $N/2$ consumidores.

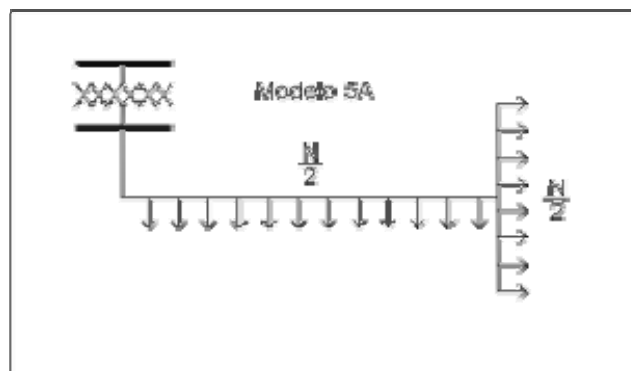


Figura 3.5 – Modelo 5A de rede de baixa tensão.

3.2.6. Modelo 5B

No modelo 5B, na Figura 3.6, um trecho possui $N/3$ consumidores e o outro possui $2N/3$ consumidores.

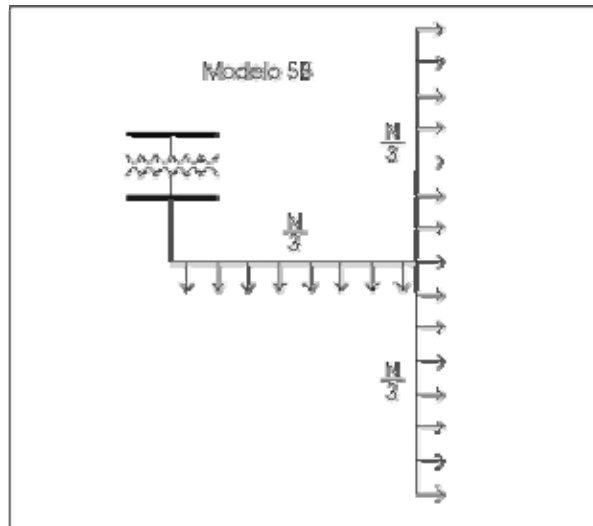


Figura 3.6 – Modelo 5B de rede de baixa tensão.

3.2.7. Modelo 6

No modelo 6, presente na Figura 3.7, a fonte está localizada no centro, sendo que uma das extremidades possui outro trecho de rede conectado.

Todos os três trechos possuem as mesmas concentrações de carga.

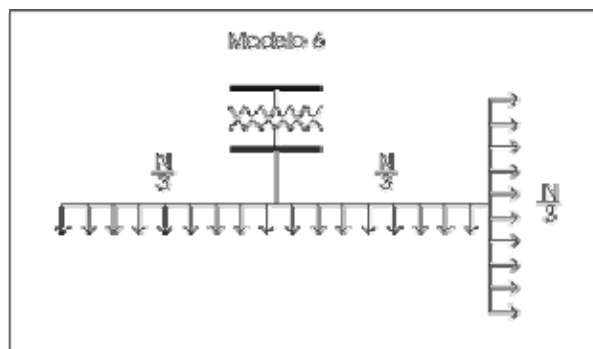


Figura 3.7 – Modelo 6 de rede de baixa tensão.

3.2.8. Modelo 7

O modelo 7, visualizado na Figura 3.8, é similar ao modelo 5A, exceto pela simetria. Ele é composto por dois modelos 5A em cada um dos lados do transformador de distribuição.

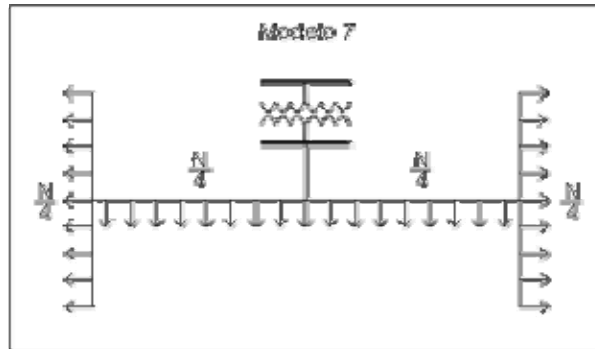


Figura 3.8 – Modelo 7 de rede de baixa tensão.

3.3. Metodologia Adotada para a Abordagem dos Modelos

Para o cálculo das quedas de tensão nos modelos de redes de baixa tensão, elaborou-se a seguinte metodologia.

Para cada um dos modelos, foram determinadas as quedas de tensão até o final das redes modeladas. Da mesma forma, calcularam-se as quedas de tensão até os centros de carga.

3.3.1. Determinação da Queda de Tensão até o Final da Rede

Para a determinação dos valores correspondentes à queda de tensão até o final das redes de baixa tensão, referentes a cada um dos modelos, subdividiram-se as estruturas em dois elementos.

Desenvolveu-se então o cálculo acerca de um Elemento A e um Elemento B, a partir dos quais todos os modelos possam ser considerados uma composição dos mesmos.

Abaixo as principais variáveis utilizadas ao longo desta seção:

- ΔV_f : queda de tensão até o final da linha;
- ΔV_f^{ex} : queda de tensão exata até o final da linha;
- ΔV_f^s : queda de tensão aproximada até o final da linha;
- L: comprimento total da rede BT;
- R_o : resistência do condutor;
- I: corrente no trecho;
- I_Σ : somatório da corrente de todos os consumidores.

3.3.1.1. Elemento A

Calcula-se a queda de tensão, tendo-se conhecimento do comprimento do condutor L , de sua resistência R_o e da corrente I , que atravessa este condutor. Tal cálculo é expresso na equação 3.1.

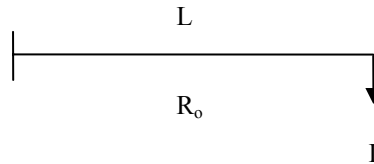


Figura 3.9 – Elemento A.

$$\Delta V_f = ILR_o \quad (3.1)$$

O Elemento A se caracteriza por apresentar somente um trecho de rede, no qual as cargas $I(x)$ estão uniformemente distribuídas ao longo do condutor de comprimento L e resistência R_o .

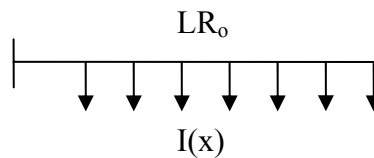


Figura 3.10 – Elemento A com cargas.

Para este caso, a queda de tensão até o final da linha pode ser calculada através da equação 2.

$$\Delta V_f = \int_0^L I(x) dx R_o = \frac{ILR_o}{2} \quad (3.2)$$

As cargas, no entanto, não estão distribuídas uniformemente. Ou seja, o número de cargas não é infinito. Existe, na verdade, um espaço entre cada carga, o que acarreta um erro com a utilização da equação 3.2.

No exemplo abaixo, este fato é evidenciado.

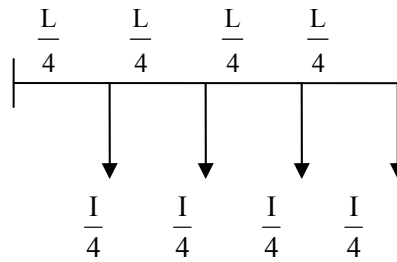


Figura 3.11 – Exemplo de cálculo do Elemento A com cargas.

Para cálculos exatos, tem-se o resultado mostrado na equação 3.3.

$$\Delta V_f^{ex} = I \cdot \frac{L}{4} R_o + \frac{3}{4} \cdot I \cdot \frac{L}{4} R_o + \frac{2}{4} \cdot I \cdot \frac{L}{4} R_o + \frac{1}{4} \cdot I \cdot \frac{L}{4} R_o = \frac{10ILR_o}{16} = 0,625ILR_o \quad (3.3)$$

Para cálculos que utilizam a equação 3.2, tem-se, como resultado, o valor encontrado na equação 3.4.

$$\Delta V_f^s = \frac{ILR_o}{2} = 0,5ILR_o \quad (3.4)$$

Para que a equação 3.2 seja utilizada, deve-se, por isso, definir um fator de correção chamado Kv_1 .

$$\Delta V_f = \Delta V_f^s \cdot Kv_1 \quad (3.5)$$

Onde: $Kv_1 = f(N)$

É mostrado, na equação 3.6, o valor do fator de correção correspondente ao exemplo ilustrado na figura 3.11.

$$Kv_1(4) = \frac{0,625}{0,5} = 1,25 \quad (3.6)$$

3.3.1.2. Elemento B

No Elemento B, a carga $I(x)$ está distribuída também uniformemente, porém com uma concentração I na sua extremidade.

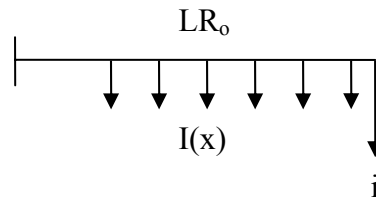


Figura 3.12 – Elemento B com cargas.

Pode-se definir a queda de tensão no final da linha através da equação 3.7.

$$\Delta V_f = \int_0^L [I(x) + i] dx R_o = \frac{ILR_o}{2} + iLR_o \quad (3.7)$$

Tendo-se $i = I$, então:

$$\Delta V_f = \frac{3}{2} ILR_o \quad (3.8)$$

Uma vez que as cargas também não estão distribuídas uniformemente, há novamente um erro com a utilização da equação 3.7.

No exemplo, ilustrado na figura 3.13, evidencia-se a presença deste erro.

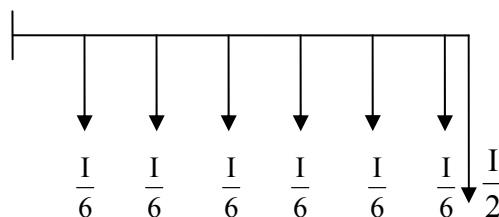


Figura 3.13 – Exemplo de cálculo do Elemento B com cargas.

Para a representação exata da queda de tensão até o final da rede BT, representada por ΔV_f^{ex} , tem-se o resultado mostrado na equação 3.9.

$$\Delta V_f^{ex} = \frac{1}{9} \cdot I \cdot \frac{L}{6} \cdot R_o + \frac{8}{6} \cdot I \cdot \frac{L}{6} \cdot R_o + \frac{7}{6} \cdot I \cdot \frac{L}{6} \cdot R_o + \frac{6}{6} \cdot I \cdot \frac{L}{6} \cdot R_o + \frac{5}{6} \cdot I \cdot \frac{L}{6} \cdot R_o + \frac{4}{6} \cdot I \cdot \frac{L}{6} \cdot R_o = \frac{39}{36} ILR_o \quad (3.9)$$

Para a representação que faz uso da equação 3.7, o resultado está ilustrado na equação 3.10.

$$\Delta V_f^s = \frac{ILR_o}{2} + \frac{1}{2} ILR_o = ILR_o \quad (3.10)$$

Introduz-se então um fator de correção chamado Kv_2 .

$$\Delta V_f = \Delta V_f^s \cdot Kv_2 \quad (3.11)$$

$$\text{Onde: } Kv_2 = f\left(\frac{N_A}{N_B}, N\right);$$

N_A : número de consumidores no Elemento A;

N_B : número de consumidores no Elemento B.

Para o exemplo, ilustrado na figura 3.13, tem-se o fator de correção de erro mostrado na equação 3.12.

$$Kv_2 = \frac{\Delta V_f^{ex}}{\Delta V_f^s} = 1,083 \quad (3.12)$$

A partir da metodologia de cálculo exposta para os Elementos A e B, chega-se às fórmulas correspondentes ao cálculo da queda de tensão até o final da linha para todos os modelos desenvolvidos.

3.3.1.3. Modelo 1

A queda de tensão, neste caso, é nula, uma vez que a rede de baixa tensão não existe.

$$\Delta V_f = 0 \quad (3.13)$$

3.3.1.4. Modelo 2

Existe, neste modelo, somente uma carga concentrada na extremidade do condutor. Por isso, a queda de tensão é dada pela equação 3.14.

$$\Delta V_f = I_{\Sigma} L R_o \quad (3.14)$$

3.3.1.5. Modelo 3

Este modelo equivale ao Elemento A.

O valor da queda de tensão até o final da linha é dado pela equação 3.15.

$$\Delta V_f = \frac{1}{2} I_{\Sigma} L R_o K_{v_1}(N) \quad (3.15)$$

3.3.1.6. Modelo 4

Este modelo, na sua constituição, verifica-se equivalente a dois Elementos A. Sendo, cada um deles, metade dos consumidores.

O valor da queda de tensão é dado pela equação 3.16.

$$\Delta V_{f_1} = \Delta V_{f_2} = \frac{1}{4} I_{\Sigma} L R_o K_{v_1} \left(\frac{N}{2} \right) \quad (3.16)$$

Onde:

ΔV_{f_1} : queda de tensão até o final da linha no ramo esquerdo do modelo representativo;

ΔV_{f_2} : queda de tensão até o final da linha no ramo direito do modelo representativo.

3.3.1.7. Modelo 5A

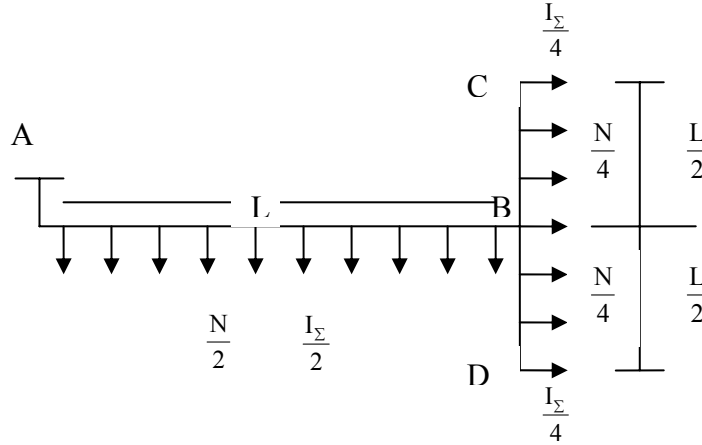


Figura 3.14 – Modelo 5A – Esquemático

O modelo 5 é composto de um Elemento B e dois Elementos A. No trecho A-B está identificado o Elemento B e nos trechos B-C e B-D estão os Elementos A.

$$\Delta V_{AC} = \Delta V_{AD} = \Delta V_{AB} + \Delta V_{BC} = \Delta V_{AB} + \Delta V_{BD}$$

$$\Delta V_{BC} = \Delta V_{BD} \text{ (Elemento A)} = \frac{\frac{I_{\Sigma}}{4} \cdot \frac{L}{2} \cdot R_o}{2} = \frac{I_{\Sigma} L R_o}{16} \cdot K_{V1} \left(\frac{N}{4} \right)$$

$$\Delta V_{AB} \text{ (Elemento B com } i = \frac{I_{\Sigma}}{2} \text{)}$$

$$\left(\frac{\frac{I_{\Sigma}}{2} \cdot L \cdot R_o}{2} + \frac{I_{\Sigma}}{2} \cdot L \cdot R_o \right) \cdot K_{V2} \left(1, \frac{N}{2} \right) = \frac{3 I_{\Sigma} L R_o}{4} \cdot K_{V2} \left(1, \frac{N}{2} \right)$$

O valor final da queda de tensão para este modelo é dado pela equação 3.18.

$$\Delta V_f = \frac{3}{4} \cdot I_{\Sigma} \cdot L \cdot R_o \cdot K_{V2} \left(1, \frac{N}{2} \right) + \frac{1}{16} \cdot I_{\Sigma} \cdot L \cdot R_o \cdot K_{V1} \left(\frac{N}{4} \right) \quad (3.17)$$

$$\frac{I_{\Sigma} L R_o}{16} \left[12 K_{V2} \left(1, \frac{N}{2} \right) + K_{V1} \left(\frac{N}{4} \right) \right] \quad (3.18)$$

3.3.1.8. Modelo 5B

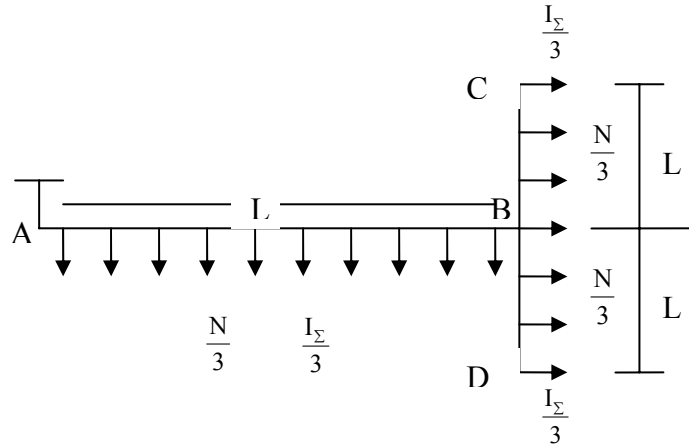


Figura 3.15 – Modelo 5B - Esquemático

Para este modelo, o valor da queda de tensão até o final da linha é dado pela equação 3.20.

$$\Delta V_{AB} = \frac{1}{2} \cdot \frac{I_{\Sigma}}{3} \cdot L \cdot R_o + \frac{2}{3} \cdot I_{\Sigma} \cdot L \cdot R_o = \frac{5I_{\Sigma}LR_o}{6} \cdot K_{V_2} \left(2, \frac{N}{3} \right) \quad (3.19)$$

$$\frac{I_{\Sigma}LR_o}{6} \left[5K_{V_2} \left(2, \frac{N}{3} \right) + K_{V_1} \left(\frac{N}{3} \right) \right] \quad (3.20)$$

3.3.1.9. Modelo 6

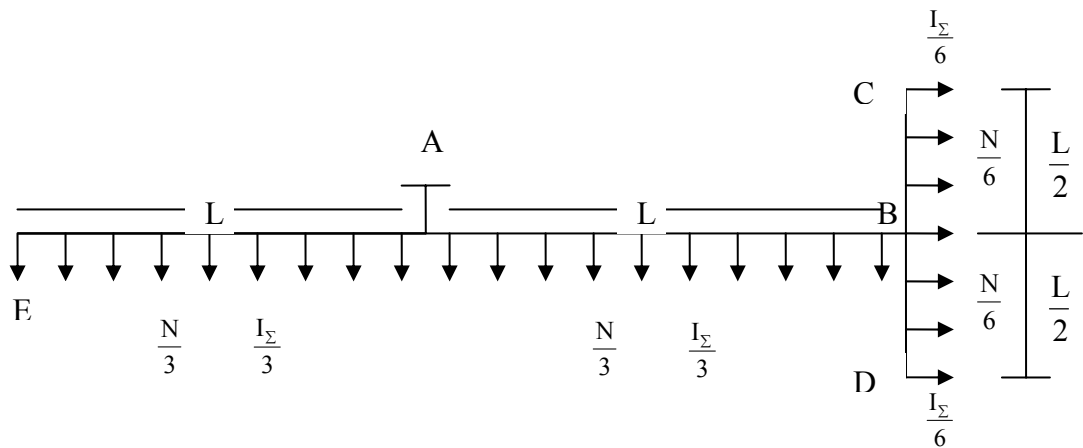


Figura 3.16 – Modelo 6 - Esquemático

Este modelo é constituído de Elementos A em três diferentes trechos e de um Elemento B em um único trecho.

Neste modelo, é bastante provável que se utilizem condutores diferentes nos trechos que o compõe. Tais condutores apresentariam valores de resistência R_o' e R_o'' . Nos trechos AB, BC e BD será considerada a utilização do condutor de resistência R_o' , e no trecho AE será considerada a utilização do condutor de resistência R_o'' .

Os valores das quedas de tensão até o final da linha, para diferentes valores de resistência, são dados pelas equações 3.24 e 3.25.

$$\Delta V_{BC} = \Delta V_{BD} = \frac{1}{2} \cdot \frac{I_\Sigma}{6} \cdot \frac{L}{2} \cdot R_o' \cdot K_{V1} \left(\frac{N}{6} \right) = \frac{I_\Sigma L R_o'}{24} \cdot K_{V1} \left(\frac{N}{6} \right) \quad (3.21)$$

$$\Delta V_{AB} = \frac{1}{2} \cdot \frac{I_\Sigma}{3} \cdot L R_o' + \frac{I_\Sigma}{3} \cdot L R_o' = \frac{1}{2} \cdot I_\Sigma L R_o' \cdot K_{V2} \left(1, \frac{N}{3} \right) \quad (3.22)$$

$$\Delta V_f' = \frac{1}{2} \cdot \frac{I_\Sigma}{3} \cdot L R_o' \cdot K_{V2} \left(1, \frac{N}{3} \right) + \frac{I_\Sigma L R_o'}{24} \cdot K_{V1} \left(\frac{N}{6} \right) \quad (3.23)$$

$$\Delta V_f' = \frac{I_\Sigma L R_o'}{24} \left[12 K_{V2} \left(1, \frac{N}{3} \right) + K_{V1} \left(\frac{N}{6} \right) \right] \quad (3.24)$$

$$\Delta V_f'' = \frac{1}{2} \cdot \frac{I_\Sigma}{3} \cdot L R_o'' = \frac{I_\Sigma L R_o''}{6} \cdot K_{V1} \left(\frac{N}{3} \right) \quad (3.25)$$

Onde:

$\Delta V_f'$: queda de tensão até o final da linha no ramo direito do modelo;

$\Delta V_f''$: queda de tensão até o final da linha no ramo esquerdo do modelo.

3.3.1.10. Modelo 7

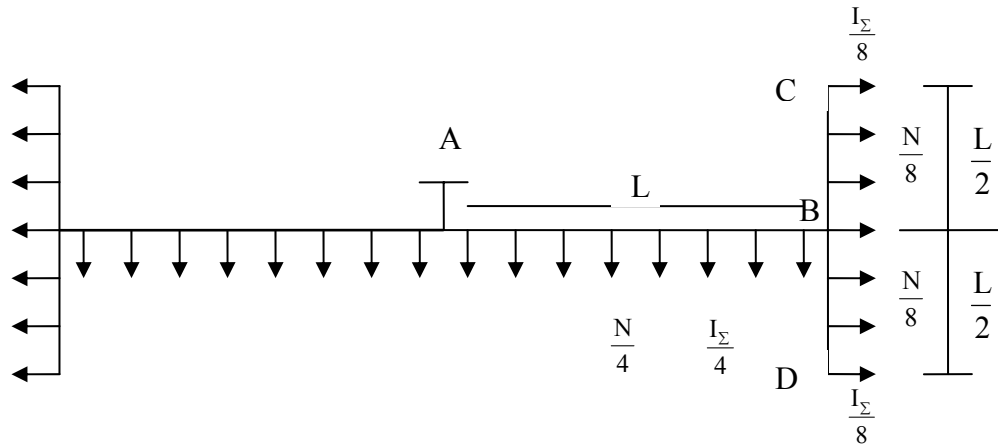


Figura 3.17 – Modelo 7 – Esquemático

Uma vez que este modelo mostra-se simétrico, irá considerar-se, para fins de cálculo de queda de tensão no final da linha, somente um dos seus lados. Este, por sua vez, é composto por um Elemento B e dois elementos A.

Portanto, para este modelo, tem-se o valor da queda de tensão até o final da linha mostrado na equação 3.29.

$$\Delta V_{BC} = \Delta V_{BD} = \frac{1}{2} \cdot \frac{I_{\Sigma}}{8} \cdot \frac{L}{2} \cdot R_o \cdot K_{V1} \left(\frac{N}{8} \right) = \frac{I_{\Sigma} L R_o}{32} \cdot K_{V1} \left(\frac{N}{8} \right) \quad (3.26)$$

$$\Delta V_{AB} = \frac{1}{2} \cdot \frac{I_{\Sigma}}{4} \cdot L \cdot R_o + \frac{I_{\Sigma}}{4} \cdot L \cdot R_o = \frac{3 I_{\Sigma} L R_o}{8} \cdot K_{V2} \left(1, \frac{N}{4} \right) \quad (3.27)$$

$$\Delta V_f = \frac{3 I_{\Sigma} L R_o}{8} \cdot K_{V2} \left(1, \frac{N}{4} \right) + \frac{I_{\Sigma} L R_o}{32} \cdot K_{V1} \left(\frac{N}{8} \right) \quad (3.28)$$

$$\frac{I_{\Sigma} L R_o}{32} \left[12 K_{V2} \left(1, \frac{N}{4} \right) + K_{V1} \left(\frac{N}{8} \right) \right] \quad (3.29)$$

3.4 Determinação da Queda de Tensão até o Centro de Carga

É bastante importante a determinação da queda de tensão até o centro de carga, uma vez que os cálculos, referentes à energia fornecida fora dos níveis adequados de tensão, são realizados tomando-se como referência o centro de carga dos modelos representativos de redes de baixa tensão.

A construção dos cálculos, referentes às quedas de tensão até os centros de carga, assim como apresentado na definição da queda de tensão até o final da linha, é realizada através da representação dos Elementos A e B. A partir deles, chega-se aos valores finais para cada um dos modelos.

Define-se, de forma generalista, a queda de tensão até o centro de carga através da equação 3.30.

$$\Delta V_c = \frac{\sum \Delta V_i P_i}{\sum P_i} \quad (3.30)$$

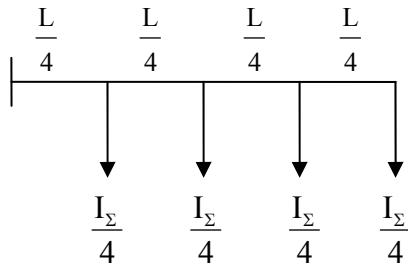
Da mesma forma como foi definida a queda de tensão final, a queda de tensão até o centro de carga para os modelos simplificados será definida com o auxílio dos Elementos essenciais A e B.

Abaixo segue uma lista das principais variáveis utilizadas nesta seção:

- ΔV_c : queda de tensão até o centro de carga;
- L: comprimento do condutor;
- R_o : resistência do condutor;
- I_Σ : somatório da corrente de todos os consumidores;
- N: número de consumidores

3.4.1. Elemento A

Primeiramente será demonstrado o cálculo da queda de tensão até o centro de carga para o Elemento A num exemplo prático aplicando-se a equação 3.30.



$$\begin{aligned}
 \Delta V_c &= \frac{\frac{4}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o \cdot \frac{I_\Sigma}{4}}{I_\Sigma} + \frac{\left(\frac{4}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o + \frac{3}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o \right) \frac{I_\Sigma}{4}}{I_\Sigma} + \\
 &\quad \frac{\left(\frac{4}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o + \frac{3}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o + \frac{2}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o \right) \frac{I_\Sigma}{4}}{I_\Sigma} + \\
 &\quad \frac{\left(\frac{4}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o + \frac{3}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o + \frac{2}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o + \frac{1}{4} I_\Sigma \cdot \frac{L}{4} \cdot R_o \right) \frac{I_\Sigma}{4}}{I_\Sigma} = \\
 &= \frac{I_\Sigma L R_o}{N^3} (4.4 + 3.3 + 2.2 + 1.1)
 \end{aligned}$$

Então para um caso geral tem-se:

$$\begin{aligned}
 \Delta V_c &= \frac{I_\Sigma L R_o}{N^3} [N \cdot N + (N-1)(N-1) + \dots + 1 \cdot 1] = \\
 &= \frac{I_\Sigma L R_o}{N^3} [N^2 + (N-1)^2 + \dots + 1^2] = \\
 &= \frac{I_\Sigma L R_o}{N^3} \cdot \frac{N(N+1)(2N+1)}{6} = \\
 &= \frac{I_\Sigma L R_o}{6N^2} (N+1)(2N+1)
 \end{aligned} \tag{3.31}$$

3.4.2. Elemento B

Novamente, será demonstrado o cálculo de um exemplo utilizando equação 3.30, desta vez calculando o Centro de Carga para o Elemento B.

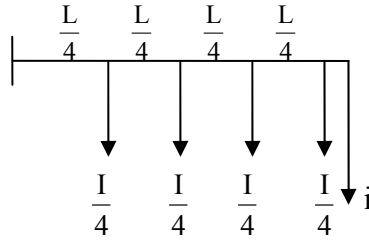


Figura 3.18 – Representação do Elemento B

$$\begin{aligned}
 \Delta V_c &= \frac{\left(\frac{4}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o \cdot \frac{I}{4}}{(I+i)} + \\
 &\frac{\left[\left(\frac{4}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o + \left(\frac{3}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o\right] \frac{I}{4}}{(I+i)} + \\
 &\frac{\left[\left(\frac{4}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o + \left(\frac{3}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o + \left(\frac{2}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o\right] \frac{I}{4}}{(I+i)} + \\
 &\frac{\left[\left(\frac{4}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o + \left(\frac{3}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o + \left(\frac{2}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o + \left(\frac{1}{4}I+i\right) \cdot \frac{L}{4} \cdot R_o\right] \left(\frac{I}{4}+i\right)}{(I+i)} = \\
 &= \frac{I^2 L R_o}{N^3} \cdot \frac{(4^2 + 3^2 + 2^2 + 1^2)}{(I+i)} + \frac{4.5 \cdot N}{N^3} \cdot \frac{I i L R_o}{(I+i)} + \frac{N i^2 L R_o \cdot N^2}{N^3 (I+i)} = \\
 &= \frac{I^2 L R_o}{6 N^2 (I+i)} (N+1)(2N+1) + \frac{(N+1)}{N} \cdot \frac{I i L R_o}{(I+i)} + \frac{i^2 L R_o}{(I+i)} \quad (3.32)
 \end{aligned}$$

3.4.3 Modelos Compostos

Agora pode-se definir a queda de tensão até o centro de carga para os modelos compostos pelos Elementos A e B.

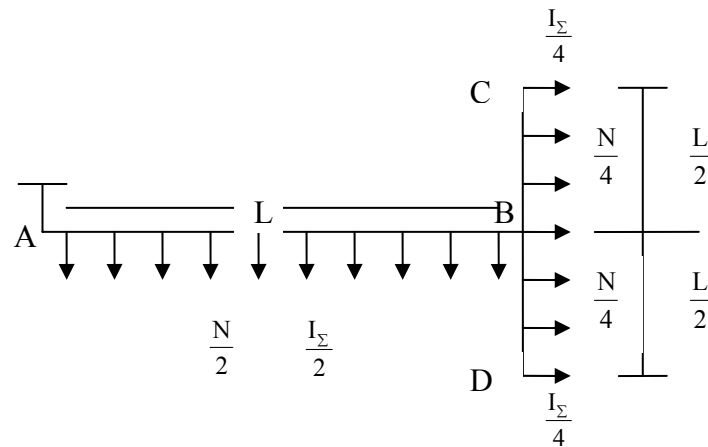


Figura 3.19 – Exemplo de um modelo composto

O Modelo anterior pode ser representado pelo conjunto dos elementos básicos A e B.

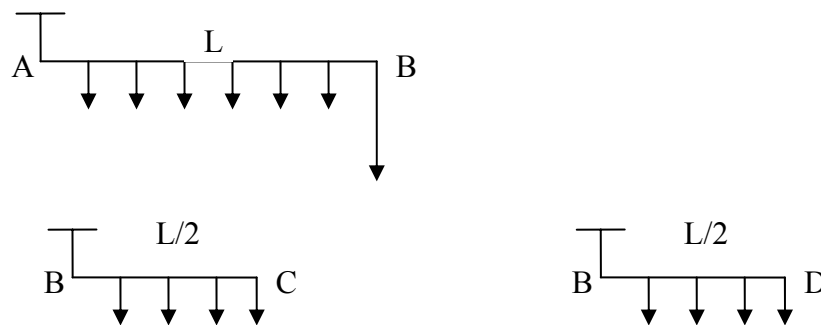


Figura 3.20 – Partes de um modelo composto

Então, a queda de Tensão até o Centro de Carga deste modelo pode ser definido pela média ponderada dos elementos que o compõem.

$$\Delta V_c = \frac{\Delta V_c' \cdot I_\Sigma + \Delta V_c'' \cdot I'' + \Delta V_c''' \cdot I'''}{I_\Sigma}$$

Onde:

$\Delta V_c'$: queda de tensão até o centro de carga do trecho AB;

$\Delta V_c''$: queda de tensão até o centro de carga do trecho BC;

I'' : corrente total no trecho BC;

$\Delta V_c''$: queda de tensão até o centro de carga no trecho BD;

I'' : corrente total no trecho BD.

3.4.3.1. Modelo 1

Para o modelo 1, representativo de uma rede de baixa tensão, tem-se o valor, da queda de tensão até o seu centro de carga, dado pela equação 3.33.

$$\Delta V_c = 0 \quad (3.33)$$

3.4.3.2. Modelo 2

Para o modelo 2, o valor da queda de tensão até o centro de carga é fornecido pela equação 3.34.

$$\Delta V_c = \Delta V_f = I_\Sigma L R_o \quad (3.34)$$

3.4.3.3. Modelo 3

O valor da queda de tensão, para este modelo, é obtido com a equação 3.35.

$$\Delta V_c = \frac{I_\Sigma L R_o}{6N^2} (N+1)(2N+1) = I_\Sigma L R_o \left(\frac{2N^2 + 3N + 1}{6N^2} \right) \quad (3.35)$$

3.4.3.4. Modelo 4

Neste modelo, o valor da queda de tensão até o seu centro de carga é dado pela equação 3.36.

$$\Delta V_c = I_\Sigma L R_o \left(\frac{2N^2 + 3N + 1}{12N^2} \right) \quad (3.36)$$

3.4.3.5. Modelo 5A

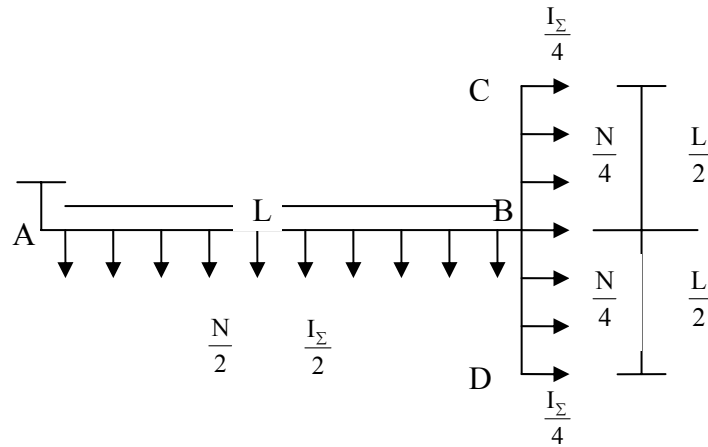


Figura 3.21 – Exemplo de Modelo 5A

Sendo $AB = \Delta V_c'$ e $BC = \Delta V_c''$

$$\Delta V_c' = \frac{\frac{I_\Sigma^2}{2} \cdot LR_o}{6 \frac{N^2}{4} \left(\frac{I_\Sigma}{2} + \frac{I_\Sigma}{2} \right)} \left(\frac{N}{2} + 1 \right) (N+1) + \frac{\left(\frac{N}{2} + 1 \right)}{\frac{N}{2}} \cdot \frac{\frac{I_\Sigma}{2} \cdot \frac{I_\Sigma}{2} \cdot LR_o}{\frac{I_\Sigma}{2} + \frac{I_\Sigma}{2}} +$$

$$\frac{\frac{I_\Sigma^2}{4} \cdot LR_o}{\frac{I_\Sigma}{2} + \frac{I_\Sigma}{2}} = \dots = I_\Sigma LR_o \left(\frac{7N^2 + 9N + 2}{12N^2} \right)$$

$$\Delta V_c'' = \frac{\frac{I_\Sigma}{4} \cdot \frac{L}{2} \cdot R_o}{6 \cdot \frac{N^2}{16}} \cdot \left(\frac{N}{4} + 1 \right) \left(\frac{N}{2} + 1 \right) = I_\Sigma LR_o \left(\frac{N^2 + 6N + 8}{24N^2} \right)$$

O valor da queda de tensão, para este modelo, pode ser obtido com a equação 3.37.

$$\Delta V_c = \frac{I_\Sigma^2 LR_o \left(\frac{7N^2 + 9N + 2}{12N^2} \right)}{I_\Sigma} + 2 \frac{I_\Sigma^2 LR_o \left(\frac{N^2 + 6N + 8}{4 \cdot 24N^2} \right)}{I_\Sigma}$$

$$I_{\Sigma}LR_o \left(\frac{29N^2 + 42N + 16}{48N^2} \right) \quad (3.37)$$

3.4.3.6. Modelo 5B

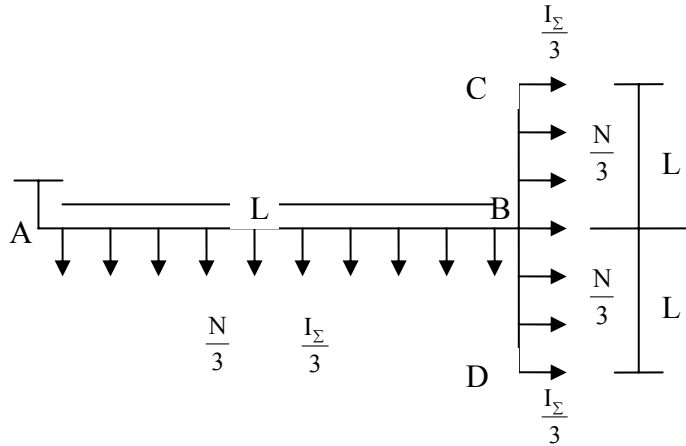


Figura 3.21 – Exemplo de Modelo 5B

Sendo $AB = \Delta V_c'$ e $BC = \Delta V_c''$

$$\Delta V_c' = \frac{\frac{I_{\Sigma}^2}{9} \cdot LR_o}{6 \cdot \frac{N^2}{9} \cdot I_{\Sigma}} \left(\frac{N}{3} + 1 \right) \left(\frac{2N}{3} + 1 \right) + \left(\frac{\frac{N}{3} + 1}{\frac{N}{3}} \right) \cdot \frac{\frac{I_{\Sigma}}{3} \cdot \frac{2I_{\Sigma}}{3} \cdot LR_o}{I_{\Sigma}} +$$

$$\frac{\frac{4}{9} I_{\Sigma}^2 \cdot LR_o}{I_{\Sigma}} = \dots = I_{\Sigma} LR_o \frac{38N^2 + 45N + 9}{54N^2}$$

$$\Delta V_c'' = \frac{\frac{I_{\Sigma}}{3} \cdot LR_o}{6 \cdot \frac{N^2}{9}} \left(\frac{N}{3} + 1 \right) \left(\frac{2N}{3} + 1 \right) = I_{\Sigma} LR_o \left(\frac{2N^2 + 9N + 9}{18N^2} \right)$$

Para este modelo, tem-se o valor da queda de tensão até o centro de carga dado pela equação 3.38.

$$\Delta V_c = \frac{\Delta V_c' \cdot I_\Sigma + 2 \cdot \Delta V_c'' \cdot \frac{1}{3} \cdot I_\Sigma}{I_\Sigma} = I_\Sigma LR_o \left[\frac{(38N^2 + 45N + 9)}{54N^2} + \frac{2}{3} \cdot \frac{(2N^2 + 9N + 9)}{18N^2} \right] =$$

$$= I_\Sigma LR_o \frac{14N^2 + 21N + 9}{18N^2} \quad (3.38)$$

3.4.3.7. Modelo 6

Tem-se o valor da queda de tensão até o centro de carga fornecido pela equação 3.39.

$$\Delta V_c = \frac{\Delta V_c' \cdot \frac{2}{3} I_\Sigma + 2 \Delta V_c'' \cdot \frac{1}{6} I_\Sigma + \Delta V_c''' \cdot \frac{1}{3} I_\Sigma}{I_\Sigma} =$$

$$= I_\Sigma \left[\left(\frac{28N^2 + 54N + 18}{108N^2} + \frac{N^2 + 9N + 18}{108N^2} \right) LR_o' + \frac{4N^2 + 18N + 18}{108N^2} LR_o'' \right] =$$

$$I_\Sigma \left[LR_o' \cdot \frac{29N^2 + 63N + 36}{108N^2} + LR_o'' \cdot \frac{4N^2 + 18N + 18}{108N^2} \right] \quad (3.39)$$

3.4.3.8. Modelo 7

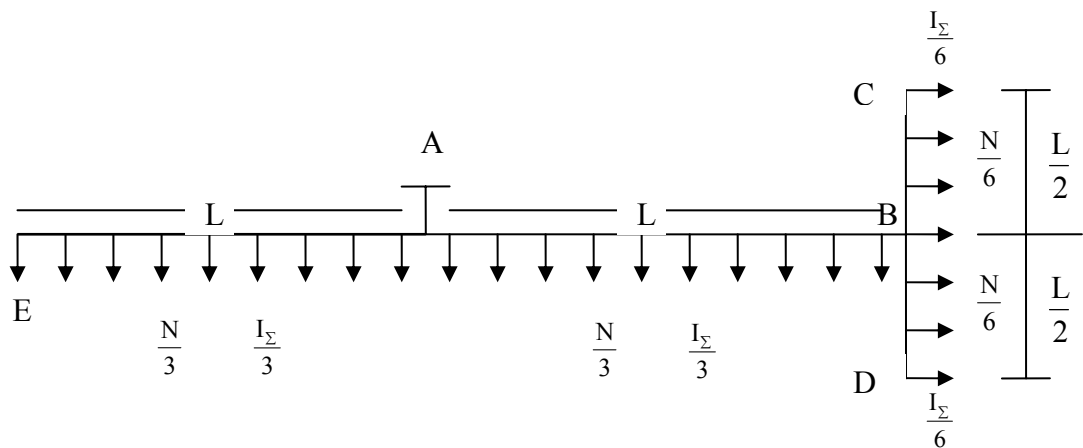


Figura 3.22 – Exemplo de Modelo 7

Onde:

$\Delta V_c'$: queda de tensão até o centro de carga no Trecho AB;

$\Delta V_c''$: queda de tensão até o centro de carga no Trecho BC;

$\Delta V_c'''$: queda de tensão até o centro de carga no Trecho AE.

ΔV_c^d : queda de tensão até o centro de carga no ramo direito;

ΔV_c^e : queda de tensão até o centro de carga no ramo esquerdo;

$$\Delta V_c' = \frac{\frac{I_\Sigma^2}{9} \cdot LR_o'}{6 \cdot \frac{N^2}{9} \cdot \frac{2}{3} \cdot I_\Sigma} \cdot \left(\frac{N}{3} + 1 \right) \left(\frac{2N}{3} + 1 \right) + \frac{\frac{N}{3} + 1}{\frac{N}{3}} \cdot \frac{\frac{I_\Sigma}{3} \cdot \frac{I_\Sigma}{3} \cdot LR_o'}{\frac{2}{3} I_\Sigma} + \frac{\frac{1}{9} I_\Sigma^2 \cdot LR_o'}{\frac{2}{3} I_\Sigma} =$$

$$= I_\Sigma LR_o' \left(\frac{14N^2 + 27N + 9}{36N^2} \right)$$

$$\Delta V_c'' = \frac{\frac{I_\Sigma}{6} \cdot \frac{L}{2} \cdot R_o'}{6 \cdot \frac{N^2}{36}} \left(\frac{N}{6} + 1 \right) \left(\frac{2N}{6} + 1 \right) = I_\Sigma LR_o' \left(\frac{N^2 + 9N + 18}{36N^2} \right)$$

$$\Delta V_c''' = \frac{\frac{I_\Sigma}{3} \cdot LR_o''}{6 \cdot \frac{N^2}{9}} \left(\frac{N}{3} + 1 \right) \left(\frac{2N}{3} + 1 \right) = I_\Sigma LR_o'' \left(\frac{2N^2 + 9N + 9}{18N^2} \right)$$

Para este último modelo representativo, o valor da queda de tensão até o centro de carga é dado pela equação 3.40.

$$\Delta V_c^d = \frac{\Delta V_c' \cdot \frac{I_\Sigma}{2} + 2 \cdot \Delta V_c'' \cdot \frac{I_\Sigma}{8}}{\frac{I_\Sigma}{2}} =$$

$$= \frac{I_\Sigma^2 LR_o' \left[\left(\frac{14N^2 + 36N + 16}{96N^2} \right) + \left(\frac{2N^2 + 24N + 64}{192 \cdot 2N^2} \right) \right]}{\frac{I_\Sigma}{2}} =$$

$$= 2I_\Sigma LR_o' \left(\frac{58N^2 + 168N + 128}{384N^2} \right)$$

$$\Delta V_c^d = \Delta V_c^e$$

$$\Delta V_c = \frac{\Delta V_c^d \cdot \frac{I_\Sigma}{2} + \Delta V_c^e \cdot \frac{I_\Sigma}{2}}{I_\Sigma} = \Delta V_c^d = \Delta V_c^e \quad (3.40)$$

3.5. Ausência e Disponibilidade de Parâmetros

Neste trabalho, um dos principais objetivos é a proposta de uma metodologia para enfrentar o problema da ausência de informações a respeito das redes de baixa tensão.

Foram consideradas disponíveis, no estudo, as informações acerca da corrente total distribuída pela rede de baixa tensão (I_Σ), do número total de consumidores conectados à rede (N) e da queda de tensão máxima permissível na rede, definida durante o projeto (ΔV_{perm}).

Sendo ΔV_{fmax} a máxima queda de tensão até o final da linha, considera-se, neste caso, que $\Delta V_{fmax} = \Delta V_{perm}$ para cada transformador de distribuição em regime de demanda máxima.

Para a definição dos valores referentes ao comprimento do condutor L, multiplicado pela sua resistência R_o (LR_o), considera-se um regime de demanda máxima; onde I_Σ é igual a corrente máxima total ($I_{\Sigma max}$) e $\Delta V_f = \Delta V_{fmax} = \Delta V_{perm}$ (sendo que ΔV_f representa a queda de tensão até o final da linha).

Uma vez feitas estas considerações, são encontrados os valores de LR_o para todos os modelos desenvolvidos, utilizando-se as equações obtidas para a queda de tensão até o final da linha.

- Modelo 1

$$\Delta V_f = 0 \quad (3.41)$$

- Modelo 2

$$LR_o = \frac{\Delta V_{perm}}{I_{\Sigma max}} \quad (3.42)$$

- Modelo 3

$$LR_o = \frac{2\Delta V_{perm}}{I_{\Sigma max} \cdot K_{V_1} \left(N \right)} \quad (3.43)$$

- Modelo 4

$$LR_o = \frac{4 \cdot \Delta V_{perm}}{I_{\Sigma max} \cdot K_{V_1} \left(\frac{N}{2} \right)} \quad (3.44)$$

- Modelo 5A

$$LR_o = \frac{16 \cdot \Delta V_{perm}}{I_{\Sigma max} \left[12K_{V_2} \left(1, \frac{N}{2} \right) + K_{V_1} \left(\frac{N}{4} \right) \right]} \quad (3.45)$$

- Modelo 5B

$$LR_o = \frac{16 \cdot \Delta V_{perm}}{I_{\Sigma max} \left[5K_{V_2} \left(2, \frac{N}{3} \right) + K_{V_1} \left(\frac{N}{3} \right) \right]} \quad (3.46)$$

- Modelo 6

$$LR_o' = \frac{24 \cdot \Delta V_{perm}}{I_{\Sigma max} \left[12K_{V_2} \left(1, \frac{N}{3} \right) + K_{V_1} \left(\frac{N}{3} \right) \right]} \quad (3.47)$$

$$LR_o'' = \frac{6 \cdot \Delta V_{perm}}{I_{\Sigma max} \cdot K_{V_1} \left(\frac{N}{3} \right)} \quad (3.48)$$

- Modelo 7

$$LR_o = \frac{32 \cdot \Delta V_{perm}}{I_{\Sigma max} \left[12K_{V_2} \left(1, \frac{N}{4} \right) + K_{V_1} \left(\frac{N}{8} \right) \right]} \quad (3.49)$$

Para todos os casos considera-se que:

$$\frac{\Delta V_t}{\Delta V_{\max}} = \frac{S_t}{S_{\max}} = \frac{I_t}{I_{\max}}$$

Por consequência,

$$\begin{aligned} \frac{\Delta V_{f_t}}{\Delta V_{f \max}} &= \frac{S_t}{S_{\max}} \\ \frac{\Delta V_{c_t}}{\Delta V_{c \max}} &= \frac{S_t}{S_{\max}} \end{aligned} \quad (3.50)$$

3.6 Queda de Tensão até o centro de carga desconhecendo-se os valores de LR_o

Como em alguns casos não existem informações completas sobre a rede secundária, agora serão redefinidas as fórmulas de cálculo de queda de tensão até o centro de carga, onde não estão disponíveis comprimento e características elétricas dos condutores utilizados.

A seguir serão redefinidas as fórmulas da seção 3.4.3 para cada um dos Modelos Simplificados de redes de baixa tensão substituindo-se os valores de LR_o pelo que foi definido na seção anterior.

Modelo 1

$$\Delta V_c = 0 \quad (3.51)$$

Modelo 2

$$\begin{aligned} \Delta V_c &= \Delta V_f = I_{\Sigma} LR_o \\ \Delta V_c &= \Delta V_f = I_{\Sigma} \frac{\Delta V_{\text{perm}}}{I_{\Sigma \max}} \end{aligned} \quad (3.52)$$

Modelo 3

$$\Delta V_c = I_\Sigma LR_o \left(\frac{2N^2 + 3N + 1}{6N^2} \right)$$

$$\Delta V_c = I_\Sigma \frac{2\Delta V_{perm}}{I_{\Sigma max} \cdot K_{V_1}(N)} \left(\frac{2N^2 + 3N + 1}{6N^2} \right) \quad (3.53)$$

Modelo 4

$$\Delta V_c = I_\Sigma LR_o \left(\frac{2N^2 + 3N + 1}{12N^2} \right)$$

$$\Delta V_c = I_\Sigma \frac{4 \cdot \Delta V_{perm}}{I_{\Sigma max} \cdot K_{V_1}\left(\frac{N}{2}\right)} \left(\frac{2N^2 + 3N + 1}{12N^2} \right) \quad (3.54)$$

Modelo 5A

$$\Delta V_c = I_\Sigma LR_o \left(\frac{29N^2 + 42N + 16}{48N^2} \right)$$

$$\Delta V_c = I_\Sigma \frac{16 \cdot \Delta V_{perm}}{I_{\Sigma max} \left[12K_{V_2}\left(1, \frac{N}{2}\right) + K_{V_1}\left(\frac{N}{4}\right) \right]} \left(\frac{29N^2 + 42N + 16}{48N^2} \right) \quad (3.55)$$

Modelo 5B

$$\Delta V_c = I_\Sigma LR_o \frac{14N^2 + 21N + 9}{18N^2}$$

$$\Delta V_c = I_\Sigma \frac{16 \cdot \Delta V_{perm}}{I_{\Sigma max} \left[5K_{V_2}\left(2, \frac{N}{3}\right) + K_{V_1}\left(\frac{N}{3}\right) \right]} \frac{14N^2 + 21N + 9}{18N^2} \quad (3.56)$$

Modelo 6

$$\Delta V_c = I_\Sigma \left[LR_o \cdot \frac{29N^2 + 63N + 36}{108N^2} + LR_o \cdot \frac{4N^2 + 18N + 18}{108N^2} \right]$$

$$\Delta V_c = I_\Sigma \left[\frac{24 \Delta V_{perm}}{I_{\Sigma max} \left[12K_{v_2} \left(1, \frac{N}{3} \right) + K_{v_1} \left(\frac{N}{3} \right) \right]} \cdot \frac{29N^2 + 63N + 36}{108N^2} + \frac{6 \Delta V_{perm}}{I_{\Sigma max} \cdot K_{v_1} \left(\frac{N}{3} \right)} \cdot \frac{4N^2 + 18N + 18}{108N^2} \right] \quad (3.57)$$

Modelo 7

$$\Delta V_c = 2I_\Sigma LR_o \left(\frac{58N^2 + 168N + 128}{384N^2} \right)$$

$$\Delta V_c = 2I_\Sigma \frac{32 \Delta V_{perm}}{I_{\Sigma max} \left[12K_{v_2} \left(1, \frac{N}{4} \right) + K_{v_1} \left(\frac{N}{8} \right) \right]} \cdot \left(\frac{58N^2 + 168N + 128}{384N^2} \right) \quad (3.58)$$

3.7 Revisão do Capítulo

Neste capítulo foram apresentados os Modelos de Rede de Baixa Tensão que possibilitam a estimação de estado, mesmo quando os dados disponíveis sobre elas são pouco confiáveis ou inexistentes. Foram feitas definições do que não está disponível, e também para possibilitar as estimações foram realizadas suposições, como por exemplo, da distribuição da carga dos consumidores ao longo da rede e entre as fases.

Foram propostas as Equações para cálculo de queda de tensão até o final da rede secundária, bem como a queda de tensão até o centro de carga. Posteriormente supondo que os valores de L e R_o, comprimento e resistência dos condutores respectivamente, não existam, estes foram estimados para cada modelo. Posteriormente os valores de LR_o foram aplicados às equações anteriormente propostas.

CAPÍTULO IV

4. REDES REAIS DE BAIXA TENSÃO

4.1. Conceitos e Características

Entende-se por rede de baixa tensão as instalações que incluem o transformador de distribuição e o circuito que este alimenta, bem como os equipamentos de proteção e conexão associados e as estruturas que o sustentam.

Uma das características destas redes é o grande número de circuitos envolvidos. Um alimentador primário urbano atende, em média, 100 transformadores de distribuição (podendo chegar a abastecer 200 transformadores).

A área de influência de cada rede é também de dimensões reduzidas, o que prejudica a previsão individualizada de cada um destes circuitos.

Há ainda outras características associadas às redes de baixa tensão, tais como a grande diversidade de topologias observadas e o seu comportamento bastante dinâmico, visto que ela é o ponto de acesso da grande maioria dos consumidores à energia, e todos os dias muitos consumidores são ligados ou desligados destas redes.

4.2. Aplicação dos Modelos Representativos

Os modelos representativos das redes de baixa tensão que foram definidos no Capítulo anterior terão sua aplicação demonstrada em situações práticas, em alguns circuitos secundários de um alimentador real.

Para cada rede de baixa tensão será apropriado um modelo representativo que mais lhe cabe.

4.3. Referências quanto à Localização

As topologias de redes de baixa tensão, mostradas no decorrer deste capítulo, referem-se ao alimentador número oito, da Subestação Pal9 da Companhia Estadual de Distribuição de Energia Elétrica, localizado na cidade de Porto Alegre no Estado do Rio Grande do Sul. Este alimentador possui uma tensão nominal de 13.8kV na sua rede primária e está situado numa região central da cidade de Porto Alegre.

4.4. Topologias Reais

Foram localizadas, pesquisadas e estudadas todas as redes de baixa tensão referentes ao AL 8 da SE PAL9 da Companhia Estadual de Distribuição de Energia Elétrica (CEEE-D).

As Figuras 14, 15 e 16 a seguir, mostram três exemplos de partes da topologia real destas redes primárias e secundárias do AL8 da PAL9, devido à grande dimensão do alimentador estudado.

Estas figuras foram criadas automaticamente até os transformadores de distribuição visto que existe um cadastro atualizado da rede primária, já a parte secundária foi gerada manualmente baseada em plantas baixa dos circuitos que compõem este alimentador. Estas plantas foram visualizadas em arquivos de programa do tipo CAD.

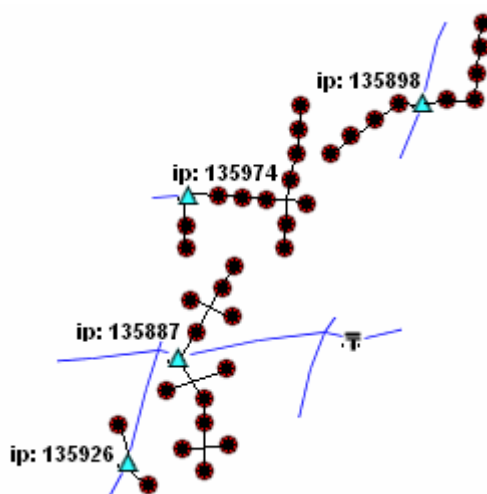


Figura 4.1 – Pal9-AL8 (Parte 1).

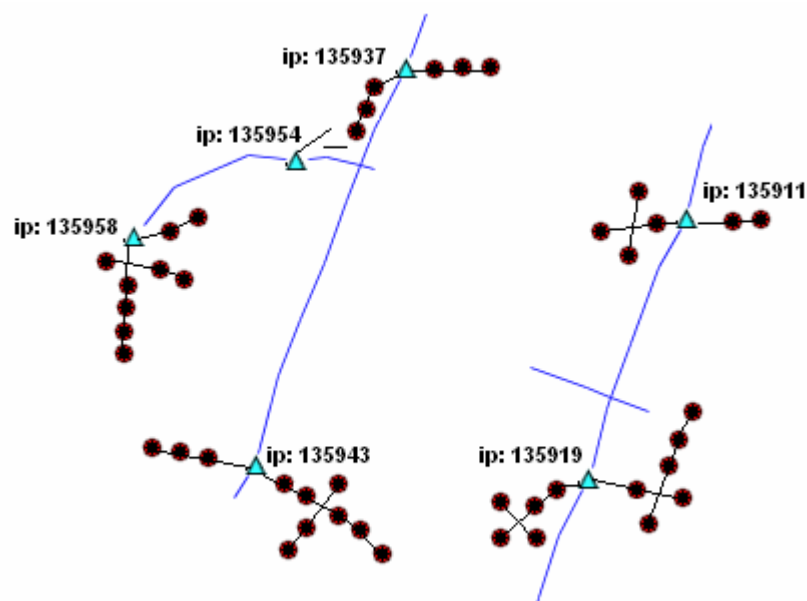


Figura 4.2 – Pal9-AL8 (Parte 2).

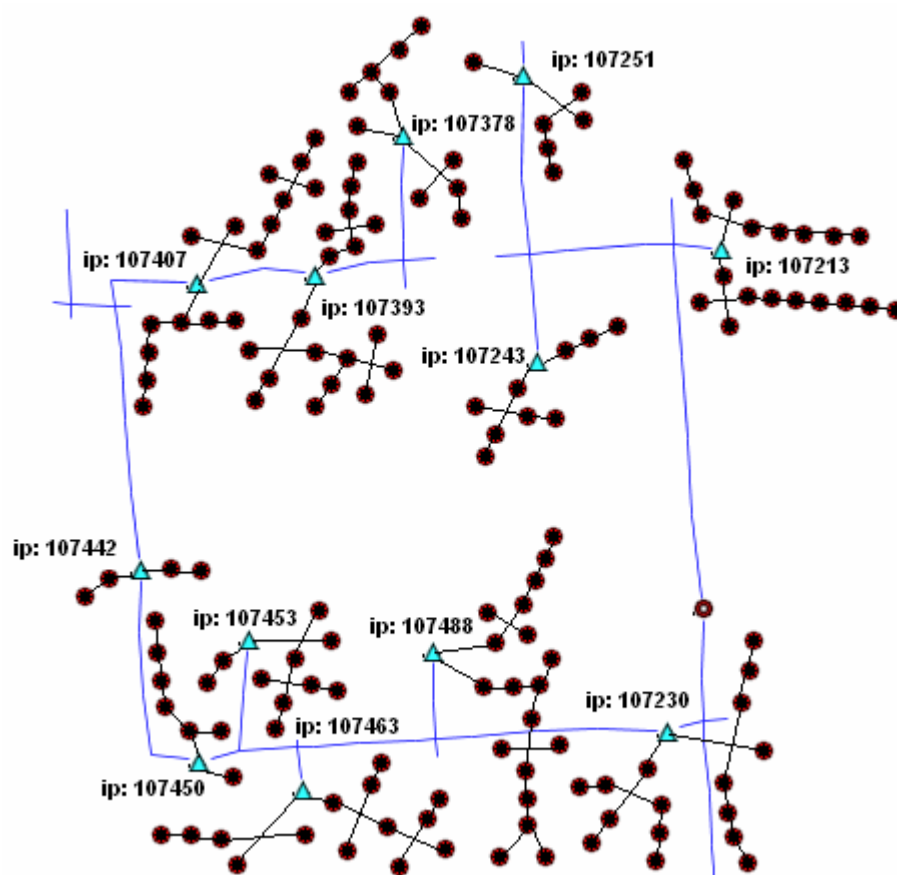


Figura 4.3 – Pal9-AL8 (Parte 3).

Os códigos contidos nas figuras 14, 15 e 16, referem-se aos identificadores de cadastro dos respectivos transformadores de distribuição.

4.5. Apropriação dos Modelos Representativos

Para a aplicação prática dos modelos representativos em redes reais de distribuição resta agora apenas a apropriação destes modelos para as topologias encontradas nos circuitos estudados na seção anterior. Esta apropriação é feita com base nas características do circuito estudado e na sua maior semelhança com algum dos modelos representativos. Este trabalho necessita de intervenção humana, visto que é feito manualmente.

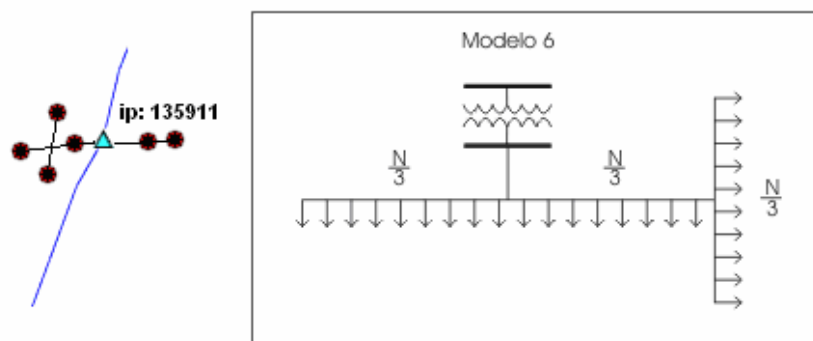


Figura 4.4 – Apropriação de um Modelo Representativo

A Figura 4.4 exemplifica como é o trabalho de apropriação de Modelos Representativos. O circuito 135911 do alimentador 8 da SE PAL9 possui 6 consumidores que estão divididos em 3 setores de 2 consumidores, o que torna esta topografia muito próxima do Modelo Representativo 6. Mesmo estando o modelo verticalmente invertido em relação à topografia, esta inversão não possui nenhum efeito para fins de cálculo.

CAPÍTULO V

5. ENERGIA FORNECIDA FORA DO PADRÃO

5.1. Fundamentação e Significado

Para uma quantificação e maximização do desempenho dos algoritmos desenvolvidos para a correção dos níveis de tensão entregues aos consumidores, verificou-se necessária a criação de um indicador, denominado Energia Fornecida Fora do Padrão (EFPF).

Para obtê-lo, utilizam-se os parâmetros de níveis de tensão determinados pelo órgão regulador (ANEEL). Embora a ANEEL utilize dois níveis de tensão para determinar a transgressão, denominados nível precário e nível crítico, podemos calcular a EFPF tanto para o nível precário como para o nível crítico.

O indicador EFPF difere bastante dos indicadores DRP e DRC, Duração Relativa da Transgressão de Tensão Precária e Duração Relativa da Transgressão de Tensão Crítica, respectivamente, visto que estes são indicadores individuais, ou seja, consideram apenas uma unidade consumidora. A utilização destes indicadores como base para a medição do desempenho dos algoritmos não seria apropriada, visto que os algoritmos são aplicados de forma global, em toda uma subestação, e a avaliação pontual não é o ideal neste caso.

O indicador ICC, Índice de Unidades Consumidoras com Tensão Crítica, é um indicador global, entretanto pode também não ser o mais apropriado, visto que neles todos os consumidores são iguais e tem o mesmo peso independente da demanda de cada um. Este indicador não é apropriado visto que as companhias de distribuição visam também a redução dos valores gastos com indenizações, sendo que estes valores são proporcionais à energia que foi fornecida em nível precário ou crítico para o consumidor em questão. Por estes motivos o índice EFPF, que leva

em consideração a quantidade de energia diferentemente dos índices da ANEEL, foi o índice mais apropriado aos interesses das concessionárias, bem como deste estudo.

O cálculo do indicador é realizado a partir da estimação da queda de tensão em cada um dos trechos (unidades consumidoras) dos modelos propostos para as redes de baixa tensão. Caso ocorra violação nos níveis de tensão, a energia fornecida neste local é calculada e acumulada no índice.

Para o cálculo deste indicador foram utilizadas duas metodologias diferentes. Primeiramente foi utilizada uma metodologia clássica, a qual tenta obter fórmulas algébricas, diferentes para cada modelo representativo, para a obtenção direta do valor de EFP por circuito de baixa tensão. Devido à complexidade do problema, nos modelos compostos (que possuem Elemento A e Elemento B) e em alguns casos não foi possível obter-se a solução. Esta metodologia é apresentada no Apêndice 1.

A abordagem alternativa que foi utilizada para a implementação efetiva do trabalho é apresentada no item a seguir. Esta abordagem utiliza um método iterativo para calcular o nível de tensão em cada consumidor. Caso o nível de tensão transgrida os níveis estipulados o consumidor em questão está com o nível de tensão fora do padrão.

5.2. Cálculo de EFP através algoritmo iterativo

Na seção 3.5 foram definidos os valores de resistência para os condutores (LR_o) para todos os modelos simplificados. Isto permite calcular o valor de Energia consumida com níveis de tensão fora do padrão da forma a ser apresentada a seguir. Os algoritmos propostos podem analisar automaticamente todas as situações, no que diz respeito aos pontos onde são transgredidos os níveis de tensão.

Todos os algoritmos seguem a mesma idéia. Percorrer toda a rede secundária consumidor a consumidor, a partir do primeiro calculando as quedas de tensão afim de encontrar os pontos críticos onde a tensão é violada por subtensão ou por sobretensão.

Encontrados os pontos críticos, que pode ser vistos na Figura 6.1 representados por k^+ e k^- . Caso eles estejam dentro dos limites da rede é feito o cálculo da quantidade de energia que é consumida entre estes limites.

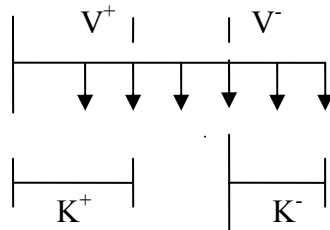


Figura 5.1: Pontos Críticos de Transgressão de Tensão

Para o entendimento dos processos referentes à obtenção da Energia Fornecida Fora de Padrão são necessárias algumas definições a respeito das variáveis envolvidas.

Abaixo, encontram-se listadas as principais variáveis utilizadas ao longo deste capítulo:

- LR_o : comprimento da rede secundária multiplicado pela resistência do condutor;
- L_oR_o : comprimento de um trecho unitário da rede secundária multiplicado pela resistência do condutor;
- V^- : nível de tensão mínima permitida pela legislação vigente;
- V^+ : nível de tensão máximo permitido pela legislação vigente;
- W^- : quantidade de energia consumida com subtensão;
- W^+ : quantidade de energia consumida com sobretensão;
- V_{BT} : nível de tensão no início da rede de baixa tensão;
- ΔV_f : queda de tensão até o final da rede de baixa tensão;
- V_f : nível de tensão medido no final da rede de baixa tensão;
- i : índice para o consumidor em que o algoritmo está realizando os cálculos;
- ΔV_i : queda de tensão até o consumidor i ;
- V_i : tensão no consumidor i ;
- N : número total de consumidores do circuito.

A seguir serão definidas as formas de cálculo de EFPF para cada um dos modelos a partir do Modelo Representativo 3. Os Modelos 1 e 2 foram excluídos da metodologia por possuírem apenas um consumidor, nestes casos a EFPF sempre será de 0 ou 100%.

A EFPF é calculada usando estimativas de carga que são feitas baseadas em curvas de carga típicas por consumidor. Por este motivo cada um dos algoritmos possui uma iteração de 0 a 23 para cada uma das horas do dia onde a carga é estimada.

5.2.1. Modelo 3

Tem-se para o Modelo 3 o seguinte valor de L_oR_o :

$$L_oR_o = \frac{LR_o}{N}$$

Para cada período de tempo “t”. Variando de 0 a 23 calcula-se a EFPF, conforme a seqüência de programação apresentada a seguir.

$$i = 0, \quad V_o = V_{BT},$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[I_{\Sigma} - \frac{I_{\Sigma}}{N}(i-1) \right] L_oR_o$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W^+ = W^+ + \frac{I_{\Sigma}}{N} \cdot V_i$$

se $V_i < V^-$

$$W^- = W^- + \frac{I_{\Sigma}}{N} \cdot V_i$$

enquanto $i < N$

5.2.2. Modelo 4

Tem-se para o Modelo 4 o seguinte valor de $L_o R_o$:

$$L_o R_o = \frac{2LR_o}{N}$$

Para cada período de tempo “t”. Variando de 0 a 23 calcula-se a EFPF, conforme a seqüência de programação apresentada a seguir.

$$i = 0, \quad V_o = V_{BT},$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[\frac{I_\Sigma}{2} - \frac{I_\Sigma}{N} (i - 1) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{2I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{2I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{N}{2}$

5.2.3. Modelo 5A

Tem-se para o Modelo 5A o seguinte valor de $L_o R_o$:

$$L_o R_o = \frac{2LR_o}{N}$$

Para cada período de tempo “t”. Variando de 0 a 23 calcula-se a EFPF, conforme a seqüência de programação apresentada a seguir.

$$i = 0, \quad V_0 = V_{BT}$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[I_\Sigma - \frac{I_\Sigma}{N} (i-1) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{N}{2}$

faça

$$i = i + 1$$

$$\Delta V_i = \left[\frac{I_\Sigma}{4} - \frac{I_\Sigma}{N} \left(i - \frac{N}{2} - 1 \right) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{2I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{2I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{3N}{4}$

5.2.4. Modelo 5B

Tem-se para o Modelo 5B o seguinte valor de $L_o R_o$:

$$L_o R_o = \frac{3LR_o}{N}$$

Para cada período de tempo “t”. Variando de 0 a 23 calcula-se a EFPF, conforme a seqüência de programação apresentada a seguir.

$$i = 0, \quad V_o = V_{BT}$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[I_\Sigma - \frac{I_\Sigma}{N}(i-1) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

$$\text{se } V_i > V^+$$

$$W_i^+ = W_i^+ + \frac{I_\Sigma}{N} \cdot V_i$$

$$\text{se } V_i < V^-$$

$$W_i^- = W_i^- + \frac{I_\Sigma}{N} \cdot V_i$$

$$\text{enquanto } i < \frac{N}{3}$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[\frac{I_\Sigma}{3} - \frac{I_\Sigma}{N} \left(i - \frac{N}{3} - 1 \right) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

$$\text{se } V_i > V^+$$

$$W_i^+ = W_i^+ + \frac{2I_\Sigma}{N} \cdot V_i$$

$$\text{se } V_i < V^-$$

$$W_i^- = W_i^- + \frac{2I_\Sigma}{N} \cdot V_i$$

$$\text{enquanto } i < \frac{2N}{3}$$

5.2.5. Modelo 6

Tem-se para o Modelo 6 os seguintes valor de L_oR_o , para cada uma das ramificações:

$$L_oR_o' = \frac{2LR_o'}{N} \quad L_oR_o'' = \frac{LR_o''}{N}$$

Para cada período de tempo “t”. Variando de 0 a 23 calcula-se a EFPF, conforme a sequência de programação apresentada a seguir.

$$i = 0, \quad V_o = V_{BT}$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[I_\Sigma - \frac{I_\Sigma}{N}(i-1) \right] L_oR_o'$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{N}{3}$

faça

$$i = i + 1$$

$$\Delta V_i = \left[\frac{I_\Sigma}{6} - \frac{I_\Sigma}{N} \left(i - \frac{N}{6} - 1 \right) \right] L_oR_o'$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{2I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{2I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{N}{2}$

$$i = 0, \quad V_o = V_{BT}$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[I_\Sigma - \frac{I_\Sigma}{N}(i-1) \right] L_o R_o "$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{N}{3}$

5.2.6. Modelo 7

Tem-se para o Modelo 7 o seguinte valor de $L_o R_o$:

$$L_o R_o = \frac{2LR_o}{N}$$

Para cada período de tempo "t". Variando de 1 a 24 calcula-se a EFPF, conforme a sequência de programação apresentada a seguir.

$$i = 0, \quad V_o = V_{BT}$$

faça

$$i = i + 1$$

$$\Delta V_i = \left[\frac{I_\Sigma}{2} - \frac{I_\Sigma}{N} \left(i - \frac{N}{2} - 1 \right) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{2I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{2I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{N}{4}$

faça

$$i = i + 1$$

$$\Delta V_i = \left[\frac{I_\Sigma}{8} - \frac{I_\Sigma}{N} \left(i - \frac{3N}{4} - 1 \right) \right] L_o R_o$$

$$V_i = V_{i-1} - \Delta V_i$$

se $V_i > V^+$

$$W_i^+ = W_i^+ + \frac{4I_\Sigma}{N} \cdot V_i$$

se $V_i < V^-$

$$W_i^- = W_i^- + \frac{4I_\Sigma}{N} \cdot V_i$$

enquanto $i < \frac{3N}{8}$

5.3. Resumo do Capítulo

Neste capítulo foi apresentado o indicador utilizado para medir o desempenho da rede no que diz respeito às transgressões nos níveis de tensão, o EFP – Energia Fornecida Fora do Padrão.

Este índice não pode ser calculado de uma maneira clássica em alguns casos como mostra o Apêndice 1, então adotada uma abordagem alternativa, onde através

de algoritmos iterativos o valor de EFP vai sendo obtido para cada consumidor onde os níveis de tensão foram transgredidos.

CAPÍTULO VI

6. MÉTODOS DE ADEQUAÇÃO DOS NÍVEIS DE TENSÃO

6.1. Contextualização

A adequação dos níveis de tensão nos sistemas de distribuição pode ocorrer de forma insatisfatória, caso não haja um adequado sincronismo entre o ajuste dos tapes dos transformadores e a determinação da lei de controle na subestação.

Visto que os tapes dos transformadores das redes não podem ser reconfigurados sob carga, estes são determinados para um nível de demanda constante. As variações de demanda ao longo do dia são, por sua vez, tratadas pelo comutador de tap sob carga da subestação.

É impossível realizar a análise correta dos níveis de tensão nas redes de distribuição sem levar em conta os meios técnicos ligados ao controle de tensão. Os equipamentos mais comuns para o controle de tensão nas subestações e redes de distribuição são:

- Transformadores: sem comutação sob carga (geralmente apresentam três posições de tap, correspondentes a 100%, 95% e 90% da tensão nominal) nas redes de distribuição; ou com comutação sob carga (controle manual ou automático através do comutador de tapes sob carga) nas subestações;
- Bancos de Capacitores: atuação com controle manual ou programável;
- Reguladores de Tensão: autotransformadores (compensação de cerca de 10% da tensão).

Destes enumerados anteriormente, os meios básicos de controle de tensão utilizados no trabalho envolvem apenas os tapes dos transformadores de distribuição e o comutador de tapes sob carga da subestação. Optou-se por esta abordagem pois os bancos de capacitores com controle manual podem causar transgressões

por sobretensão em momentos de demanda mínima. Já os equipamentos mais modernos com controle automático, bem como os reguladores de tensão possuem custos mais elevados.

Então, os meios básicos de controle de tensão podem ter a sua configuração otimizada de forma a utilizar outros meios para o controle de tensão, apenas nos casos mais críticos onde eles realmente se fazem necessários.

Para o entendimento acerca da determinação do perfil de tensão para os consumidores de um sistema de distribuição, faz-se necessária a análise de alguns conceitos básicos do sistema.

A figura 6.1 mostra um alimentador, com sua rede de distribuição de média tensão, um transformador de distribuição, com as respectivas cargas ligadas no lado de baixa tensão, e o perfil de tensão, de forma simplificada, ao longo de toda a rede de distribuição.

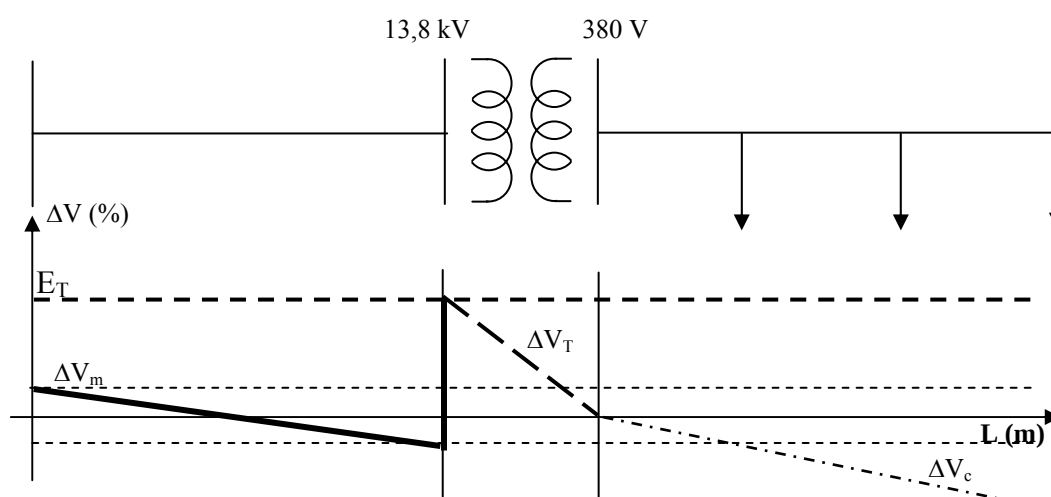


Figura 6.1 – Perfil simplificado de tensão em uma rede de distribuição.

Onde:

E_T : elevação de tensão;

ΔV : queda de tensão;

ΔV_m : queda de tensão na rede primária;

ΔV_c : queda de tensão até o centro de carga do transformador de distribuição;

ΔV_T : queda de tensão interna do transformador de distribuição;

L : comprimento da rede em metros.

No ponto onde está localizado o transformador de distribuição, o nível de tensão sofre um aumento (E_T), em função do tap do transformador e uma conseqüente queda (ΔV_T) devido à sua reatância interna.

Geralmente é desejável manter o nível de tensão na subestação o mais alto possível para o período de demanda máxima. Por conseguinte, os transformadores de distribuição, no início da rede de média tensão, apresentam tap 1 - 100% (mais baixo). Ao longo do alimentador, os transformadores podem passar para os taps 2 e 3, com o objetivo de compensar a queda de tensão na rede de média tensão. Há o cuidado, no entanto, de não se provocar uma sobretensão no período de demanda mínima.

Em concordância com as questões salientadas acima, mostra-se necessária a análise da variação de tensão para os dois regimes extremos, correspondentes à demanda máxima e à demanda mínima.

Analisando-se desta maneira, todos os outros regimes podem ser considerados intermediários. Logicamente que se as exigências correspondentes à qualidade de tensão forem satisfeitas para os regimes extremos, serão satisfeitas também para os regimes intermediários.

6.2. Ajuste dos Taps dos Transformadores de Distribuição

A Figura 6.2 mostra as alterações de tensão, de forma detalhada, no sistema de distribuição. Inclui-se, na figura, a rede de média tensão, os terminais de baixa tensão dos transformadores de distribuição e as redes de baixa tensão.

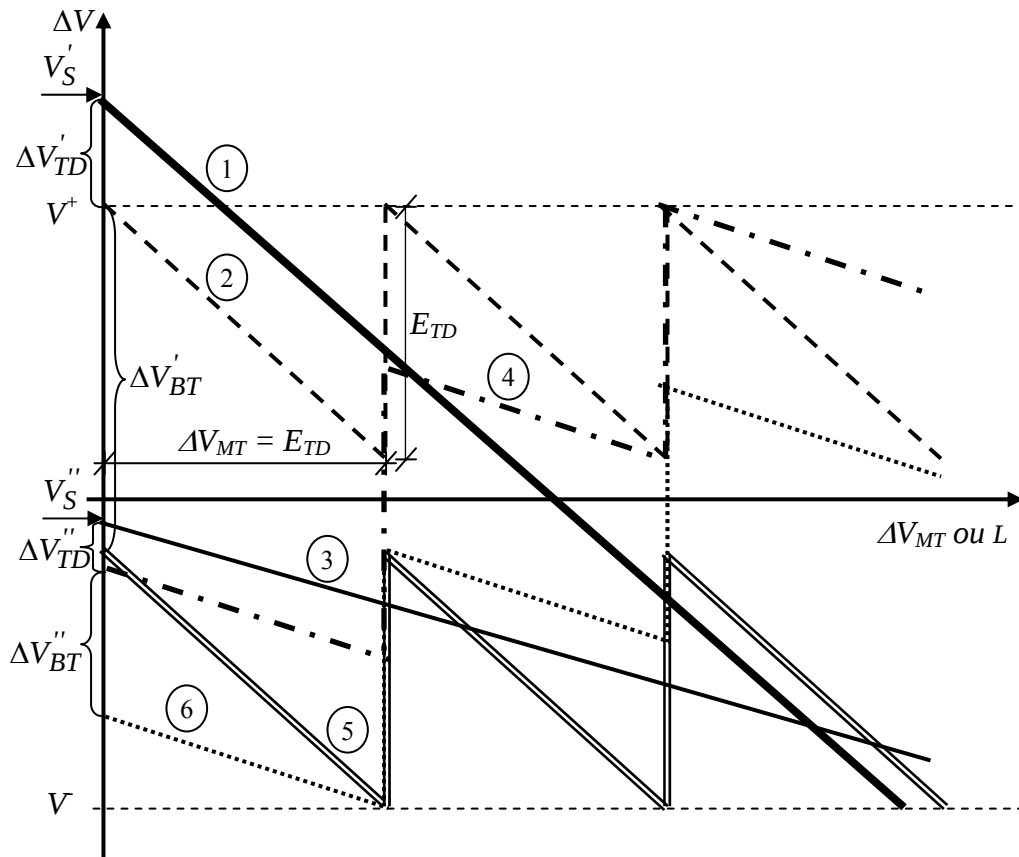


Figura 6.2 – Exemplo detalhado de perfil de tensão em redes de distribuição.

—	(1) tensão na rede primária de distribuição (em caso de demanda máxima);
- - -	(2) tensão nos terminais de BT dos TDs (em caso de demanda máxima);
—	(3) tensão na rede primária de distribuição (em caso de demanda mínima);
- . -	(4) tensão nos terminais de BT dos TDs (em caso de demanda mínima);
==	(5) tensão no final das redes de BT (em caso de demanda máxima);
.....	(6) tensão no final das redes de BT (em caso de demanda mínima);
V_{MT}	queda de tensão na rede primária de distribuição;
L	comprimento do alimentador.

Constata-se, na figura 6.2, que as alterações nos taps dos transformadores de distribuição ocorrem sempre que a queda na rede de média tensão ultrapassa o valor do degrau do tap de regulação de tensão.

As condições ótimas de operação dos equipamentos de uma instalação ocorrem quando se trabalha com a tensão nominal. Como, na prática, é impossível obter tal condição para todos os consumidores, propõe-se escolher o regime de funcionamento dos meios básicos de controle de tensão de tal forma que se mantenha o regime de tensão, o mais próximo do nominal, no ponto chamado

“centro de carga da rede de BT de cada TD”. Desta forma, é possível calcular, a partir da Equação 6.1, a queda de tensão do TD n até o centro de carga de sua rede de baixa tensão, ponderada pela potência, para cada instante t.

$$\Delta V_{Cn_t} = \frac{\sum_{m_n=1}^{M_n} \Delta V_{m_nt} P_{m_nt}}{\sum_{m_n=1}^{M_n} P_{m_nt}}, t = 1, \dots, T \quad (6.1)$$

onde:

ΔV_{Cn_t} = queda de tensão até o centro de carga;

$t = 1, \dots, T$ = período de tempo considerado;

ΔV_{m_nt} = queda de tensão percentual a partir do terminal de BT do TD n até o consumidor m, no instante t;

P_{m_nt} = potência do consumidor m do TD n no instante t;

Mn = quantidade de consumidores de BT do TD n.

Esta é a forma clássica de cálculo de centro de carga, entretanto como neste trabalho propõe-se a utilização de Modelos Representativos das Redes de baixa tensão, também existe uma forma simplificada de realizar este cálculo. O método utilizado na implementação do trabalho foi o descrito na seção 4.6, onde existe uma expressão simplificada para cálculo de centro de carga, para cada modelo simplificado das redes secundárias.

As condições ótimas de operação dos equipamentos de uma instalação ocorrem quando se trabalha com a tensão nominal. Na prática, no entanto, não é possível obter esta condição para todos os consumidores.

Por isso propõe-se a escolha do regime de funcionamento dos meios básicos de controle de tensão de uma forma tal, que se mantenha o regime de tensão, o mais próximo do nominal, no centro de carga da rede de baixa tensão de cada transformador de distribuição.

Desta forma, garante-se, na média, o melhor nível de tensão para o conjunto dos consumidores de cada transformador de distribuição. Entretanto deve-se

lembrar que os taps dos transformadores de distribuição não podem ser configurados para troca automática, conforme a queda de tensão ou demanda da rede.

Como a demanda varia constantemente ao longo do dia, e por consequência as quedas de tensão, a escolha do regime de carga influencia na configuração de tapes que serão utilizados nos transformadores. A seção a seguir discute as questões relativas à escolha deste regime de demanda.

6.3. Escolha do regime de demanda para ajuste dos tapes

A escolha do regime de demanda a ser utilizado para o ajuste dos tapes é um problema de otimização bastante complexo. O regime de demanda a ser escolhido deve levar em consideração vários fatores como as quedas em cada um dos alimentadores e também em cada um dos transformadores de distribuição.

Como o ajuste de tapes baseado na queda de tensão leva em consideração quedas na rede primária e secundária, deve-se ter consciência de que a curva de carga destes alimentadores pode ser muito diferente, assim como a curva de carga do próprio TD pode ser bem diferente do alimentador em que ele se encontra. Todos estes fatores tornam a escolha de um regime de demanda global para a subestação computacionalmente difícil, visto que esta variável deveria satisfazer em média todos os transformadores de distribuição.

Nos testes realizados em Software, os regimes de demanda máxima e mínima não surtiram bons resultados. O regime de demanda médio teve bons resultados em alguns casos, mas não em todos, no caso do alimentador 8 da SE Porto Alegre 9 (seção 5.3), o indicador de EFPF piorou em relação à configuração já instalada. Estes fatos tornaram necessária a utilização de um método diferenciado para a escolha dos tapes dos transformadores de distribuição.

6.4. Escolha dos tapes baseado em EFPF

Já que está se buscando otimizar o índice EFPF, então é recomendável utilizar uma abordagem para a escolha dos tapes dos transformadores de distribuição baseado no índice mínimo de EFPF.

O esforço computacional adicionado ao algoritmo de cálculo de EFPF para acumular em cada transformador a EFPF por tap configurado é muito pequeno. Depois deste passo basta verificar em cada transformador qual foi o tap que obteve o melhor resultado, com isto sempre se tem a solução ótima de configuração para o indicador EFPF. O Apêndice 3 mostra como ficaram os algoritmos finais para a escolha dos tapes dos transformadores.

Portanto, depois de realizada a escolha adequada dos taps dos transformadores de distribuição, deve-se partir para a definição da lei de controle de tensão na subestação.

Esta lei se encarrega de compensar as alterações diárias de carga e os níveis de queda de tensão nas redes de distribuição, já que os transformadores de distribuição não têm um controle automático de tensão. Os dois ajustes, nos tapes dos transformadores e no comutador de tapes sob carga da SE, trabalham de forma complementar.

6.5. Lei de Controle de Tensão na Subestação

6.5.1. Análise Geral da Rede de Distribuição

Na subestação, através do comutador de tap sob carga, são tratadas as variações de demanda ao longo do dia. A lei de controle de tensão na SE consiste na elevação de tensão necessária para satisfazer a queda de tensão nos pontos de consumo, em função da demanda, ou corrente instantânea.

Para a resolução deste problema também foram implementadas duas metodologias. A primeira mais clássica leva em consideração pontos críticos, em horários de demanda máxima e mínima, ao longo do alimentador. Estes pontos seriam o transformador mais próximo da SE, o último transformador com menor tap, o primeiro transformador com maior tap, etc. Os outros transformadores em regiões menos críticas não precisam ser analisados, visto que se as condições de nível de tensão forem boas nos pontos extremos, nos intermediários também serão.

Embora esta abordagem também proporcione bons resultados, ela não é ideal, visto que ela leva em consideração uma linearidade dos tapes ao longo do alimentador, ou seja, a medida que se afasta da SE os tapes tem a sua configuração

somente crescente. Sabe-se que devido a abordagem para o ajuste dos tapes que leva em consideração não apenas as quedas na rede primária, mas também as quedas na rede secundária, isto nem sempre pode acontecer. Nada impede que um transformador com um grande consumidor ligado diretamente a ele (sem quedas na rede secundária) esteja configurado num tap inferior aos transformadores ao redor deste na rede primária.

Esta metodologia foi implementada para testes e possui um bom desempenho, ela pode ser analisada no Apêndice 2. Entretanto, pelos motivos anteriormente citados foi desenvolvida outra abordagem para o ajuste da Lei de Controle da SE.

6.6. Ajuste da Lei de Controle na SE

A metodologia que seleciona os transformadores críticos para ajustar a tensão nestes pontos extremos, pode não ter um desempenho tão bom na redução das infrações de nível de tensão na rede, graças à distribuição não necessariamente uniforme dos tapes ao longo do alimentador. Neste caso a metodologia mais eficaz deve utilizar uma média das quedas de tensão nos transformadores de distribuição, visando atender o maior número possível de consumidores dentro dos níveis permissíveis de tensão.

A abordagem a ser apresentada para escolha da lei de controle de tensão nas SEs deve corresponder às condições reais que ocorrem nos sistemas de distribuição. É certo que vários ALs, contendo diversos TDs e redes de BT têm diferentes valores de queda de tensão. Além disso, as curvas de carga de diferentes consumidores podem ser bastante heterogêneas. Tudo isto precisa ser levado em conta no processo de desenvolvimento dos algoritmos de controle de tensão.

Parte-se da premissa de que as condições ótimas de controle de tensão acontecem quando, para cada instante de tempo, é possível garantir o aumento no nível de tensão na barra da subestação igual ao valor (com o inverso do sinal, no entanto) da queda de tensão ponderada pela potência em relação a todos os centros de carga, de todos os transformadores de distribuição que recebem energia da subestação em questão.

São analisadas abaixo considerações, as quais podem ser utilizadas para a definição dos níveis de tensão desejáveis nas subestações para os regimes de demanda máxima e mínima.

Na figura 6.3, há um alimentador convencional de uma subestação e os transformadores de distribuição instalados ao longo da rede.

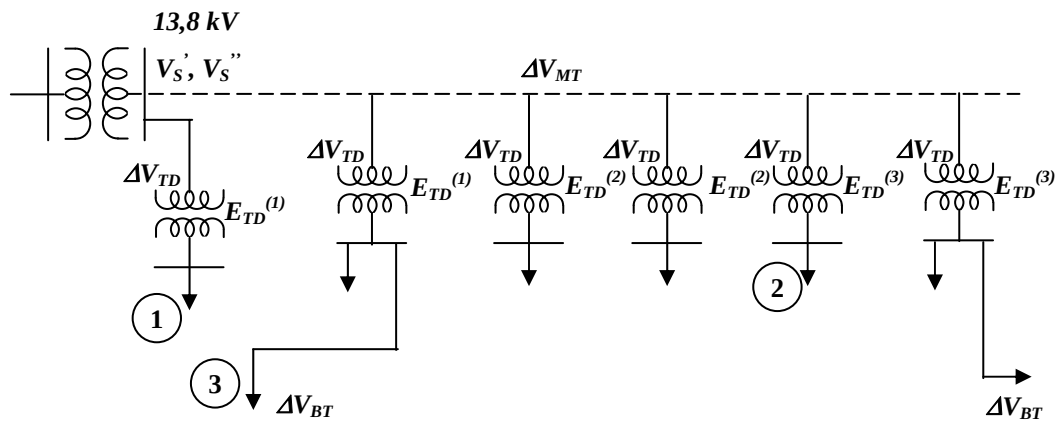


Figura 6.3 – Alimentador convencional de uma rede de distribuição.

A partir da análise da Figura 6.3, obtém-se a Equação 6.2.

$$\Delta V = E_{TD} - \Delta V_{MT} - \Delta V_{TD} - \Delta V_C \quad (6.2)$$

onde:

ΔV : corresponde à queda de tensão total até o centro de carga da rede BT;

E_{TD} : corresponde ao aumento no nível de tensão, devido à variação de tap do transformador de distribuição;

ΔV_{MT} : corresponde à queda na rede de média tensão;

ΔV_{TD} : corresponde à queda na tensão interna do transformador de distribuição;

ΔV_C : corresponde à queda na rede de baixa tensão até o centro de carga.

Com a Equação 6.2 é possível obter o valor total da queda de tensão até o centro de carga do transformador, considerando todas as quedas, e inclusive a elevação de tensão do tap, caso exista. A elevação de tensão na subestação deve compensar estas quedas, em cada um dos transformadores, através da média ponderada pela potência das quedas de tensão até o centro de carga destes, conforme mostra a Equação 6.3.

$$E_{Tse} = - \frac{\sum_{t=1}^N \Delta V_{C_t} P_t}{\sum_{t=1}^N P_t} \quad (6.3)$$

Onde: E_{Tse} = elevação de tensão na SE;

t = transformador;

N = total de transformadores;

ΔV_{C_t} = queda de tensão até o centro de carga do transformador t;

P_t = potência do transformador t.

Para obtermos a Lei de Controle de Tensão na SE ainda são necessários mais alguns passos. A Equação anteriormente definida representa a elevação de tensão necessária, instantaneamente, entretanto é necessário ter esta elevação ao longo das 24 horas.

Para tanto é necessário que se tenham as curvas de carga de cada transformador de distribuição. No caso deste trabalho, as curvas de carga são obtidas através das leituras mensais dos consumidores e da curva típica de carga associada a cada um deles (KÖNIG, 2004). A Figura 6.4 mostra a elevação de tensão em função do tempo para a SE Porto Alegre 9.

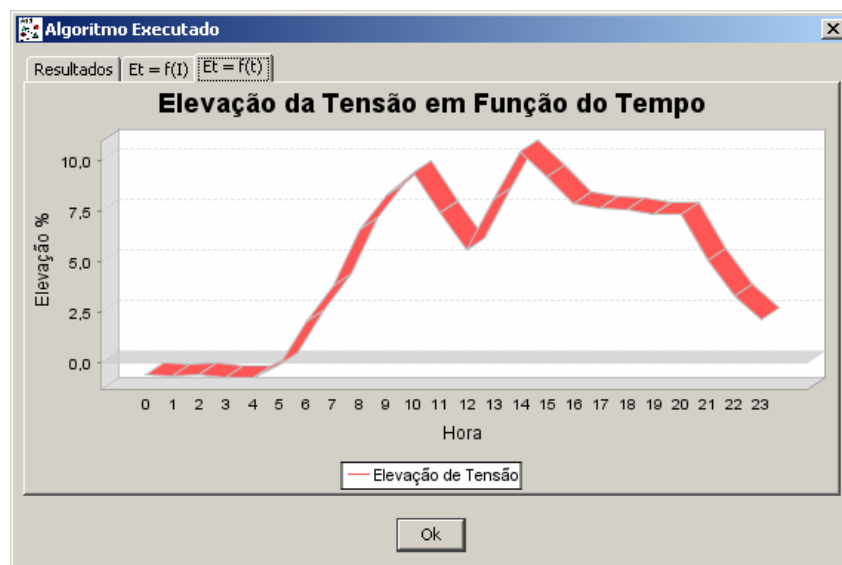


Figura 6.4 – Elevação da Tensão em Função do Tempo

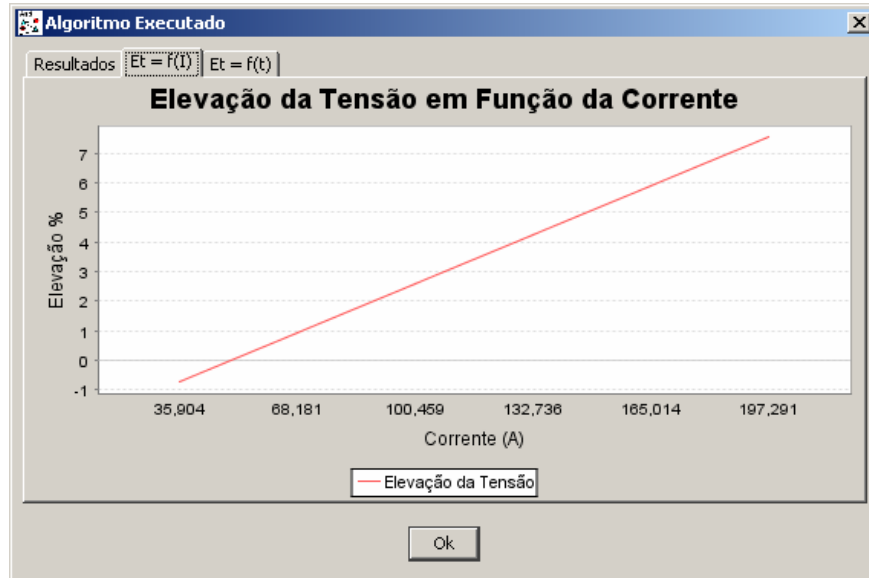


Figura 6.5 – Elevação da Tensão em Função da Corrente

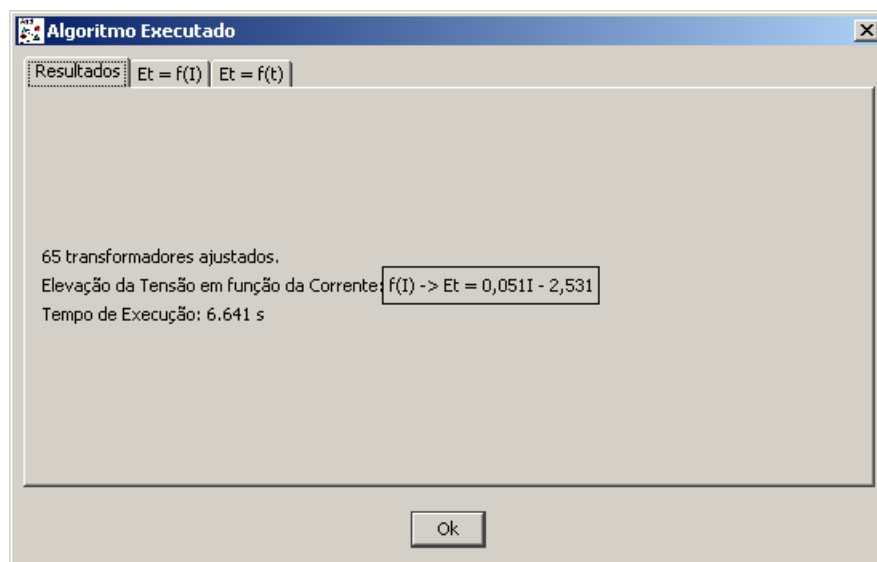


Figura 6.6 – Equação da Elevação da Tensão em Função do Tempo

Até agora os valores obtidos são da elevação ao longo do tempo, entretanto a programação realizada no computador da subestação é em função da corrente e não do tempo. Contudo, para cada tempo t já existe um valor calculado da corrente na barra da SE, que foi obtido através do cálculo do fluxo de potência (BERNARDON, 2004).

Através da aplicação do método dos mínimos quadrados é possível obter a função da elevação de tensão em função da corrente, conforme a reta mostrada na Figura 6.5, e a função mostrada na Figura 6.6.

Como já foi dito anteriormente este algoritmo deve ser utilizado em conjunto com o de ajuste de tapes nos transformadores de distribuição, visto que ele tem como objetivo compensar as variações nas quedas de tensão, devido à curva de demanda ao longo do dia. Entretanto, após cada ajuste na Lei de Controle de Tensão, deve-se executar novamente o ajuste dos tapes dos transformadores, visto que em alguns casos algum reajuste pode ser necessário devido justamente à Lei de Controle de Tensão na SE.

Nota-se então que existe dependência mútua na execução dos dois algoritmos, tornando este ajuste iterativo até que se atinja uma convergência de resultados. Contudo, devido ao fato de existirem apenas 3 posições nos tapes dos transformadores a convergência normalmente já é obtida na primeira iteração.

6.7. Resumo do Capítulo

Este Capítulo conclui a parte teórica deste trabalho. Ele utiliza todas as metodologias desenvolvidas ao longo da dissertação para enfim, ajustar os meios básicos de controle de tensão.

Numa introdução foi contextualizado o problema, logo partiu-se para o ajuste dos tapes dos transformadores de distribuição, o qual chega num impasse que é a determinação do nível de demanda para ajuste do tap. A seguir foi mostrada uma solução alternativa que ajusta o tap buscando minimizar o indicador EFP para o transformador de distribuição.

Finalmente foi demonstrada uma solução para a definição da Lei de Controle de Tensão na Subestação, sendo que esta deve ser utilizada em conjunto com o ajuste nos transformadores de distribuição, pois ela tem a função de corrigir as quedas de tensão que ocorrem com a variação da demanda ao longo do dia.

CAPÍTULO VII

7. RESULTADOS OBTIDOS

7.1. Software Desenvolvido

As concessionárias de distribuição de energia elétrica hoje em dia, devem cumprir uma série de legislações quanto à qualidade da energia que é disponibilizada aos consumidores finais. Os órgãos reguladores com muita frequência impõem multas pesadas às concessionárias que infringem os níveis aceitáveis de tensão. Por estes motivos o controle de tensão tornou-se ainda mais primordial e a utilização correta dos meios básicos de controle de tensão, pode ser um modo barato de fornecer uma energia com maior qualidade aos consumidores.

Com o objetivo de implementar todas os algoritmos aqui propostos, e principalmente de proporcionar uma ferramenta útil e que realmente auxilie as distribuidoras de energia elétrica no controle de tensão foi desenvolvido o ASD (Análise de Sistemas de Distribuição) – Controle de Tensão.

O Software faz parte de um projeto de P&D da UFSM – PPGE, com a Companhia Estadual de Distribuição de Energia Elétrica (CEEE-D), o qual iniciou em 2006 e terá seu término no final de 2007. O projeto prevê a implementação do Software e o estudo da Subestação Porto Alegre 9, sugerindo reconfiguração nos meios de controle de tensão.

O ASD foi desenvolvido em linguagem Java, totalmente orientado a objetos, utilizando uma interface interativa para o desenho dos alimentadores, permitindo que o usuário altere a rede como bem desejar, conforme mostra a Figura 7.1. O sistema foi modelado e concebido utilizando padrões de projeto, o que o tornou totalmente modular, ou seja, o ASD é uma plataforma onde algoritmos que realizam cálculos em redes de distribuição podem ser facilmente plugados.

Outra característica importante é a integração com a base de dados da concessionária CEEE, onde todos os dados referentes a topologia da rede primária, bem como leituras de consumo mensais, foram lidas diretamente a partir dos bancos de dados da CEEE-D. Como a base de dados da concessionária não possui no seu cadastro qual a fase em que os consumidores estão conectados, o ASD precisa considerar todos os consumidores como trifásicos e com equilíbrio entre as fases.

Como alicerce para os cálculos realizados pelos algoritmos desenvolvidos, existe a estimação de estado. Os algoritmos para cálculo do Fluxo de Potência foram propostos por BERNARDON (2004). Esta estimação de estado torna-se muito mais efetiva se utilizada em conjunto com a estimação de demanda horária dos transformadores, baseado em curvas típicas por classes de consumidores, este algoritmo foi proposto por KÖNIG (2004).

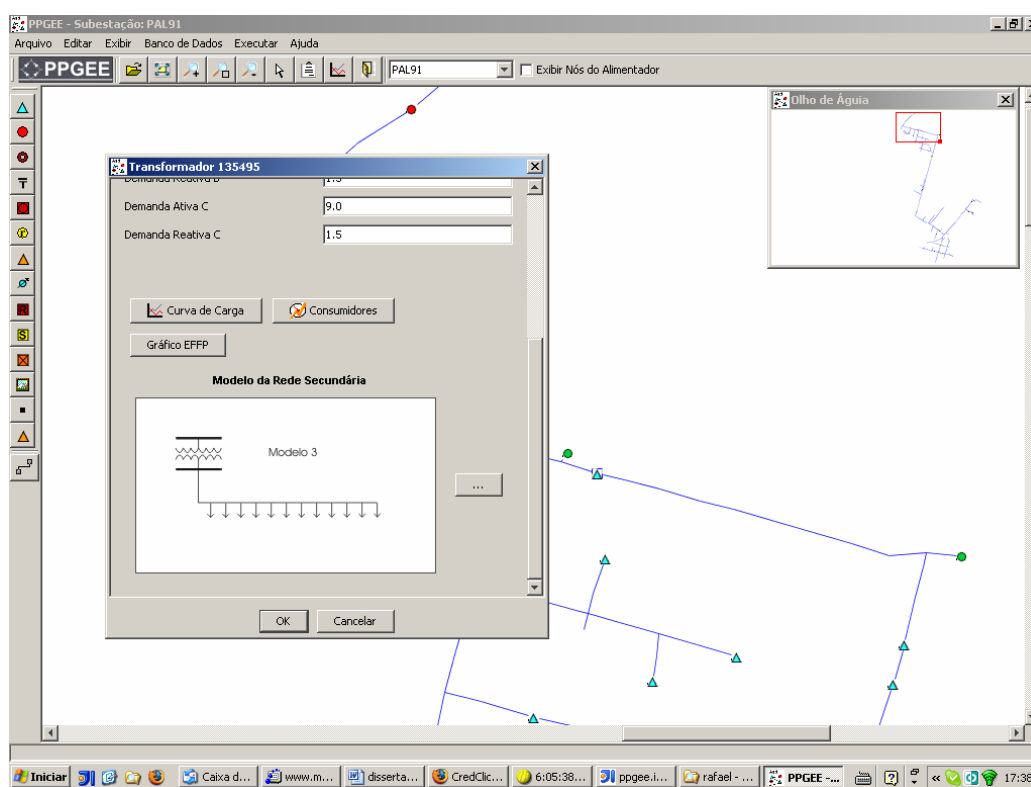


Figura 7.1 – ASD – Controle de Tensão

Nas seções a seguir serão apresentadas as funcionalidades do Software ASD.

7.3. Energia Fornecida Fora de Padrão

O primeiro passo na execução do sistema, antes mesmo do ajuste dos meios básicos de controle de tensão deve ser estimar o estado em termos de EFP que a rede se encontra com a configuração atual. Foi informado pela CEEE-D que todos os transformadores de distribuição da SE PAL9, estão configurados com o tap 2.

Num primeiro instante é necessário realizar a configuração do sistema. Ele permite que se escolha o mês do ano para fazer a simulação, conforme a Figura 7.2. Na aba “Parâmetros de Queda de Tensão” é possível configurar os valores de queda e elevação de tensão máximos, conforme a Figura 7.3, bem como a queda de tensão máxima permissível por padrões de projeto, e a queda de tensão máxima interna do transformador de distribuição.

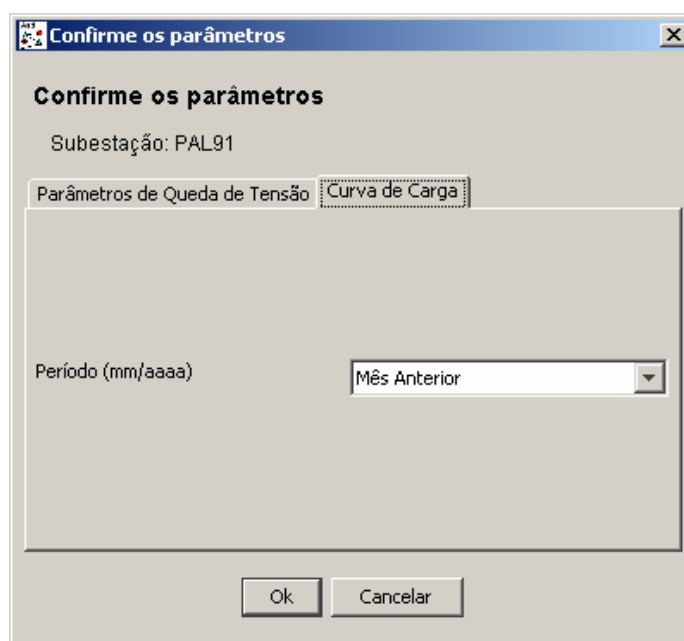


Figura 7.2 – Parâmetros EFP - 1.

Figura 7.3 – Parâmetros EFPF 2.

Tão logo seja concluída a configuração de parâmetros, já pode-se iniciar a execução do algoritmo.

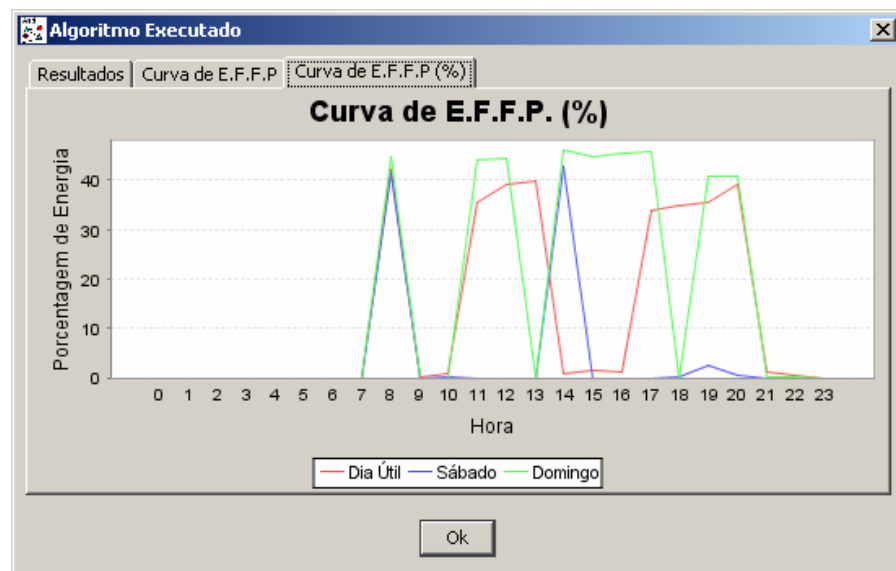


Figura 7.4 – Curva de EFPF.

A Figura 7.4. mostra a curva de Energia Fornecida Fora do Padrão ao longo das 24 horas dividida entre dias úteis, sábados e domingos. O valor no eixo das ordenadas está o valor de EFPF percentual, em relação ao total de energia distribuída naquele período.

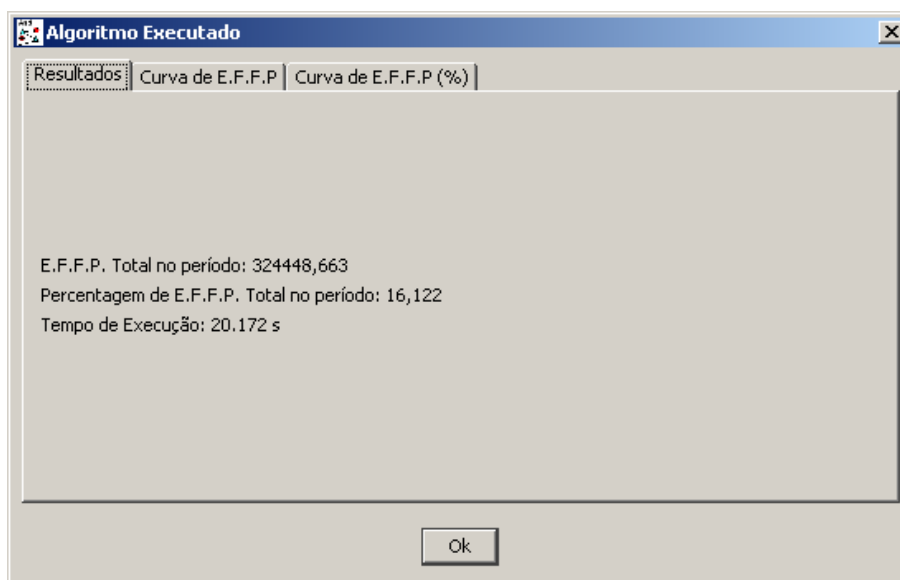


Figura 7.5 – Resultados - EFPF

A Figura 7.5 mostra uma tela de resultados que exibe um resultado de 16,122% de EFPF. Segundo estudos realizados pelo departamento de planejamento da CEEE-D, foi feita uma estimativa de que em torno de 20% da energia fornecida nos alimentadores da capital gaúcha esteja com níveis inadequados de tensão. O ASD-CT obteve um resultado bastante satisfatório neste caso, inclusive dentro expectativa da própria concessionária.

7.4. Ajuste dos tapes dos Transformadores de Distribuição

Já foi obtido o desempenho quanto aos níveis de tensão do alimentador com a configuração atual. Agora é hora de iniciar a otimização dos meios básicos de controle de tensão. Os tapes dos transformadores de distribuição serão configurados conforme o método descrito na seção 6.4.

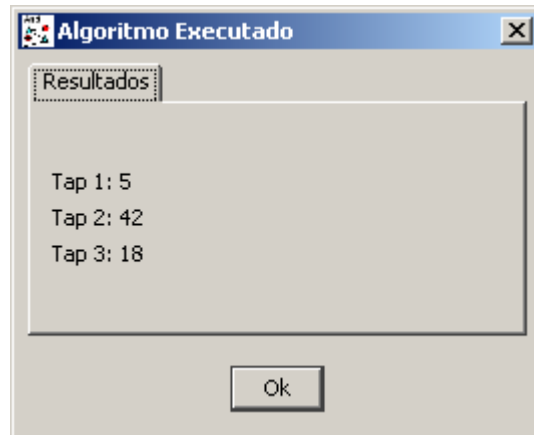


Figura 7.6 – Ajuste nos tapes

A Figura 7.6 mostra o resultado da execução do algoritmo. Este algoritmo não possui parâmetros, visto que a sua execução já foi realizada dentro do cálculo de EFPF, restando para esta fase apenas trocar os tapes dos transformadores para os que tiveram menor valor de EFPF acumulada no período.

Note também que a maioria dos tapes (42 deles) permaneceu na sua configuração anterior, que é o tap 2.

7.5. Lei de Controle de Tensão na SE

O próximo passo dentro do programa agora é realizar o ajuste da Lei de Controle de Tensão na SE.

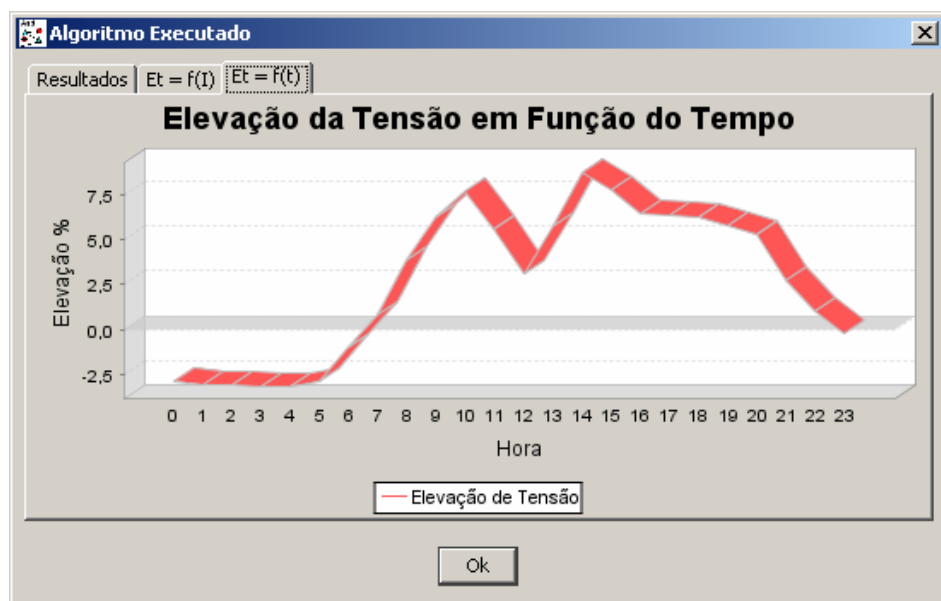


Figura 7.7 – Gráfico da Elevação da Tensão em Função do Tempo.

O resultado da execução pode ser visto na Figura 7.7, onde vê-se a importância do controle de tensão automático na SE. Nos horários de pico chega a ser necessário uma elevação de mais de 7,5%, sendo que nos horários de demanda mínima, é necessário uma redução de tensão para não haver sobretensão nos transformadores com tap maior do que 1.

7.6. Melhora do Indicador EFP após ajustes do Controle de Tensão

Finalmente, após os ajustes em tapes de transformadores de distribuição e Lei de Controle de Tensão na subestação, chega o momento de avaliar o resultado obtido pelos algoritmos neste caso em estudo.

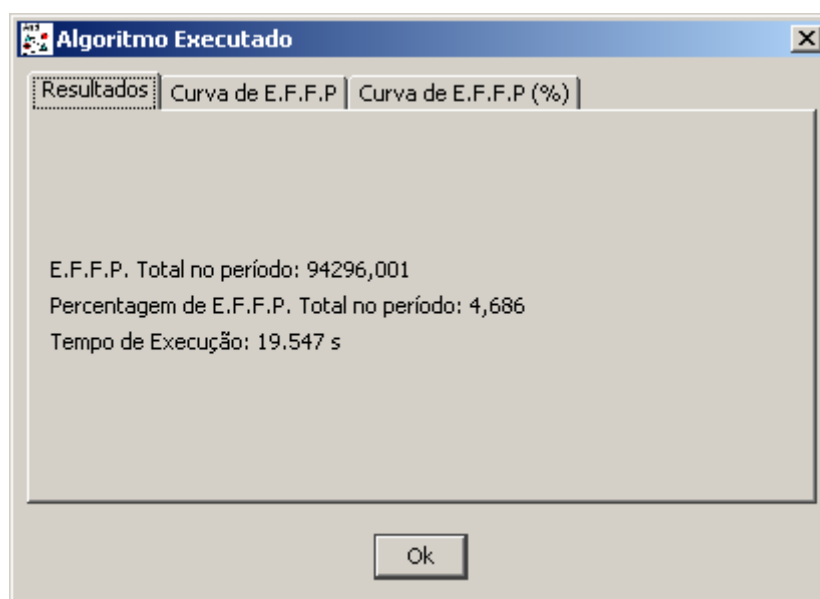


Figura 7.8 – Resultados de EFP após ajuste

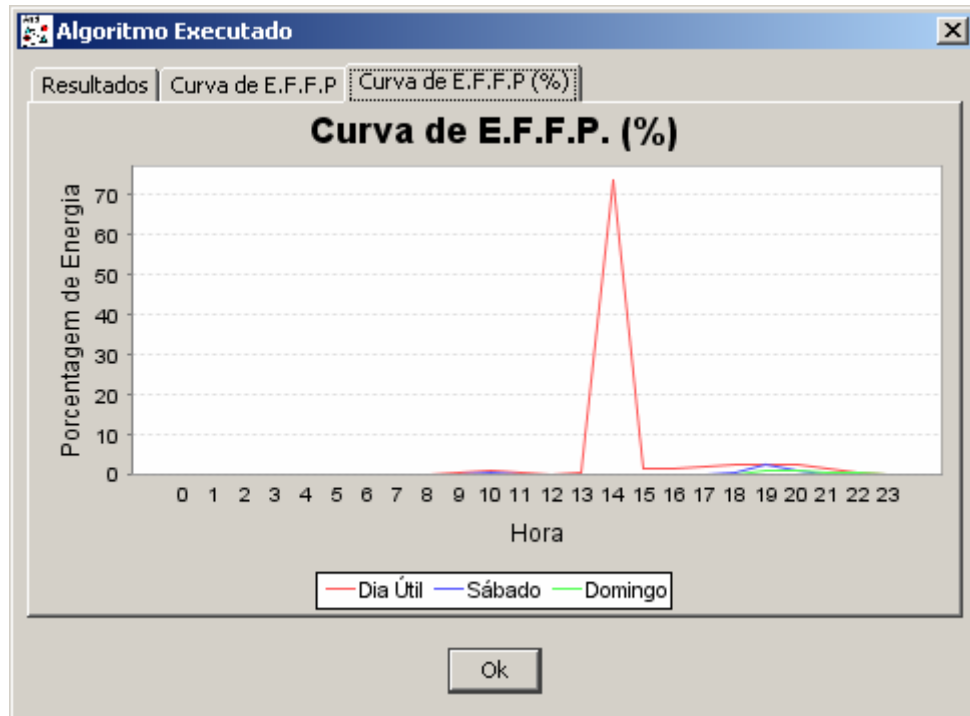


Figura 7.9 – Curva de EFPF após ajuste

Como pode ser visto houve uma redução de 16,122% para 4,686% (com cerca de 94.296,00 kWh de EFPF) do valor total percentual de EFPF no período estudado. Neste caso houve uma melhora bastante significativa, mas é importante ressaltar que os algoritmos sempre conseguem algum percentual de melhora no indicador EFPF, seja ele pequeno ou grande, visto que o objetivo é otimizar o indicador.

7.7. Resumo do Capítulo

Neste capítulo foi apresentada a ferramenta ASD, que foi desenvolvida utilizando os conceitos mais modernos da computação e implementa os algoritmos propostos nesta dissertação.

Realizou-se uma demonstração de uso do programa utilizando uma subestação real da Companhia Estadual de Distribuição de Energia Elétrica em Porto Alegre. Calculou-se o índice com a configuração de tapes informada pela CEEE-D, onde foi feito o ajuste dos meios básicos de controle de tensão. O resultado do indicador EFPF comparado com a configuração atual, obteve uma redução bastante significativa.

CAPÍTULO VIII

8. CONCLUSÃO

A busca de um fornecimento de energia com maior qualidade e redução dos custos com indenizações são metas das empresas distribuidoras de energia, e também deste trabalho. O trabalho focou seus esforços em desenvolver metodologias que satisfizessem as limitações das concessionárias, fornecendo ferramentas úteis e de possível implantação para que o aumento da qualidade da energia possa ser alcançado.

A ausência de dados confiáveis sobre as redes de baixa tensão, tornou impossível a estimação de estado no modo clássico, utilizando fluxo de potência. Foi necessária a construção de modelos representativos destas redes de baixa tensão, onde estes modelos simplificados possibilitaram a estimação de estado de uma forma também mais direta.

A utilização dos tapes dos transformadores de distribuição e do comutador de tap sob carga da subestação como meios básicos para o controle de tensão possibilita que a empresa de distribuição de energia reduza seus custos operacionais. Estes meios já estão disponíveis, não necessitando de nenhum investimento adicional em equipamentos. Esta dissertação mostra que o ajuste adequado destes meios de controle de tensão, pode auxiliar de forma significativa na correção dos níveis de tensão aos consumidores.

A configuração dos tapes dos transformadores de distribuição, levando-se em consideração as quedas de tensão na rede secundária, torna o processo de ajuste de tapes muito mais preciso. É necessário ressaltar que nem sempre os níveis de tensão poderão ser mantidos utilizando-se apenas estes meios de controle de tensão, então a solução mais satisfatória, em qualquer caso, é realizar o ajuste dos

tapes baseado no centro de carga das redes de baixa tensão, considerando todas as quedas até este ponto.

Em complemento ao ajuste dos tapes dos transformadores de distribuição também foi proposto uma forma de se definir a Lei de Controle de Tensão na Subestação. A regulação de tensão na SE é muito importante, visto que ela é realizada automaticamente pelo comutador de tapes sob carga, e tem a capacidade de suprir as variações de queda de tensão ao longo do dia.

A Lei de Controle de Tensão é calculada a partir da média ponderada pela potência das quedas de tensão até o centro de carga das redes de baixa tensão, ou seja, novamente são levadas em consideração as quedas na rede secundária. Como a queda considerada já inclui a possível elevação de tensão imposta pelo transformador de distribuição, fica visível a interação entre os dois algoritmos.

Para que fosse possível medir o desempenho das redes no que diz respeito à transgressão nos níveis de tensão, foi proposto um indicador de Energia Fornecida Fora do Padrão – EFPF, que difere dos indicadores propostos pela ANEEL. Este indicador é mais adequado, para os objetivos deste trabalho, por considerar o desempenho de toda a rede e também por quantificar a energia e não o número de consumidores.

O indicador EFPF tornou possível testar as melhoras obtidas em redes de distribuição reais, a partir dos ajustes nos tapes e Lei de Controle de Tensão da SE propostos. A introdução deste indicador trouxe também pequenas alterações no algoritmo de ajuste de tapes que passou a ter o objetivo exclusivo de minimizá-lo.

Como resultado de todos estes estudos e metodologias foi desenvolvido um Software, o ASD-CT, o qual implementou em linguagem Java, utilizando modernos conceitos de engenharia de software, todos os algoritmos aqui propostos. O ASD-CT foi testado em alimentadores reais da Companhia Estadual de Distribuição de Energia Elétrica (CEEE-D), onde demonstrou usabilidade e melhorias significativas nestes alimentadores.

Espera-se que o Software ASD-CT, que também faz parte de um projeto de P&D junto à distribuidora CEEE-D, contribua com a melhora da qualidade da energia fornecida aos consumidores, assim que este software tiver o seu uso iniciado.

8.1. Contribuições do trabalho

O trabalho desenvolvido teve as seguintes contribuições:

- Desenvolvimento de modelos simplificados das redes de baixa tensão, tornando possível a estimação de estado em redes de baixa tensão onde existe ausência de dados;
- Metodologia para o ajuste dos tapes dos transformadores de distribuição e Lei de Controle de Tensão na SE que leva em consideração as quedas da rede secundária;
- Desenvolvimento de um indicador para medir o desempenho das redes quanto às transgressões nos níveis de tensão;
- Desenvolvimento de um Software que implementa os algoritmos propostos neste trabalho, com boa usabilidade e integração com os dados das concessionárias.

8.2. Perspectivas para Trabalhos Futuros

- A apropriação dos modelos simplificados para as redes de baixa tensão é feito manualmente. O desenvolvimento de um algoritmo para a apropriação automática dos modelos seria muito útil para a implantação do sistema, visto que esta é a única parte que ainda permanece manual;
- O algoritmo de determinação da Lei de Controle de Tensão na SE, leva em consideração que o comutador de tapes sob carga faz infinitos chaveamentos ao longo do dia. Sabe-se que cada chaveamento realizado decrementa a vida útil do equipamento, então o desenvolvimento de um algoritmo que leve em consideração esta característica e consiga balancear o número de chaveamentos realizados e a EFPF é muito importante para aplicação prática.

9. Referências Bibliográficas

ANEEL – Agência Nacional de Energia Elétrica. Resolução Nº 505 de 26 de novembro de 2001, Nº 676 de 19 de dezembro de 2003.

ANEEL – Agência Nacional de Energia Elétrica. Nota Técnica nº 0035/2007 de 22 de Junho de 2007.

BACKLUND, Y. & J. A. Bubenko. Computer-Aided Distribution System Planning. Electrical Power & Energy Systems, Vol. 1, Nº 1, páginas 35-45.

BARAN M., and KELLY A., A branch current based state estimation method for distribution systems, IEEE Transaction on Power Systems, V. 10, N 1, 1995, pp. 483-491.

BARAN E., and HSU M., Volt/Var control at distribution substations, IEEE Transaction on Power Systems, V. 14, N 1, 1999, pp. 312-318.

BERNARDON, D.P.; KÖNIG, A.L.; COMASSETTO, L.; BATISTA, P.C. POPOV, V.A.; CANHA, L.N.; ABAIDE, A.R.; FARRET, F.A.. Emprego de novas Metodologias e Algoritmos mais eficientes para Modelagem das Cargas Elétricas e Estimação de Estados em Sistemas de Distribuição. XVI SENDI – Seminário Nacional de Distribuição de Energia Elétrica, Brasília / DF, novembro/2004.

BONHOMME A., CORTINAS D., BOULANGER J., and FRAISSE L., A new voltage control system to facilitate the connection of dispersed generation to 5th distribution networks, CIRED 2001, conference publication N 482, pp. 4.8.1-4.8.5.

BURANDT, R. P. et al. Present-Worth Approach for Optimizing Distribution Transformer and secondary Designs for Serve Growing Loads. AIEE Transactions – Part 3, New York, 79: 199-204, 1960.

CALOVIC M., Modeling and analysis of under load tap changing transformer control system, IEEE Transaction on Power Apparatus and Systems, V. 103, N 7, 1984, pp. 1909-1915.

CANHA, L. N. ET AL. Problemas de Controle de Tensão em Redes de Distribuição com Fontes Distribuídas de Geração e Armazenamento de Energia. Disponível em <www.aneel.gov.br>. Acesso em 02 de agosto de 2004.

CANHA, L. N. ET AL. Multicriterial Analysis for Optimal Location of Distributed Energy Sources Considering the Power System Reaction. IX SYMPOSIUM OF SPECIALISTS IN ELECTRIC OPERATIONAL AND EXPANSION PLANNING - SEPOPE, Rio de Janeiro, 2004.

CESPEDES R., New method for the analysis of distribution networks. IEEE Transaction on Power Delivery, V. 5, N 1, 1990, pp. 391-396.

CHANG S., MARKS G., and KATO K., Optimal real time voltage control IEEE Transaction on Power Systems, V. 5, N 3, 1990, pp. 750-758.

CHEN T., CHEN M., HWANG P., KATAS P., and CHEBI E., Distribution System power flow analysis - a rigid approach, IEEE Transaction on Power Delivery, V. 8, N 4, 1993, pp. 1966-1972.

CHOI J., and KIM J., Advanced voltage regulation method of power distribution system interconnected with dispersed storage and generation systems, IEEE Transaction on Power Delivery, V. 16, N 2, 2001, pp. 329-334.

FAN J., and SALEMAN S., The effect of integration of wind farms into utility network on voltage control due to the co-ordination of AVC relays. Proceedings of

the power system control, operation and management, APSCOM-97, Hong-Kong, 1977, pp. 260-265.

HAUBRICH H., Daniels G., Clemens G., Konings H., Brouns S., Direks H., and Dreuw H., Voltage control in medium voltage distribution networks, CIRED97, IEE1997, Conference publication 438, pp. 6.42.1-6.42.5.

JARDINI J.; TAHAN C.; GOUVEA M. et al. Curvas de Carga Típicas e Agregação dos Consumidores Residenciais, Comerciais e Industriais de Baixa Tensão. Encontro Luso-Afro Brasileiro de Planejamento e Exploração de Redes de Energia, ST7-7, 1999.

JIANG F., Bo Z., CHIN P., A new automatic voltage control relay for an-balanced distribution system, IEEE summer meeting, 2000, pp. 2301-2304.

KASZTENNY B., ROSOLOWSKI E., IZYKOWSKI J., SAHA M., and HILLSTROM B., Fuzzy logic controller for on-load transformer tap changer, IEEE Transaction on Power Delivery, V. 13, N 1, 1998, pp. 164-170.

KÖNIG, A. L. Desenvolvimento de algoritmos e ferramenta computacional para modelagem de cargas e redes secundárias de distribuição. Dissertação (Mestrado) Universidade Federal de Santa Maria, 2004, pp. 209.

LEON R., and LU C., Improved feeder voltage calculations results with telemetered data, IEEE Transaction on Power Delivery, V. 11, N 4, 1996, pp. 1914-1920.

LIANG R., and CHENG C., Dispatch of main transformer ULTC and capacitors in distribution system, IEEE Transaction on Power Delivery, V. 16, N 4, 2001, pp. 625-630.

LIANG R., and WANG Y., Fuzzy-based reactive power and voltage control in a distribution system, IEEE Transaction on Power Delivery, V. 18, N 2, 2003, pp. 610-618.

LU F., HSU Y., Fuzzy dynamic programming approach to reactive power /voltage control in a distribution substations, IEEE Transaction on Power Systems, V. 12, N 2, 1997, pp. 681-688.

MÉFFE, A.; OLIVEIRA, C.C.B.; CASOLARI, R.P. et al. Desenvolvimento de Modelos de Classificação para o Estabelecimento de Padrões Típicos de Representação de Redes de Baixa Tensão para Cálculo de Perdas Técnicas. Projeto P&D USP – RGE, r.01, set, 2001.

MIRANDA V., MANUEL A., MATOS A., and SARAIVA J, Fuzzy load flow - new algorithm incorporating uncertain generation and load representation Proceedings of the PSCC90, Gras, Austria, 1990, pp. 621- 627.

ROSA, Luiz Pinguelli; TOLMASQUIM, Maurício Tiomno; PIRES, José Cláudio Linhares. A Reforma do Setor Elétrico no Brasil e no Mundo: Uma visão Crítica. Relume Dumará. Coppe, UFRJ 1998.

SQUAIELLA, Davi J. F. Planejamento Agregado em Redes de Distribuição Secundária – Modelo Alternativo para empresas com Cadastro de Redes Reduzido. Dissertação (Mestrado) Universidade de São Paulo, Escola Politécnica, 2004.

STANGARLIN, W. Henrique; EKEL, Petr Ya.; FARRET, Felix A.; POPOV, Vladimir A. Improving Efficiency of Voltage Control in Distribution Systems. SBQEE 96 – Seminário Brasileiro da Qualidade de Energia Elétrica. Universidade Federal de Santa Maria, 1996.

SU C., and LIN C., Fuzzy-based voltage/reactive power scheduling for voltage security improvement and loss reduction, IEEE Transaction on Power Delivery, V. 16, N 2, 2001, pp. 319-123.

THORNLEY V., and HISCOCK A., Improved voltage quality through advances in voltage control techniques, Proceedings of the International Conference Developments in Power System Protection, London 2001, Conference publication N 479, pp. 355-358.

WANG Z., and ALVORADO F., Interval arithmetic in power flow analysis, IEEE Transaction on Power Systems, V. 7, N 3, 1992, pp. 1341-1349.

WAGNER W, KEYHAN A., Hao S., and Wong T., A rule based approach to decentralized voltage control, IEEE Transaction on Power Systems, V. 5, N 2, 1990, pp. 643-651.

APÊNDICE I

A1. Método Clássico para o Cálculo de EFPF

A1.1. Metodologia

Para o entendimento dos processos referentes à obtenção da Energia Fornecida Fora de Padrão são necessárias algumas definições a respeito das variáveis envolvidas.

Abaixo, encontram-se listadas as principais variáveis utilizadas ao longo deste capítulo:

- V^- : nível de tensão mínima permitida pela legislação vigente;
- V^+ : nível de tensão máximo permitido pela legislação vigente;
- W^- : quantidade de energia consumida com subtensão;
- W^+ : quantidade de energia consumida com sobretensão;
- V_{BT} : nível de tensão no início da rede de baixa tensão;
- ΔV_f : queda de tensão até o final da rede de baixa tensão;
- V_f : nível de tensão medido no final da rede de baixa tensão.

Na figura A1.1, há um esboço genérico de uma rede de baixa tensão, desde o transformador de distribuição até o consumidor final.

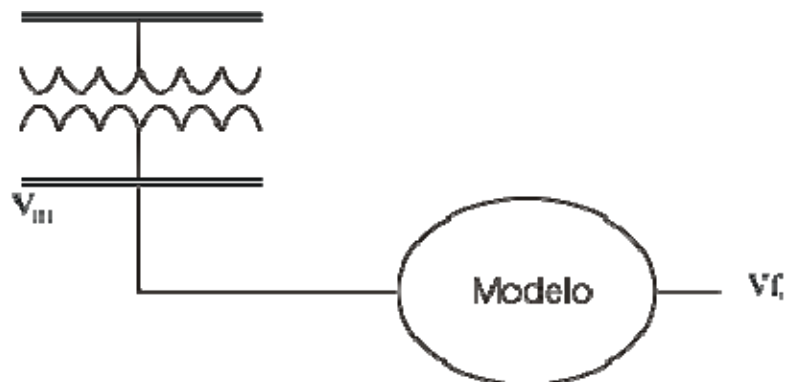


Figura A1.1 – Esboço genérico de uma rede de baixa tensão.

A tensão entregue ao consumidor final pode ser visualizada na equação A1.1.

$$V_{BT} - \Delta V_f = V_f \quad (A1.1)$$

A1.2. Modelo 1

Para o cálculo da Energia Fornecida Fora de Padrão referente ao modelo representativo mais simples, deve-se ter conhecimento das situações em que o indicador (a EFP) pode ser observado.

A1.2.1. Situação 1

Ocorre quando a tensão no início da rede de baixa tensão é menor do que a permitida. Então, toda a energia consumida é considerada fora de padrão, verificando-se um exemplo de subtensão.

$$V_{BT} < V^- \rightarrow W_t^- (100\%) \quad (A1.2)$$

A1.2.2. Situação 2

Ocorre quando a tensão no início da rede de baixa tensão está dentro dos limites permitidos. Desta forma, nenhuma energia é consumida fora de padrão.

$$V^- \leq V_{BT} \leq V^+ \rightarrow 0 \quad (A1.3)$$

A1.2.3. Situação 3

Ocorre quando a tensão no início da rede de baixa tensão é maior do que a permitida. Por isso, toda a energia consumida é considerada fora de padrão. Há, neste caso, um exemplo de sobretensão.

$$V_{BT} > V^+ \rightarrow W_t^+ (100\%) \quad (A1.4)$$

A1.3. Modelo 2

O cálculo da Energia Fornecida Fora de Padrão, para este modelo, é bastante similar ao apresentado no modelo anterior.

Deve-se, entretanto, considerar a tensão no final da rede e, não mais, no início da rede, pois, neste modelo, existem perdas a serem consideradas no decorrer da rede de baixa tensão.

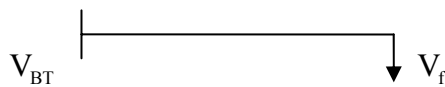


Figura A1.2 – Representação de rede de baixa tensão.

A1.3.1. Situação 1

Acontece quando a tensão no final da rede de baixa tensão é menor do que a permitida. Toda a energia consumida então é considerada fora de padrão, registrando-se um exemplo de subtensão.

$$V_f < V^- \rightarrow W_t^- (100\%) \quad (A1.5)$$

A1.3.2. Situação 2

Acontece quando a tensão no final da rede de baixa tensão está dentro dos limites permitidos. Logo, nenhuma energia é consumida fora de padrão.

$$V^- \leq V_f \leq V^+ \rightarrow 0 \quad (A1.6)$$

A1.3.3. Situação 3

Acontece quando a tensão no início da rede de baixa tensão é maior do que a permitida. Assim, a energia consumida é considerada fora de padrão. Há, neste caso, um exemplo de sobretensão.

$$V_f > V^+ \rightarrow W_t^+ (100\%) \quad (A1.7)$$

A1.4. Cálculo de EFP para as unidades elementares

Assim como nas seções anteriores para facilitar os cálculos da EFP, os modelos serão divididos nas unidades elementares denominadas “Elemento A” e “Elemento B”.

A1.4.1. Cálculo de EFP no Elemento A

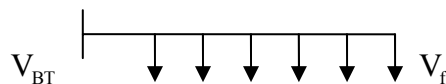


Figura A1.3 – Elemento estrutural simplificado “A”.

Para o cálculo da EFP neste caso existem seis situações possíveis conforme mostra o Quadro A1.1.

Situação	V_{BT}	V_f	W
1	$> V^- < V^+$	$> V^- < V^+$	0
2	$> V^+$	$> V^- < V^+$	W^+
3	$< V^-$	$< V^-$	$W^- (100\%)$
4	$> V^+$	$> V^+$	$W^+ (100\%)$
5	$> V^- < V^+$	$< V^-$	W^-
6	$> V^+$	$< V^-$	$W^+ \quad W^-$

Quadro A1.1 – Possíveis situações no cálculo de EFP.

Dentre estas 6 situações as mais críticas são a 2, 5 e 6, onde a quantidade de energia consumida com níveis de tensão fora do padrão depende diretamente das quedas na rede secundária. Nas situações restantes as quedas na rede secundária não influenciam no cálculo da EFP, visto que a mesma sempre assumirá valores inteiros, 0 ou 100%.

A1.4.1.1 Situação 2

Na situação 2 o nível de tensão inicial está acima do permitido, entretanto o nível de tensão final está em um nível permitido. Logo, existirá um ponto de distância ℓ do transformador dentro da rede secundária onde a queda de tensão é suficiente para que a tensão volte a um nível adequado. Após este ponto as unidades consumidoras estarão com um nível de tensão adequado.

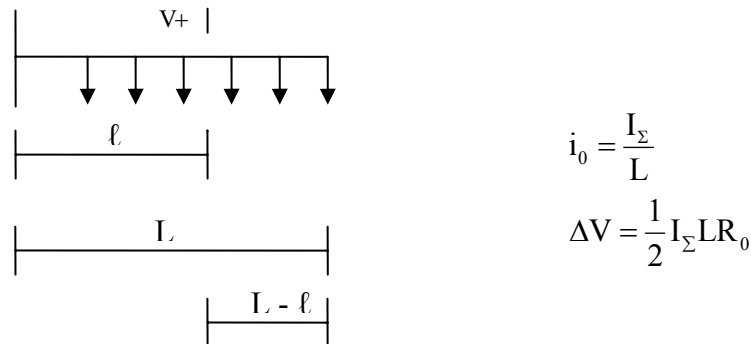


Figura A1.4 – Situação 2 – Elemento A

$$\Delta V_\ell = \left[\frac{1}{2} I_\Sigma \frac{\ell}{L} \ell R_0 + I_\Sigma \frac{(L \cdot \ell)}{L} \ell R_0 \right] \quad (A1.8)$$

Entretanto o valor de ΔV_ℓ pode ser definido de maneira exata, considerando-se que ΔV_ℓ é a queda de tensão até o ponto de violação dos níveis permissíveis de tensão, e que V_{BT} e V^+ são valores conhecidos.

$$\Delta V_\ell = V_{BT} - V^+ \quad (A1.9)$$

Por isso a outra parte da equação também deve ser definida exatamente através da introdução do coeficiente de correção. Por isso:

$$\Delta V_\ell = \left[\frac{1}{2} I_\Sigma \frac{\ell}{L} \ell R_0 + I_\Sigma \frac{(L-\ell)}{L} \ell R_0 \right] \cdot K_{V_2}$$

$$K_{V_2} = f\left(\frac{N-k}{k}, k\right)$$

onde:

N: número total de consumidores.

k: número de consumidores até o ponto de violação dos níveis permissíveis de tensão.

$$\Delta V_\ell = \left[\frac{1}{2} I_\Sigma \frac{\ell^2}{L} R_0 + I_\Sigma \ell R_0 - I_\Sigma \frac{\ell^2}{L} R_0 \right] \cdot K_{V_2}$$

$$\Delta V_\ell - I \ell R_0 K_{V_2} + \frac{1}{2} I_\Sigma \frac{\ell^2}{L} R_0 K_{V_2} = 0$$

$$\Delta V_\ell - I L R_0 \frac{\ell}{L} K_{V_2} + \frac{1}{2} I_\Sigma L R_0 \left(\frac{\ell}{L} \right)^2 K_{V_2} = 0$$

Suponha:

$$x = \frac{\ell}{L} \quad ; \quad a = \frac{1}{2} I L R_0 \cdot K_{V_2} = \Delta V_f \cdot K_{V_2} \quad ; \quad b = I L R_0 \cdot K_{V_2} = 2 \Delta V_f K_{V_2} \quad ; \quad c = \Delta V_\ell$$

Então temos:

$$ax^2 - bx + c = 0$$

$$x^2 - \frac{b}{a}x + \frac{c}{a} = 0$$

$$x^2 - 2x + \frac{\Delta V_\ell}{\Delta V_f \cdot K_{V_2}} = 0$$

$$\frac{\ell}{L} = 1 - \sqrt{1 - \frac{\Delta V_\ell}{\Delta V_f \cdot K_{V_2}}}$$

Mas como: $\frac{\ell}{L} = \frac{k}{N}$

$$\frac{k}{N} = 1 - \sqrt{1 - \frac{\Delta V_\ell}{\Delta V_f \cdot K_{V_2}}}$$

Considerando que $\Delta V_\ell = \Delta V_{BT} - V^+$ temos:

$$k^+ = \left[N - N \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f \cdot K_{V_2}}} \right]$$

Onde [a] significa menor valor inteiro mais próximo de “a”.

$$K_{V_2} = f\left(\frac{N - k^+}{k^+}, k^+\right)$$

ΔV_f já foi definido anteriormente.

Finalmente:

$$W^+ = k^+ \frac{I_\Sigma \cdot V_{BT}}{N} \quad (A1.10)$$

A1.4.1.2. Situação 5

Na situação 5 o nível de tensão inicial está dentro do permitido, entretanto o nível de tensão final está abaixo do nível mínimo permitido. Logo, existirá um ponto ℓ dentro da rede secundária onde a queda de tensão é suficiente para que a tensão passe da faixa adequada para inadequada, deixando as unidades consumidoras após este ponto com um nível de tensão baixo.

Considerando que:

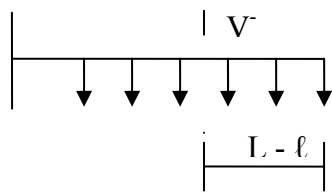
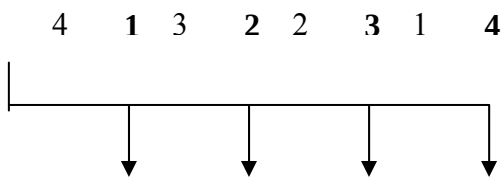


Figura A1.5 – Situação 5 – Elemento A

$$\frac{L - \ell}{L} = 1 - \frac{\ell}{L} = 1 - \left(1 - \sqrt{1 - \frac{\Delta V_{\ell}}{\Delta \cdot K_{V_2}}} \right) = \sqrt{1 - \frac{\Delta V_{\ell}}{\Delta \cdot K_{V_2}}}$$

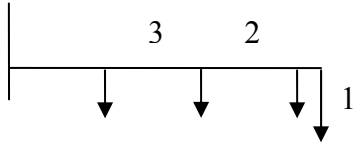
Exemplo prático:



Verifica-se no nó 3:

$$\Delta V_{\ell} = V_{BT} - V_3 = 9$$

Define-se K_{V_2}



$$\Delta V^{\text{ex}} = 9$$

$$\Delta V = \frac{1}{2} \cdot 3 \cdot 3 \cdot 1 + 1 \cdot 3 \cdot 1 = 7,5. K_{V_2} = \frac{9}{7,5} = 1,2$$

$$K^+ = \left[4 - 4 \sqrt{1 - \frac{9}{8 \cdot 1,2}} \right] =$$

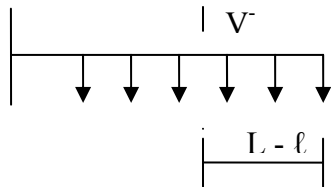
$$\Delta V_f = \frac{1}{2} \cdot 4 \cdot 4 = 8$$

$$K^+ = \left[4 \cdot 4 \sqrt{0,0625} \right] = \left[4 - 4 \cdot 0,25 \right] = \left[3 \right] = 3$$

Logo a Solução existe.

Tem-se:

$$K_t^- = \left[N \sqrt{1 - \frac{\Delta V_{\ell t}}{\Delta V_{ft} K_{V_2}}} \right]$$



Onde: $\Delta V_{\ell} = V_{BT} - V^-$

(a] significa menor valor inteiro mais próximo ao “a”

$$K_{V_2} = + \left(\frac{k^-}{N - k^-}, N - k^- \right)$$

Neste caso:

$$W^- = K^- \frac{I_{\Sigma} \cdot (V_{BT} - \Delta V_f)}{N} \quad (\text{A1.10})$$

A1.4.1.3. Situação 6

A situação 6 ocorre quando existe energia fornecida fora do padrão por sobretensão e por subtensão. É necessário se encontrar dois pontos, o primeira onde a queda de tensão é suficiente para a tensão voltar a um nível adequado e outro onde a queda de tensão torna a tensão abaixo do nível mínimo permitido.

Esta situação é a mais crítica de todas e também a menos comum, entretanto não pode ser descartada de todas as hipóteses. Uma rede secundária nesta situação está mal dimensionada para a carga que está recebendo, e os meios básicos de controle de tensão não são suficientes para restabelecer a tensão a um nível adequado. Claro que se isto ocorre somente na hora de demanda máxima, os meios básicos podem melhorar os níveis de tensão nas outras horas do dia.

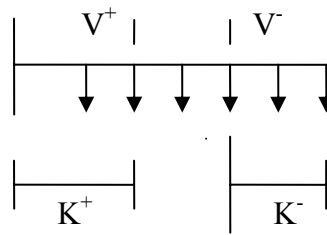


Figura A1.6 – Situação 6 – Elemento A

É necessário definir os dois pontos:

$$K^+ = \left[N - N \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f K_{V_2}}} \right] \quad (A1.11)$$

Onde:

$$K_{V_2} = f \left(\frac{N - k^+}{k^+}, k^+ \right) \quad (A1.12)$$

E a partir da Equação A1.11 calcula-se W_t^+ baseado na energia consumida pelos consumidores a montante do ponto k^+ .

$$k^- = \left[N \sqrt{1 - \frac{V_{BT} - V^-}{\Delta V_f K_{V_2}}} \right] \quad (A1.13)$$

Onde:

$$K_{V_2} = f \left(\frac{k^-}{N - k^-}, N - k^- \right) \quad (A1.14)$$

E a partir da Equação A1.12 calcula-se W_t^- baseado na energia consumida pelos consumidores a jusante do ponto k^- .

No entanto definir diretamente o valor de K_{V_2} é impossível. Isto pode ser realizado através de cálculos iterativos onde se teria primeiramente $K_{V_2} = 1$. Logo, calcula-se k^+ ou k^- e depois na próxima iteração recalcula-se K_{V_2} , k^+ ou k^- e assim sucessivamente.

Este processo é bastante complexo, mas é possível simplificar os cálculos através de alteração de K_{V_2} para K_{V_1} ou considerar $K_{V_2} = 1$.

Agora esta avaliação pode ser estendida para os outros modelos. Levando-se em conta as mesmas características de cada situação, serão definidas apenas os pontos críticos de violação por sobretensão e/ou subtensão (k^+ e k^-).

A1.3.2. Elemento B

Antes de se definir para os modelos compostos, a partir do modelo 5, deve-se definir as fórmulas de quantidade de energia fornecida com níveis de tensão fora do padrão para o Elemento B, em todas as 3 situações possíveis.

A1.4.2.1. Situação 2

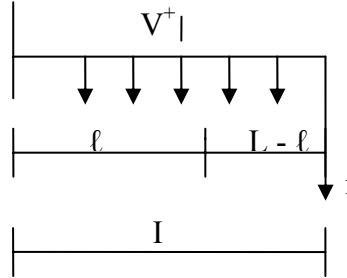


Figura A1.7 – Situação 2 – Elemento B

$$\Delta V_{\ell} = \left\{ \frac{1}{2} I \frac{\ell}{L} \ell R_o + I \frac{(L-\ell)}{L} \ell R_o + i \ell R_o \right\} K v_2$$

Desconsiderando-se $K v_2$:

$$\frac{1}{2} I L R_o \left(\frac{\ell}{L} \right)^2 - (I + i) L R_o \left(\frac{\ell}{L} \right) + \Delta V_{\ell} = 0$$

$$x = \frac{\ell}{L} \quad , \quad a = \frac{1}{2} I L R_o = \Delta V_f'$$

$$b = (I + i) L R_o = I L R_o + i L R_o = 2 \Delta V_f' + \Delta V_f''$$

$$c = \Delta V_{\ell}$$

$$x^2 + \frac{b}{a} x + \frac{c}{a} = 0$$

$$x = \frac{b}{a} - \sqrt{\frac{b^2}{4a^2} - \frac{c}{a}}$$

$$x = \frac{2 \Delta V_f' + \Delta V_f''}{2 \Delta V_f'} - \sqrt{\frac{4 (\Delta V_f')^2 + 4 \Delta V_f' \Delta V_f'' + (\Delta V_f'')^2}{4 (\Delta V_f')^2} - \frac{\Delta V_{\ell}}{\Delta V_f'}}$$

$$= 1 + \frac{\Delta V_f''}{2 \Delta V_f'} - \sqrt{1 + \frac{\Delta V_f''}{\Delta V_f'} + \frac{(\Delta V_f'')^2}{4 (\Delta V_f')^2} - \frac{\Delta V_{\ell}}{\Delta V_f'}}$$

$$\frac{\ell}{L} = \frac{k^+}{N}, \quad k^+ = x.N$$

Então:

$$k^+ = N + N \frac{\Delta V_f''}{2\Delta V_f'} - N \sqrt{1 + \frac{\Delta V_f''}{\Delta V_f'} + \frac{(\Delta V_f'')^2}{4(\Delta V_f')^2} - \frac{\Delta V_\ell}{\Delta V_f'}}$$

Se $I = i$, temos:

$$\Delta V_f'' = 2\Delta V_f'$$

$$k^+ = \left[N + N - N \sqrt{1 + 2 + 1 - \frac{\Delta V_\ell}{\Delta V_f'}} \right]$$

Considerando Kv_2 :

$$k^+ = \left[2N - N \sqrt{4 - \frac{\Delta V_\ell}{\Delta V_f' \cdot Kv_2 \left(\frac{2N - k^+}{k^+}, k^+ \right)}} \right] \quad (A1.15)$$

Então:

$$W^+ = k^+ \frac{I_\Sigma \cdot V_{BT}}{2N} \quad (A1.16)$$

A1.4.2.1. Situação 5

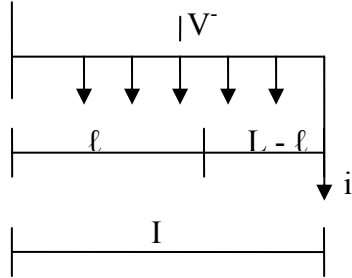


Figura A1.8 – Situação 5 – Elemento B

Desconsiderando-se \$Kv_2\$ temos:

$$k^- = \left[-N \frac{\Delta V_f''}{2\Delta V_f'} + N \sqrt{1 + \frac{\Delta V_f''}{\Delta V_f'} + \frac{(\Delta V_f'')^2}{4(\Delta V_f')^2} - \frac{\Delta V_\ell}{\Delta V_f'}} \right]$$

Se \$I = i\$, então \$\Delta V_f'' = 2\Delta V_f'\$

Considerando-se \$Kv_2\$ temos:

$$k^- = \left[N \sqrt{4 - \frac{V_{BT} - V^-}{\Delta V_f' \cdot Kv_2 \left(\frac{k^-}{N - k^-}, N - k^- \right)}} - N \right] \quad (A1.17)$$

Então:

$$W^- = \left(k^- \frac{I_\Sigma}{2N} + \frac{I_\Sigma}{2} \right) (V_{BT} - \Delta V_f) \quad (A1.18)$$

$$\text{Onde: } \Delta V_f = \Delta V_f \cdot Kv_2 \left(\frac{i}{I} = 1, N \right) \quad (A1.19)$$

A1.3.2.1. Situação

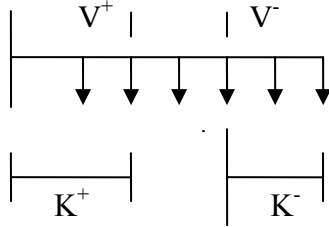


Figura A1.9 – Situação 6 – Elemento B

Nesta situação faz-se apenas a união (soma) dos dois outros casos.

A1.4 – Calculo de EFP para o Modelos de Redes BT

Agora que já foram definidas as equações para o cálculo da EFP nos modelos elementares, e em todas as situações, parte-se para a definição das equações diretamente nos modelos simplificados.

A1.4.1 Modelo 3

Situação 2

$$W^+ = k^+ \frac{I_{\Sigma} \cdot V_{BT}}{N} \quad (A1.20)$$

$$\text{Onde: } k^+ = \left[N - N \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f \cdot K_{V_2}}} \right] \quad (A1.21)$$

Situação 5

$$W^- = k^- \frac{I_{\Sigma} (V_{BT} - V^+)}{\Delta V_f \cdot K_{V_2}} \quad (A1.22)$$

$$\text{Onde: } k^- = \left[N \sqrt{1 - \frac{V_{BT} - V^-}{\Delta V_f \cdot K v_2}} \right] \quad (\text{A1.23})$$

Situação 6

$$W^+ = k^+ \frac{I_\Sigma (V_{BT} - \Delta V_f)}{N} \quad (\text{A1.24})$$

$$W^- = k^- \frac{I_\Sigma (V_{BT} - \Delta V_f)}{N} \quad (\text{A1.25})$$

$$\text{Onde: } k^+ = \left[N - N \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f \cdot K v_2}} \right] \quad (\text{A1.26})$$

$$k^- = \left[N \sqrt{1 - \frac{V_{BT} - V^-}{\Delta V_f \cdot K v_2}} \right] \quad (\text{A1.27})$$

A1.4.2. Modelo 4

Situação 2

$$W^+ = 2k^+ \frac{I_\Sigma V_{BT}}{N} \quad (\text{A1.28})$$

Onde:

$$k^+ = \left[\frac{N}{2} - \frac{N}{2} \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f \cdot K v_2}} \right] \quad (\text{A1.29})$$

Situação 5

$$W^- = 2k^- \frac{I_\Sigma (V_{BT} - \Delta V_f)}{N} \quad (\text{A1.30})$$

Onde:
$$k^- = \left[\frac{N}{2} \sqrt{1 - \frac{V_{BT} - V^-}{\Delta V_f \cdot K_{V_2}}} \right] \quad (A1.31)$$

Situação 6

$$W^+ = 2k^+ \frac{I_\Sigma \cdot V_{BT}}{N} \quad (A1.32)$$

$$W^- = 2k^- \frac{I_\Sigma \cdot (V_{BT} - \Delta V_f)}{N} \quad (A1.33)$$

Onde:

$$k^+ = \left[\frac{N}{2} - \frac{N}{2} \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f \cdot K_{V_2}}} \right] \quad (A1.34)$$

$$k^- = \left[\frac{N}{2} \sqrt{1 - \frac{V_{BT} - V^+}{\Delta V_f \cdot K_{V_2}}} \right] \quad (A1.35)$$

A1.4.3. Modelo 5A

O Modelo 5A, é o primeiro modelo composto de Elemento A e B. Neste caso deve-se considerar duas diferentes situações: Quando o ponto crítico k^+ está dentro do Elemento B (Situação 2A), e quando ele está dentro do Elemento A (Situação 2B).

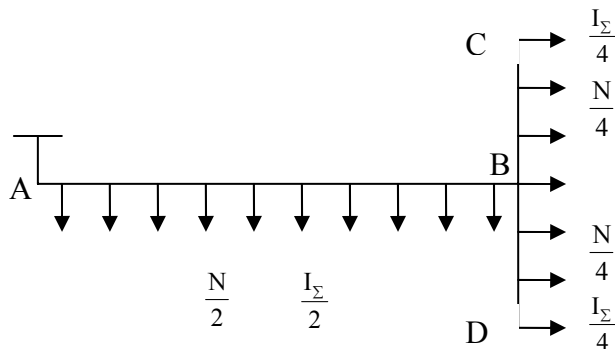


Figura A1.10 – Modelo 5A - EFPF

Neste caso $I = i = \frac{I_{\Sigma}}{2}$

Situação 2A

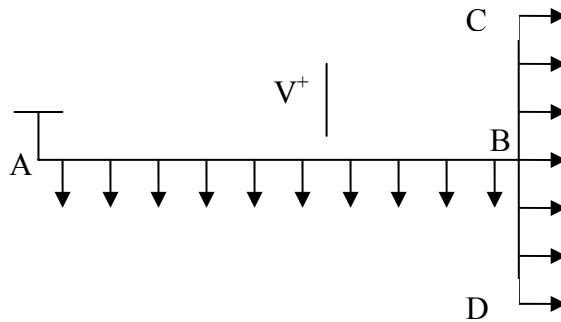


Figura A1.11 – Modelo 5A – Situação 2A

Neste caso $I = i = \frac{I_{\Sigma}}{2}$

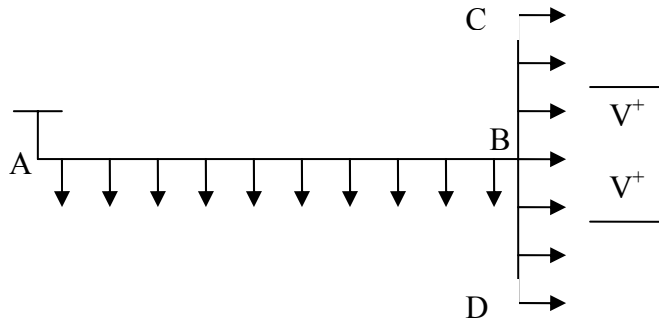
Então:

$$k^+ = 2 \cdot \frac{N_{\Sigma}}{2} - \frac{N_{\Sigma}}{2} \sqrt{4 - \frac{V_{BT} - V^+}{\frac{1}{2} \cdot \frac{I_{\Sigma}}{2} \cdot R_o L_{AB} \cdot K v_2 \left(\frac{N - k^+}{k^+}, k^+ \right)}}$$

$$k^+ = N_{\Sigma} - \frac{N_{\Sigma}}{2} \sqrt{4 - \frac{V_{BT} - V^+}{\frac{1}{4} I_{\Sigma} R_o L K v_2 \left(\frac{N - k^+}{k^+}, k^+ \right)}} \quad (A1.36)$$

$$W^+ = k^+ \frac{I_{\Sigma}}{N} V_{BT} \quad (A1.37)$$

Situação 2B

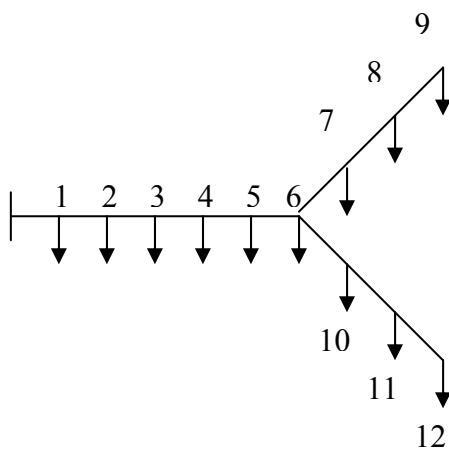


$$\Delta V_{\ell} = \Delta V_{AB} + \Delta V_{B\ell}$$

$$k^+ = \frac{N}{2} + 2 \left[\frac{N}{4} - \frac{N}{4} \sqrt{1 - \frac{V_{B\ell}}{\Delta V_{BC} \cdot K_{V_2}}} \right] = \left[N - \frac{N}{2} \sqrt{1 - \frac{V_{BT} - V^+ - \Delta V_{AB}}{\Delta V_{BC} \cdot K_{V_2}}} \right]$$

$$= \left[N - \frac{N}{2} \sqrt{1 - \frac{V_{BT} - V^+ - \frac{3}{4} I_{\Sigma} L R_o \cdot K_{V_{2(AB)}} \left(1, \frac{N}{2} \right)}{\frac{1}{16} I_{\Sigma} L R_o \cdot K_{V_{2(BC)}}}} \right]$$

Exemplo para teste:



Calcular a EFP para o Modelo dado acima, considerando-se:

$$R_o = 1$$

$$L_o = 1$$

$$\frac{I_z}{N} = 1$$

$$\Delta V_{0-1} = 12.1.1 = 12$$

$$\Delta V_{0-2} = 12.1.1 + 11.1.1 = 23$$

$$\Delta V_{0-3} = 33$$

$$\Delta V_{0-4} = 42$$

$$\Delta V_{0-5} = 50$$

$$\Delta V_{0-6} = 57$$

Verifica-se o “nó” 3

Isto significa que $V_{BT} - V = \Delta V_{0-3} = 33$

Neste caso:

$$K = 2 \cdot \frac{12}{2} - \frac{12}{2} \sqrt{4 - \frac{33}{\frac{1}{4} \cdot 12 \cdot 1 \cdot 6 \cdot K_{V_2} \left(\frac{12-3}{3}, 3 \right)}} = 12 - 6 \sqrt{4 - \frac{33}{18 K_{V_2}(3,3)}}$$

Supondo que $K_{V_2} = 1$

$$K = \left[12 - 6 \sqrt{4 - \frac{33}{18}} \right] = \left[12 - 6 \sqrt{2,167} \right] = \left[12 - 6 \cdot 1,472 \right] = \left[3,2 \right] = 3$$

$$\Delta V_{0-7} = 57 + 3 = 60$$

$$\Delta V_{0-8} = 62$$

$$\Delta V_{0-9} = 63$$

Verifica-se o “nó” 8 e 11

Então $V_{BT} - V = \Delta V_{0-8} = 62$

Então

$$K = 12 - 6 \sqrt{1 - \frac{62 - \frac{3}{4} \cdot 12 \cdot 1 \cdot 6 \cdot K_{V_2}(1,6)}{\frac{1}{16} \cdot 12 \cdot 1 \cdot 6 \cdot K_{V_2}\left(\frac{1}{2}, 2\right)}} =$$

$$= 12 - 6 \sqrt{1 - \frac{62 - 54K_{V_2(AB)}}{4,5K_{V_2(BC)}}} =$$

Supondo:

$$K_{V_2(AB)} = 1 \quad , \quad K_{V_2(BC)} = 1$$

$$K = 12 - 6 \sqrt{1 - \frac{8}{4,5}} = 12 - 6\sqrt{-0,777} \quad \text{Sem Solução}$$

Num processo de cálculo a definição dos valores de K_{V_1} e K_{V_2} é impossível. Por outro lado a utilização de $K_{V_1}=1$ e $K_{V_2}=1$ ou alguns valores médios, por exemplo $K_{V_1}=1,1$ e $K_{V_2}=1,05$ pode provocar grandes erros, especialmente quando o número de nós não é grande. Por isso, para a avaliação do valor de energia consumida com níveis de tensão fora do padrão será necessário utilizar uma abordagem alternativa.

APÊNDICE II

A2. Determinação da Lei de Controle de Tensão na SE pela análise de Casos Extremos

As regras para o controle de tensão na subestação podem ser definidas analisando-se alguns casos críticos sob o ponto de vista do regime de tensão, conforme descrito na metodologia abaixo.

A2.1. Caso 1

Supõe-se que V'_S corresponda ao nível de tensão na subestação, em regime de demanda máxima. Espera-se, portanto, as maiores quedas de tensão.

Para manter-se a tensão dentro dos níveis permissíveis, o tap do transformador da subestação está em uma posição que proporciona um aumento no valor da tensão.

O primeiro transformador de distribuição (transformador para serviços internos e auxiliares da subestação), do alimentador em questão, fornece energia a algumas cargas diretamente do terminal de baixa tensão. Desta forma, a única queda de tensão que deve ser considerada ocorre devido à queda interna no transformador de distribuição ($\Delta V'_{TD}$).

Nestas circunstâncias, é necessário garantir-se que a tensão de fornecimento para as cargas ligadas no transformador não irá ultrapassar o limite máximo permitido.

Supondo-se então que V^+ corresponda à máxima tensão permissível no ponto de consumo, através do uso da equação 44, pode-se chegar à equação 45.

$$V_S' + E_{TD} - \Delta V_{TD}' \leq V^+ \rightarrow V_S' \leq V^+ - E_{TD} + \Delta V_{TD}' \quad (\text{A2.1})$$

A2.2. Caso 2

Sabe-se que V_S'' corresponde ao nível de tensão na subestação, em regime de demanda mínima.

É necessário então garantir, aos consumidores ligados no primeiro transformador com maior aumento de tensão (tap 3), que a tensão de fornecimento não ultrapasse o limite máximo permitido.

Esta condição pode ser entendida da seguinte forma. Os taps dos transformadores de distribuição são definidos de forma a compensar as quedas na rede de média tensão, para o caso de ocorrência de demanda máxima.

Naturalmente, no caso de demanda mínima, as quedas na rede de média tensão são menores do que o aumento de tensão nos transformadores de distribuição.

Este contexto pode fazer com que o nível de tensão, no transformador de distribuição com maior aumento de tensão (e mínima queda na rede de média tensão), ultrapasse o limite permitido.

Neste caso, consideram-se as quedas de tensão devido à distância até o transformador de distribuição ($\Delta V_{MT}''$) e à reatância interna do transformador ($\Delta V_{TD}''$). Sendo V^+ a máxima tensão permissível no ponto de consumo e levando-se em consideração a equação A2.1, obtem-se a equação A2.2.

$$V_S'' + E_{TD} - \Delta V_{MT}'' - \Delta V_{TD}'' \leq V^+ \rightarrow V_S'' \leq V^+ - E_{TD} + \Delta V_{MT}'' + \Delta V_{TD}'' \quad (\text{A2.2})$$

A2.3. Caso 3

Em regime de demanda mínima, deve-se assegurar aos consumidores mais distantes, conectados ao último transformador de distribuição com tap 1, que a tensão de fornecimento não ultrapasse o limite mínimo permitido.

Neste caso, as quedas de tensão a serem consideradas correspondem à distância até o último transformador com tap 1 ($\Delta V_{MT}''$), à reatância interna do transformador ($\Delta V_{TD}''$) e à queda na rede de baixa tensão, desde o transformador de distribuição até os pontos de consumo no interior da instalação do consumidor ($\Delta V_{BT}''$).

Desta forma, com base na equação A2.1, chega-se à equação A2.3.

$$\begin{aligned} V_S'' + E_{TD} - \Delta V_{MT}'' - \Delta V_{TD}'' - \Delta V_{BT}'' &\geq V^- \rightarrow \\ V_S'' &\geq V^- - E_{TD} + \Delta V_{MT}'' + \Delta V_{TD}'' + \Delta V_{BT}'' \end{aligned} \quad (A2.3)$$

A2.4. Análise de um Exemplo

Para realizar-se a análise de um exemplo, supõe-se que, em regime de demanda máxima, $\Delta V_{MT}' = 12\%$; $\Delta V_{TD}' = 3\%$; $\Delta V_{BT}' = 6\%$.

Se $\frac{I_{máx}}{I_{min}} = 3$, então, em regime de demanda mínima, tem-se que

$$\Delta V_{MT}'' = 4\%; \Delta V_{TD}'' = 1\% \text{ e } \Delta V_{BT}'' = 2\% .$$

Realizando-se os cálculos, de acordo com as equações 45, 46 e 47, tem-se que:

- Caso 1: $V_S' \leq 5 - 0 + 3 = 8\%$;
- Caso 2: $V_S'' \leq 5 - 9 + \frac{9}{3} + \frac{3}{3} = 0\%$;
- Caso 3: $V_S'' \geq -5 - 0 + \frac{4,5}{3} + \frac{3}{3} + \frac{6}{3} = -0,5\%$.

São definidos então três pontos, conforme ilustra a Figura A2.1.

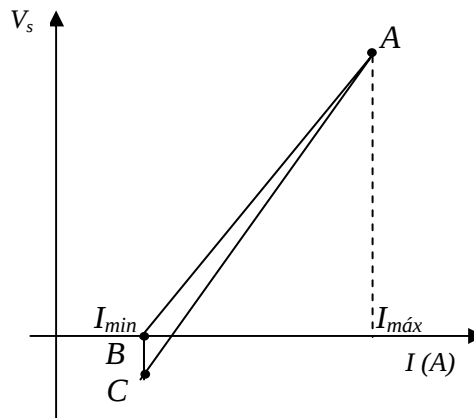


Figura A2.1 - Definição da lei de controle de tensão na SE.

O ponto (A) corresponde ao regime de demanda máxima.

Os dois outros pontos (B e C) correspondem ao regime de demanda mínima.

Assim, qualquer reta, dentro do triângulo ABC, define a lei de controle de tensão para o transformador da subestação, em função de sua carga.

APÊNDICE III

A3. Análise de Sistemas de Distribuição – Controle de Tensão – Códigos Fonte

O Software ASD-CT foi totalmente desenvolvido em Java, utilizando os conceitos de orientação a objetos bem como padrões de projetos, já bastante difundidos, que visam a coesão dos módulos, entretanto mantendo o baixo acoplamento. A modularidade facilita a manutenção bem como torna mais simples a expansão do Software.

A plataforma ASD é um Software para análise de sistemas de distribuição, onde vários algoritmos são “plugáveis”. A plataforma fornece aos algoritmos formas de manipular e executar cálculos baseados em informações contidas em cada nó, bem como faz a leitura dos bancos de dados da concessionária. Esta plataforma vem sendo desenvolvida há 4 anos, e o seu código fonte além de ser extenso, foge um pouco do escopo deste trabalho.

Serão demonstrados os códigos fontes dos dois algoritmos que utilizam a plataforma, o de Ajuste de Tapes, a determinação da Lei de Controle de Tensão na SE, e o algoritmo para cálculo da EFP. Também serão demonstrados os códigos fontes dos Modelos Representativos de Redes de Baixa Tensão, onde estão contidos os cálculos referentes a estes Modelos, como queda de tensão até o centro de carga, EFP, ...

A3.1 – Algoritmo para Ajuste de Tapes (br.com.mega.ppgee.plugins.AjusteTaps)

```
package br.com.mega.ppgee.plugins;

import br.com.mega.ppgee.interfaces.Algoritmo;
import br.com.mega.ppgee.interfaces.BarraProgresso;
import br.com.mega.ppgee.core.*;
import br.com.mega.ppgee.beans.Repositorio;
import br.com.mega.ppgee.beans.Context;
import br.com.mega.ppgee.beans.CalculoDemandasBean;
import br.com.mega.ppgee.swing.BarraProgressoEngodo;
```

```

import br.com.mega.ppgee.swing.DialogParams;

import java.util.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 28/04/2005
 * Time: 09:47:53
 * To change this template use File | Settings | File Templates.
 */
public class AjusteTaps implements Algoritmo {

    private static final String OPCA_O_CALCULAR = "Calcular Valores DMax e DMin";
    private static final String OPCA_O_FIXO = "Utilizar horas fixas de DMax e DMin";
    private String valorTipo;
    private String valorHoraDmin;
    private String valorHoraDmax;
    private String valorPeriodo;
    private double valorTensaoAL;
    private String tipoDiaDMin;
    private String tipoDiaDMax;
    public static final String CONFIGURACAO_DMAX = "Configuração na Demanda Máxima";
    public static final String CONFIGURACAO_DMIN = "Configuração na Demanda Mínima";
    public static final String QUEDA_INTERNA_TRAFO = "Queda de tensão interna do transformador";

    public AjusteTaps() {
    }

    private Map params = new HashMap();

    private void criaParams(String[] alimentadores) {
        List l = new ArrayList();
        Repositorio rep = Context.getContext().getRepositorio();

        Parametro p = new Parametro();
        p.setNome("Elevação de Tensão Máxima (%)" );
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setValor(new Double(- Constantes.ELEVACAO_MAXIMA_CONSUMIDOR));
        l.add(p);

        p = new Parametro();
        p.setNome("Queda de Tensão Máxima (%)" );
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setValor(new Double(Constantes.QUEDA_MAXIMA_CONSUMIDOR));
        l.add(p);

        p = new Parametro();
        p.setNome("Queda de Tensão Máxima Permissível na rede de BT (%)" );
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setValor(new Double(Constantes.QUEDA_PERMISSIVEL_BT));
        l.add(p);

        p = new Parametro();
    }

```

```

(%)");
    p.setNome("Queda de Tensão Máxima interna do transf. de distr.
    p.setTipo(Parametro.TIPO.DOUBLE);
    p.setValor(new Double(Constants.QUEDA_TD));
    l.add(p);

    params.put("Parâmetros de Queda de Tensão", l);

    l = new ArrayList();
    p = new Parametro();
    p.setNome("Modo de Execução");
    p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
    List opcoes = new ArrayList(2);
    opcoes.add(OPCAO_CALCULAR);
    opcoes.add(OPCAO_FIXO);
    p.setOpcoes(opcoes);
    l.add(p);

    p = new Parametro();
    p.setNome("Hora de demanda mínima");
    p.setValor("3");
    p.setTipo(Parametro.TIPO.STRING);
    l.add(p);

    p = new Parametro();
    p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
    p.setNome("Dia da Dem. Mínima");
    List x = new ArrayList(3);
    x.add(AtualizarCargasCurvasTipicas.V_DIA_UTIL);
    x.add(AtualizarCargasCurvasTipicas.V_SABADO);
    x.add(AtualizarCargasCurvasTipicas.V_DOMINGO);
    p.setOpcoes(x);
    l.add(p);

    p = new Parametro();
    p.setNome("Hora de demanda máxima");
    p.setValor("19");
    p.setTipo(Parametro.TIPO.STRING);
    l.add(p);

    p = new Parametro();
    p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
    p.setNome("Dia da Dem. Máxima");
    x = new ArrayList(3);
    x.add(AtualizarCargasCurvasTipicas.V_DIA_UTIL);
    x.add(AtualizarCargasCurvasTipicas.V_SABADO);
    x.add(AtualizarCargasCurvasTipicas.V_DOMINGO);
    p.setOpcoes(x);
    l.add(p);

    params.put("Ajuste de Taps", l);

    l = new ArrayList();

    p = new Parametro();
    p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
    p.setNome("Período");
    p.setOpcoes(rep.getPeriodos(alimentadores));
    l.add(p);

    p = new Parametro();

```

```

        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setNome("Tensão Nominal do Alimentador (kV)");
        p.setValor(new Double(13.8));
        l.add(p);

        params.put("Fluxo de Potência", l);
    }

    private void initParams() {
        List l = (List) params.get("Ajuste de Taps");

        Parametro p = (Parametro) l.get(0);
        valorTipo = (String) p.getValor();

        p = (Parametro) l.get(1);
        valorHoraDmin = (String) p.getValor();

        p = (Parametro) l.get(2);
        tipoDiaDMin = (String) p.getValor();

        p = (Parametro) l.get(3);
        valorHoraDmax = (String) p.getValor();

        p = (Parametro) l.get(4);
        tipoDiaDMax = (String) p.getValor();

        l = (List) params.get("Fluxo de Potência");

        p = (Parametro) l.get(0);
        valorPeriodo = (String) p.getValor();

        p = (Parametro) l.get(1);
        valorTensaoAL = ((Double) p.getValor()).doubleValue();

        l = (List) params.get("Parâmetros de Queda de Tensão");

        p = (Parametro) l.get(0);
        Constantes.ELEVACAO_MAXIMA_CONSUMIDOR = - ((Double)
p.getValor()).doubleValue();

        p = (Parametro) l.get(1);
        Constantes.QUEDA_MAXIMA_CONSUMIDOR = ((Double)
p.getValor()).doubleValue();

        p = (Parametro) l.get(2);
        Constantes.QUEDA_PERMISSIVEL_BT = ((Double)
p.getValor()).doubleValue();

        p = (Parametro) l.get(3);
        Constantes.QUEDA_TD = ((Double) p.getValor()).doubleValue();
    }

    private Algoritmo algoritmoFluxo = new FluxoPotencia2();
    private Algoritmo algoritmoCurvasTipicas = new
AtualizarCargasCurvasTipicas();

    public Map execute(Grafo grafo, BarraProgresso barra) {
        initParams();
        CalculoDemandasBean bean = Context.getContext().getDemandasBean();

```

```

        long millis = System.currentTimeMillis();
        Map result = new TreeMap();
        barra.setMaximo(24 + 2);
        barra.setValor(0);
        if (valorTipo.equals(OPCAO_CALCULAR)) {
            CalculoDemandasBean.ConfiguracaoAlimentador[] res =
bean.calculaHorasDemandas(grafo, valorTensaoAL, valorPeriodo, barra);
            minima = res[0];
            maxima = res[1];
        } else {
            maxima = new CalculoDemandasBean.ConfiguracaoAlimentador();
            maxima.tipoDia = tipoDiaDMax;
            maxima.hora = Integer.parseInt(valorHoraDmax);
            maxima.valorPeriodo = valorPeriodo;
            maxima.tensaoAL = valorTensaoAL;

            minima = new CalculoDemandasBean.ConfiguracaoAlimentador();
            minima.tipoDia = tipoDiaDMin;
            minima.hora = Integer.parseInt(valorHoraDmin);
            minima.valorPeriodo = valorPeriodo;
            minima.tensaoAL = valorTensaoAL;
            barra.setValor(24);
        }
        Vertice source = grafo.getSubestacao();
        AtributoDerivado att = new AtributoDerivado(CONFIGURACAO_DMAX,
maxima, "", false);
        source.getComponente().setValorAtributo(CONFIGURACAO_DMAX, att);
        att = new AtributoDerivado(CONFIGURACAO_DMIN, minima, "", false);
        source.getComponente().setValorAtributo(CONFIGURACAO_DMIN, att);
        int[] taps = configuraTaps(grafo);
        barra.setValor(24 + 2);
        millis = System.currentTimeMillis() - millis;
        result.put(taps[0] + " transformadores ajustados para o tap 1",
null);
        result.put(taps[1] + " transformadores ajustados para o tap 2",
null);
        result.put(taps[2] + " transformadores ajustados para o tap 3",
null);
        result.put("Tempo de Execução", ((double) millis)/1000.0 + " s");
        return result; //To change body of implemented methods use File |
Settings | File Templates.
    }

```

```

    private int[] configuraTaps(Grafo grafo) {
        Repositorio rep = Context.getContext().getRepository();
        int[] nTaps = new int[] { 0, 0, 0};
        BarraProgressoEngodo engodo = new BarraProgressoEngodo();
        algoritmoFluxo.getParams(grafo);
        Map params = algoritmoCurvasTipicas.getParams(grafo);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 0,
valorPeriodo);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 1,
maxima.tipoDia);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 2,
Integer.toString(maxima.hora));
        algoritmoCurvasTipicas.execute(grafo, engodo);
        algoritmoFluxo.execute(grafo, engodo);
        Iterator it = grafo.getVertices().values().iterator();
        while (it.hasNext()) {
            try {
                Vertice v = (Vertice) it.next();

```

```

        if (v.getComponente() instanceof Transformador) {
            Transformador trafo = (Transformador)
v.getComponente();
            double tensao = 0.0;
            tensao = ((Double)
trafo.getValorAtributo(FluxoPotencia.MODULO_TENSAO).getValor()).doubleValue
());
            AtributoDerivado att =
trafo.getValorAtributo(CentroDeCarga.DEMANDA_MAXIMA);
            if (att == null) {
                double[] d =
rep.getDemandaMaximaTrafo(trafo.getVertice().getId());
                if (d == null) {
                    att = new
AtributoDerivado(CentroDeCarga.DEMANDA_MAXIMA, new Double(0), "kW", true);
trafo.setValorAtributo(CentroDeCarga.DEMANDA_MAXIMA, att);
                } else {
                    att = new
AtributoDerivado(CentroDeCarga.DEMANDA_MAXIMA, new Double(d[0]), "kW",
true);
trafo.setValorAtributo(CentroDeCarga.DEMANDA_MAXIMA, att);
                }
            }
            double demandaMaxima = ((Double)
att.getValor()).doubleValue();
            double queda = 0;
            if (demandaMaxima != 0) {
                CentroDeCarga.calculaCCTransformador(trafo,
Constantes.QUEDA_PERMISSIVEL_BT, demandaMaxima,
trafo.getNumConsumidores());
                double S = trafo.getDemanda();
                double quedaCC = ((Double)
trafo.getValorAtributo(CentroDeCarga.QUEDA_CC).getValor()).doubleValue();
                double quedaInternaTrafo = (S / demandaMaxima) *
Constantes.QUEDA_TD;
                queda = (((valorTensaoAL - tensao) / valorTensaoAL)
* 100) + (quedaInternaTrafo) + quedaCC;
            }
            if (queda > 10) {
                nTaps[2]++;
                trafo.setTap(3);
            } else if (queda > 5) {
                nTaps[1]++;
                trafo.setTap(2);
            } else {
                nTaps[0]++;
                trafo.setTap(1);
            }
        }
    } catch (Exception e) { }
}
return nTaps;
}

private CalculoDemandasBean.ConfiguracaoAlimentador maxima;
private CalculoDemandasBean.ConfiguracaoAlimentador minima;

public static double getQuedaInternaTrafo(Transformador trafo) {
    Repositorio rep = Context.getContext().getRepositorio();

```

```

        AtributoDerivado att =
trafo.getValorAtributo(CentroDeCarga.DEMANDA_MAXIMA);
        if (att == null) {
            double[] d =
rep.getDemandaMaximaTrafo(trafo.getVertice().getId());
            if (d == null) {
                att = new AtributoDerivado(CentroDeCarga.DEMANDA_MAXIMA,
new Double(0), "kW", true);
                trafo.setValorAtributo(CentroDeCarga.DEMANDA_MAXIMA, att);
            } else {
                att = new AtributoDerivado(CentroDeCarga.DEMANDA_MAXIMA,
new Double(d[0]), "kW", true);
                trafo.setValorAtributo(CentroDeCarga.DEMANDA_MAXIMA, att);
            }
        }
        double demandaMaxima = ((Double) att.getValor()).doubleValue();
        double S = trafo.getDemanda();
        double quedaInternaTrafo = (S / demandaMaxima) *
Constantes.QUEDA_TD;
        return quedaInternaTrafo;
    }

    public Map getParams(Grafo grafo) {
        criaParams(grafo.getIdentificadorGrafo());
        return params;
    }

    public Algoritmo getNext() {
        return null;
    }

    public boolean isParamsPadrao() {
        return true;
    }

    public void trataTelaParametros(DialogParams dialog) {
    }

    public boolean isPrintable() {
        return false; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String[] getPrintResults() {
        return new String[0]; //To change body of implemented methods use
File | Settings | File Templates.
    }
}

```

A3.2 Algoritmo para ajuste da Lei de Controle de Tensão na SE (br.com.mega.ppgee.plugins.CentrodeCarga)

```

package br.com.mega.ppgee.plugins;

import br.com.mega.ppgee.interfaces.Algoritmo;

```

```

import br.com.mega.ppgee.interfaces.BarraProgresso;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.core.*;
import br.com.mega.ppgee.swing.BarraProgressoEngodo;
import br.com.mega.ppgee.swing.DialogParams;
import br.com.mega.ppgee.chart.DefinicaoChart;
import br.com.mega.ppgee.excecoes.ExcecaoAlgoritmo;
import br.com.mega.ppgee.beans.Repositorio;
import br.com.mega.ppgee.beans.Context;
import java.util.*;
import java.text.DecimalFormat;
import java.text.NumberFormat;
import java.awt.geom.Point2D;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 22/03/2005
 * Time: 17:26:20
 * To change this template use File | Settings | File Templates.
 */
public class CentroDeCarga implements Algoritmo {

    public static final String DEMANDA_MAXIMA = "Demanda Máxima";
    private static final String ALFA = "Alfa";
    public static final String QUEDA_CC = "Queda de Tensão no Centro de
Carga";
    public static final String PARAM_PERIODO = "Período (mm/aaaa)";
    private static final String TENSAO_AL = "Tensão na saída do Alimentador
(kv)";
    private double tensaoAL = 13.8;
    private String periodo;
    private String tipoDia;
    private Algoritmo algoritmoFluxo = new FluxoPotencia2();
    private Algoritmo algoritmoCurvasTipicas = new
AtualizarCargasCurvasTipicas();
    private static final String QUEDA_FIM_BT = "Queda de Tensão no Final da
Rede BT";

    public CentroDeCarga() {
    }

    private void criaParams(String[] als) {
        params = new TreeMap();
        Parametro p = null;
        List l = new ArrayList(3);

        p = new Parametro();
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setNome(TENSAO_AL);
        p.setValor(new Double(13.8));
        l.add(p);

        Repositorio rep = Context.getContext().getRepositorio();
        p = new Parametro();
        p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
        p.setNome(PARAM_PERIODO);
        p.setOpcoes(rep.getPeriodos(als));
        l.add(p);

        p = new Parametro();

```

```

        p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
        p.setNome(AtualizarCargasCurvasTipicas.PARAM_DIA);
        List x = new ArrayList(3);
        x.add(AtualizarCargasCurvasTipicas.V_DIA_UTIL);
        x.add(AtualizarCargasCurvasTipicas.V_SABADO);
        x.add(AtualizarCargasCurvasTipicas.V_DOMINGO);
        p.setOpcoes(x);
        l.add(p);

        params.put("Fluxo de Potência", 1);
        l = new ArrayList(2);

        p = new Parametro();
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setNome("Queda de Tensão máxima permissível BT (%)");
        p.setValor(new Double(5.0));
        l.add(p);

        p = new Parametro();
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setNome("Queda de Tensão no TD (%)");
        p.setValor(new Double(2.5));
        l.add(p);

        params.put("Centro de Carga", 1);
    }

    private double quedaTensaoCC_AL = 0.0;
    private double somaPotencia = 0.0;

    private static DecimalFormat nf = (DecimalFormat)
    NumberFormat.getInstance(new Locale("pt", "BR"));
    static {
        nf.applyPattern("0.000");
    }

    public Map execute(Grafo grafo, BarraProgresso barra) {
        initParams();
        long millis = System.currentTimeMillis();
        Map result = new TreeMap();
        BarraProgressoEngodo engodo = new BarraProgressoEngodo();
        barra.setMaximo(24);
        barra.setValor(0);
        algoritmoFluxo.getParams(grafo);
        int nAjustados = 0;
        List quedasCGAl = new ArrayList(24);
        List correntesAl = new ArrayList(24);
        try {
            for (int i = 0; i < 24; i++) {
                Map params = algoritmoCurvasTipicas.getParams(grafo);
                setValor(params, "Ajuste de Cargas", 0, periodo);
                setValor(params, "Ajuste de Cargas", 1, tipoDia);
                setValor(params, "Ajuste de Cargas", 2,
Integer.toString(i));
                algoritmoCurvasTipicas.execute(grafo, engodo);
                algoritmoFluxo.execute(grafo, engodo);
                double[] res = calculaCentrodeCarga(grafo);
                quedasCGAl.add(new Double(res[0]));
                correntesAl.add(new Double(res[1]));
                nAjustados = (int) res[2];
            }
        }
    }

```

```

        barra.setValor(i + 1);
    }
} catch (Exception e) {
    e.printStackTrace();
    result.put(e.getMessage(), null);
    return result;
}
}
milis = System.currentTimeMillis() - milis;
if (nAjustados > 0) {
    double[] as = calculaA1A2(quedasCGA1, correntesA1);
    if (as[1] >= 0) {
        result.put("Elevação da Tensão em função da Corrente: f(I)
-> Et = "
                    + nf.format(as[0]) + "I + " + nf.format(as[1]),
null);
    } else {
        result.put("Elevação da Tensão em função da Corrente: f(I)
-> Et = "
                    + nf.format(as[0]) + "I - " + nf.format(-as[1]),
null);
    }
    result.put(nAjustados + " transformadores ajustados.", null);
    Object[][] chart = new Object[24][3];
    Object[][] chart2 = new Object[6][3];
    double pulo = ((maxI - minI) / 24);
    int j = 0;
    for (int i = 0; i < 24; i++) {
        chart[i][1] = "Elevação de Tensão";
        chart[i][0] = new Integer(i);
        chart[i][2] = quedasCGA1.get(i);

        if ((i % 4) == 0) {
            chart2[j][1] = "Elevação da Tensão";
            double corrente = minI + (pulo * i);
            chart2[j][0] = nf.format(corrente);
            chart2[j][2] = new Double((as[0] * corrente) + as[1]);
            j++;
        }
    }
    //definição do gráfico da elevação em função do tempo
    DefinicaoChart def = new DefinicaoChart();
    def.setNome("Elevação da Tensão em Função do Tempo");
    def.setNomeEixoX("Hora");
    def.setGrafico3d(true);
    def.setNomeEixoY("Elevação %");
    def.setTipo(DefinicaoChart.LINHAS);
    def.setDados(chart);
    result.put("Et = f(t)", def);

    //definição do gráfico da elevação em função da corrente
    def = new DefinicaoChart();
    def.setNome("Elevação da Tensão em Função da Corrente");
    def.setNomeEixoX("Corrente (A)");
    def.setGrafico3d(false);
    def.setNomeEixoY("Elevação %");
    def.setTipo(DefinicaoChart.LINHAS);
    def.setDados(chart2);
    result.put("Et = f(I)", def);
    Point2D p1 = new Point2D.Double(minI * 13.8, (as[0] * minI) +
as[1]);

```

```

        Point2D p2 = new Point2D.Double(maxI * 13.8, (as[0] * maxI) +
as[1]);
        Subestacao subestacao = (Subestacao)
grafo.getSubestacao().getComponente();
        subestacao.setLeiControleTensao(new
Subestacao.LeiControleTensao(p1, p2));
    } else {
        result.put("Não foi possível encontrar informações de carga
sobre os transformadores.", null);
    }
    result.put("Tempo de Execução", ((double) millis)/1000.0 + " s");
    return result;
}

double minI = Double.MAX_VALUE;
double maxI = 0;

private double[] calculaA1A2(List elevacao, List correntes) {
    minI = Double.MAX_VALUE;
    maxI = 0;
    double somI = 0;
    double somI2 = 0;
    double somE = 0;
    double somIE = 0;
    double minE = Double.MAX_VALUE;
    for (int i = 0; i < 24; i++) {
        Double cor = (Double) correntes.get(i);
        somI += cor.doubleValue();
        somI2 += (cor.doubleValue() * cor.doubleValue());
        Double el = (Double) elevacao.get(i);
        somE += el.doubleValue();
        somIE += (el.doubleValue() * cor.doubleValue());
        if (cor.doubleValue() < minI) {
            minI = cor.doubleValue();
            minE = el.doubleValue();
        }
        if (cor.doubleValue() > maxI) {
            maxI = cor.doubleValue();
        }
    }
    double a1 = ((somE * somI) - (24 * somIE)) / ((somI * somI) - (24 *
somI2));
    double a2 = minE - a1*minI;
    return new double[] { a1, a2 };
}

private double[] calculaCentrodeCarga(Grafo grafo) {
    quedaTensaoCC_AL = 0;
    somaPotencia = 0;
    Repositorio rep = Context.getContext().getRepositorio();
    Iterator it = grafo.getVertices().values().iterator();
    List trafos = new LinkedList();
    while (it.hasNext()) {
        Vertice v = (Vertice) it.next();
        if (v.getComponente() instanceof Transformador) {
            trafos.add(v.getComponente());
        }
    }
    it = trafos.iterator();
    int ajustados = 0;
    while (it.hasNext()) {

```

```

        Transformador t = (Transformador) it.next();
        AtributoDerivado dMax = t.getValorAtributo(DEMANDA_MAXIMA);
        if (dMax != null) {
            double demanda = ((Double) dMax.getValor()).doubleValue();
            calculaCCTransformador(t, Constantes.QUEDA_PERMISSIVEL_BT,
demanda, t.getNumConsumidores());
            calculaCCAlimentador(t);
            ajustados++;
        } else {
            double[] d =
rep.getDemandaMaximaTrafo(t.getVertice().getId());
            if (d != null) {
                AtributoDerivado at = new
AtributoDerivado(DEMANDA_MAXIMA, new Double(d[0]), "KW", true);
                t.getAtributos().put(DEMANDA_MAXIMA, at);
                t.setNumConsumidores((long) d[1]);
                calculaCCTransformador(t,
Constantes.QUEDA_PERMISSIVEL_BT, d[0], (long) d[1]);
                calculaCCAlimentador(t);
                ajustados++;
            }
        }
    }
    Subestacao sub = (Subestacao)
grafo.getSubestacao().getComponente();
    double corrente = ((Double)
sub.getValorAtributo(FluxoPotencia.CORRENTE).getValor()).doubleValue();
    return new double[] { (quedaTensaoCC_AL / somaPotencia), corrente,
ajustados };
}

private void initParams() {
    List l = (List) params.get("Fluxo de Potência");

    Parametro p = (Parametro) l.get(0);
    Double d = (Double) p.getValor();
    this.tensaoAL = d.doubleValue();

    p = (Parametro) l.get(1);
    this.periodo = (String) p.getValor();

    p = (Parametro) l.get(2);
    this.tipoDia = (String) p.getValor();

    l = (List) params.get("Centro de Carga");

    p = (Parametro) l.get(0);
    d = (Double) p.getValor();
    Constantes.QUEDA_PERMISSIVEL_BT = d.doubleValue();

    p = (Parametro) l.get(1);
    Constantes.QUEDA_TD = ((Double) p.getValor()).doubleValue();
}

private int calculaCCAlimentador(Transformador t) {
    int tap = 1;
    try {
        double S = t.getDemanda();
        somaPotencia += S;
        double demandaMaxima = ((Double)
t.getValorAtributo(DEMANDA_MAXIMA).getValor()).longValue();

```

```

        if (demandaMaxima == 0.0) return tap;
        double quedaCC = ((Double)
t.getValorAtributo(QUEDA_CC).getValor()).doubleValue();
        double tensao = ((Double)
t.getValorAtributo(FluxoPotencia.MODULO_TENSAO).getValor()).doubleValue();
        double quedaMT = (1 - (tensao / tensaoAL)) * 100;
        double quedaTD = Constantes.QUEDA_TD * (S / demandaMaxima);
        double d = (quedaMT + quedaTD + quedaCC);

        int valorTap = t.getTap();
        d = d - ((valorTap - 1) * 5);
        System.out.println("Queda Total: " + d);
        quedaTensaoCC_AL += (d * S);
        return tap;
    } catch (Exception e) {
        //throw new ExcecaoAlgoritmo("O Fluxo de potência ainda não foi
executado.");
        return 1;
    }
}

public static void calculaCCTransformador(Transformador t, double
deltaVBT, double demandaMaxima, long numConsumidores) {
    t.setNumConsumidores(numConsumidores);
    ModeloRedeSecundaria m = t.getModelo();
    m.setDeltaVPerm(deltaVBT);
    m.setnConsumidores(numConsumidores);
    m.setsMax(demandaMaxima);
    AtributoDerivado att = new AtributoDerivado(QUEDA_CC, new
Double(t.getModelo().getDeltaVCC(t.getDemanda())), "%", true);
    t.setValorAtributo(QUEDA_CC, att);

    att = new AtributoDerivado(QUEDA_FIM_BT, new
Double(t.getModelo().getDeltaV(t.getDemanda())), "%", true);
    t.setValorAtributo(QUEDA_FIM_BT, att);
}

/*private static double getDeltaVCC(Transformador t, double deltaVBT) {
    double alfa = ((Double)
t.getValorAtributo(ALFA).getValor()).doubleValue();
    double demandaMaxima = ((Double)
t.getValorAtributo(DEMANDA_MAXIMA).getValor()).doubleValue();
    double S = Math.sqrt(Math.pow(t.getDemandaAtivaA() +
t.getDemandaAtivaB() + t.getDemandaAtivaC(), 2.0) +
Math.pow(t.getDemandaReativaA() + t.getDemandaReativaB() +
t.getDemandaReativaC(), 2.0));
    return alfa * deltaVBT * (S / demandaMaxima);
}

private static double getDeltaVFim(Transformador t, double deltaVBT) {
    double demandaMaxima = ((Double)
t.getValorAtributo(DEMANDA_MAXIMA).getValor()).doubleValue();
    double S = Math.sqrt(Math.pow(t.getDemandaAtivaA() +
t.getDemandaAtivaB() + t.getDemandaAtivaC(), 2.0) +
Math.pow(t.getDemandaReativaA() + t.getDemandaReativaB() +
t.getDemandaReativaC(), 2.0));
    return deltaVBT * (S / demandaMaxima);
} */

private static double getAlfa(long nConsumidores) {
    int deltaV = 0;

```

```

        for (int i = 1; i <= nConsumidores; i++) {
            deltaV += i;
        }
        long preVc = 0;
        long anterior = 0;
        for (long n = nConsumidores; n > 0; n--) {
            preVc += (n + anterior);
            anterior = (n + anterior);
        }
        double deltaC = ((double) preVc) / ((double) nConsumidores);
        return deltaC / (double) deltaV;
    }

    private Map params = new HashMap();
    public Map getParams(Grafo grafo) {
        criaParams(grafo.getIdentificadorGrafo());
        return params;
    }

    public Algoritmo getNext() {
        return null;
    }

    public boolean isParamsPadrao() {
        return true;
    }

    public void trataTelaParametros(DialogParams dialog) {
    }

    public boolean isPrintable() {
        return false; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String[] getPrintResults() {
        return new String[0]; //To change body of implemented methods use
File | Settings | File Templates.
    }

    public static void setValor(Map params, String lista, int posicao,
Object valor) {
        List l = (List) params.get(lista);
        Parametro p = (Parametro) l.get(posicao);
        p.setValor(valor);
    }
}

```

A3.3 – Algoritmo para cálculo da EFPF

(br.com.mega.ppgee.plugins.EnergiaFornecidaForaDoPadrao)

```

package br.com.mega.ppgee.plugins;

import br.com.mega.ppgee.interfaces.*;
import br.com.mega.ppgee.core.*;
import br.com.mega.ppgee.swing.DialogParams;
import br.com.mega.ppgee.swing.BarraProgressoEngodo;
import br.com.mega.ppgee.chart.DefinicaoChart;
import br.com.mega.ppgee.beans.Repositorio;
import br.com.mega.ppgee.beans.Context;
import br.com.mega.ppgee.util.Navigator;
import br.com.mega.ppgee.util.SemAcaoPreFixada;
import java.util.*;
import java.text.DecimalFormat;
import java.text.NumberFormat;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 15/09/2005
 * Time: 11:37:04
 * To change this template use File | Settings | File Templates.
 */
public class EnergiaFornecidaForaDoPadrao implements Algoritmo {

    private Algoritmo algoritmoFluxo = new FluxoPotencia2();
    private Algoritmo algoritmoCurvasTipicas = new
AtualizarCargasCurvasTipicas();
    public static final String PARAM_PERIODO = "Período (mm/aaaa)";
    private String valorPeriodo;

    private void initParams() {
        List l = (List) params.get("Curva de Carga");
        Parametro p = (Parametro) l.get(0);
        valorPeriodo = (String) p.getValor();

        l = (List) params.get("Parâmetros de Queda de Tensão");

        p = (Parametro) l.get(0);
        Constantes.ELEVACAO_MAXIMA_CONSUMIDOR = - ((Double)
p.getValor()).doubleValue();

        p = (Parametro) l.get(1);
        Constantes.QUEDA_MAXIMA_CONSUMIDOR = ((Double)
p.getValor()).doubleValue();

        p = (Parametro) l.get(2);
        Constantes.QUEDA_PERMISSIVEL_BT = ((Double)
p.getValor()).doubleValue();

        p = (Parametro) l.get(3);
        Constantes.QUEDA_TD = ((Double) p.getValor()).doubleValue();
    }

    private static DecimalFormat nf = (DecimalFormat)
NumberFormat.getInstance(new Locale("pt", "BR"));
    static {
        nf.applyPattern("0.000");
    }

    private static String dia = "Dia Útil";
    private static int hora = 0;

```

```

public Map execute(Grafo grafo, BarraProgresso barra) {
    initParams();
    long millis = System.currentTimeMillis();
    Map result = new TreeMap();
    barra.setMaximo(24 * 3);
    barra.setValor(0);
    Object[][] chart = new Object[24*3][3];
    Object[][] chart2 = new Object[24*3][3];
    double effpTotal = 0.0;
    double energiaTotal = 0.0;
    Repositorio rep = Context.getContext().getRepository();
    Subestacao sub = (Subestacao)
grafo.getSubestacao().getComponente();
    int[] numTiposDia = rep.getNumTiposDia(valorPeriodo);
    AlgoritmoEFP calculaEFP = new
AlgoritmoEFP(grafo.getTensaoNominalAl());

    //dia útil
    BarraProgressoEngodo engodo = new BarraProgressoEngodo();
    dia = "Dia Útil";
    for (int i = 0; i < 24; i++) {
        hora = i;
        algoritmoFluxo.getParams(grafo);
        Map params = algoritmoCurvasTipicas.getParams(grafo);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 0,
valorPeriodo);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 1, "Dia
Útil");
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 2,
Integer.toString(i));
        algoritmoCurvasTipicas.execute(grafo, engodo);
        algoritmoFluxo.execute(grafo, engodo);
        double effp = ((Double)
Navigator.percorreArvore(grafo.getSubestacao(), new Double(0),
SemAcaoPreFixada.INSTANCE, calculaEFP)).doubleValue();
        chart[i][1] = "Dia Útil";
        chart[i][0] = new Integer(i);
        chart[i][2] = new Double(effp);
        double P = ((Double)
sub.getValorAtributo(FluxoPotencia2.POTENCIA_ATIVA).getValor()).doubleValue
());
        double Q = ((Double)
sub.getValorAtributo(FluxoPotencia2.POTENCIA_REATIVA).getValor()).doubleVal
ue());
        double s = Math.sqrt((P * P) + (Q * Q));
        chart2[i][1] = "Dia Útil";
        chart2[i][0] = new Integer(i);
        chart2[i][2] = new Double((effp / s) * 100);

        effpTotal += (effp * numTiposDia[0]);
        energiaTotal += (s * numTiposDia[0]);
        barra.setValor(barra.getValor() + 1);
    }

    //sábado
    dia = "Sábado";
    for (int i = 0; i < 24; i++) {
        hora = i;
        algoritmoFluxo.getParams(grafo);

```

```

        Map params = algoritmoCurvasTipicas.getParams(grafo);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 0,
valorPeriodo);
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 1,
"Sábado");
        CentroDeCarga.setValor(params, "Ajuste de Cargas", 2,
Integer.toString(i));
        algoritmoCurvasTipicas.execute(grafo, engodo);
        algoritmoFluxo.execute(grafo, engodo);
        double effp = ((Double)
Navigator.percorreArvore(grafo.getSubestacao(), new Double(0),
SemAcaoPreFixada.INSTANCE, calculaEFPF).doubleValue());
        chart[i + 24][1] = "Sábado";
        chart[i + 24][0] = new Integer(i);
        chart[i + 24][2] = new Double(effp);
        effpTotal += (effp * numTiposDia[1]);

        double P = ((Double)
sub.getValorAtributo(FluxoPotencia2.POTENCIA_ATIVA).getValor()).doubleValue
());
        double Q = ((Double)
sub.getValorAtributo(FluxoPotencia2.POTENCIA_REATIVA).getValor()).doubleVal
ue());
        double s = Math.sqrt((P * P) + (Q * Q));
        chart2[i + 24][1] = "Sábado";
        chart2[i + 24][0] = new Integer(i);
        chart2[i + 24][2] = new Double((effp / s) * 100);

        energiaTotal += (s * numTiposDia[1]);
        barra.setValor(barra.getValor() + 1);
    }

//domingo
dia = "Domingo";
for (int i = 0; i < 24; i++) {
    hora = i;
    algoritmoFluxo.getParams(grafo);
    Map params = algoritmoCurvasTipicas.getParams(grafo);
    CentroDeCarga.setValor(params, "Ajuste de Cargas", 0,
valorPeriodo);
    CentroDeCarga.setValor(params, "Ajuste de Cargas", 1,
"Domingo");
    CentroDeCarga.setValor(params, "Ajuste de Cargas", 2,
Integer.toString(i));
    algoritmoCurvasTipicas.execute(grafo, engodo);
    algoritmoFluxo.execute(grafo, engodo);
    double effp = ((Double)
Navigator.percorreArvore(grafo.getSubestacao(), new Double(0),
SemAcaoPreFixada.INSTANCE, calculaEFPF).doubleValue());
    chart[i + 48][1] = "Domingo";
    chart[i + 48][0] = new Integer(i);
    chart[i + 48][2] = new Double(effp);
    effpTotal += (effp * numTiposDia[2]);

    double P = ((Double)
sub.getValorAtributo(FluxoPotencia2.POTENCIA_ATIVA).getValor()).doubleValue
());
    double Q = ((Double)
sub.getValorAtributo(FluxoPotencia2.POTENCIA_REATIVA).getValor()).doubleVal
ue());
    double s = Math.sqrt((P * P) + (Q * Q));

```

```

        chart2[i + 48][1] = "Domingo";
        chart2[i + 48][0] = new Integer(i);
        chart2[i + 48][2] = new Double((effp / s) * 100);

        energiaTotal += (s * numTiposDia[2]);
        barra.setValor(barra.getValor() + 1);
    }

    //definição do gráfico
    DefinicaoChart def = new DefinicaoChart();
    def.setNome("Curva de E.F.F.P.");
    def.setNomeEixoX("Hora");
    def.setGrafico3d(false);
    def.setNomeEixoY("Energia (kWh)");
    def.setTipo(DefinicaoChart.LINHAS);
    def.setDados(chart);
    result.put("Curva de E.F.F.P", def);

    //definição do gráfico 2
    def = new DefinicaoChart();
    def.setNome("Curva de E.F.F.P. (%");
    def.setNomeEixoX("Hora");
    def.setGrafico3d(false);
    def.setNomeEixoY("Porcentagem de Energia");
    def.setTipo(DefinicaoChart.LINHAS);
    def.setDados(chart2);
    result.put("Curva de E.F.F.P (%)", def);

    result.put("E.F.F.P. Total no período", nf.format(effpTotal));
    result.put("Porcentagem de E.F.F.P. Total no período",
    nf.format((effpTotal / energiaTotal) * 100));

    milis = System.currentTimeMillis() - milis;
    result.put("Tempo de Execução", ((double) milis) / 1000.0 + " s");
    Map<Double, Transformador> treeDest = new TreeMap<Double,
Transformador>();
    for (Map.Entry<Transformador, Double[][]> e : tree.entrySet()) {
        Double[][] v = e.getValue();
        double[] values = new double[] { v[0][0], v[1][0], v[2][0] };
        Arrays.sort(values);
        if (values[0] == v[0][0]) {
            e.getKey().setMelhorTap(1);
        }
        if (values[0] == v[1][0]) {
            e.getKey().setMelhorTap(2);
        }
        if (values[0] == v[2][0]) {
            e.getKey().setMelhorTap(3);
        }
        treeDest.put(e.getValue()[e.getKey().getMelhorTap() - 1][0],
e.getKey());
    }
    for (Double d : treeDest.keySet()) {
        Transformador t = treeDest.get(d);
        Double[] ds = tree.get(t)[t.getMelhorTap() - 1];
        double percTotal = ds[0] * 100.0 / effpTotal;
        double percDistr = ds[0] * 100.0 / ds[1];
        System.out.println("EFPF: " + d + " (" + percDistr + "%) " +
percTotal + ", Trafo " + t.getVertice().getId());
    }
}

```

```

        return result; //To change body of implemented methods use File |
Settings | File Templates.
    }

    static Map<Transformador, Double[][]> tree = new HashMap<Transformador,
Double[][]>();

    static class AlgoritmoEFP implements AcaoPosFixada {

        private double tensaoAL;

        public AlgoritmoEFP(double tensaoAL) {
            this.tensaoAL = tensaoAL;
        }

        public Object execute(Navigable n, Object params) {
            double efp = 0;
            if (params != null) {
                efp = ((Double) params).doubleValue();
            }
            if (n instanceof Vertice) {
                Vertice v = (Vertice) n;
                if (v.getComponente() instanceof Transformador) {
                    Transformador t = (Transformador) v.getComponente();
                    double tensao = ((Double)
t.getValorAtributo(FluxoPotencia2.MODULO_TENSAO).getValor()).doubleValue();
                    double quedaInterna =
AjusteTaps.getQuedaInternaTrafo(t);
                    double demandaMaxima = ((Double)
t.getValorAtributo(CentroDeCarga.DEMANDA_MAXIMA).getValor()).doubleValue();
                    CentroDeCarga.calculaCCTransformador(t,
Constantes.QUEDA_PERMISSIVEL_BT, demandaMaxima, t.getNumConsumidores());
                    double quedaMT = (1.0 - (tensao / tensaoAL)) * 100.0;
                    double elevacao = - ((t.getTap() - 1) * 5);
                    double efpTrafo =
t.getModelo().getEFP(t.getDemanda(), quedaMT + quedaInterna + elevacao);
                    if (efpTrafo / t.getDemanda() > 1.0001) {
                        t.getModelo().getEFP(t.getDemanda(), quedaMT +
quedaInterna + elevacao);
                        System.out.println("Erro trafo: " +
t.getVertice().getId());
                    }
                    efp += efpTrafo;
                    Double[][] efpTemp = tree.get(t);
                    if (efpTemp == null) {
                        efpTemp = new Double[][] { {0.0, 0.0}, {0.0, 0.0},
{0.0, 0.0} };
                    }

                    double efpTap1 = t.getModelo().getEFP(t.getDemanda(),
quedaMT + quedaInterna);
                    efpTemp[0][0] += efpTap1;
                    efpTemp[0][1] += t.getDemanda();

                    double efpTap2 = t.getModelo().getEFP(t.getDemanda(),
quedaMT + quedaInterna -5);
                    efpTemp[1][0] += efpTap2;
                    efpTemp[1][1] += t.getDemanda();

                    double efpTap3 = t.getModelo().getEFP(t.getDemanda(),
quedaMT + quedaInterna -10);

```

```

        effpTemp[2][0] += effpTap3;
        effpTemp[2][1] += t.getDemanda();

        tree.put(t, effpTemp);
        if (dia.equals("Dia Útil")) {
            if (t.getChart() == null) {
                t.setChart(new Object[24*3][3]);
            }
            t.getChart()[hora][0] = new Integer(hora);
            t.getChart()[hora][1] = "EFPF";
            t.getChart()[hora][2] = new Double(effpTrafo *
100.0/ t.getDemanda());

            t.getChart()[hora + 24][0] = new Integer(hora);
            t.getChart()[hora + 24][1] = "Queda MT";
            t.getChart()[hora + 24][2] = new Double(quedaMT);

            t.getChart()[hora + 48][0] = new Integer(hora);
            t.getChart()[hora + 48][1] = "Queda Fim de Linha";
            t.getChart()[hora + 48][2] = new Double(quedaMT +
quedaInterna + elevacao + t.getModelo().getDeltaV(t.getDemanda()));
        }
    }
    }
    return new Double(effp);
}

public Object operaResultados(Object resultadoParcial, Object
resultadoRamo) {
    double d1 = 0;
    if (resultadoParcial != null) {
        d1 = ((Double) resultadoParcial).doubleValue();
    }
    double d2 = 0;
    if (resultadoRamo != null) {
        d2 = ((Double) resultadoRamo).doubleValue();
    }
    return new Double(d1 + d2);
}

private Map params = new HashMap();

private void criaParams(String[] als) {
    Repositorio rep = Context.getContext().getRepository();

    List l = new ArrayList(1);
    Parametro p = new Parametro();
    p.setNome("Elevação de Tensão Máxima (%");
    p.setTipo(Parametro.TIPO.DOUBLE);
    p.setValor(new Double(- Constantes.ELEVACAO_MAXIMA_CONSUMIDOR));
    l.add(p);

    p = new Parametro();
    p.setNome("Queda de Tensão Máxima (%");
    p.setTipo(Parametro.TIPO.DOUBLE);
    p.setValor(new Double(Constantes.QUEDA_MAXIMA_CONSUMIDOR));
    l.add(p);

    p = new Parametro();
    p.setNome("Queda de Tensão Máxima Permissível na rede de BT (%");

```

```

        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setValor(new Double(Constants.QUEDA_PERMISSIVEL_BT));
        l.add(p);

        p = new Parametro();
        p.setNome("Queda de Tensão Máxima interna do transf. de distr.
(%)");
        p.setTipo(Parametro.TIPO.DOUBLE);
        p.setValor(new Double(Constants.QUEDA_TD));
        l.add(p);

        params.put("Parâmetros de Queda de Tensão", l);

        p = new Parametro();
        p.setTipo(Parametro.TIPO.MULTIPLA_ESCOLHA);
        p.setNome(PARAM_PERIODO);
        p.setOpcoes(rep.getPeriodos(als));
        l = new ArrayList(1);
        l.add(p);
        params.put("Curva de Carga", l);
    }

    public Map getParams(Grafo grafo) {
        criaParams(grafo.getIdentificadorGrafo());
        return params;
    }

    public Algoritmo getNext() {
        return null;
    }

    public boolean isParamsPadrao() {
        return true;
    }

    public void trataTelaParametros(DialogParams dialog) { }

    public boolean isPrintable() {
        return false; //To change body of implemented methods use File |
Settings | File Templates.
    }
    public String[] getPrintResults() {
        return new String[0]; //To change body of implemented methods use
File | Settings | File Templates.
    }
}

```

A3.4 – Modelos Representativos

```

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.util.ModelagemRedesSec;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 01/09/2005

```

```

* Time: 17:02:13
* To change this template use File | Settings | File Templates.
*/
public class ElementoA {

    private long nConsumidores;
    private double coefDeltaV;
    private double sMax;

    public void setsMax(double sMax) {
        this.sMax = sMax;
    }

    public long getnConsumidores() {
        return nConsumidores;
    }

    public void setnConsumidores(long nConsumidores) {
        this.nConsumidores = nConsumidores;
    }

    private double getC(double deltaVfj, double s) {
        double somLRof = ModelagemRedesSec.getNLRo((int) nConsumidores);
        return somLRof * deltaVfj / getDeltaV(s);
    }

    public double getNConsumidor(double deltaV, double s) {
        double c = getC(deltaV, s);
        double j_1 = (-1 + Math.sqrt(1 + (4 * c))) / 2.0;
        double j_2 = (-1 - Math.sqrt(1 + (4 * c))) / 2.0;
        if (j_1 > 0.0) {
            return j_1;
        } else {
            return j_2;
        }
    }

    public double getDeltaV(double s) {
        return s * coefDeltaV / sMax;
    }

    public void setCoefDeltaV(double coefDeltaV) {
        this.coefDeltaV = coefDeltaV;
    }
}

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.util.ModelagemRedesSec;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 01/09/2005
 * Time: 17:02:23
 * To change this template use File | Settings | File Templates.
 */
public class ElementoB {

    private long nConsumidores;
    private long nConsumidoresRamo;

```

```

    public long getnConsumidoresRamo() {
        return nConsumidoresRamo;
    }

    public void setnConsumidoresRamo(long nConsumidoresRamo) {
        this.nConsumidoresRamo = nConsumidoresRamo;
    }

    public long getnConsumidores() {
        return nConsumidores;
    }

    public void setnConsumidores(long nConsumidores) {
        this.nConsumidores = nConsumidores;
    }

    public double getDeltaV(double deltaVTotal) {
        double n = ModelagemRedesSec.getNLRo(nConsumidores,
nConsumidoresRamo);
        double nTotal = ModelagemRedesSec.getNLRo((int) (nConsumidoresRamo
/ 2)) + n;
        return n * deltaVTotal / nTotal;
    }

    private double getC(double deltaV, double deltaVTotal) {
        return (deltaV * ModelagemRedesSec.getNLRo(nConsumidores,
nConsumidoresRamo) / getDeltaV(deltaVTotal)) - nConsumidoresRamo;
    }

    public double getNConsumidor(double deltaV, double deltaVTotal) {
        double c = getC(deltaV, deltaVTotal);
        double j1 = ((-1 -nConsumidoresRamo) + Math.sqrt(Math.pow(1 +
nConsumidoresRamo, 2.0) + 4 * c)) / 2.0;
        double j2 = ((-1 -nConsumidoresRamo) - Math.sqrt(Math.pow(1 +
nConsumidoresRamo, 2.0) + 4 * c)) / 2.0;
        if (j1 >= 0.0 && j2 >= 0.0) {
            return Math.min(j1, j2);
        } else {
            return Math.max(j1, j2);
        }
    }
}

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 09:48:10
 * To change this template use File | Settings | File Templates.
 */
public class Modelo1 implements ModeloRedeSecundaria {

```

```

    public double getDeltaV(double s) {
        return 0;
    }

    public double getDeltaVCC(double s) {
        return 0;
    }

    public Icon getIcon() {
        return new
ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_1_b.gif"));
    }

    public double getEFFP(double s, double quedaPrimCons) {
        if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
            return s;
        }
        if (quedaPrimCons > Constantes.QUEDA_MAXIMA_CONSUMIDOR) {
            return s;
        }
        return 0;
    }

    public int getIdModelo() {
        return 1; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String getNomeModelo() {
        return "Modelo 1";
    }

    public void setnConsumidores(long nConsumidores) {
    }

    public void setsMax(double sMax) {
    }

    public void setDeltaVPerm(double deltaVPerm) {
    }

}

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;

import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 09:51:46
 * To change this template use File | Settings | File Templates.

```

```

*/
public class Modelo2 implements ModeloRedeSecundaria {

    private double deltaVPerm;
    private double sMax;

    public Modelo2() { }

    public double getDeltaV(double s) {
        return s * (deltaVPerm / sMax);
    }

    public double getDeltaVCC(double s) {
        return s * (deltaVPerm / sMax);
    }

    public Icon getIcon() {
        return new
        ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_2_b.gif"));
    }

    public double getEFFF(double s, double quedaPrimCons) {
        double deltaV = getDeltaV(s);
        if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
            if ((-(quedaPrimCons - Constantes.ELEVACAO_MAXIMA_CONSUMIDOR))
> deltaV) {
                return s;
            }
        }
        if (deltaV + quedaPrimCons > Constantes.QUEDA_MAXIMA_CONSUMIDOR) {
            return s;
        }
        return 0;
    }

    public int getIdModelo() {
        return 2; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String getNomeModelo() {
        return "Modelo 2";
    }

    public void setnConsumidores(long nConsumidores) {
        //To change body of implemented methods use File | Settings | File
Templates.
    }

    public double getDeltaVPerm() {
        return deltaVPerm;
    }

    public void setDeltaVPerm(double deltaVPerm) {
        this.deltaVPerm = deltaVPerm;
    }

    public double getsMax() {
        return sMax;
    }
}

```

```

        public void setsMax(double sMax) {
            this.sMax = sMax;
        }
    }

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;

import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 09:57:30
 * To change this template use File | Settings | File Templates.
 */
public class Modelo3 implements ModeloRedeSecundaria {

    private double sMax;
    private double deltaVPerm;
    private long nConsumidores;

    public Modelo3() { }

    public double getsMax() {
        return sMax;
    }

    public void setsMax(double sMax) {
        this.sMax = sMax;
    }

    public double getDeltaVPerm() {
        return deltaVPerm;
    }

    public void setDeltaVPerm(double deltaVPerm) {
        this.deltaVPerm = deltaVPerm;
    }

    public long getnConsumidores() {
        return nConsumidores;
    }

    public void setnConsumidores(long nConsumidores) {
        this.nConsumidores = nConsumidores;
    }

    public double getDeltaV(double s) {
        return s * (deltaVPerm / sMax);
    }

    public double getDeltaVCC(double s) {
        if (nConsumidores == 0) {
            return 0.0;
        }
    }

```

```

    }
    return s * getLRO() * (((2.0 * nConsumidores * nConsumidores) +
(3.0 * nConsumidores) + 1) / (6.0 * nConsumidores * nConsumidores));
}

    private double getLRO() {
        return (2 * deltaVPerm) / (sMax *
ModelagemRedesSec.getKv(nConsumidores));
    }

    public Icon getIcon() {
        return new
ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_3_b.gif"));
    }

    public double getEFFP(double s, double quedaPrimCons) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        double effp = 0.0;
        if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
            double n = getNConsumidor(-(quedaPrimCons -
Constantes.ELEVACAO_MAXIMA_CONSUMIDOR), s);
            int i = (int) n;
            if (i > 0) {
                if (i > nConsumidores) {
                    i = (int) nConsumidores;
                }
                effp += (i * s) / nConsumidores;
            }
        }
        double n = getNConsumidor(Constantes.QUEDA_MAXIMA_CONSUMIDOR -
quedaPrimCons, s);
        int i = (int) n;
        if (i > 0 && i <= nConsumidores) {
            effp += ((nConsumidores - i) * s) / nConsumidores;
        }
        return effp;
    }

    public int getIdModelo() {
        return 3; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String getNomeModelo() {
        return "Modelo 3";
    }

    private double getC(double deltaVfj, double s) {
        double somLRof = ModelagemRedesSec.getNLRO((int) nConsumidores);
        return somLRof * deltaVfj / getDeltaV(s);
    }

    private double getNConsumidor(double deltaV, double s) {
        double c = getC(deltaV, s);
        double j_1 = (-1 + Math.sqrt(1 + (4 * c))) / 2.0;
        double j_2 = (-1 - Math.sqrt(1 + (4 * c))) / 2.0;
        if (j_1 > 0.0) {
            return j_1;
        } else {

```

```

        return j_2;
    }
}

public static void main(String[] args) {
    Modelo3 m = new Modelo3();
    m.setDeltaVPerm(5);
    m.setnConsumidores(10);
    m.setsMax(400);
    System.out.println("deltaV: " + m.getDeltaV(300));
    System.out.println("LRO: " + m.getLRO());
    System.out.println("deltaVCC: " + m.getDeltaVCC(300));

    for (int i = 0; i < 5; i++) {
        System.out.println("j = " + m.getNConsumidor(i, 350));
    }
}
}

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;

import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 12:26:01
 * To change this template use File | Settings | File Templates.
 */
public class Modelo4 implements ModeloRedeSecundaria {

    private double sMax;
    private double deltaVPerm;
    private long nConsumidores;

    public Modelo4() { }

    public double getDeltaV(double s) {
        return s * (deltaVPerm / sMax);
    }

    public double getDeltaVCC(double s) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        return s * getLRO() * (((2.0 * nConsumidores * nConsumidores) +
(3.0 * nConsumidores) + 1.0) / (12.0 * nConsumidores * nConsumidores));
    }

    private double getLRO() {
        return (4 * deltaVPerm) / (sMax *
ModelagemRedesSec.getKv(nConsumidores / 2));
    }
}

```

```

    public Icon getIcon() {
        return new
        ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_4_b.gif"));
    }

    public double getEFP(double s, double quedaPrimCons) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        if (nConsumidores == 1) {
            double deltaV = getDeltaV(s);
            if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
                if ((-(quedaPrimCons -
                Constantes.ELEVACAO_MAXIMA_CONSUMIDOR)) > deltaV) {
                    return s;
                }
            }
            if (deltaV + quedaPrimCons >
            Constantes.QUEDA_MAXIMA_CONSUMIDOR) {
                return s;
            }
        }
        double effp = 0.0;
        if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
            double n = getNConsumidor(-(quedaPrimCons -
            Constantes.ELEVACAO_MAXIMA_CONSUMIDOR), s);
            int i = (int) n;
            if (i > 0) {
                if (i > nConsumidores) {
                    i = (int) nConsumidores;
                }
                effp += (i * s) / nConsumidores;
            }
        }
        double n = getNConsumidor(Constantes.QUEDA_MAXIMA_CONSUMIDOR -
        quedaPrimCons, s);
        int i = (int) n;
        if (i > 0 && i <= (nConsumidores / 2)) {
            effp += ((nConsumidores - (i * 2)) * s) / nConsumidores;
        }
        return effp;
    }

    private double getC(double deltaVfj, double s) {
        double somLRof = ModelagemRedesSec.getNLRo((int) nConsumidores /
        2);
        return somLRof * deltaVfj / getDeltaV(s);
    }

    private double getNConsumidor(double deltaV, double s) {
        double c = getC(deltaV, s);
        double j_1 = (-1 + Math.sqrt(1 + (4 * c))) / 2.0;
        double j_2 = (-1 - Math.sqrt(1 + (4 * c))) / 2.0;
        if (j_1 > 0.0) {
            return j_1;
        } else if (j_2 > 0.0) {
            return j_2;
        }
        return 0.0;
    }
}

```

```

        public int getIdModelo() {
            return 4; //To change body of implemented methods use File |
Settings | File Templates.
        }

        public String getNomeModelo() {
            return "Modelo 4";
        }

        public double getsMax() {
            return sMax;
        }

        public void setsMax(double sMax) {
            this.sMax = sMax;
        }

        public double getDeltaVPerm() {
            return deltaVPerm;
        }

        public void setDeltaVPerm(double deltaVPerm) {
            this.deltaVPerm = deltaVPerm;
        }

        public long getnConsumidores() {
            return nConsumidores;
        }

        public void setnConsumidores(long nConsumidores) {
            this.nConsumidores = nConsumidores;
        }
    }

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.excecoes.ExcecaoAlgoritmo;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;

import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 14:38:04
 * To change this template use File | Settings | File Templates.
 */
public class Modelo5A implements ModeloRedeSecundaria {

    protected double sMax;
    protected double deltaVPerm;
    protected long nConsumidores;
    private ElementoB elementoB = new ElementoB();
    private ElementoA elementoA = new ElementoA();

    public Modelo5A() { }

```

```

    public double getsMax() {
        return sMax;
    }

    public void setsMax(double sMax) {
        this.sMax = sMax;
        elementoA.setsMax(sMax / 4);
    }

    public double getDeltaVPerm() {
        return deltaVPerm;
    }

    public void setDeltaVPerm(double deltaVPerm) {
        this.deltaVPerm = deltaVPerm;
    }

    public long getnConsumidores() {
        return nConsumidores;
    }

    public void setnConsumidores(long nConsumidores) {
        this.nConsumidores = nConsumidores;
        elementoB.setnConsumidores(nConsumidores / 2);
        elementoB.setnConsumidoresRamo(nConsumidores / 2);
        elementoA.setnConsumidores(nConsumidores / 4);
        double nLROB = ModelagemRedesSec.getNLRO(nConsumidores / 2,
nConsumidores / 4);
        double nLROA = ModelagemRedesSec.getNLRO((int) nConsumidores / 4);
        elementoA.setCoefDeltaV(deltaVPerm * nLROA / (nLROA + nLROB));
    }

    public double getDeltaV(double s) {
        return s * (deltaVPerm / sMax);
    }

    public double getDeltaVCC(double s) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        return s * getLRO() * (((29.0 * nConsumidores * nConsumidores) +
(42.0 * nConsumidores) + 16.0) / (48.0 * nConsumidores * nConsumidores));
    }

    private double getLRO() {
        return (16.0 * deltaVPerm) / (sMax * ((12.0 *
ModelagemRedesSec.getKv(nConsumidores / 2, nConsumidores / 2.0)) +
ModelagemRedesSec.getKv(nConsumidores / 4)));
    }

    public Icon getIcon() {
        return new
ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_5A_b.gif"));
    }

    public int getIdModelo() {
        return 51; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String getNomeModelo() {

```

```

        return "Modelo 5A";
    }

    public double getC() {
        return 0.0;
    }

    public double getEFPF(double s, double quedaPrimCons) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        double efpf = 0.0;
        double queda = Constantes.QUEDA_MAXIMA_CONSUMIDOR - quedaPrimCons;
        double quedaFim = getDeltaV(s);
        //testa se pode haver efpf por sobretensão
        if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
            //calcula a partir de qual nível de queda a tensão estará
            dentro do padrão no elemento B
            double quedaProc = -(quedaPrimCons -
            Constantes.ELEVACAO_MAXIMA_CONSUMIDOR);
            //consumidor a partir do qual a tensão estará dentro do padrão
            no elemento B
            double n = elementoB.getNConsumidor(quedaProc, quedaFim);
            int i = (int) n;
            //verica se o consumidor está dentro do elemento A
            if (i > 0 && i <= (nConsumidores / 2)) {
                efpf += (i * s) / nConsumidores;
            } //senão talvez possa estar no elemento B
            } else if (i > (nConsumidores / 2)) {
                //soma-se a energia do elemento A como EFPF
                efpf += s / 2;
                double deltaVB = elementoB.getDeltaV(quedaFim);
                //calcula a partir de qual nível de queda a tensão estará
                dentro do padrão no elemento A
                quedaProc -= deltaVB;
                //consumidor a partir do qual a tensão estará dentro do
                padrão no elemento A
                n = (int) elementoA.getNConsumidor(quedaProc, s);
                i = (int) n;
                //se o consumidor estiver fora da rede então toda a energia
                estará fora do padrão
                if (i > elementoA.getnConsumidores()) {
                    efpf += s / 2;
                } //senão a EFPF será considerada até o consumidor i * 2,
                pois são dois elementos A.
                } else {
                    efpf += 2 * (i * s) / (nConsumidores);
                }
            }
        }
        //se a queda final for menor que a queda máxima permitida então não
        existirá EFPF por subtensão
        if (quedaFim < queda) {
            return efpf;
        } //senão vamos calcular a EFPF
        } else {
            double quedaProc = queda;
            //vamos encontrar qual consumidor possui a queda de tensão
            necessária para a energia ficar fora do padrão
            double n = elementoB.getNConsumidor(quedaProc, quedaFim);
            int i = (int) n;

```

```

        //se o consumidor estiver dentro do elemento A
        if (i > 0 && i <= (nConsumidores / 2)) {
            //adicionamos a EFP do elemento B
            effp += (((nConsumidores / 2) - i) * s) / nConsumidores;
            //e tambem a EFP dos dois elementos A
            effp += s / 2;
        }
        //senão se o consumidor estiver nos elementos A
        } else if (i > (nConsumidores / 2)) {
            double deltaVB = elementoB.getDeltaV(quedaFim);
            quedaProc -= deltaVB;
            //consumidor a partir do qual a energia estará fora do
            padrão
            n = (int) elementoA.getNConsumidor(quedaProc, s);
            i = (int) n;
            //se o consumidor estiver no intervalo coberto calculamos a
            EFP, senão toda a energia está dentro do padrão
            if (i > 0 && i < (nConsumidores / 4)) {
                effp += 2 * ((nConsumidores / 4) - i) * s /
                (nConsumidores);
            }
        }
    }
    if (effp > s) {
        throw new ExcecaoAlgoritmo("Bug no cálculo EFP Modelo5A");
    }
    return effp;
}
}

```

```
package br.com.mega.ppgee.modelos;
```

```
import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.excecoes.ExcecaoAlgoritmo;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;
```

```
import javax.swing.*;
```

```
/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 15:08:23
 * To change this template use File | Settings | File Templates.
 */
```

```
public class Modelo5B implements ModeloRedeSecundaria {
```

```
    private double sMax;
    private double deltaVPerm;
    private long nConsumidores;
    private ElementoA elementoA = new ElementoA();
    private ElementoB elementoB = new ElementoB();
```

```
    public Modelo5B() { }
```

```
    public double getsMax() {
        return sMax;
    }
}

```

```

    public void setsMax(double sMax) {
        this.sMax = sMax;
        elementoA.setsMax(sMax / 3);
    }

    public double getDeltaVPerm() {
        return deltaVPerm;
    }

    public void setDeltaVPerm(double deltaVPerm) {
        this.deltaVPerm = deltaVPerm;
    }

    public long getnConsumidores() {
        return nConsumidores;
    }

    public void setnConsumidores(long nConsumidores) {
        this.nConsumidores = nConsumidores;
        elementoA.setnConsumidores(nConsumidores / 3);
        elementoB.setnConsumidores(nConsumidores / 3);
        elementoB.setnConsumidoresRamo((2 * nConsumidores) / 3);
        double nLROB = ModelagemRedesSec.getNLRO(nConsumidores / 3, (2 *
nConsumidores) / 3);
        double nLROA = ModelagemRedesSec.getNLRO((int) (2 * nConsumidores)
/ 3);
        elementoA.setCoefDeltaV(deltaVPerm * nLROA / (nLROA + nLROB));
    }

    public double getDeltaV(double s) {
        return s * (deltaVPerm / sMax);
    }

    public double getDeltaVCC(double s) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        return s * getLRO() * (((42.0 * nConsumidores * nConsumidores) +
(63.0 * nConsumidores) + 27.0) / (54.0 * nConsumidores * nConsumidores));
    }

    private double getLRO() {
        return (16.0 * deltaVPerm) / (sMax * ((5.0 *
ModelagemRedesSec.getKv(nConsumidores / 3, 2.0 * nConsumidores / 3.0)) +
(ModelagemRedesSec.getKv(nConsumidores / 3))));
    }

    public Icon getIcon() {
        return new
ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_5B_b.gif"));
    }

    public double getEFFP(double s, double quedaPrimCons) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        double effp = 0.0;
        double queda = Constantes.QUEDA_MAXIMA_CONSUMIDOR - quedaPrimCons;
        double quedaFim = getDeltaV(s);
        //testa se pode haver effp por sobretensão
        if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {

```

```

        //calcula a partir de qual nível de queda a tensão estará
dentro do padrão no elemento B
        double quedaProc = - (quedaPrimCons -
Constantes.ELEVACAO_MAXIMA_CONSUMIDOR);
        //consumidor a partir do qual a tensão estará dentro do padrão
no elemento B
        double n = elementoB.getNConsumidor(quedaProc, quedaFim);
        int i = (int) n;
        //verica se o consumidor está dentro do elemento A
        if (i > 0 && i <= (nConsumidores / 3)) {
            effp += (i * s) / nConsumidores;
        //senão talvez possa estar no elemento B
        } else if (i > (nConsumidores / 3)) {
            //soma-se a energia do elemento A como EFPF
            effp += s / 3;
            double deltaVB = elementoB.getDeltaV(quedaFim);
            //calcula a partir de qual nível de queda a tensão estará
dentro do padrão no elemento A
            quedaProc -= deltaVB;
            //consumidor a partir do qual a tensão estará dentro do
padrão no elemento A
            n = (int) elementoA.getNConsumidor(quedaProc, s);
            i = (int) n;
            //se o consumidor estiver fora da rede então toda a energia
estará fora do padrão
            if (i > elementoA.getnConsumidores()) {
                effp += 2*s/3;
            //senão a EFPF será considerada até o consumidor i * 2,
pois são dois elementos A.
            } else {
                effp += 2 * (i * s) / (nConsumidores / 3);
            }
        }
    }
    //se a queda final for menor que a queda máxima permitida então não
existirá EFPF por subtensão
    if (quedaFim < queda) {
        return effp;
    //senão vamos calcular a EFPF
    } else {
        double quedaProc = queda;
        //vamos encontrar qual consumidor possui a queda de tensão
necessária para a energia ficar fora do padrão
        double n = elementoB.getNConsumidor(quedaProc, quedaFim);
        int i = (int) n;
        //se o consumidor estiver dentro do elemento A
        if (i > 0 && i <= (nConsumidores / 3)) {
            //adicionamos a EFPF do elemento B
            effp += (((nConsumidores / 3) - i) * s) / nConsumidores;
            //e também a EFPF dos dois elementos A
            effp += 2 * s / 3;
        //senão se o consumidor estiver nos elementos A
        } else if (i > (nConsumidores / 3)) {
            double deltaVB = elementoB.getDeltaV(quedaFim);
            quedaProc -= deltaVB;
            //consumidor a partir do qual a energia estará fora do
padrão
            n = (int) elementoA.getNConsumidor(quedaProc, s);
            i = (int) n;
            //se o consumidor estiver no intervalo coberto calculamos a
EFPF, senão toda a energia está dentro do padrão

```

```

        if (i > 0 && i < (nConsumidores / 3)) {
            effp += 2 * ((nConsumidores / 3) - i) * s /
(nConsumidores);
        }
    }
    if (effp > s) {
        throw new ExcecaoAlgoritmo("Bug no cálculo EFPF Modelo5A");
    }
    return effp;
}

public int getIdModelo() {
    return 52; //To change body of implemented methods use File |
Settings | File Templates.
}

public String getNomeModelo() {
    return "Modelo 5B";
}
}

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.plugins.Constantes;
import br.com.mega.ppgee.excecoes.ExcecaoAlgoritmo;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;

import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 29/07/2005
 * Time: 15:21:26
 * To change this template use File | Settings | File Templates.
 */
public class Modelo6 implements ModeloRedeSecundaria {

    private double sMax;
    private double deltaVPerm;
    private long nConsumidores;

    private ElementoA elementoAEsq = new ElementoA();
    private ElementoB elementoBDir = new ElementoB();
    private ElementoA elementoADir = new ElementoA();

    public Modelo6() { }

    public double getsMax() {
        return sMax;
    }

    public void setsMax(double sMax) {
        this.sMax = sMax;
        elementoAEsq.setsMax(sMax / 3);
        elementoADir.setsMax(sMax / 6);
    }

```

```

    }

    public double getDeltaVPerm() {
        return deltaVPerm;
    }

    public void setDeltaVPerm(double deltaVPerm) {
        this.deltaVPerm = deltaVPerm;
    }

    public long getnConsumidores() {
        return nConsumidores;
    }

    public void setnConsumidores(long nConsumidores) {
        this.nConsumidores = nConsumidores;
        elementoAEsq.setnConsumidores(nConsumidores / 3);
        elementoBDir.setnConsumidores(nConsumidores / 3);
        elementoBDir.setnConsumidoresRamo(nConsumidores / 3);
        elementoADir.setnConsumidores(nConsumidores / 6);
    }

    public double getDeltaV(double s) {
        return s * (deltaVPerm / sMax);
    }

    public double getDeltaVCC(double s) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        double p1 = getLRo1() * (((29.0 * nConsumidores * nConsumidores) +
(63.0 * nConsumidores) + 36.0) / (108.0 * nConsumidores * nConsumidores));
        double p2 = getLRo2() * (((4.0 * nConsumidores * nConsumidores) +
(18.0 * nConsumidores) + 18.0) / (108.0 * nConsumidores * nConsumidores));
        return s * (p1 + p2);
    }

    private double getLRo1() {
        return (24.0 * deltaVPerm) / (sMax * ((12.0 *
ModelagemRedesSec.getKv(nConsumidores / 3, nConsumidores / 2.0)) +
ModelagemRedesSec.getKv(nConsumidores / 3)));
    }

    private double getLRo2() {
        return 6.0 * deltaVPerm / (sMax *
ModelagemRedesSec.getKv(nConsumidores / 3));
    }

    public Icon getIcon() {
        return new
ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_6_b.gif"));
    }

    public double getEFFP(double s, double quedaPrimCons) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        double effp = 0.0;
        double queda = Constantes.QUEDA_MAXIMA_CONSUMIDOR - quedaPrimCons;
        double quedaFim = getDeltaV(s);

```

```

    /**
     * primeiramente vamos calcular a EFP do lado direito como se
     fosse um modelo 5
     */

    //testa se pode haver efp por sobretensão
    if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
        //calcula a partir de qual nível de queda a tensão estará
        dentro do padrão no elemento B
        double quedaProc = -(quedaPrimCons -
        Constantes.ELEVACAO_MAXIMA_CONSUMIDOR);
        //consumidor a partir do qual a tensão estará dentro do padrão
        no elemento B
        double n = elementoBDir.getNConsumidor(quedaProc, quedaFim);
        int i = (int) n;
        //verica se o consumidor está dentro do elemento A
        if (i > 0 && i <= (nConsumidores / 3)) {
            efp += (i * s) / nConsumidores;
        //senão talvez possa estar no elemento B
        } else if (i > (nConsumidores / 3)) {
            //soma-se a energia do elemento A como EFP
            efp += s / 3;
            double deltaVB = elementoBDir.getDeltaV(quedaFim);
            //calcula a partir de qual nível de queda a tensão estará
            dentro do padrão no elemento A
            quedaProc -= deltaVB;
            //consumidor a partir do qual a tensão estará dentro do
            padrão no elemento A
            n = (int) elementoADir.getNConsumidor(quedaProc, s);
            i = (int) n;
            //se o consumidor estiver fora da rede então toda a energia
            estará fora do padrão
            if (i > elementoADir.getnConsumidores()) {
                efp += s / 3;
            //senão a EFP será considerada até o consumidor i * 2,
            pois são dois elementos A.
            } else {
                efp += 2 * (i * s) / (nConsumidores);
            }
        }
    }

    //se a queda final for menor que a queda máxima permitida então não
    existirá EFP por subtensão
    if (quedaFim > queda) {
        double quedaProc = queda;
        //vamos encontrar qual consumidor possui a queda de tensão
        necessária para a energia ficar fora do padrão
        double n = elementoBDir.getNConsumidor(quedaProc, quedaFim);
        int i = (int) n;
        //se o consumidor estiver dentro do elemento A
        if (i > 0 && i <= (nConsumidores / 3)) {
            //adicionamos a EFP do elemento B
            efp += (((nConsumidores / 3) - i) * s) / nConsumidores;
            //e também a EFP dos dois elementos A
            efp += s / 3;
        //senão se o consumidor estiver nos elementos A
        } else if (i > (nConsumidores / 3)) {
            double deltaVB = elementoBDir.getDeltaV(quedaFim);
            quedaProc -= deltaVB;
            //consumidor a partir do qual a energia estará fora do
            padrão
        }
    }

```

```

        n = (int) elementoADir.getNConsumidor(quedaProc, s);
        i = (int) n;
        //se o consumidor estiver no intervalo coberto calculamos a
        EFP, senão toda a energia está dentro do padrão
        if (i > 0 && i < (nConsumidores / 6)) {
            effp += 2 * ((nConsumidores / 6) - i) * s /
(nConsumidores);
        }
    }
}

/**
 * Agora finalmente vamos calcular a EFP do lado esquerdo
utilizando o ElementoA
 */

//testa se pode haver effp por sobretensão
if (quedaPrimCons < Constantes.ELEVACAO_MAXIMA_CONSUMIDOR) {
    //calcula a partir de qual nível de queda a tensão estará
dentro do padrão no elemento B
    double quedaProc = -(quedaPrimCons -
Constantes.ELEVACAO_MAXIMA_CONSUMIDOR);
    //consumidor a partir do qual a tensão estará dentro do padrão
no elemento A
    int i = (int) elementoADir.getNConsumidor(quedaProc, s);
    //se o consumidor estiver fora da rede então toda a energia
estará fora do padrão
    if (i > elementoADir.getnConsumidores()) {
        effp += s / 3;
        //senão a EFP será considerada até o consumidor i * 2, pois
são dois elementos A.
    } else {
        effp += (i * s) / (nConsumidores);
    }
}

//se a queda final for menor que a queda máxima permitida então não
existirá EFP por subtensão
if (quedaFim > queda) {
    double quedaProc = queda;
    //vamos encontrar qual consumidor possui a queda de tensão
necessária para a energia ficar fora do padrão
    int i = (int) elementoADir.getNConsumidor(quedaProc, s);
    //se o consumidor estiver no intervalo coberto calculamos a
EFP, senão toda a energia está dentro do padrão
    if (i > 0 && i < (nConsumidores / 3)) {
        effp += ((nConsumidores / 3) - i) * s / (nConsumidores);
    }
}

if (effp > s) {
    throw new ExcecaoAlgoritmo("Bug no cálculo EFP Modelo 6");
}

return effp;
}

public int getIdModelo() {
    return 6; //To change body of implemented methods use File |
Settings | File Templates.
}

public String getNomeModelo() {

```

```

        return "Modelo 6";
    }
}

package br.com.mega.ppgee.modelos;

import br.com.mega.ppgee.Asd;
import br.com.mega.ppgee.interfaces.ModeloRedeSecundaria;
import br.com.mega.ppgee.util.ModelagemRedesSec;

import javax.swing.*;

/**
 * Created by IntelliJ IDEA.
 * User: milbradt
 * Date: 01/08/2005
 * Time: 09:25:12
 * To change this template use File | Settings | File Templates.
 */
public class Modelo7 extends Modelo5A implements ModeloRedeSecundaria {

    public Modelo7() { }

    public double getDeltaVCC(double s) {
        if (nConsumidores == 0) {
            return 0.0;
        }
        double p = ((30.0 * nConsumidores * nConsumidores) + (96.0 *
nConsumidores) + 96.0) / (384.0 * nConsumidores * nConsumidores);
        return s * getLRO() * p;
    }

    private double getLRO() {
        return (32.0 * deltaVPerm) / (sMax * (12.0 *
ModelagemRedesSec.getKv(nConsumidores / 4, nConsumidores / 4.0)) +
ModelagemRedesSec.getKv(nConsumidores / 8));
    }

    public Icon getIcon() {
        return new
ImageIcon(Asd.class.getResource("icones/modelos/modelo_bt_7_b.gif"));
    }

    public int getIdModelo() {
        return 7; //To change body of implemented methods use File |
Settings | File Templates.
    }

    public String getNomeModelo() {
        return "Modelo 7";
    }
}

```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)