UTILIZAÇÃO DE OPENGL E LINGUAGEM ORIENTADA A OBJETOS PARA O DESENVOLVIMENTO DE UM PACOTE GRÁFICO PARA ANÁLISE DE REDES

Thiago Trezza Borges

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

Aprovada por:

_____
Prof. José Luiz Rezende Pereira, Ph.D.

_____
Prof. Sandoval Carneiro Junior, Ph.D.

_____
Prof. Paulo Augusto Nepomuceno Garcia, D.Sc.

_____
Prof. Edimar José de Oliveira, D.Sc.

JUIZ DE FORA, MG – BRASIL
AGOSTO DE 2006

BORGES, THIAGO TREZZA

Utilização de OpenGL e Linguagem Orientada a Objetos para o Desenvolvimento de um Pacote Gráfico para análise de Redes [Juiz de Fora] 2006

XV, xxx p. 29,7 cm, (UFJF, M.Sc., Engenharia Elétrica, 2006)

Tese – Universidade Federal de Juiz de Fora, Faculdade de Engenharia

1. Fluxo de Potência Trifásico
2. Interface Gráfica
3. Sistemas Elétricos Trifásicos
4. Programação Orientada a Objetos
5. OpenGL

I. UFJF II. Título (Série)

Aos meus pais, Isolina e Severino,
a minha namorada Carol,
aos meus familiares,
aos professores,
aos amigos do Labspot.

AGRADECIMENTOS

Ao amigo e orientador José Luiz Rezende Pereira pela orientação, dedicação, incentivo e colaboração na realização deste trabalho.

Ao amigo Leandro Ramos de Araujo pela ajuda imprescindível durante todo trabalho

Aos professores Paulo Augusto N. Garcia , Helio Antonio, Edimar José de Oliveira e Márcio de Pinho Vinagre dirijo meus agradecimentos por seus comentários, sugestões e discussões técnicas que permitiram um melhor aprimoramento do trabalho.

Aos amigos do LABSPOT – Laboratório de Sistemas de Potência da Universidade Federal de Juiz de Fora (Rafael, Leonardo, Ângelo, Weberson, Ivo Michelle, Magda), dirijo meus sinceros agradecimentos pela ajuda, incentivo e pelos bons momentos que passei com vocês dentro e fora do laboratório..

Aos companheiros da pós-graduação pela amizade e apoio ao desenvolvimento deste trabalho. Aos amigos do C.S. pelo apoio e momentos de descontração.

Ao corpo docente da Faculdade de Engenharia/UFJF tanto de graduação quanto de pos graduação, pela dedicação na transferência dos conhecimentos.

Aos meus amigos e familiares, pelo apoio e incentivo durante toda a realização do curso.

Resumo da Dissertação apresentada à UFJF como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UTILIZAÇÃO DE OPENGL E LINGUAGEM ORIENTADA A OBJETOS PARA O DESENVOLVIMENTO DE UM PACOTE GRÁFICO PARA ANÁLISE DE REDES

Thiago Trezza Borges

Agosto / 2006

Orientadores: José Luiz Rezende Pereira

Programa: Engenharia Elétrica

Este trabalho propõe o desenvolvimento de um pacote gráfico, que é constituído de modelos matemáticos dos componentes de redes elétricas bem como de modelos gráficos de interface, para análise dos sistemas elétricos de Potência (SEP) em regime permanente.

A primeira etapa deste trabalho consiste no desenvolvimento e implementação de um programa computacional para o cálculo de fluxo de potência trifásico em coordenadas polares, usando linguagem orientada a objetos.

Em seguida, foram desenvolvidos os modelos gráficos para a concepção de uma interface gráfica para representação do SEP. Esses modelos foram implementados utilizando a biblioteca OpenGL, que é uma poderosa ferramenta gráfica permitindo um grande desempenho gráfico do aplicativo.

Posteriormente, as duas ferramentas desenvolvidas foram agregadas em uma única plataforma computacional, criando-se assim um pacote gráfico para análise de redes elétricas.

Esta nova aplicação foi modelada utilizando os conceitos de modelagem orientada a objetos (MOO) e as implementações computacionais foram feitas utilizando o Microsoft Visual C++ 6.0. Os resultados obtidos mostram as vantagens, robustez e modularidade da aplicação desenvolvida, permitindo a criação de diversas outras aplicações sobre a base já desenvolvida.

Abstract of Thesis presented to UFJF as a partial fulfillment of requirements for the degree of Master of Science (M.Sc.)

THE OPENGL AND OBJECT-ORIENTED LANGUAGE USAGE IN THE DEVELOPMENT OF A NETWORK ANALISES GRAPHIC PACKAGE

Thiago Trezza Borges

August / 2006

Supervisors: José Luiz Rezende Pereira

Department: Electrical Engineering

This work proposes the development of a graphic package having mathematical models of electrical networks as well as graphical interface models to analyze the Electrical Power Systems (EPS) in steady state.

The first part of this work comprises the development and implementation of a computer program to calculate the Three-Phase Power Flow in Polar coordinates, using object-oriented language.

The second part is dedicated to the development of models for the graphical interface to represent the EPS. These models were implemented using OpenGL library, which is a robust graphical tool that provides a great performance for this application.

In the third part, both tools were added in a single computational platform, creating a graphic package to analyze electrical networks.

The new application was modeled using object-oriented modeling (OOM) concepts and the computational implementations were made using the Microsoft Visual C++ 6.0. The results obtained show the advantages, robustness and modularity of the proposed application, allowing the creation of other applications over the base already developed.

SUMÁRIO

Lista de Figuras

Índice de Tabelas

Capítulo 1 Introdução

1.1 Considerações Iniciais

O setor elétrico nacional e mundial passa atualmente por grandes transformações. A mudança do modelo cooperativo para o modelo competitivo impôs novas filosofias para a operação e o planejamento dos sistemas elétricos. Além disso, o rápido aumento da demanda de energia tem obrigado os sistemas a operarem nos seus limites de capacidade, e por outro lado, a tentativa de expansão enfrenta problemas ambientais, sociais e principalmente financeiros.

Dentro deste ambiente competitivo, as empresas distribuidoras de energia elétrica procuram operar seus sistemas de forma cada vez mais otimizada, buscando a redução de custos assim como a redução das perdas de energia. Como conseqüência, novas filosofias de planejamento e operação passaram a ser adotas, destacando-se: (i) Operação integrada da subtransmissão e da distribuição; (ii) Controle em tempo real dos sistemas de distribuição e (iii) Utilização extensiva de programas computacionais tanto no planejamento quanto na operação dos sistemas de distribuição. Com isso, tornou-se importante a utilização de modelos matemáticos mais precisos onde as características dos sistemas de distribuição devem ser corretamente representadas, destacando-se os desequilíbrios e a representação da geração dispersa e distribuída.

Também sistemas de transmissão, com a utilização cada vez maior de dispositivos FACTS ("*Flexible AC Transmission System*") e linhas de potência natural elevada (LPNE), não podem ser mais considerados equilibrados. Portanto, a modelagem monofásica desses sistemas utilizando somente a componente de seqüência positiva não consegue representar os desequilíbrios. Neste caso, tanto para a simulação dos sistemas de distribuição quanto para os sistemas de transmissão deve-se contemplar a formulação trifásica desequilibrada.

Para uma perfeita análise, planejamento e operação destes sistemas, tanto de distribuição quanto de transmissão é indispensável a utilização de programas computacionais. Um correto aproveitamento, interpretação e análise das informações está ligada diretamente à implementação computacional.

Contudo, tradicionalmente as ferramentas computacionais usadas para simulação do comportamento dos sistemas elétricos foram historicamente desenvolvidas por

diferentes grupos, de tal forma que integração entre elas torna-se inviável. Assim, uma análise completa de um sistema elétrico é muito demorada, demandando a utilização de programas diferentes para cada tipo de estudo (Fluxo de Potência, Curto-Circuito, Estabilidade, etc.), com bases de dados normalmente incompatíveis.

Desta forma, os conceitos de Modelagem Orientada a Objetos (MOO) vêm cada vez mais sendo aplicados no desenvolvimento de ferramentas computacionais para os Sistemas Elétricos de Potência (ARAUJO, 2000). A MOO permite que o projeto seja independente, desenvolvido de forma modular, podendo a qualquer momento ser modificado ou que novas instâncias sejam criadas.

Outro problema das aplicações para Sistemas de Potência consiste na interface com o usuário. Normalmente esses programas computacionais apresentam interfaces pouco amigáveis, uma vez que são aplicações antigas e foram concebidas em ambientes e linguagens com poucos recursos gráficos (ex. FORTRAN). Estas aplicações concebidas nessas linguagens dificultam a inserção e interpretação dos dados, análise dos resultados e confecção de relatórios uma vez que a entrada e saída de dados são feitas em arquivos texto com o formato de tabelas. Além disso, o estudo de Sistemas Elétricos não é simples, e se a interface com o usuário não for feita de forma clara os diversos estudos podem se tornar confusos e trabalhosos.

1.2 Revisão Bibliográfica

Na subseção 1.2.1 será apresentada a revisão bibliográfica sobre fluxo de potência trifásico e na subseção 1.2.2 a revisão a utilização de interface gráfico com o usuário (GUI – Grafical User Interface) em aplicação de Sistemas de Potência.

1.2.1 Fluxo de Potência Trifásico

O fluxo de potência é a ferramenta mais utilizada em estudos dos sistemas elétricos de potência. Seus resultados e suas análises são aplicados no planejamento da expansão, na operação dos sistemas, na otimização dos sistemas elétricos, na análise de estabilidade, nos estudos de contingências, no controle e análise de sistemas em tempo

real, em projetos de várias espécies. Constantemente são desenvolvidos e discutidos diversos algoritmos, utilizando as mais variadas metodologias.

Dentre os mais conhecidos, destacam-se os métodos de Newton-Raphson em coordenadas polares (TINNEY e HART, 1967; MONTICELLI, 1983) e o método Desacoplado Rápido (STOTT e ALSAC, 1974). A eficiência destes métodos na solução de sistemas de transmissão é indiscutível. Todavia, em sistemas de distribuição desequilibrados, as simplificações adotadas na modelagem dos sistemas (seqüência positiva) não permitem a obtenção de resultados realísticos. Além disso, para sistemas com relação R/X das linhas elevada, característica comum aos sistemas de distribuição, o método apresentado em STOTT e ALSAC (1974) apresenta dificuldade de convergência

Para solucionar o problema de representação de sistemas desequilibrados, foi proposta a formulação trifásica para o problema do fluxo de carga. Em WASLEY e SHLASH (1974) e BIRT, GRAFFY e MacDONALD (1976) foram apresentadas extensões trifásicas para os tradicionais métodos descritos em TINNEY e HART (1967) e STOTT e ALSAC (1967). Porém, os acoplamentos mútuos e a necessidade de constantes refatorações da matriz Jacobiana na forma trifásica, tornaram estes métodos extremamente complexos.

Uma metodologia trifásica, específica para sistemas de distribuição, foi descrita em KERSTING e MENDIVE (1976). Nesta formulação, explora-se a característica radial dos sistemas de distribuição, sendo a solução do problema obtida adotando-se a teoria dos circuitos Ladder. Este método, o qual consiste em varreduras sucessivas do nó fonte em direção aos nós terminais e vice-versa, mostrou-se eficiente na solução de sistemas radiais sem a presença de derivações (ramos laterais). Contudo para sistemas com ramificações laterais, é necessária a realização de iterações auxiliares para cada um destes ramos.

Como os programas de fluxo de potência trifásico exigem grandes requisitos computacionais, diversos pesquisadores optaram por algoritmos que utilizam modelagem de seqüência positiva. Assim sendo, RAJICIC e BOSE (1988) apresentaram um modelo de fluxo de carga desacoplado modificado, no qual utilizam-se técnicas de compensação para solucionar os problemas causados por ramos onde a relação R/X é elevada.

Em SHIRMOHAMMADI (1988) foi proposto um método para solução de sistemas radiais e fracamente malhados. Neste método, o sistema é primeiramente

convertido em um sistema estritamente radial, sendo em seguida aplicado um procedimento que consiste na aplicação direta das leis de Kirchhoff em dois passos. No primeiro passo, partindo dos nós terminais em direção ao nó fonte, calculam-se as correntes nos ramos ("Backward Sweep").

Em CHEN et al. (1991) uma formulação Zbus, onde o método de Gauss é aplicado, foi descrita. Aplicando o princípio da superposição, considera-se neste caso que a tensão em cada barra é resultante de dois componentes: tensões especificadas para barras do tipo PV e injeções de correntes para barras do tipo PQ.

Em DAS et al. (1994, 1995), os módulos das tensões nodais são escritos em função do somatório das potências ativa e reativa das cargas e em função do somatório das perdas. Posteriormente, a partir do nó fonte em direção aos nós terminais ("Forward Sweep"), determina-se a solução do fluxo de carga.

Com o decorrer dos anos, devido ao grande desenvolvimento da informática, os engenheiros e pesquisadores voltaram a considerar as formulações trifásicas. Além disso, incorporaram novas funções, como análise em tempo real e estimação de estados, aos programas de fluxo de carga.

Um algoritmo trifásico desacoplado, explorando a característica radial dos sistemas de distribuição, no qual um esquema de ordenação dos ramos laterais é aplicado visando a redução do número de equações é descrito em ZIMMERMAN (1995). Contudo, esse trabalho é deficiente na representação de unidades de cogeração.

Formulações que adotam o método de Newton-Raphson e são baseadas nas equações de injeção de corrente foram apresentadas em LIN et al. (1999) e MOON et al. (1999). Porém esses métodos não permitem a inclusão de barras do tipo PV e as simplificações adotadas não permitem a representação de equipamentos de controle.

Em GARCIA (2001) apresentou-se uma formulação baseada na metodologia proposta em COSTA et al. (1999), desenvolvida para sistemas em EAT e UAT, que se mostrou 30% mais rápida que o método de Newton-Raphson convencional. Na formulação proposta, as equações das correntes injetadas, para cada fase, são escritas em coordenadas retangulares o que resulta numa matriz Jacobiana formada por blocos (6 x 6) muito próxima da matriz admitância de barras, sendo a diferença determinada pelo modelo de carga adotado. A metodologia criada foi denominada fluxo de potência pelo método de injeção de correntes trifásico – MICT. Porém o referido método, em sua modelagem, utiliza uma consideração que é válida apenas para sistemas equilibrados ou solidamente aterrados em todas as barras, consideradas as tensões de neutro sempre são

nulas em todas as barras do sistema. Sabe-se que esta consideração é incorreta para sistemas desequilibrados, uma vez que os mesmos possuem tensões de neutro diferentes de zero, exceto em neutros solidamente aterrados (condição teórica de impedância de aterramento igual a zero).

Existem diversas configurações para sistemas trifásicos. Em SHORT et al. (2002) é simulado e construído um sistema de distribuição a cinco condutores. Já em WARD et al. (2003) é realizada uma análise de sistemas de distribuição a cinco condutores, comparando-se suas características com as dos sistemas a quatro condutores.

Em CIRIC et al. (2003) é proposto um algoritmo de fluxo de potência para redes radiais de distribuição trifásicas, a quatro condutores, considerando aterramento de neutro, baseado na técnica forward-backward. Esta técnica pode ser classificada como um método de soma de correntes, método de soma de potência, e método de soma de admitâncias. Neste algoritmo, tanto o fio neutro, quanto a terra são explicitamente representados. Porém, este método não se comporta bem quando aplicado a sistemas reticulados, apresenta dificuldade de convergência em alguns casos, especialmente para sistemas com relação R/X elevada, e não possibilita a correta representação de controles e de geração dispersa.

Em PENIDO (2004) foi apresentada uma formulação para o fluxo de potência para sistemas trifásicos a quatro condutores (três fases e o neutro), utilizando o método de Newton-Raphson para solucionar o conjunto de equações de injeção de corrente em coordenadas retangulares. O equacionamento proposto resulta em um sistema de equações não-lineares com dimensão 8*n*, onde *n* é o número de barras do sistema. Também foram modelados os equipamentos com representação explícita de neutros e aterramentos. A metodologia proposta foi utilizada para análise de sistemas equilibrados ou desequilibrados, para sistemas radiais ou reticulados, com cargas ou ramais monofásicos, bifásicos e trifásicos, podendo ser utilizada em sistemas de transmissão, subtransmissão e distribuição, e para sistemas de grande porte.

1.2.2 Modelagem Orientada a Objetos (MOO) e Interface Gráfica com o Usuário

Na literatura foram publicados diversos trabalhos para modelar sistemas elétricos utilizando a Modelagem Orientada a Objetos, destacam-se as seguintes publicações, (NEYER e WU, 1990; HAKAVIC e HOLEN, 1994; ZHOU, 1996; ESQUIVEL et alii,1998; MANZONI et alii, 1999;AGOSTINI et alli, 2000, etc.).

As primeiras aplicações das técnicas de orientação a objetos na modelagem de redes elétricas apareceram na década de 90 (Neyer e Wu, 1990; Hakavic e Holen, 1994; Zhou, 1996; Esquivel *et alii*, 1998; Manzoni *et alii*, 1999; Agostini *et alii*, 2000).

A referência (Neyer e Wu, 1990) mostra o uso de POO aplicada ao problema de fluxo de potência, onde é apresentada a conceituação básica, mas as classes ainda não foram colocadas de forma modular.

É apresentado em (Hakavic e Holen, 1994) a montagem da matriz admitância de barras utilizando técnicas de esparsidade, resolvendo-se o sistema de equações lineares utilizando rotinas da biblioteca NAG, desenvolvidas em FORTRAN (NAG Fortran Library, 1990).

O artigo (Zhou, 1996) descreve o uso de POO aplicada ao problema de fluxo de potência, onde o uso excessivo de classes "templates" é claramente detectado e a modularidade da estrutura de classes não é caracterizada.

Uma descrição sobre a inclusão de dispositivos FACTS (Oliveira *et alii*, 2000) em sistemas de potência usando POO é apresentada em (Esquivel *et alii*, 1998). A principal desvantagem da modelagem apresentada é a inclusão das funções de solução de sistemas lineares dentro da classe Load_Flow, comprometendo a modularidade do protótipo.

A referência (Manzoni *et alii*, 1999) descreve o uso de POO aplicada ao problema de estabilidade dinâmica de sistemas elétricos potência. A estrutura hierárquica das classes deste protótipo é melhorada em (Agostini *et alii*, 2000), sendo esta a primeira aplicação em sistemas de potência que utiliza a técnica de modelagem de objetos (TMO) (Rumbaugh *et alii*, 1994).

Em (ARAUJO, 2000) todos os problemas detectados nos artigos anteriores foram corrigidos e a estrutura de dados mostrou-ser eficiente para a modelagem matemática de redes. No entanto, alguns dados e procedimentos referentes a metodologia de cálculo aplicada no trabalho estão presentes nas classes dos elementos que compõe a rede, dificultando o entendimento, manutenção e modularidade da aplicação.

Uma das primeiras referencias na literatura sobre aplicações para sistemas de potencia com interface gráfica pode ser vista em (Foley *et alii*, 1993). Este trabalho demonstra um aplicativo orientado a objetos, programa sobre o ambiente X Window do Unix. No entanto, a aplicação não apresenta métodos numéricos próprios, sendo estes executados por rotinas externas ao programa.

LI E SHAHIDEHPOUR, 1993 mostram uma aplicação puramente de interface gráfica baseada em PC utilizando a plataforma Microsoft Windows 3.1. No trabalho é ressaltada a importância da representação gráfica dos sistemas elétricos em detrimento a aplicações com informações textuais. Da mesma forma que (Foley *et alii*, 1993), a aplicação de calculo fica externa a interface, prejudicando assim o desempenho computacional da aplicação

A referencia (FLINN, 1995) apresenta uma proposta de aplicação, com uma base de dados única para a realização de diversos estudos elétricos (fluxo de potencia, curto-circuito, estabilidade, etc.). Neste mesmo trabalho são mostradas as facilidades de visualização dos resultados obtidos com as simulações utilizando código de cores nos elementos.

Uma forma de utilização da linguagem orientada a objetos sobre o ambiente Windows para o desenvolvimento de uma interface amigável com o usuário pode ser vista em (HATZIAGYRIOU et alli, 1998). Outro ponto de destaque nesse trabalho ‘e a utilização de ferramentas conhecidas como RAD (*“Rapid Application Modelling”*), que são ferramentas com as quais se constroem aplicações gráficas arrastando e posicionando componentes com o mouse, sem a necessidade de programação “escrita”. Além disso, o autor fala sobre a programação modularizada, onde vários grupos podem trabalhar sobre uma mesma aplicação, mantendo sempre a mesma base.

Em (ABORESHAID e BASOUDAM, 1999) um aplicativo gráfico é apresentado com base no Microsoft Visual Basic e tem como objetivo facilitar o aprendizado de sistemas elétricos, em especial de confiabilidade.

Em (KOBER et alli, 2003) as vantagens da modelagem orientada a objetos (MOO) são demonstradas, nos casos onde a aplicação computacional é desenvolvida por vários grupos em conjunto (modularidade da aplicação). Além disso é mostrado que um aplicativo moderno deve conter a parte gráfica e a parte de cálculo em um único pacote a fim de tornar a aplicação mais rápida e eficiente. Para a implementação da parte gráfica, o autor utiliza as funcionalidades da biblioteca gráfica OpenGL.

1.3 Objetivos da Dissertação

Este trabalho propõe o desenvolvimento, utilizando os conceitos da (MOO) (JONES, 2000), de uma aplicação computacional para análise de Sistemas Elétricos de

Potência integrada com uma Interface Gráfica com o Usuário (GUI) utilizando para tal a linguagem de programação C++.

O método desenvolvido para a análise de redes foi o Fluxo de Potência Trifásico em Coordenadas Polares (FPTP) que é uma extensão método de fluxo de potência monofásico em coordenadas polares (Monticelli, 1983).

A interface gráfica foi concebida utilizando-se a tecnologia biblioteca gráfica OpenGL (NEIDER e DAVIS,1996, WRIGHT e SWEET,1996) o que possibilitou uma grande robustez gráfica da aplicação.

1.4 Organização do Texto

A tese é divida em 5 capítulos, incluindo o capítulo de introdução, além de 4 apêndices, que serão resumidamente descritos a seguir.

No capítulo 2 é desenvolvida a modelagem do Fluxo de Potência Trifásico Polar – FPTP. Também é apresentada a modelagem dos diversos equipamentos que compõe o sistema elétrico (linhas, geradores, transformadores, cargas, etc.). No final é apresentado um pequeno sistema teste para que o leitor possa reproduzir passo-a-passo o método.

A modelagem das classes criadas para o desenvolvimento de uma plataforma de multiaplicativos e também a interface gráfica são descritas no capítulo 3. Estas classes foram projetadas para acomodar qualquer metodologia além dos dados gráficos dos diversos elementos do sistema. Como resultado, é apresentado o aplicativo desenvolvido utilizando a biblioteca gráfica OpenGL e o Microsoft Visual C++.

O capítulo 4 é didicado aos resultados dos sistemas testes utilizados para validar a metodologia matemática proposta neste trabalho.

No capítulo 5 são apresentadas as principais conclusões deste trabalho assim como sugestões para trabalhos futuros.

Visando uma apresentação mais didática deste trabalho, alguns conceitos importantes foram disponibilizados nos apêndices. No Apêndice A apresenta-se os conceitos de MOO e UML amplamente utilizados nesta dissertação. Este apêndice será fundamental para uma perfeita compreensão dos gráficos, fluxogramas e dos modelos utilizados.

No Apêndice B são mostradas todas as derivadas que foram deduzidas para o cálculo do fluxo de potência e que complementam o entendimento do Capítulo 2

No Apêndice C são apresentados os fundamentos da biblioteca gráfica OpenGL que foi utilizada no desenvolvimento das interface gráfica da aplicação.

No Apêndice D são apresentadas todas as características e funcionalidades aplicativo desenvolvido no Microsoft Visual C++.

1.5 Publicações Decorrentes Deste Trabalho

"Ambiente Gráfico para Análise de Fluxo de Potência Trifásico Utilizando OpenGL". XV Congresso Brasileiro de Automática, 2004, Gramado. Anais do XV CBA, 2004. v. 1. p. 1-6.

"Ambiente Gráfico para Análise de Fluxo de Potência Trifásico Utilizando OpenGL", *I Semana de Potência, Automação e Controle, Juiz de Fora, MG, Outubro 2003.*

"Análise de Sistemas de Distribuição Utilizando Modelagem Orientada a Objetos", *I Semana de Potência, Automação e Controle, Juiz de Fora, MG, Outubro 2003.*

1.6 Convenções e Nomenclaturas Utilizadas

Neste item são apresentadas algumas das convenções e nomenclaturas utilizadas nesta dissertação, com o objetivo de tornar mais fácil a leitura do trabalho e evitar possíveis interpretações errôneas do texto.

Na Tabela 1.1 é apresentada a convenção utilizada para distinção dos tipos de variáveis utilizadas:

Tabela 1.1 – Convenções adotadas para escrita de variáveis

Tipo da variável	Tipo de escrita	Exemplo
Escalar Real	Minúscula em itálico	x
Escalar Complexo	Maiúscula em itálico	X
Vetor	Minúscula em negrito	$\mathbf{x}$
Matriz	Maiúscula em negrito	$\mathbf{X}$

Todos os vetores são considerados como vetores colunas. Um vetor linha é representado pelo transposto de um vetor coluna (Exemplo $\mathbf{c}^t$).

Na tabela a seguir apresenta-se os símbolos utilizados para designar funções ou operações, com seus respectivos significados.

Tabela 1.2 – Convenções adotadas para funções e operações

Símbolo	Exemplo	Significado
t sobrescrito em itálico	$\mathbf{A}^t$	Matriz transposta de **A**
* sobrescrito	X^*	Conjugado do complexo X
u.m.	10 u.m	Unidades Monetárias
abs()	abs*(X)*	Valor absoluto (módulo) do complexo X
| |	$\|X\|$	Valor absoluto (módulo) do complexo X
arg()	arg(X)	Argumento (ângulo) em radianos do complexo X.
"Re" subscrito	X_{Re}	Parte real do complexo X
$\Re(\ )$	$\Re(X)$	Parte real do complexo X
"Im" subscrito	X_{Im}	Parte imaginária do complexo X
$\Im(\ )$	$\Im(X)$	Parte imaginária do complexo X
"s" sobrescrito	V^s	Variável de fase, s={a,b,c}
"t" sobrescrito	V^t	Variável de fase, t={a,b,c}
"~" acima	$\widetilde{X}_{\mathrm{Re}}$	Derivada da função Lagrangeana em relação a variável $\widetilde{X}_{\mathrm{Re}} = \dfrac{\partial L}{\partial X_{\mathrm{Re}}}$

A seguir são mostrados os símbolos mais freqüentes utilizados para designar variáveis, grandezas ou entidades matemáticas.

Tabela 1.3 – Convenções adotadas para funções e operações

Símbolo	Exemplo	Significado
0	**0**	Matriz ou vetor nulo
a	a	Complexo $e^{j\frac{2\pi}{3}}$
j	j	Complexo unitário, igual a uma das raízes quadradas de -1 (a raiz positiva), ou seja, $j=+\sqrt{-1}$
C	C	Capacitância
L	L	Indutância
R, r	R, r	Resistência
x	x	Reatância
z	z	Impedância (z=r+jx)
g	g	Condutância
b	b	Susceptância
y	y	Admitância (y=g+jb)
v	v	Módulo da tensão
i	i	Módulo da corrente
θ	θ	Ângulo em radianos
V	V	Tensão complexa $V=ve^{j\theta}$
I	I	Corrente complexa $I=ie^{j\theta}$
P	P	Potência Ativa
Q	Q	Potência Reativa
S	S	Potência Aparente (S=P+jQ)
Δ	Δx	Pequeno desvio de uma variável em relação ao valor de regime permanente
J	**J**	Matriz Jacobiana
H	**H**	Matriz Hessiana
L	L	Função Lagrangeana

As derivadas de funções complexas em relação aos parâmetros reais **x** podem ser calculadas diretamente utilizando-se as seguintes propriedades (1.1):

$$\frac{\partial \Re[f]}{\partial x_1} = \Re\left[\frac{\partial f}{\partial x_1}\right] \qquad \frac{\partial \Im[f]}{\partial x_1} = \Im\left[\frac{\partial f}{\partial x_1}\right]$$
$$\frac{\partial^2 \Re[f]}{\partial x_1 \partial x_2} = \Re\left[\frac{\partial^2 f}{\partial x_1 \partial x_2}\right] \qquad \frac{\partial^2 \Im[f]}{\partial x_1 \partial x_2} = \Im\left[\frac{\partial^2 f}{\partial x_1 \partial x_2}\right] \tag{1.1}$$

Seja **A** uma matriz ou vetor de funções, como apresentado em (1.2):

$$\mathbf{A} = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,n} \\ f_{2,1} & f_{2,2} & & f_{2,n} \\ \vdots & & \ddots & \\ f_{m,1} & f_{m,2} & & f_{m,n} \end{bmatrix} \tag{1.2}$$

As derivadas matriciais de primeira ordem em relação a uma variável x_1 são dadas por (1.3):

$$\frac{\partial \mathbf{A}}{\partial x_1} = \mathbf{A}'_{x_1} = \begin{bmatrix} \frac{\partial f_{1,1}}{\partial x_1} & \frac{\partial f_{1,2}}{\partial x_1} & \cdots & \frac{\partial f_{1,n}}{\partial x_1} \\ \frac{\partial f_{2,1}}{\partial x_1} & \frac{\partial f_{2,2}}{\partial x_1} & & \frac{\partial f_{2,n}}{\partial x_1} \\ \vdots & & \ddots & \\ \frac{\partial f_{m,1}}{\partial x_1} & \frac{\partial f_{m,2}}{\partial x_1} & & \frac{\partial f_{m,n}}{\partial x_1} \end{bmatrix} \tag{1.3}$$

As derivadas matriciais de segunda ordem em relação ao par de variáveis (x_1, x_2) são dadas por (1.4):

$$\frac{\partial^2 \mathbf{A}}{\partial x_1 \partial x_2} = \mathbf{A}''_{x_1,x_2} = \begin{bmatrix} \frac{\partial^2 f_{1,1}}{\partial x_1 \partial x_2} & \frac{\partial^2 f_{1,2}}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f_{1,n}}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f_{2,1}}{\partial x_1 \partial x_2} & \frac{\partial^2 f_{2,2}}{\partial x_1 \partial x_2} & & \frac{\partial^2 f_{2,n}}{\partial x_1 \partial x_2} \\ \vdots & & \ddots & \\ \frac{\partial^2 f_{m,1}}{\partial x_1 \partial x_2} & \frac{\partial^2 f_{m,2}}{\partial x_1 \partial x_2} & & \frac{\partial^2 f_{m,n}}{\partial x_1 \partial x_2} \end{bmatrix} \tag{1.4}$$

As funções $\Re(\ )$ e $\Im(\ )$ matriciais são definidas por (1.5)

$$\Re(\mathbf{A}) = \begin{bmatrix} \Re(f_{1,1}) & \Re(f_{1,2}) & \cdots & \Re(f_{1,n}) \\ \Re(f_{2,1}) & \Re(f_{2,2}) & & \Re(f_{2,n}) \\ \vdots & & \ddots & \\ \Re(f_{m,1}) & \Re(f_{m,2}) & & \Re(f_{m,n}) \end{bmatrix}$$

$$\Im(\mathbf{A}) = \begin{bmatrix} \Im(f_{1,1}) & \Im(f_{1,2}) & \cdots & \Im(f_{1,n}) \\ \Im(f_{2,1}) & \Im(f_{2,2}) & & \Im(f_{2,n}) \\ \vdots & & \ddots & \\ \Im(f_{m,1}) & \Im(f_{m,2}) & & \Im(f_{m,n}) \end{bmatrix} \quad \textbf{(1.5)}$$

Capítulo 2 Fluxo de Potência Trifásico em Coordenadas Polares

2.1 Introdução

O cálculo de fluxo de potência em uma rede de energia elétrica consiste basicamente na determinação dos estados da rede (tensões e ângulos nas barras), da distribuição dos fluxos e de outras grandezas de interesse. Nesse tipo de problema, a modelagem do sistema é estática, significando que a rede é representada por um conjunto de equações algébricas. Esse tipo de representação é utilizado em situações nas quais as variações com o tempo são suficientemente lentas para que se possa ignorar os efeitos transitórios (Monticelli, 1983).

Para a maioria dos programas de análise de sistemas de potência em estado permanente, o desbalanço do sistema é ignorado utilizando-se modelagem de seqüência positiva como descrita em (Monticelli, 1983). Estas simplificações foram adotadas principalmente devido às limitações dos recursos computacionais da época e foi extremamente difundida nas décadas de 70 e 80, sendo muito utilizada até os dias de hoje. No entanto, na prática, as cargas não são completamente balanceadas e existem linhas desequilibradas devido à linhas não transpostas (ARRILLAGA, 1983) e à linhas de potência natural elevada LPNE (GOMES, 2000).

Entre os efeitos do desbalanço do sistema de potência pode-se citar a corrente de seqüência negativa, que causa aquecimento do rotor da máquina e a corrente de seqüência zero que causa má operação de relés e aumenta as perdas em linhas não transpostas.

Tendo em vista a disponibilidade de recursos computacionais adequados e à crescente demanda por programas computacionais que representam adequadamente as redes desequilibradas, apresenta-se neste capítulo uma extensão do método de fluxo de potência monofásico em coordenadas polares (MONTICELLI, 1983) para sistemas trifásicos desequilibrados. Esta metodologia tem grande aplicabilidade em sistemas de distribuição e subtransmissão de energia elétrica. A formulação trifásica é desenvolvida a partir das equações de fluxo de potência, que são escritas em coordenadas polares.

2.2 Desenvolvimento Matemático

Neste tópico é apresentado o desenvolvimento matemático das equações algébricas para a resolução do Fluxo de Potência Trifásico em Coordenadas Polares (FPTP) para sistemas equilibrados e desequilibrados.

2.2.1 Equações Básicas do Fluxo de Potência

Considerando um sistema trifásico, a corrente que flui da barra *k* para a barra *m* é dada por:

$$I_{km}^{s} = \sum_{t \in \alpha_p} Y^{st}(E_k^t - E_m^t) + jB_{shunt}E_k^s \qquad \textbf{(2.1)}$$

onde:

Y^{st} : Admitância série da linha k-m

E_k^t : Fasor de tensão na barra *k*

E_m^t : Fasor de tensão na barra *m*

B_{shunt} : Susceptância em derivação na barra *k*

$s,t \in \alpha$

$\alpha = \{a,b,c\}$

Multiplicando a corrente I_{km} pela tensão na barra k, encontramos a potência complexa conjugada que flui da barra k para barra m.

$$S^s = P^s + jQ^s \qquad \textbf{(2.2)}$$

$$S^s = E_k^s I_k^{s^*} \qquad \textbf{(2.3)}$$

$$S^{s^*} = E_k^{s^*} I_k^s \qquad \textbf{(2.4)}$$

$$S^{s*} = v_k^s e^{-j\theta_k^s}\left[\sum_{t \in \alpha} Y^{st}(v_k^t e^{j\theta_k^t} - v_m^t e^{j\theta_m^t}) + jB_{shunt}v_k^s e^{j\theta_k^s}\right] \qquad \textbf{(2.5)}$$

Desenvolvendo a Equação 2.5 temos:

$$S^{s*} = \sum_{t \in \alpha} Y^{st}(v_k^s v_k^t e^{-j\theta_k^s} e^{j\theta_k^t} - v_k^s v_m^t e^{-j\theta_k^s} e^{j\theta_m^t}) + jB_{shunt}v_k^s v_k^{s-j\theta_k^s} e^{j\theta_k^t} \qquad \textbf{(2.6)}$$

$$S^{s*} = \sum_{t\in\alpha} Y^{st} \begin{Bmatrix} v_k^s v_k^t \left[\cos(-\theta_k^s + \theta_k^t) + jsen(-\theta_k^s + \theta_k^t)\right] \\ -v_k^s v_m^t \left[\cos(-\theta_k^s + \theta_m^t) + jsen(-\theta_k^s + \theta_m^t)\right] \end{Bmatrix} \quad \textbf{(2.7)}$$
$$+ jB_{shunt} v_k^{s2} \left[\cos(-\theta_k^s + \theta_k^s) + jsen(-\theta_k^s + \theta_k^s)\right]$$

$$S^{s*} = \sum_{t\in\alpha_k} (G^{st} + jB^{st}) \begin{Bmatrix} v_k^s v_k^t \left[\cos(-\theta_k^s + \theta_k^t) + jsen(-\theta_k^s + \theta_k^t)\right] \\ -v_k^s v_m^t \left[\cos(-\theta_k^s + \theta_m^t) + jsen(-\theta_k^s + \theta_m^t)\right] \end{Bmatrix} \quad \textbf{(2.8)}$$
$$+ jB_{shunt} v_k^{s2} \left[\cos(-\theta_k^s + \theta_k^s) + jsen(-\theta_k^s + \theta_k^s)\right]$$

onde:

S^s : Potência complexa da *fase s*

v_k^s : Módulo da tensão da *fase s* na barra k

θ_k^s : Ângulo do fasor tensão na *fase s* na barra k

Separando a Equação 2.8 em partes real e imaginária temos:

$$P^s = \sum_{t\in\alpha} G^{st} v_k^s v_k^t \cos(\theta_k^s - \theta_k^t) + B^{st} v_k^s v_k^t sen(\theta_k^s - \theta_k^t) - G^{st} v_k^s v_m^t \cos(\theta_k^s - \theta_m^t) \quad \textbf{(2.9)}$$
$$- B^{st} v_k^s v_m^t sen(\theta_k^s - \theta_m^t)$$

$$Q^s = \sum_{t\in\alpha_k} G^{st} v_k^s v_k^t sen(\theta_k^s - \theta_k^t) - B^{st} v_k^s v_k^t \cos(\theta_k^s - \theta_k^t) - G^{st} v_k^s v_m^t sen(\theta_k^s - \theta_m^t) \quad \textbf{(2.10)}$$
$$+ B^{st} v_k^s v_m^t \cos(\theta_k^s - \theta_m^t) + B_{shunt} v_k^{s2}$$

onde:

Pc_k^s : Potência ativa da carga na fase s da barra k

Pg_k^s : Potência ativa gerada na fase s da barra k

Qc_k^s : Potência reativa da carga na fase s da barra k

Qg_k^s : Potência reativa da carga na fase s da barra k

G_{ki}^{st} : Condutância série do ramo $k - i$

B_{ki}^{st} : Susceptância série do ramo $k - i$

Ω_k : Conjunto de barras conectadas diretamente à barra k.

2.2.2 Solução do Sistema de Equações Utilizando o Método de Newton Raphson

Para a solução de sistemas de equações não-lineares é comumente utilizado o Método de Newton-Rapshon. Escrevendo-se as equações para todas as barras do sistema, das três fase (a,b,c), resulta em um sistema de equações não-lineares de ordem seis vezes o número de barras(6*n*). Este algoritmo é bastante utilizado pela sua simplicidade e ótimo desempenho computacional.

Considerando o sistema de equações dado por:

$$g(x) = 0 \qquad \textbf{(2.11)}$$

$$g(x) = [g_1(x), g_2(x), ..., g_n(x)]^t$$

$$x = [x_1, x_2, ..., x_n]^t \qquad \textbf{(2.12)}$$

sendo $g(x)$ a função vetorial e x o vetor das incógnitas.

A linearização da função vetorial $g(x)$ para $x = x^v$ é dada pelos dois primeiros termos da série de Taylor:

$$g(x^v + \Delta x^v) \cong g(x^v) + J(x^v)\Delta x^v \qquad \textbf{(2.13)}$$

Sendo J a matriz jacobiana dada por:

$$J = \frac{\partial g}{\partial x} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n} \end{bmatrix} \qquad \textbf{(2.14)}$$

O vetor de correção Δx é calculado impondo-se que:

$$J(x)\Delta x = -g(x) \qquad \textbf{(2.15)}$$

O problema do Fluxo de Potência Trifásico Polar, na forma matricial, é representado como mostra a equação a Equação **(2.16)** .

$$\begin{bmatrix} \ddots & & \vdots & & \iddots \\ \cdots & \dfrac{\partial Q_k^{\alpha}}{\partial \theta_k^{\beta}} \quad \dfrac{\partial Q_k^{\alpha}}{\partial V_k^{\beta}} & \cdots & \dfrac{\partial Q_k^{\alpha}}{\partial \theta_m^{\beta}} \quad \dfrac{\partial Q_k^{\alpha}}{\partial V_m^{\beta}} & \cdots \\ & \dfrac{\partial P_k^{\alpha}}{\partial \theta_k^{\beta}} \quad \dfrac{\partial P_k^{\alpha}}{\partial V_k^{\beta}} & & \dfrac{\partial P_k^{\alpha}}{\partial \theta_m^{\beta}} \quad \dfrac{\partial P_k^{\alpha}}{\partial V_m^{\beta}} & \\ \cdots & \dfrac{\partial Q_m^{\alpha}}{\partial \theta_k^{\beta}} \quad \dfrac{\partial Q_m^{\alpha}}{\partial V_k^{\beta}} & \cdots & \dfrac{\partial Q_m^{\alpha}}{\partial \theta_m^{\beta}} \quad \dfrac{\partial Q_m^{\alpha}}{\partial V_m^{\beta}} & \cdots \\ & \dfrac{\partial P_m^{\alpha}}{\partial \theta_k^{\beta}} \quad \dfrac{\partial P_m^{\alpha}}{\partial V_k^{\beta}} & & \dfrac{\partial P_m^{\alpha}}{\partial \theta_m^{\beta}} \quad \dfrac{\partial P_m^{\alpha}}{\partial V_m^{\beta}} & \\ \iddots & & \vdots & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta\theta_k^{\alpha} \\ \Delta V_k^{\alpha} \\ \vdots \\ \Delta\theta_m^{\alpha} \\ \Delta V_m^{\alpha} \\ \vdots \end{bmatrix} = - \begin{bmatrix} \vdots \\ \Delta P_k^{\alpha} \\ \Delta Q_k^{\alpha} \\ \vdots \\ \Delta P_m^{\alpha} \\ \Delta Q_m^{\alpha} \\ \vdots \end{bmatrix} \qquad \textbf{(2.16)}$$

Para a solução deste sistema de equações adota-se um processo iterativo, como fica esquematizado na Figura 2.1.

As derivadas que aparecem na matriz jacobiana são mostradas no Apêndice B.

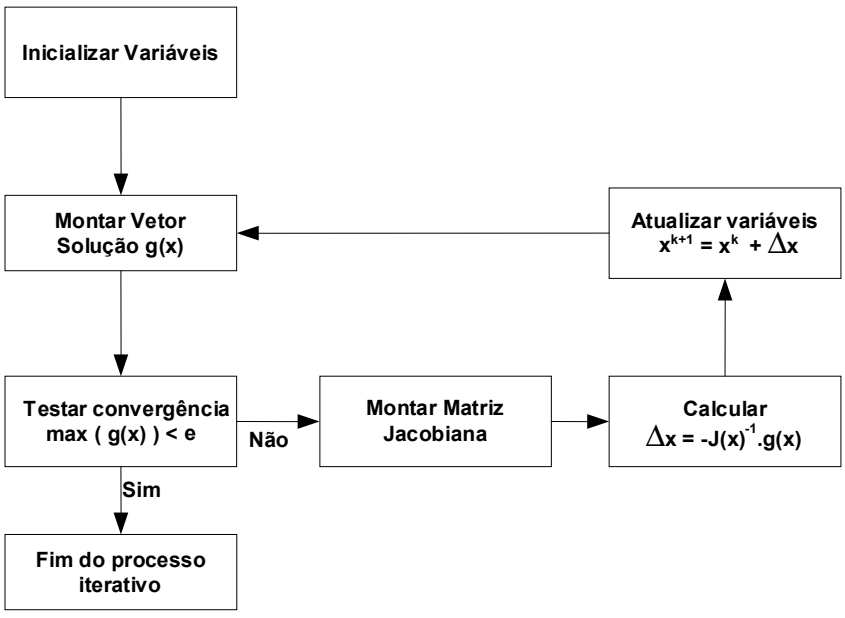

Figura 2.1- Algoritmo de Solução do Problema

2.3 Modelagem dos Equipamentos da Rede

Cada componente do sistema de potência – linhas, cargas, geradores, reatores, transformadores, etc. – deve ser analisado para se determinar os parâmetros, necessários para o cálculo do fluxo de potência trifásico.

Neste caso, a implementação matemática para a análise multifásica dos sistemas elétricos necessita considerar o desequilíbrio dos sistemas físicos reais.

A seguir será mostrada a modelagem trifásica dos elementos que compõem um sistema elétrico de potência.

2.3.1 Modelo de Linhas

Para a análise em regime permanente, as linhas trifásicas são modeladas por seus parâmetros de indutâncias, capacitâncias e resistências através de um circuito π equivalente a parâmetros concentrados. Este circuito pode ser observado na Figura 2.2 enquanto na Figura 2.3 ilustra-se o circuito π equivalente de uma linha de transmissão trifásica na forma matricial.

Como pode ser observado este modelo permite a representação de linhas com parâmetros assimétricos, muito comuns em sistemas de distribuição e em LPNE (GOMES, 2000).

A Equação **(2.17)** representa a matriz de impedância série e a Equação **(2.18)** representa a matriz admitância "shunt" da linha.

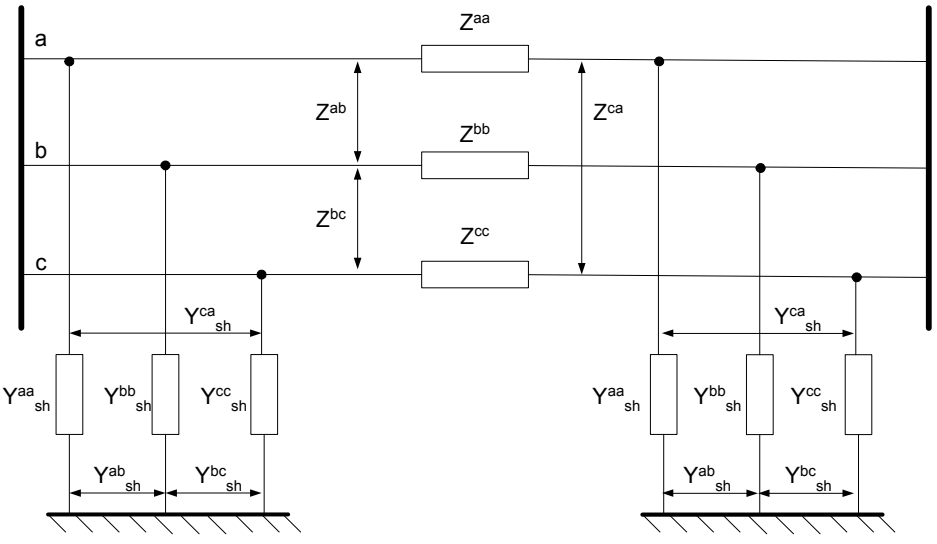

Figura 2.2 - Circuito Equivalente π de uma linha trifásica a parâmetros concentrados

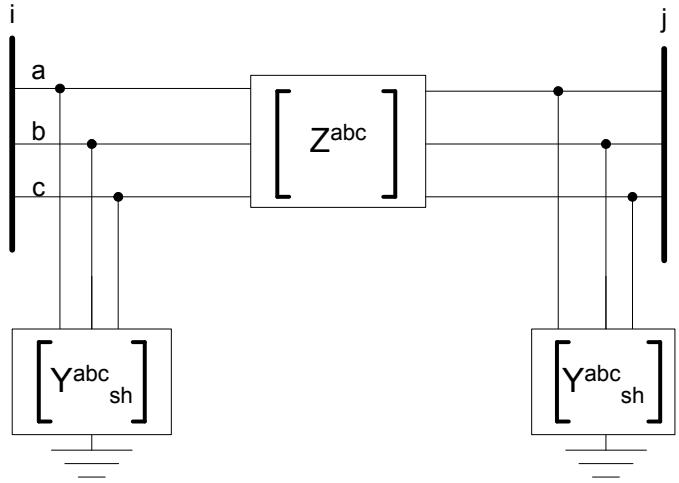

Figura 2.3 - Circuito π equivalente de uma linha de transmissão trifásica na forma matricial

$$\left[Z_{ij}^{abc}\right] = \begin{bmatrix} Z_{ij}^{aa} & Z_{ij}^{ab} & Z_{ij}^{ac} \\ Z_{ij}^{ba} & Z_{ij}^{bb} & Z_{ij}^{bc} \\ Z_{ij}^{ca} & Z_{ij}^{cb} & Z_{ij}^{cc} \end{bmatrix} = \begin{bmatrix} r_{ij}^{aa} & r_{ij}^{ab} & r_{ij}^{ac} \\ r_{ij}^{ba} & r_{ij}^{bb} & r_{ij}^{bc} \\ r_{ij}^{ca} & r_{ij}^{cb} & r_{ij}^{cc} \end{bmatrix} + j \begin{bmatrix} x_{ij}^{aa} & x_{ij}^{ab} & x_{ij}^{ac} \\ x_{ij}^{ba} & x_{ij}^{bb} & x_{ij}^{bc} \\ x_{ij}^{ca} & x_{ij}^{cb} & x_{ij}^{cc} \end{bmatrix} \quad \textbf{(2.17)}$$

$$\left[Y_{shij}^{abc}\right] = j \begin{bmatrix} b_{shij}^{aa} & b_{shij}^{ab} & b_{shij}^{ac} \\ b_{shij}^{ba} & b_{shij}^{bb} & b_{shij}^{bc} \\ b_{shij}^{ca} & b_{shij}^{cb} & b_{shij}^{cc} \end{bmatrix} \quad \textbf{(2.18)}$$

Entretanto, é comum em sistemas de distribuição a presença de derivações monofásicas e bifásicas. Para representar esses elementos, considera-se que a fase não existente possui uma impedância série infinita. Matematicamente, adota-se o artifício de substituir a impedância própria da fase inexistente por um número de valor elevado, como pode ser visto na Equação **(2.19)**. Dessa forma, para um ramo bifásico constituído pelas fases a e c, tem-se:

$$\left[Z_{ij}^{abc}\right] = \begin{bmatrix} Z_{ij}^{aa} & 0 & Z_{ij}^{ac} \\ 0 & 10^{15} & 0 \\ Z_{ij}^{ca} & 0 & Z_{ij}^{cc} \end{bmatrix} \quad \textbf{(2.19)}$$

Para a matriz admitância do circuito π equivalente, substitui-se a susceptância por zero.

As contribuições das linhas trifásicas no fluxo de potência são mostradas nas Equações **(2.9)** e. **(2.10)**.

2.3.2 Modelo de Cargas

Para a análise de sistemas elétricos de potência, deve-se considerar dois tipos básicos de conexão para as cargas:

- Conexão Estrela aterrada;
- Conexão delta.

Outro aspecto importante a ser considerado é a representação dos efeitos da tensão sobre as cargas do sistema, visto que o nível de carregamento da rede é diretamente proporcional ao nível de tensão (Garcia, 2001). Além disso, para a análise de sistemas de distribuição, deve-se considerar a presença de cargas monofásicas e bifásicas. Neste trabalho serão apresentadas somente as cargas com conexão estrela aterrada devido a sua simplicidade de representação.

A Figura 2.4 mostra uma representação esquemática de cargas monofásicas, bifásicas e trifásicas conectadas em estrela-aterrada. Matematicamente, consideram-se essas cargas através da injeção da potência, que é um parâmetro especificado.

Para representar o efeito da tensão, optou-se por considerar o modelo de carga do tipo polinomial (Monticelli, 1983). Assim, tem-se:

$$S^s = P^s + jQ^s \qquad \textbf{(2.20)}$$

$$P^s = (P^s)^{sp}(A_p + B_p V^s + C_p (V^s)^2) \qquad \textbf{(2.21)}$$

$$Q^s = (Q^s)^{sp}(A_q + B_q V^s + C_q (V^s)^2) \qquad \textbf{(2.22)}$$

$$A_p + B_p + C_p = 1 \qquad \textbf{(2.23)}$$

$$A_q + B_q + C_q = 1 \qquad \textbf{(2.24)}$$

onde:

s = {a,b,c}

$(P^s)^{sp}, (Q^s)^{sp}$: Potência ativa e reativa especificadas.

A_p, A_q: Parcela da carga ativa e reativa modelada como potência constante.

B_p, B_q: Parcela da carga ativa e reativa modelada como corrente constante.

C_p, C_q: Parcela da carga ativa e reativa modelada como impedância constante.

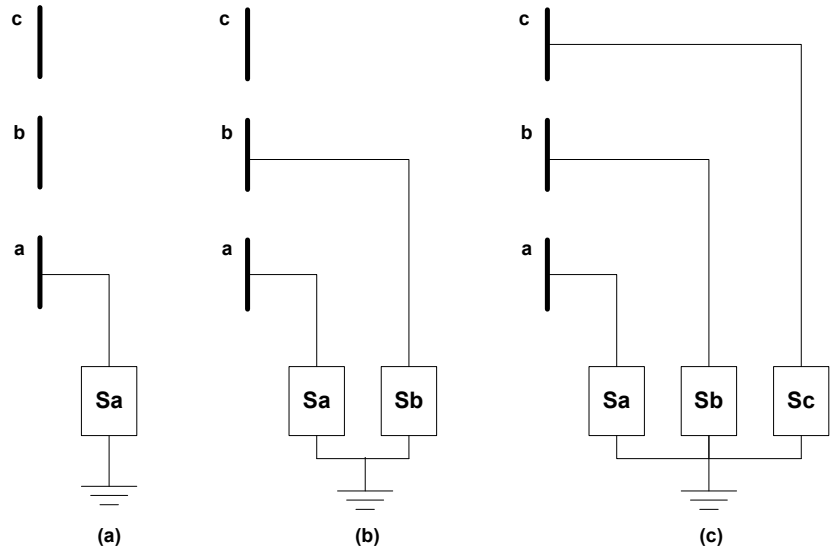

Figura 2.4 - Esquema da ligação para a carga ligada em estrela: (a) Monofásica; (b) Bifásica; (c) Trifásica

As contribuições de carga em Y aterrada conectada na barra k no vetor independente e na matriz jacobiana podem ser vistos nas Equações **(2.25)** e **(2.26)** respectivamente.

$$g(x) = \begin{bmatrix} P_k^{\,a} \\ P_k^{\,b} \\ P_k^{\,c} \\ Q_k^{\,a} \\ Q_k^{\,b} \\ Q_k^{\,c} \end{bmatrix} \qquad \textbf{(2.25)}$$

$$J(x)=\begin{bmatrix} \ddots & & & & & & \iddots \\ & 0 & 0 & 0 & (P_k^a)^{sp}(B_p+2C_pV^a) & 0 & 0 \\ & 0 & 0 & 0 & 0 & (P_k^b)^{sp}(B_p+2C_pV^b) & 0 \\ & 0 & 0 & 0 & 0 & 0 & (P_k^c)^{sp}(B_p+2C_pV^c) \\ & 0 & 0 & 0 & (Q_k^a)^{sp}(B_q+2C_qV^a) & 0 & 0 \\ & 0 & 0 & 0 & 0 & (Q_k^b)^{sp}(B_q+2C_qV^b) & 0 \\ & 0 & 0 & 0 & 0 & 0 & (Q_k^c)^{sp}(B_q+2C_qV^c) \\ \iddots & & & & & & \ddots \end{bmatrix} \quad \textbf{(2.26)}$$

2.3.3 Modelo de Máquina

Neste trabalho, as máquinas tanto síncronas quanto de indução são modeladas como uma injeção de potência ativa e reativa em uma barra do sistema, como mostra a Figura 2.5 .

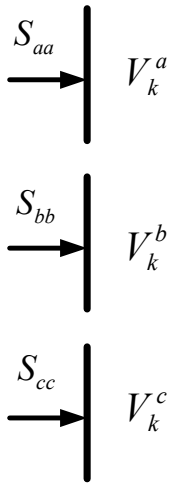

Figura 2.5 - Modelo de máquina trifásica

Neste estudo de fluxo de potência realizado, as máquinas podem ser classificadas em três tipos:

- Referência: A barra onde a máquina está conectada é a barra de folga (barra swing).
- Fixo: A máquina injeta potência ativa e reativa constante na barra
- Controle: A máquina controla a tensão em uma determinada barra do sistema

2.3.3.1 Máquina Tipo Referência

A máquina do tipo referência é aquela que está conectada à barra de folga (barra swing) do sistema.

Para representá-la no fluxo de potência deve-se fixar o módulo de sua tensão (normalmente 1pu) e o ângulo (0˚) . Além disso, deve-se eliminar as equações das potências injetadas na barra. Isso pode ser feito de duas formas:

1. Inserindo um número elevado (ex. 10^{10}) na diagonal referente à barra da matriz jacobiana
2. Zerando a linha e a coluna referente à barra e colocando o número um na diagonal.

No vetor independente deve-se zerar as contribuições da barra de referência

A Equação **(2.27)** mostra o sistema de equações para uma máquina tipo referencia conectada a barra *k*. Utilizou-se o segundo método para eliminar a equação da barra de folga.

$$\begin{bmatrix} \ddots & & & & & & \vdots & & & & & & & \iddots \\ & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \\ \cdots & & & \vdots & & & \ddots & & & \vdots & & & & \cdots \\ & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial P_m^a}{\partial \theta_m^a} & \frac{\partial P_m^a}{\partial \theta_m^b} & \frac{\partial P_m^a}{\partial \theta_m^c} & \frac{\partial P_m^a}{\partial V_m^a} & \frac{\partial P_m^a}{\partial V_m^b} & \frac{\partial P_m^a}{\partial V_m^c} & \\ & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial P_m^b}{\partial \theta_m^a} & \frac{\partial P_m^b}{\partial \theta_m^b} & \frac{\partial P_m^b}{\partial \theta_m^c} & \frac{\partial P_m^b}{\partial V_m^a} & \frac{\partial P_m^b}{\partial V_m^b} & \frac{\partial P_m^b}{\partial V_m^c} & \\ & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial P_m^c}{\partial \theta_m^a} & \frac{\partial P_m^c}{\partial \theta_m^b} & \frac{\partial P_m^c}{\partial \theta_m^c} & \frac{\partial P_m^c}{\partial V_m^a} & \frac{\partial P_m^c}{\partial V_m^b} & \frac{\partial P_m^c}{\partial V_m^c} & \\ & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial Q_m^a}{\partial \theta_m^a} & \frac{\partial Q_m^a}{\partial \theta_m^b} & \frac{\partial Q_m^a}{\partial \theta_m^c} & \frac{\partial Q_m^a}{\partial V_m^a} & \frac{\partial Q_m^a}{\partial V_m^b} & \frac{\partial Q_m^a}{\partial V_m^c} & \\ & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial Q_m^b}{\partial \theta_m^a} & \frac{\partial Q_m^b}{\partial \theta_m^b} & \frac{\partial Q_m^b}{\partial \theta_m^c} & \frac{\partial Q_m^b}{\partial V_m^a} & \frac{\partial Q_m^b}{\partial V_m^b} & \frac{\partial Q_m^b}{\partial V_m^c} & \\ & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\partial Q_m^c}{\partial \theta_m^a} & \frac{\partial Q_m^c}{\partial \theta_m^b} & \frac{\partial Q_m^c}{\partial \theta_m^c} & \frac{\partial Q_m^c}{\partial V_m^a} & \frac{\partial Q_m^c}{\partial V_m^b} & \frac{\partial Q_m^c}{\partial V_m^c} & \\ \iddots & & & & & & \vdots & & & & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta\theta_k^a \\ \Delta\theta_k^b \\ \Delta\theta_k^c \\ \Delta V_k^a \\ \Delta V_k^b \\ \Delta V_k^c \\ \vdots \\ \Delta\theta_m^a \\ \Delta\theta_m^b \\ \Delta\theta_m^c \\ \Delta V_m^a \\ \Delta V_m^b \\ \Delta V_m^c \\ \vdots \end{bmatrix} = - \begin{bmatrix} \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ P_m^a \\ P_m^b \\ P_m^c \\ Q_m^a \\ Q_m^b \\ Q_m^c \\ \vdots \end{bmatrix} \quad \textbf{(2.27)}$$

2.3.3.2 Máquina Tipo Fixo

Este tipo de máquina injeta uma potência ativa e reativa em uma determinada barra do sistema, independente da tensão do barramento no qual está conectada. A

Equação (2.28) mostra a contribuição das máquinas do tipo fixo no sistema de equações.

$$g(x)=\begin{bmatrix} -P_k^{\,a} \\ -P_k^{\,b} \\ -P_k^{\,c} \\ -Q_k^{\,a} \\ -Q_k^{\,b} \\ -Q_k^{\,c} \end{bmatrix} \qquad \textbf{(2.28)}$$

Como as potências ativa e reativa da máquina, neste estudo, não dependem da tensão e ângulo da barra, a mesma não contribui com derivadas na matriz Jacobiana.

2.3.3.3 Máquina Tipo Controle

A máquina do tipo controle permite controlar o nível tensão em uma determinada barra do sistema através do controle de excitação do seu estator, controlando assim a potência reativa injetada na barra de controle. A equação de controle é mostrada na Equação (2.29).

$$V_m^s - V_{esp}^s = 0 \qquad \textbf{(2.29)}$$

O sistema de equações para uma máquina na barra *k* controlando a tensão na barra *m* pode ser visto na Equação **(2.30)**.

$$\left[\begin{array}{cccccccccccc:ccc} & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & -1 & 0 & 0 \\ & & & & & & & & & & & & 0 & -1 & 0 \\ & & & & & & & & & & & & 0 & 0 & -1 \\ & & & & & & JAC & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right] \cdot \begin{bmatrix} \Delta\theta_k^a \\ \Delta\theta_k^b \\ \Delta\theta_k^c \\ \Delta V_k^a \\ \Delta V_k^b \\ \Delta V_k^c \\ \Delta\theta_m^a \\ \Delta\theta_m^b \\ \Delta\theta_m^c \\ \Delta V_m^a \\ \Delta V_m^b \\ \Delta V_m^c \\ \hdashline \Delta Q_k^a \\ \Delta Q_k^b \\ \Delta Q_k^c \end{bmatrix} = - \begin{bmatrix} P_k^a \\ P_k^b \\ P_k^c \\ Q_k^a \\ Q_k^b \\ Q_k^c \\ P_m^a \\ P_m^b \\ P_m^c \\ Q_m^a \\ Q_m^b \\ Q_m^c \\ \hdashline V_m^a - V_{esp}^a \\ V_m^b - V_{esp}^b \\ V_m^c - V_{esp}^c \end{bmatrix} \qquad \textbf{(2.30)}$$

Após cada resolução de sistema de equações, deve atualizar não só os módulos das tensões e os ângulos das barras. Neste caso, deve-se também atualizar a potência reativa injetada, cujo o incremento é calculado na Equação **(2.30)**. Essa atualização pode ser vista na Equação **(2.31)**

$$\begin{aligned} Q_k^{a(h)} &= Q_k^{a(h-1)} + \Delta Q_k^a \\ Q_k^{b(h)} &= Q_k^{b(h-1)} + \Delta Q_k^b \\ Q_k^{c(h)} &= Q_k^{c(h-1)} + \Delta Q_k^c \end{aligned} \qquad \textbf{(2.31)}$$

onde:

h: Iteração

2.3.4 Transformador

O modelo convencional de transformador é feito em termos de componentes simétricas (Stevenson, 1986), porém, o mesmo não pode ser utilizado corretamente para sistemas desequilibrados, uma vez que na sua concepção supõe-se que o sistema de potência é suficientemente balanceado. Especialmente para sistemas de distribuição é necessário considerar uma modelagem que leve em consideração os possíveis desequilíbrios.

Com a modelagem apresentada neste trabalho pode-se representar qualquer combinação de ligações de transformadores de dois enrolamentos, conexões em estrela (Y) ou delta (Δ). A modelagem pode ser usada para uma unidade trifásica onde existem mútuas entre as fases, Figura 2.6 ou blocos de unidades monofásicas sem mútuas entre as fases, Figura 2.7. Em SILVA, (2004) é apresentada uma abordagem detalhada sobre a modelagem de transformadores.

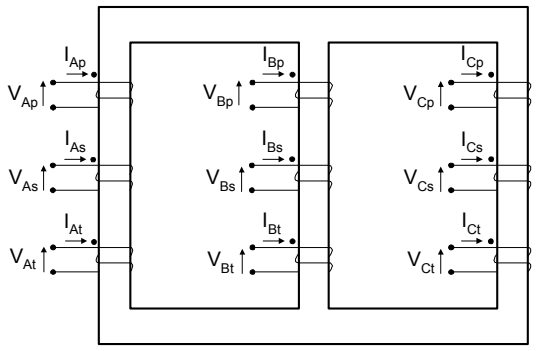

Figura 2.6- Transformador Trifásico

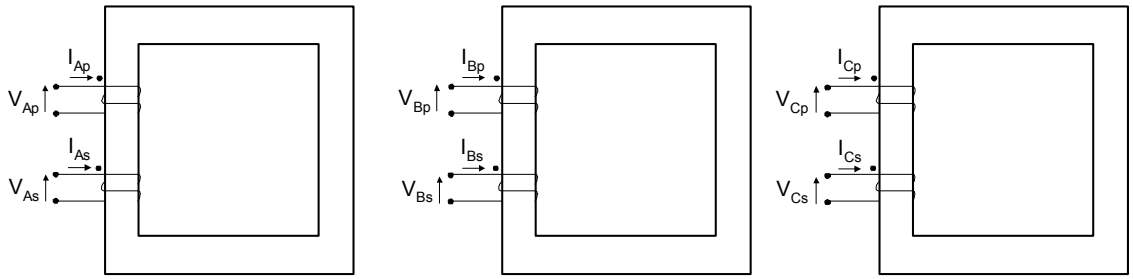

Figura 2.7 – Transformador Trifásico Constituído por Bancos Monofásicos

O transformador trifásico pode ser modelado por uma matriz contendo as impedâncias primitivas dos ramos, dada pelas Equações (2.32). a .(2.41)

$$Z_{prim} = \begin{bmatrix} Z_p & Z_{ps} & Z_{pt} \\ Z_{sp} & Z_s & Z_{st} \\ Z_{tp} & Z_{ts} & Z_t \end{bmatrix} \quad \textbf{(2.32)}$$

onde:

p, s e t: Representam as grandezas do primário, secundário e terciário respectivamente.

$Z_p, Z_s, Z_t, Z_{ps}, Z_{pt}, Z_{sp}, Z_{st}, Z_{tp}$ e Z_{ts}: São submatrizes 3X3 contendo as impedâncias primitivas dos ramos que são mostradas nas Equações de (2.33) a (2.41)

$$Z_p = \begin{bmatrix} Z_p^{aa} & Z_p^{ab} & Z_p^{ac} \\ Z_p^{ba} & Z_p^{bb} & Z_p^{bc} \\ Z_p^{ca} & Z_p^{cb} & Z_p^{cc} \end{bmatrix} \quad \textbf{(2.33)}$$

$$Z_s = \begin{bmatrix} Z_s^{aa} & Z_s^{ab} & Z_s^{ac} \\ Z_s^{ba} & Z_s^{bb} & Z_s^{bc} \\ Z_s^{ca} & Z_s^{cb} & Z_s^{cc} \end{bmatrix} \quad \textbf{(2.34)}$$

$$Z_t = \begin{bmatrix} Z_t^{aa} & Z_t^{ab} & Z_t^{ac} \\ Z_t^{ba} & Z_t^{bb} & Z_t^{bc} \\ Z_t^{ca} & Z_t^{cb} & Z_t^{cc} \end{bmatrix} \quad \textbf{(2.35)}$$

$$Z_{ps} = \begin{bmatrix} Z_{ps}^{aa} & Z_{ps}^{ab} & Z_{ps}^{ac} \\ Z_{ps}^{ba} & Z_{ps}^{bb} & Z_{ps}^{bc} \\ Z_{ps}^{ca} & Z_{ps}^{cb} & Z_{ps}^{cc} \end{bmatrix} \quad \textbf{(2.36)}$$

$$Z_{pt} = \begin{bmatrix} Z_{pt}^{aa} & Z_{pt}^{ab} & Z_{pt}^{ac} \\ Z_{pt}^{ba} & Z_{pt}^{bb} & Z_{pt}^{bc} \\ Z_{pt}^{ca} & Z_{pt}^{cb} & Z_{pt}^{cc} \end{bmatrix} \quad \textbf{(2.37)}$$

$$Z_{sp} = \begin{bmatrix} Z_{sp}^{aa} & Z_{sp}^{ab} & Z_{sp}^{ac} \\ Z_{sp}^{ba} & Z_{sp}^{bb} & Z_{sp}^{bc} \\ Z_{sp}^{ca} & Z_{sp}^{cb} & Z_{sp}^{cc} \end{bmatrix} \quad \textbf{(2.38)}$$

$$Z_{st} = \begin{bmatrix} Z_{st}^{aa} & Z_{st}^{ab} & Z_{st}^{ac} \\ Z_{st}^{ba} & Z_{st}^{bb} & Z_{st}^{bc} \\ Z_{st}^{ca} & Z_{st}^{cb} & Z_{st}^{cc} \end{bmatrix} \quad \textbf{(2.39)}$$

$$Z_{tp} = \begin{bmatrix} Z_{tp}^{aa} & Z_{tp}^{ab} & Z_{tp}^{ac} \\ Z_{tp}^{ba} & Z_{tp}^{bb} & Z_{tp}^{bc} \\ Z_{tp}^{ca} & Z_{tp}^{cb} & Z_{tp}^{cc} \end{bmatrix} \quad \textbf{(2.40)}$$

$$Z_{ts} = \begin{bmatrix} Z_{ts}^{aa} & Z_{ts}^{ab} & Z_{ts}^{ac} \\ Z_{ts}^{ba} & Z_{ts}^{bb} & Z_{ts}^{bc} \\ Z_{ts}^{ca} & Z_{ts}^{cb} & Z_{ts}^{cc} \end{bmatrix} \quad \textbf{(2.41)}$$

Para que sejam obtidos resultados precisos nas simulações, todas estas impedâncias devem ser determinadas. Sendo assim, para determinar todos os elementos da matriz da equação (2.32) serão feitas as seguintes considerações em relação as indutâncias próprias e mútuas do transformador:

- Indutâncias próprias dos enrolamentos primário, secundário e terciário.

$$L_p^a = L_p^b = L_p^c \quad (2.42)$$

$$L_s^a = L_s^b = L_s^c \quad (2.43)$$

$$L_t^a = L_t^b = L_t^c \quad (2.44)$$

- Indutâncias mútuas entre os enrolamentos primário e secundário, primário e terciário e secundário e terciário de mesma fase.

$$L_{ps}^{aa} = L_{ps}^{bb} = L_{ps}^{cc} \quad (2.45)$$

$$L_{pt}^{aa} = L_{pt}^{bb} = L_{pt}^{cc} \quad (2.46)$$

$$L_{st}^{aa} = L_{st}^{bb} = L_{st}^{cc} \tag{2.47}$$

- Indutâncias mútuas entre os enrolamentos de fases diferentes.

$$L_{pp}^{ab} = L_{pp}^{bc} = L_{pp}^{ca} \tag{2.48}$$

$$L_{ss}^{ab} = L_{ss}^{bc} = L_{ss}^{ca} \tag{2.49}$$

$$L_{tt}^{ab} = L_{tt}^{bc} = L_{tt}^{ca} \tag{2.50}$$

$$L_{ps}^{ab} = L_{ps}^{ac} = L_{ps}^{ba} = L_{ps}^{bc} = L_{ps}^{ca} = L_{ps}^{cb} \tag{2.51}$$

$$L_{st}^{ab} = L_{st}^{ac} = L_{st}^{ba} = L_{st}^{bc} = L_{st}^{ca} = L_{st}^{cb} \tag{2.52}$$

$$L_{pt}^{ab} = L_{pt}^{ac} = L_{pt}^{ba} = L_{pt}^{bc} = L_{pt}^{ca} = L_{pt}^{cb} \tag{2.53}$$

O circuito magnético da Figura 2.6 e Figura 2.7 pode ser representado pelo seu análogo elétrico como mostra a Figura 2.8, onde cada perna do circuito magnético é modelada por uma relutância R_1 em série com uma força magnetomotriz, que é dada pelo produto do número de espiras pela corrente elétrica (*NI*).

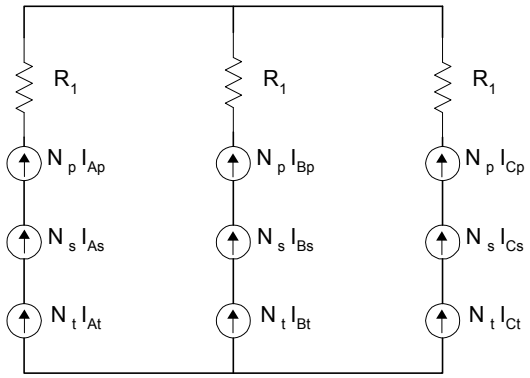

Figura 2.8 – Circuito Elétrico Equivalente

Utilizando-se as definições de indutâncias próprias e mútuas e considerando-se que o fluxo magnetizante de projeto leva a uma relutância constante , as indutâncias denotadas nas equações de (2.42) a (2.53) são determinadas de acordo com as equações:

- Indutâncias próprias.

$$L_f^m = Ld_f^m + \frac{2}{3}\frac{N_f^2}{R_1} \tag{2.54}$$

- Indutâncias mútuas de mesma fase.

$$L_{fg}^{mm} = \frac{2}{3}\frac{N_f N_g}{R_1},\ \text{com}\ f \neq g \tag{2.55}$$

- Indutâncias mútuas de fases diferentes.

$$L_{fg}^{mn} = -\frac{1}{3}\frac{N_f N_g}{R_1},\ \text{com}\ m \neq n \tag{2.56}$$

onde:

m, n : Representa as fases A,B ou C.

f, g : Representa o primário (p), secundário (s) ou terciário (t) do transformador.

Ld : É a indutância de dispersão.

Utilizando a Equação (2.57), explicitando R_1, substituindo nas equações (2.54), (2.55) e (2.56) e multiplicando-as por $j\omega$, as reatâncias primitivas serão calculadas pelas equações (2.59), (2.60) e (2.61).

$$M = \frac{2}{3}\frac{N_p N_s}{R_1} \tag{2.58}$$

$$jX_f^m = jXd_f^m + \frac{N_f^2}{N_p N_s} jX_M \tag{2.59}$$

$$jX_{fg}^{mm} = \frac{N_f N_g}{N_p N_s} jX_M,\ \text{com}\ f \neq g \tag{2.60}$$

$$jX_{fg}^{mn} = -\frac{1}{2}\frac{N_f N_g}{N_p N_s} jX_M,\ \text{com}\ m \neq n \tag{2.61}$$

onde:

jXd : É a reatância de dispersão.

jX_M : É a reatância de magnetização vista do primário.

Com as manipulações feitas anteriormente pode-se determinar a matriz de impedância primitiva da Equação (2.62) bastando-se conhecer as reatâncias de dispersão, a reatância de magnetização e as resistências dos enrolamentos. Estes parâmetros são facilmente obtidos pelos ensaios de curto-circuito e de circuito aberto ou fornecidos pelos fabricantes. O número de espiras dos enrolamentos dos transformadores normalmente não é conhecido, mas pode-se considerar uma divisão

linear da tensão de 1 V/espira, desta forma o número de espiras é igual à própria tensão na bobina.

A matriz de admitância primitiva é dada pela equação (2.63).

$$Y_{prim} = \left(Z_{prim}\right)^{-1} \tag{2.63}$$

2.3.4.1 Representação das Impedâncias de Aterramento de Transformadores

Em configurações nas quais se faz necessário o aterramento do neutro dos transformadores, como as ligações Yaterrado-Yaterrado, Yaterrado-delta, delta-Yaterrado, a matriz é representada pela Equação (2.64).

$$Z_{prim} = \left[\begin{array}{ccc:c} Z_p & Z_{ps} & Z_{pt} & \\ Z_{sp} & Z_s & Z_{st} & \\ Z_{tp} & Z_{ts} & Z_t & \\ \hdashline & & & Z_{terra} \end{array}\right] \tag{2.64}$$

onde:

Z_{terra} : É a matriz diagonal que contem os valores das impedâncias de aterramento.

2.3.4.2 Cálculo da Matriz Admitância Primitiva em PU

Caso seja necessário representar os transformadores em valores por unidade, conhecendo-se as bases do sistema em uso e as impedâncias primitivas dos ramos pode-se determinar a matriz impedância primitiva em p.u., como mostra a Equação (2.65).

$$Z_{prim} = \begin{bmatrix} Z_p \dfrac{I_{p,b}}{V_{p,b}} & Z_{ps} \dfrac{I_{s,b}}{V_{p,b}} & Z_{pt} \dfrac{I_{t,b}}{V_{p,b}} \\ Z_{sp} \dfrac{I_{p,b}}{V_{s,b}} & Z_s \dfrac{I_{s,b}}{V_{s,b}} & Z_{st} \dfrac{I_{t,b}}{V_{s,b}} \\ Z_{tp} \dfrac{I_{p,b}}{V_{t,b}} & Z_{ts} \dfrac{I_{s,b}}{V_{t,b}} & Z_t \dfrac{I_{t,b}}{V_{t,b}} \end{bmatrix} \tag{2.65}$$

onde:

$V_{p,b}, V_{s,b}, V_{t,b}, I_{p,b}, I_{s,b}$ e $I_{t,b}$: São as tensões e correntes de fase básicas do primário, secundário e terciário respectivamente.

2.3.4.3 Matriz de Incidência Nodal

Os transformadores encontrados no sistema elétrico podem ser conectados de várias maneiras. Para representar uma conexão específica do transformador é utilizada a matriz de incidência nodal (ARRILLAGA, 1983). Esta matriz pode ser determinada pela Equação (2.66).

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{b1} & a_{b2} & \cdots & a_{bm} \end{bmatrix} \tag{2.66}$$

onde:

$a_{pq} = 1$, se a corrente no ramo pq está saindo do nó.

$a_{pq} = -1$, se a corrente no ramo pq está chegando no nó.

$a_{pq} = 0$, se o nó p não está conectado ao nó q.

Como exemplo, consideremos o transformador trifásico descrito pela Figura 2.6, com seu primário ligado em delta e o seu secundário em estrela solidamente aterrado como mostra a Figura 2.9, onde as setas indicam o sentido da corrente elétrica nos ramos e os pontos indicam as polaridades das bobinas.

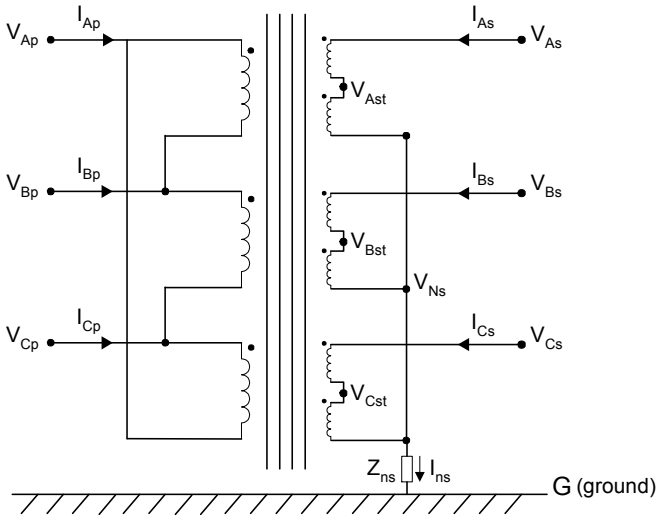

Figura 2.9 – Transformador conectado em delta-estrela aterrada

A matriz de incidência nodal para este exemplo é dada pela equação (2.67), onde as linhas da matriz representam os ramos e as colunas os nós.

Neste tipo de conexão existe uma diferença angular de 30º entre os ângulos das tensões de fase do primário e secundário. No modelo apresentado neste artigo esta diferença de fase é obtida naturalmente.

$$A = \begin{array}{c} \\ V_{ap}V_{bp} \\ V_{bp}V_{cp} \\ V_{cp}V_{ap} \\ \hline V_{as}V_{ns} \\ V_{bs}V_{ns} \\ V_{cs}V_{ns} \\ \hline V_{ns}G \end{array} \begin{array}{c} \begin{array}{ccc|ccc|c} V_{ap} & V_{bp} & V_{ac} & V_{as} & V_{bs} & V_{cs} & V_{ns} \end{array} \\ \left[\begin{array}{ccc|ccc|c} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \tag{2.67}$$

2.3.4.4 Matriz Admitância de Barras para o Transformador

A matriz que representa o transformador conectado é dada pela equação (2.68), onde Y_{barra} é a matriz admitância de barras e A^t é a transposta da matriz A.

$$Y_{barra} = A^t Y_{prim} A \tag{2.68}$$

Para a inserção desta matriz em programas trifásicos a três fios basta eliminar o neutro utilizando um método de eliminação de nós, caso a matriz Y_{barra} tenha dimensão superior a 3x3.

A partir da matriz Y_{barra} do transformador pode-se representá-lo no FPTP utilizando-se a mesma modelagem das linhas de transmissão. (Item 2.3.1)

2.3.5 Regulador de Tensão

Um dos problemas mais comuns em sistemas de distribuição é a manutenção dos níveis de tensão dentro de limites aceitáveis. O Regulador de Tensão é utilizado para manter-se regulada a tensão na carga, através de um comutador de derivações em carga monofásico, instalados em longas linhas de distribuição em pontos estratégicos e/ou em

subestações. São montados em bancos trifásicos, conexão delta, delta aberto ou estrela aterrada, permitindo a regulação da tensão em cada linha individualmente ou através de um sistema mestre comandado, onde a tensão é regulada para as três fases em um mesmo nível, substituindo os reguladores de tensão trifásicos com inúmeras vantagens. (Araujo, 2005)

No presente trabalho, os reguladores são representados por unidades monofásicas conectadas em estrela. A Figura 2.10 ilustra o modelo adotado e o modelo π-equivalente poder ser observado na Figura 2.11.

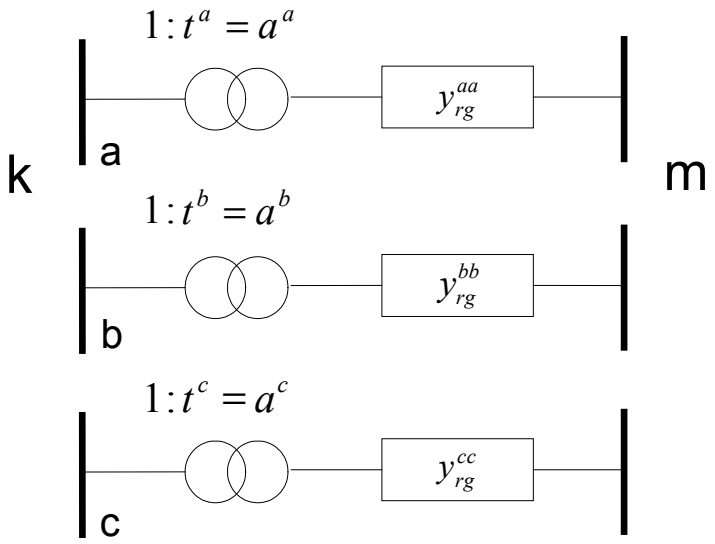

Figura 2.10 – Diagrama Esquemático do Regulador de Tensão

onde:

a^s é variável relativa ao controle de tensão, variação do tape sob carga

y^{ss} é a admitância série do regulador.

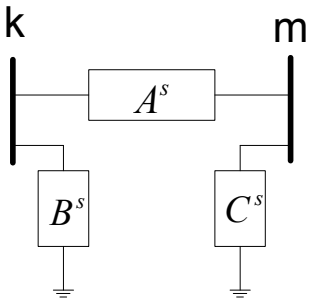

Figura 2.11 - Modelo π-equivalente de um regulador de tensão trifásico

onde:

$$\begin{aligned} A^s &= a^s y_{rg}^{ss} \\ B^s &= a^s \left(a^s - 1\right) y_{rg}^{ss} \\ C^s &= \left(1 - a^s\right) y_{rg}^{ss} \end{aligned} \quad \textbf{(2.69)}$$

Escrevendo-se as equações das correntes que fluem da barra *k* para a barra *m* e vice-versa temos:

$$I_{km}^s = \left(a^s\right)^2 y_{rg}^{ss} E_k^s - a^s y_{rg}^s E_m^s \quad (2.70)$$

$$I_{mk}^s = -a^s y_{rg}^{ss} E_k^s + y_{rg}^{ss} E_m^s \quad (2.71)$$

Multiplicando a corrente I_{km} pela tensão E_k obtém-se a potência ativa e reativa que flui da barra k para barra m:

$$P_{km}^s = \left(V_k^s\right)^2 \left(a^s\right)^2 g_{rg}^{ss} - a^s V_k^s V_m^s g_{rg}^{ss} \cos\left(\theta_{km}^s\right) - a^s V_k^s V_m^s b_{rg}^{ss} \operatorname{sen}\left(\theta_{km}^s\right) \quad (2.72)$$

$$Q_{km}^s = -\left(V_k^s\right)^2 \left(a^s\right)^2 b_{rg}^{ss} + a^s V_k^s V_m^s b_{rg}^{ss} \cos\left(\theta_{km}^s\right) - a^s V_k^s V_m^s g_{rg}^{ss} \operatorname{sen}\left(\theta_{km}^s\right) \quad (2.73)$$

Da mesma forma, multiplicado a corrente I_{mk} pela tensão E_m obtém-se:

$$P_{mk}^s = \left(V_m^s\right)^2 g_{rg}^{ss} - a^s V_k^s V_m^s g_{rg}^{ss} \cos\left(\theta_{km}^s\right) + a^s V_k^s V_m^s b_{rg}^{ss} \operatorname{sen}\left(\theta_{km}^s\right) \quad (2.74)$$

$$Q_{mk}^s = -\left(V_m^s\right)^2 b_{rg}^{ss} + a^s V_k^s V_m^s b_{rg}^{ss} \cos\left(\theta_{km}^s\right) + a^s V_k^s V_m^s g_{rg}^{ss} \operatorname{sen}\left(\theta_{km}^s\right) \quad (2.75)$$

A fim de se controlar a tensão em uma determinada barra do sistema, deve-se inserir no sistema a equação de controle de tensão. Esta equação pode ser vista na Equação (2.76)

$$V_k^s - V_{esp}^s = 0 \quad (2.76)$$

As contribuições na matriz jacobiana e no vetor independente de um regulador entre a barra k e m controlando a tensão na barra m podem ser vistas na Equação (2.77).

$$
\left[\begin{array}{cccccccccccc:ccc}
 & & & & & & & & & & & & \frac{\partial P_{km}^{a}}{\partial a^{a}} & 0 & 0 \\
 & & & & & & & & & & & & 0 & \frac{\partial P_{km}^{b}}{\partial a^{b}} & 0 \\
 & & & & & & & & & & & & 0 & 0 & \frac{\partial P_{km}^{c}}{\partial a^{c}} \\
 & & & & & & & & & & & & \frac{\partial Q_{km}^{a}}{\partial a^{a}} & 0 & 0 \\
 & & & & & & & & & & & & 0 & \frac{\partial Q_{km}^{b}}{\partial a^{b}} & 0 \\
 & & & & & & & & & & & & 0 & 0 & \frac{\partial Q_{km}^{c}}{\partial a^{c}} \\
 & & & & & JAC & & & & & & & \frac{\partial P_{mk}^{a}}{\partial a^{a}} & 0 & 0 \\
 & & & & & & & & & & & & 0 & \frac{\partial P_{mk}^{b}}{\partial a^{b}} & 0 \\
 & & & & & & & & & & & & 0 & 0 & \frac{\partial P_{mk}^{c}}{\partial a^{c}} \\
 & & & & & & & & & & & & \frac{\partial Q_{mk}^{a}}{\partial a^{a}} & 0 & 0 \\
 & & & & & & & & & & & & 0 & \frac{\partial Q_{mk}^{b}}{\partial a^{b}} & 0 \\
 & & & & & & & & & & & & 0 & 0 & \frac{\partial Q_{mk}^{c}}{\partial a^{c}} \\
\hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{array}\right]
\cdot
\left[\begin{array}{c}
\Delta\theta_k^a \\ \Delta\theta_k^b \\ \Delta\theta_k^c \\ \Delta V_k^a \\ \Delta V_k^b \\ \Delta V_k^c \\ \Delta\theta_m^a \\ \Delta\theta_m^b \\ \Delta\theta_m^c \\ \Delta V_m^a \\ \Delta V_m^b \\ \Delta V_m^c \\ \hdashline \Delta a^a \\ \Delta a^b \\ \Delta a^c
\end{array}\right]
= -
\left[\begin{array}{c}
P_k^a \\ P_k^b \\ P_k^c \\ Q_k^a \\ Q_k^b \\ Q_k^c \\ P_m^a \\ P_m^b \\ P_m^c \\ Q_m^a \\ Q_m^b \\ Q_m^c \\ \hdashline V_m^a - V_{esp}^a \\ V_m^b - V_{esp}^b \\ V_m^c - V_{esp}^c
\end{array}\right]
\qquad (2.77)
$$

2.3.6 Compensador Estático de Reativos (CER)

Os avanços na tecnologia de eletrônica de potência, em conjunto com avançadas metodologias de controle, tornaram possível o desenvolvimento de dispositivos FACTS (Flexible AC Transmission System). Entre eles podemos destacar o compensador estático de reativo (CER). Este equipamento é um importante componente para controle da tensão nodal, sendo formado por um grupo de capacitores e indutores em derivação controlados por chaveamento contínuo de tiristores.

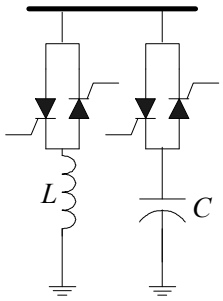

Figura 2.12- Diagrama esquemático do CER

Do ponto de vista operacional, o CER pode ser visto como uma reatância shunt variável, gerando ou absorvendo potência reativa, ajustada automaticamente em resposta à variação das condições de operação do sistema. (Passos Filho, 2000)

Este equipamento pode ter várias funções no sistema, tanto em regime permanente como em regime dinâmico devido à sua alta velocidade de controle.

O CER foi usado inicialmente na indústria para compensação de flutuações de tensão causadas por cargas como fornos a arco, passando a ser usado em sistemas de potência graças a sua capacidade de resolver problemas de rede sempre que se necessita de uma ação de controle rápido da potência reativa (Oliveira, 2005).

A curva da Figura 2.13 mostra a característica V/Q do CER em estado permanente. A faixa de controle linear é determinada pela susceptância máxima do indutor e pela susceptância total devido aos bancos de capacitores em serviço.

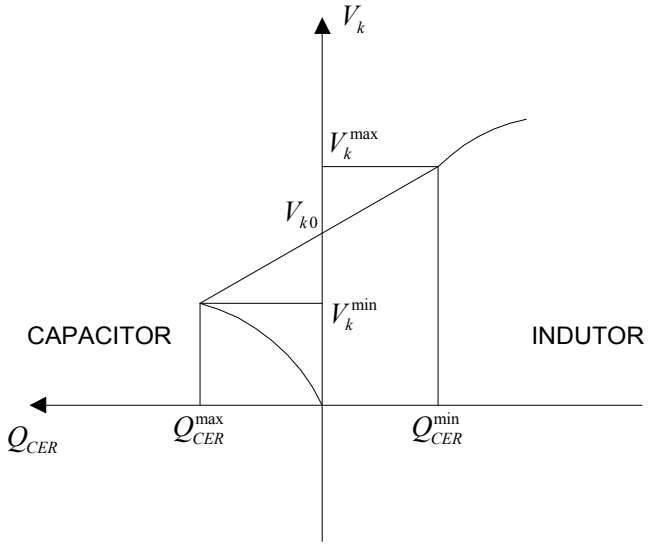

Figura 2.13 - Característica V/Q do CER

Do ponto de vista prático, a inclinação da reta de controle r, a tensão de referencia V_{k0} a susceptância mínima $b_{CER}^{\min}$ e a susceptância máxima $b_{CER}^{\max}$ são conhecidos.

$$r = tg\alpha = \frac{V_{\max} - V_{\min}}{Q_{\max} - Q_{\min}} \qquad (2.78)$$

onde:

r: Coeficiente de inclinação da curva característica

α: Ângulo de condução

$V_k^{\min}, V_k^{\max}$: Representam os valores mínimo e máximo de tensão na barra controlada, respectivamente.

V_{k0} Representa a tensão de referência do CER;

$Q_{CER}^{\min}, Q_{CER}^{\max}$ Representam os limites de compensação estática indutiva e capacitiva, respectivamente.

$b_{CER}^{\min}, b_{CER}^{\max}$ Representam os limites da susceptância do CER, correspondentes à capacidade máxima de absorção e de injeção de potência reativa, respectivamente.

Neste trabalho será adotada a modelagem proposta por (Passos Filho, 2000) e (Garcia, 2001), que consideram a potência reativa injetada na barra do CER como uma nova variável de estado adicional. Para tornar o sistema possível, deve-se inserir uma nova equação de controle que represente o equipamento.

Como pode ser visto na Figura 2.13, o CER possui três regiões de operação definidas. São elas:

- Região Linear: a potência reativa injetada na barra é função da tensão na barra controlada.
- Região Capacitiva: o equipamento se comporta como um capacitor
- Região Indutiva: o equipamento se comporta como um indutor.

2.3.6.1 Modelagem da Região Linear

Nesta região tem-se controle do módulo da tensão na barra k que pode variar entre Vmin e Vmax obedecendo a reta de inclinação r, como pode ser visto.

A equação de controle de potência reativa injetada é dada por:

$$V_k^s = V_{k0}^s + r.Q_{Gk}^s \tag{2.79}$$

A equações de controle pode ser escrita como:

$$y^s = V_{k0}^s + r.Q_{Gk}^s - V_k^s \tag{2.80}$$

As contribuições na matriz jacobiana e no vetor independente podem ser vistos na equação :

$$\left[\begin{array}{cccccccccccc|ccc} & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & -1 & 0 & 0 \\ & & & & & & & & & & & & 0 & -1 & 0 \\ & & & & & & & & & & & & 0 & 0 & -1 \\ & & & & & JAC & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ & & & & & & & & & & & & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -r \end{array}\right] \cdot \left[\begin{array}{c} \Delta\theta_k^a \\ \Delta\theta_k^b \\ \Delta\theta_k^c \\ \Delta V_k^a \\ \Delta V_k^b \\ \Delta V_k^c \\ \Delta\theta_m^a \\ \Delta\theta_m^b \\ \Delta\theta_m^c \\ \Delta V_m^a \\ \Delta V_m^b \\ \Delta V_m^c \\ \hline \Delta Q_{Gk}^a \\ \Delta Q_{Gk}^b \\ \Delta Q_{Gk}^c \end{array}\right] = - \left[\begin{array}{c} P_k^a \\ P_k^b \\ P_k^c \\ Q_k^a \\ Q_k^b \\ Q_k^c \\ P_m^a \\ P_m^b \\ P_m^c \\ Q_m^a \\ Q_m^b \\ Q_m^c \\ \hline V_{k0}^a + r.Q_{Gk}^a - V_k^a \\ V_{k0}^b + r.Q_{Gk}^b - V_k^b \\ V_{k0}^c + r.Q_{Gk}^c - V_k^c \end{array}\right] \tag{2.81}$$

2.3.6.2 Modelagem da Região Indutiva/Capacitiva

Nestas regiões o equipamento perde sua capacidade de controle pois fisicamente não consegue mais aumentar a sua susceptância shunt b, chegando em um de seus limites ($[b_{CER}^{\min}, b_{CER}^{\max}]$). Neste ponto o CER se comporta como uma susceptância fixa localizada na barra em ambos os casos (indutiva/capacitiva). As equações dadas referem-se às regiões indutivas e capacitivas.

$$Q_{Gk}^s = B_{min}.\left(V_k^s\right)^2 \tag{2.82}$$

$$Q_{Gk}^{s} = B_{\max} . \left(V_k^s \right)^2 \tag{2.83}$$

As contribuições para a matriz jacobiana e para vetor independente podem ser vistos na equação (2.74) a seguir:

$$\left[\begin{array}{ccccccccccccc|ccc}
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & JAC & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & & & & & & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 2V_k^a B_{\min/\max} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 2V_k^b B_{\min/\max} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 2V_k^b B_{\min/\max} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{array}\right]
\cdot
\left[\begin{array}{c}
\Delta\theta_k^a \\ \Delta\theta_k^b \\ \Delta\theta_k^c \\ \Delta V_k^a \\ \Delta V_k^b \\ \Delta V_k^c \\ \Delta\theta_m^a \\ \Delta\theta_m^b \\ \Delta\theta_m^c \\ \Delta V_m^a \\ \Delta V_m^b \\ \Delta V_m^c \\ \hline \Delta Q_{Gk}^a \\ \Delta Q_{Gk}^b \\ \Delta Q_{Gk}^c
\end{array}\right]
= -
\left[\begin{array}{c}
P_k^a \\ P_k^b \\ P_k^c \\ Q_k^a \\ Q_k^b \\ Q_k^c \\ P_m^a \\ P_m^b \\ P_m^c \\ Q_m^a \\ Q_m^b \\ Q_m^c \\ \hline B_{\min/\max} V_k^a - Q_{Gk}^a \\ B_{\min/\max} V_k^b - Q_{Gk}^b \\ B_{\min/\max} V_k^c - Q_{Gk}^c
\end{array}\right] \tag{2.84}$$

2.3.7 Compensação Série Controlada a Tiristores (CSCT)

A Compensação Série Controlada a Tiristores normalmente é composta por um capacitor série, da mesma forma que em um esquema de compensação de linhas de corrente alternada, em conjunto com reatores e tiristores. Com isso, o equipamento é capaz de alterar sua impedância equivalente, impondo diferentes níveis de compensação reativa na linha CA.

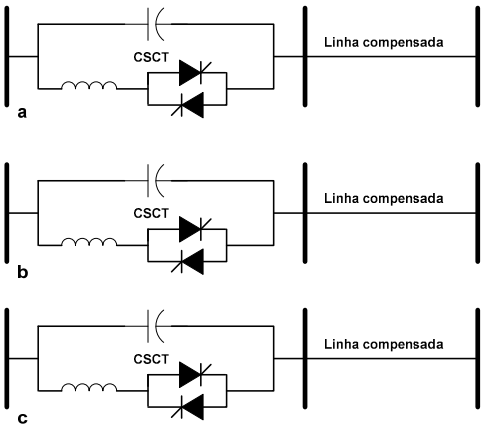

Figura 2.14 – Diagrama Esquemático do CSCT

Por ser um dispositivo de atuação rápida o CSCT pode ser empregado no amortecimento de oscilações eletromecânicas nos sistemas de potência. Além disso, a compensação série controlada pode ser utilizada com o objetivo de controlar o fluxo de potência ativa em uma linha do sistema elétrico através da variação de sua reatância série. Outra aplicação é o controle de tensão em uma determinada barra variando a impedância de uma determinada linha do sistema.

Neste trabalho será apresentada a modelagem do CSCT para o controle do fluxo de potência de uma determinada linha e para o controle de tensão em barras. O equipamento será modelado por uma susceptância variável em cada fase, conectada entre duas barras, de modo a manter a potência ativa ou o módulo da tensão segundo um valor de referência. Essa susceptância é considerada uma nova variável de estado.

2.3.7.1 Controle de Tensão em Barras

Neste modo de controle o CSCT controla o módulo da tensão em uma determinada barra do sistema. Neste caso a compensação que está localizada entre a barra k e m controla o módulo da tensão na barra m. O equacionamento é mostrado abaixo.

$$V_m^s - V_{esp}^s = 0 \qquad (2.85)$$

$$b_{km}^{ss} = b_{km}^{ss} + B_{TCSC}^s \qquad (2.86)$$

onde:

b_{km}^{ss}: susceptância série da linha *km* da *fase s*

B_{TCSC}^s: susceptância equivalente do CSCT

As contribuições para matriz jacobiana e para vetor independente podem ser vistos na Equação (2.87)

$$\begin{bmatrix} JAC & \begin{matrix} \frac{\partial P^a_{km}}{\partial B^a_{CSCT}} & 0 & 0 \\ 0 & \frac{\partial P^b_{km}}{\partial B^b_{CSCT}} & 0 \\ 0 & 0 & \frac{\partial P^c_{km}}{\partial B^c_{CSCT}} \\ \frac{\partial Q^a_{km}}{\partial B^a_{CSCT}} & 0 & 0 \\ 0 & \frac{\partial Q^b_{km}}{\partial B^b_{CSCT}} & 0 \\ 0 & 0 & \frac{\partial Q^c_{km}}{\partial B^c_{CSCT}} \\ \frac{\partial P^a_{mk}}{\partial B^a_{CSCT}} & 0 & 0 \\ 0 & \frac{\partial P^b_{mk}}{\partial B^b_{CSCT}} & 0 \\ 0 & 0 & \frac{\partial P^c_{mk}}{\partial B^c_{CSCT}} \\ \frac{\partial Q^a_{mk}}{\partial B^a_{CSCT}} & 0 & 0 \\ 0 & \frac{\partial Q^b_{mk}}{\partial B^b_{CSCT}} & 0 \\ 0 & 0 & \frac{\partial Q^c_{mk}}{\partial B^c_{CSCT}} \end{matrix} \\ \begin{matrix} 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{matrix} & \begin{matrix} 0&0&0 \\ 0&0&0 \\ 0&0&0 \end{matrix} \end{bmatrix} \cdot \begin{bmatrix} \Delta\theta^a_k \\ \Delta\theta^b_k \\ \Delta\theta^c_k \\ \Delta V^a_k \\ \Delta V^b_k \\ \Delta V^c_k \\ \Delta\theta^a_m \\ \Delta\theta^b_m \\ \Delta\theta^c_m \\ \Delta V^a_m \\ \Delta V^b_m \\ \Delta V^c_m \\ \hline \Delta B^a_{CSCT} \\ \Delta B^b_{CSCT} \\ \Delta B^c_{CSCT} \end{bmatrix} = - \begin{bmatrix} P^a_k \\ P^b_k \\ P^c_k \\ Q^a_k \\ Q^b_k \\ Q^c_k \\ P^a_m \\ P^b_m \\ P^c_m \\ Q^a_m \\ Q^b_m \\ Q^c_m \\ \hline V^a_m - V^a_{esp} \\ V^b_m - V^b_{esp} \\ V^c_m - V^c_{esp} \end{bmatrix} \quad (2.87)$$

2.3.7.2 Controle do Fluxo de Potência Ativa em Linhas

Neste modo de controle o CSCT controla a potência ativa em uma linha do sistema. Neste caso hipotético, a compensação que está localizada entre a barra k e m O equacionamento é mostrado abaixo.

$$P^s_{km} - P^s_{esp} = 0 \quad (2.88)$$

$$b^{ss}_{km} = b^{ss}_{km} + B^s_{CSCT} \quad (2.89)$$

onde:

P^s_{km} : potência ativa que flui da barra k para barra m

P^{s}_{esp} : potência ativa especificada da barra k para barra m

b^{ss}_{km} : susceptância série da linha *km* da *fase s*

B^{s}_{TCSC} : susceptância equivalente do CSCT

As contribuições na matriz jacobiana e no vetor independente podem ser vistos na Equação (2.90)

$$\left[\begin{array}{c|ccc} & \frac{\partial P^{a}_{km}}{\partial B^{a}_{CSCT}} & 0 & 0 \\ & 0 & \frac{\partial P^{b}_{km}}{\partial B^{b}_{CSCT}} & 0 \\ & 0 & 0 & \frac{\partial P^{c}_{km}}{\partial B^{c}_{CSCT}} \\ & \frac{\partial Q^{a}_{km}}{\partial B^{a}_{CSCT}} & 0 & 0 \\ & 0 & \frac{\partial Q^{b}_{km}}{\partial B^{b}_{CSCT}} & 0 \\ & 0 & 0 & \frac{\partial Q^{c}_{km}}{\partial B^{c}_{CSCT}} \\ JAC & \frac{\partial P^{a}_{mk}}{\partial B^{a}_{CSCT}} & 0 & 0 \\ & 0 & \frac{\partial P^{b}_{mk}}{\partial B^{b}_{CSCT}} & 0 \\ & 0 & 0 & \frac{\partial P^{c}_{mk}}{\partial B^{c}_{CSCT}} \\ & \frac{\partial Q^{a}_{mk}}{\partial B^{a}_{CSCT}} & 0 & 0 \\ & 0 & \frac{\partial Q^{b}_{mk}}{\partial B^{b}_{CSCT}} & 0 \\ & 0 & 0 & \frac{\partial Q^{c}_{mk}}{\partial B^{c}_{CSCT}} \\ \hline \frac{\partial P^{a}_{km}}{\partial \theta^{a}_{k}} \; \frac{\partial P^{a}_{km}}{\partial \theta^{b}_{k}} \; \frac{\partial P^{a}_{km}}{\partial \theta^{c}_{k}} \; \frac{\partial P^{a}_{km}}{\partial V^{a}_{k}} \; \frac{\partial P^{a}_{km}}{\partial V^{b}_{k}} \; \frac{\partial P^{a}_{km}}{\partial V^{c}_{k}} \; \frac{\partial P^{a}_{km}}{\partial \theta^{a}_{m}} \; \frac{\partial P^{a}_{km}}{\partial \theta^{b}_{m}} \; \frac{\partial P^{a}_{km}}{\partial \theta^{c}_{m}} \; \frac{\partial P^{a}_{km}}{\partial V^{a}_{m}} \; \frac{\partial P^{a}_{km}}{\partial V^{b}_{m}} \; \frac{\partial P^{a}_{km}}{\partial V^{c}_{m}} & \frac{\partial P^{a}_{km}}{\partial B^{a}_{CSCT}} & 0 & 0 \\ \frac{\partial P^{b}_{km}}{\partial \theta^{a}_{k}} \; \frac{\partial P^{b}_{km}}{\partial \theta^{b}_{k}} \; \frac{\partial P^{b}_{km}}{\partial \theta^{c}_{k}} \; \frac{\partial P^{b}_{km}}{\partial V^{a}_{k}} \; \frac{\partial P^{b}_{km}}{\partial V^{b}_{k}} \; \frac{\partial P^{b}_{km}}{\partial V^{c}_{k}} \; \frac{\partial P^{b}_{km}}{\partial \theta^{a}_{m}} \; \frac{\partial P^{b}_{km}}{\partial \theta^{b}_{m}} \; \frac{\partial P^{b}_{km}}{\partial \theta^{c}_{m}} \; \frac{\partial P^{b}_{km}}{\partial V^{a}_{m}} \; \frac{\partial P^{b}_{km}}{\partial V^{b}_{m}} \; \frac{\partial P^{b}_{km}}{\partial V^{c}_{m}} & 0 & \frac{\partial P^{b}_{km}}{\partial B^{b}_{CSCT}} & 0 \\ \frac{\partial P^{c}_{km}}{\partial \theta^{a}_{k}} \; \frac{\partial P^{c}_{km}}{\partial \theta^{b}_{k}} \; \frac{\partial P^{c}_{km}}{\partial \theta^{c}_{k}} \; \frac{\partial P^{c}_{km}}{\partial V^{a}_{k}} \; \frac{\partial P^{c}_{km}}{\partial V^{b}_{k}} \; \frac{\partial P^{c}_{km}}{\partial V^{c}_{k}} \; \frac{\partial P^{c}_{km}}{\partial \theta^{a}_{m}} \; \frac{\partial P^{c}_{km}}{\partial \theta^{b}_{m}} \; \frac{\partial P^{c}_{km}}{\partial \theta^{c}_{m}} \; \frac{\partial P^{c}_{km}}{\partial V^{a}_{m}} \; \frac{\partial P^{c}_{km}}{\partial V^{b}_{m}} \; \frac{\partial P^{c}_{km}}{\partial V^{c}_{m}} & 0 & 0 & \frac{\partial P^{c}_{km}}{\partial B^{c}_{CSCT}} \end{array}\right] \cdot \left[\begin{array}{c} \Delta\theta^{a}_{k} \\ \Delta\theta^{b}_{k} \\ \Delta\theta^{c}_{k} \\ \Delta V^{a}_{k} \\ \Delta V^{b}_{k} \\ \Delta V^{c}_{k} \\ \Delta\theta^{a}_{m} \\ \Delta\theta^{b}_{m} \\ \Delta\theta^{c}_{m} \\ \Delta V^{a}_{m} \\ \Delta V^{b}_{m} \\ \Delta V^{c}_{m} \\ \hline \Delta B^{a}_{CSCT} \\ \Delta B^{b}_{CSCT} \\ \Delta B^{c}_{CSCT} \end{array}\right] = - \left[\begin{array}{c} P^{a}_{k} \\ P^{b}_{k} \\ P^{c}_{k} \\ Q^{a}_{k} \\ Q^{b}_{k} \\ Q^{c}_{k} \\ P^{a}_{m} \\ P^{b}_{m} \\ P^{c}_{m} \\ Q^{a}_{m} \\ Q^{b}_{m} \\ Q^{c}_{m} \\ \hline P^{a}_{km} - P^{a}_{esp} \\ P^{b}_{km} - P^{b}_{esp} \\ P^{c}_{km} - P^{c}_{esp} \end{array}\right] \qquad (2.91)$$

2.4 Exemplo Numérico

A seguir, na Figura 2.15 é apresentado um sistema de três barras a fim de demonstrar a utilização do método proposto nesse capítulo.

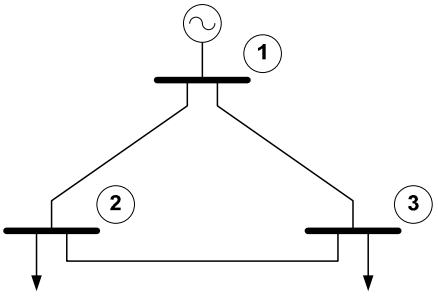

Figura 2.15 - Sistema Exemplo

Na Tabela 2.1 apresentam-se os dados das barras: tipo, módulo e ângulo da tensão. O módulo da tensão está em pu, enquanto o ângulo em radianos.

Na Tabela 2.2 são apresentados os dados de carga em cada barra do sistema. Estes dados também estão em pu.

Tabela 2.1 - Dados das barras

	Tipo	**V_a (pu)**	**$\angle_a$ (rad)**	**V_b (pu)**	**$\angle_b$ (rad)**	**V_c (pu)**	**$\angle_c$ (rad)**
Barra 1	Ref	1	0	1	-2.0944	1	2.0944
Barra 2	PQ	1	0	1	-2.0944	1	2.0944
Barra 3	PQ	1	0	1	-2.0944	1	2.0944

Tabela 2.2 - Dados das Cargas

	P_a	**Q_a**	**P_b**	**Q_b**	**P_c**	**Q_c**
Barra 1	0,00	0,00	0,00	0,00	0,00	0,00
Barra 2	0,40	0,20	0,50	0,10	0,45	0,15
Barra 3	0,80	0,30	0,90	0,35	0,70	0,28

Os dados de impedância de seqüência abc das linhas estão em pu como pode ser visto na Tabela 2.3

Tabela 2.3 - Dados das Linhas

	De	**Para**	**Z_{aa}**	**Z_{bb}**	**Z_{cc}**	**Z_{ab}**	**Z_{ac}**	**Z_{ca}**
Linha 1-2	1	2	0.01+0.1j	0.01+0.1j	0.01+0.1j	0.05j	0.05j	0.05j
Linha 2-3	2	3	0.01+0.1j	0.01+0.1j	0.01+0.1j	0.05j	0.05j	0.05j
Linha 1-3	1	3	0.01+0.1j	0.01+0.1j	0.01+0.1j	0.05j	0.05j	0.05j

No exemplo apresentado fez a consideração da Equação (2.93):

$$Z_{ab} = Z_{ba} \qquad (2.92)$$
$$Z_{ac} = Z_{ca}$$
$$Z_{bc} = Z_{cb}$$

2.4.1 Primeira Iteração

Calcula-se o vetor independente g(x)

$$g(x) = \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ -0.40000 \\ -0.50000 \\ -0.45000 \\ -0.20000 \\ -0.10000 \\ -0.15000 \\ -0.80000 \\ -0.90000 \\ -0.70000 \\ -0.30000 \\ -0.35000 \\ -0.28000 \end{bmatrix} \quad (2.93)$$

Como o módulo do máximo valor de g(x) não é menor que a tolerância ($\varepsilon = 1e^{-6}$) o processo iterativo continua.

Calculando a matriz jacobiana tem-se como resultado a Matriz (2.94)

$$Jac = \begin{bmatrix}
1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 5.2945 & 9.4223 & -7.0244 & 28.9660 & 2.6711 & 6.8243 & -2.6472 & -4.7111 & 3.5122 & -14.4830 & -1.3356 & -3.4122 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -7.0244 & 5.2945 & 9.4223 & 6.8243 & 28.9660 & 2.6711 & 3.5122 & -2.6472 & -4.7111 & -3.4122 & -14.4830 & -1.3356 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 9.4223 & -7.0244 & 5.2945 & 2.6711 & 6.8243 & 28.9660 & -4.7111 & 3.5122 & -2.6472 & -1.3356 & -3.4122 & -14.4830 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 28.9660 & 2.6711 & 6.8243 & -5.2945 & -9.4223 & 7.0244 & -14.4830 & -1.3356 & -3.4122 & 2.6472 & 4.7111 & -3.5122 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 6.8243 & 28.9660 & 2.6711 & 7.0244 & -5.2945 & -9.4223 & -3.4122 & -14.4830 & -1.3356 & -3.5122 & 2.6472 & 4.7111 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 2.6711 & 6.8243 & 28.9660 & -9.4223 & 7.0244 & -5.2945 & -1.3356 & -3.4122 & -14.4830 & 4.7111 & -3.5122 & 2.6472 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -2.6472 & -4.7111 & 3.5122 & -14.4830 & -1.3356 & -3.4122 & 5.2945 & 9.4223 & -7.0244 & 28.9660 & 2.6711 & 6.8243 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 3.5122 & -2.6472 & -4.7111 & -3.4122 & -14.4830 & -1.3356 & -7.0244 & 5.2945 & 9.4223 & 6.8243 & 28.9660 & 2.6711 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -4.7111 & 3.5122 & -2.6472 & -1.3356 & -3.4122 & -14.4830 & 9.4223 & -7.0244 & 5.2945 & 2.6711 & 6.8243 & 28.9660 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -14.4830 & -1.3356 & -3.4122 & 2.6472 & 4.7111 & -3.5122 & 28.9660 & 2.6711 & 6.8243 & -5.2945 & -9.4223 & 7.0244 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -3.4122 & -14.4830 & -1.3356 & -3.5122 & 2.6472 & 4.7111 & 6.8243 & 28.9660 & 2.6711 & 7.0244 & -5.2945 & -9.4223 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -1.3356 & -3.4122 & -14.4830 & 4.7111 & -3.5122 & 2.6472 & 2.6711 & 6.8243 & 28.9660 & -9.4223 & 7.0244 & -5.2945
\end{bmatrix} \quad (2.94)$$

Resolvendo a Equação 2.17 encontrados o vetor de correção Δx mostrado no Vetor (2.95)

$$\Delta x = \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ -0.02358 \\ -0.01400 \\ -0.00992 \\ -0.02227 \\ -0.03657 \\ -0.02007 \\ -0.02725 \\ -0.01959 \\ -0.01217 \\ -0.02812 \\ -0.04322 \\ -0.02347 \end{bmatrix} \quad (2.95)$$

Atualizando as tensões e ângulos temos o resultado mostrado na Tabela 2.4

Tabela 2.4 - Resultados atualizados (1˚ Iteração)

	Tipo	V_a	$\angle_a$	V_b	$\angle_b$	V_c	$\angle_c$
Barra 1	Ref	1.00000	0.00000	1.00000	-2.09440	1.00000	2.09440
Barra 2	PQ	0.97642	-0.02227	0.98600	-2.13096	0.99008	2.07433
Barra 3	PQ	0.97275	-0.02812	0.98042	-2.13761	0.98783	2.07093

2.4.2 Segunda Iteração

Calcula-se o vetor independente g(x)

$$g(x) = \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ -0.00805 \\ -0.00588 \\ -0.00470 \\ -0.00986 \\ -0.01314 \\ -0.00609 \\ -0.02494 \\ -0.02082 \\ -0.01341 \\ -0.01294 \\ -0.02432 \\ -0.00834 \end{bmatrix} \quad (2.96)$$

Como o módulo do máximo valor de g(x) não é menor que a tolerância (ε =1e-6) o processo iterativo continua.

Calculando a matriz jacobiana tem-se o resultado da Matriz (2.97).

$$Jac = \begin{bmatrix}
1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 4.7682 & 9.1619 & -6.8441 & 27.8060 & 2.7011 & 6.6122 & -2.5020 & -4.5717 & 3.4334 & -13.7710 & -1.3727 & -3.2871 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -6.8291 & 4.7192 & 9.3325 & 6.6662 & 28.2480 & 2.4555 & 3.4345 & -2.5151 & -4.6620 & -3.3011 & -14.0170 & -1.2406 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 9.3230 & -7.0652 & 4.7922 & 2.6023 & 6.5480 & 28.5380 & -4.6536 & 3.5546 & -2.5722 & -1.3228 & -3.2324 & -14.1740 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 28.0790 & 2.7394 & 6.6785 & -5.4397 & -9.0336 & 6.7762 & -14.1560 & -1.4002 & -3.3276 & 2.4338 & 4.4822 & -3.3916 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 6.8272 & 28.4720 & 2.4801 & 6.6681 & -5.6414 & -9.2400 & -3.3935 & -14.2970 & -1.2559 & -3.3409 & 2.4659 & 4.6053 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 2.6651 & 6.6410 & 28.5330 & -9.1031 & 6.9663 & -5.6352 & -1.3598 & -3.2969 & -14.3480 & 4.5268 & -3.4850 & 2.5409 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -2.6575 & -4.5716 & 3.3897 & -13.7410 & -1.3191 & -3.3134 & 4.3534 & 9.1253 & -6.8021 & 27.6960 & 2.6829 & 6.5889 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 3.3726 & -2.6898 & -4.6479 & -3.3362 & -13.9830 & -1.1902 & -6.7850 & 4.2940 & 9.2876 & 6.6088 & 28.1680 & 2.4063 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -4.6554 & 3.5133 & -2.6636 & -1.2828 & -3.2784 & -14.1560 & 9.2952 & -7.0707 & 4.5350 & 2.6088 & 6.4736 & 28.5370 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -14.0730 & -1.3379 & -3.3466 & 2.5948 & 4.5076 & -3.3561 & 27.9830 & 2.7365 & 6.6701 & -5.7850 & -8.9466 & 6.7193 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -3.4167 & -14.1820 & -1.2022 & -3.2931 & 2.6521 & 4.6018 & 6.7939 & 28.0670 & 2.4359 & 6.6002 & -5.9683 & -9.1746 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -1.3137 & -3.3249 & -14.2980 & 4.5456 & -3.4641 & 2.6372 & 2.6819 & 6.6030 & 28.3390 & -9.0420 & 6.9322 & -5.8530
\end{bmatrix} \quad (2.97)$$

Resolvendo a Equação 2.17 encontrados o vetor de correção Δx mostrado no Vetor (2.98).

$$\Delta x = \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ -0.00084 \\ -0.00115 \\ -0.00018 \\ -0.00038 \\ -0.00063 \\ -0.00033 \\ -0.00099 \\ -0.00144 \\ -0.00019 \\ -0.00062 \\ -0.00092 \\ -0.00048 \end{bmatrix} \quad (2.98)$$

Atualizando as tensões e ângulos temos o resultado mostrado na Tabela 2.5

Tabela 2.5 - Resultados atualizados (2° Iteração)

	Tipo	$\mathbf{V_a}$	$\angle_a$	$\mathbf{V_b}$	$\angle_b$	$\mathbf{V_c}$	$\angle_c$
Barra 1	Ref	1.00000	0.00000	1.00000	-2.09440	1.00000	2.09440
Barra 2	PQ	0.97558	-0.02265	0.98485	-2.13159	0.98990	2.07399
Barra 3	PQ	0.97177	-0.02874	0.97898	-2.13853	0.98764	2.07045

2.4.3 Terceira Iteração

Calcula-se o vetor independente g(x)

$$g(x) = \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ -0.00002 \\ 0.00000 \\ -0.00003 \\ -0.00004 \\ -0.00001 \\ -0.00011 \\ -0.00004 \\ -0.00001 \end{bmatrix} \quad (2.99)$$

Como o módulo do máximo valor de g(x) não é menor que a tolerância ($\varepsilon = 1e^{-6}$) o processo iterativo continua.

Calculando a matriz jacobiana tem-se o resultado a Matriz (2.100).

$$Jac = \begin{bmatrix}
1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 4.7552 & 9.1533 & -6.8379 & 27.7690 & 2.6978 & 6.6057 & -2.4965 & -4.5670 & 3.4307 & -13.7450 & -1.3719 & -3.2833 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -6.8195 & 4.7065 & 9.3223 & 6.6543 & 28.1950 & 2.4495 & 3.4304 & -2.5080 & -4.6568 & -3.2939 & -13.9810 & -1.2382 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 9.3212 & -7.0659 & 4.7864 & 2.6000 & 6.5371 & 28.5340 & -4.6524 & 3.5559 & -2.5697 & -1.3225 & -3.2250 & -14.1690 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 28.0430 & 2.7394 & 6.6731 & -5.4390 & -9.0146 & 6.7689 & -14.1450 & -1.4014 & -3.3244 & 2.4260 & 4.4710 & -3.3884 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 6.8209 & 28.4260 & 2.4745 & 6.6529 & -5.6352 & -9.2282 & -3.3896 & -14.2810 & -1.2537 & -3.3336 & 2.4553 & 4.5992 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 2.6651 & 6.6377 & 28.5220 & -9.0935 & 6.9589 & -5.6381 & -1.3609 & -3.2943 & -14.3460 & 4.5210 & -3.4812 & 2.5380 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -2.6582 & -4.5670 & 3.3853 & -13.7150 & -1.3163 & -3.3104 & 4.3218 & 9.1152 & -6.7942 & 27.6530 & 2.6789 & 6.5820 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 3.3658 & -2.6899 & -4.6418 & -3.3302 & -13.9460 & -1.1856 & -6.7731 & 4.2639 & 9.2751 & 6.5944 & 28.1110 & 2.3982 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -4.6546 & 3.5132 & -2.6651 & -1.2810 & -3.2734 & -14.1500 & 9.2931 & -7.0723 & 4.5203 & 2.6070 & 6.4598 & 28.5350 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -14.0580 & -1.3365 & -3.3442 & 2.5933 & 4.4978 & -3.3511 & 27.9450 & 2.7364 & 6.6643 & -5.7997 & -8.9235 & 6.7103 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -3.4135 & -14.1600 & -1.1977 & -3.2836 & 2.6492 & 4.5949 & 6.7859 & 28.0000 & 2.4282 & 6.5818 & -5.9741 & -9.1605 \\
0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & -1.3130 & -3.3238 & -14.2950 & 4.5410 & -3.4599 & 2.6382 & 2.6827 & 6.5986 & 28.3250 & -9.0307 & 6.9236 & -5.8644
\end{bmatrix} \quad (2.100)$$

Resolvendo a Equação 2.17 encontrados o vetor de correção Δx mostrado no Vetor (2.101).

$$\Delta x = \begin{bmatrix}
0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\
0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\
-0.00001 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000
\end{bmatrix} \quad (2.101)$$

Atualizando as tensões e ângulos temos o resultado mostrado na Tabela 2.6

Tabela 2.6 - Resultados atualizados (3° Iteração)

	Tipo	$\mathbf{V_a}$	$\mathbf{\angle_a}$	$\mathbf{V_b}$	$\mathbf{\angle_b}$	$\mathbf{V_c}$	$\mathbf{\angle_c}$
Barra 1	Ref	1.00000	0.00000	1.00000	-2.09440	1.00000	2.09440
Barra 2	PQ	0.97558	-0.02265	0.98484	-2.13159	0.98990	2.07399
Barra 3	PQ	0.97176	-0.02874	0.97897	-2.13854	0.98765	2.07045

2.4.4 Última Iteração

Calcula-se o vetor independente g(x)

$$g(x) = \begin{bmatrix} 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \\ 0.00000 \end{bmatrix} \qquad (2.102)$$

Como o módulo do máximo valor de g(x) é menor que a tolerância ($\varepsilon = 1e^{-6}$) o processo iterativo termina.

O resultado do caso proposto neste item do trabalho é apresentado na Tabela 2.6.

2.5 Conclusões Parciais

Neste capítulo foi apresentada uma nova formulação para o cálculo do fluxo de potência trifásico (FPTP) para sistemas equilibrados e desequilibrados. Além disso, foram modelados os diversos tipos que equipamentos que compõe o sistema elétrico, tais como linhas, cargas, transformadores, máquinas, compensador estático de reativos (CER), Compensação Série Controlada a Tiristores (CSCT), Reguladores de Tensão, etc. Este método se mostrou de fácil implementação, principalmente de controles, pois a metodologia já trabalha com as variáveis que normalmente são controladas (Ex. Tensão), não necessitando de transformações adicionais durante o processo iterativo.

Capítulo 3 Modelagem Computacional

3.1 Introdução

A análise do fluxo de potência é uma ferramenta importante nos estudos dos sistemas de distribuição de energia elétrica. Todavia, um perfeito aproveitamento e análise das informações obtidas, estão ligados a implementação computacional. Desta forma, o programa deve ser implementado observando-se a estrutura de funcionamento dos sistemas elétricos (GARCIA, 2001).

Os conceitos de Modelagem Orientada a Objetos vêm cada vez mais sendo aplicados no desenvolvimento de ferramentas computacionais para sistemas de potência (NEYER E WU, 1990; ZHOU, 1996; ESQUIVEL et. Al., 1998; MANZONI et. Al.,1998; AGOSTINI et. Al., 2002; ARAUJO et. Al., 2002). A MOO permite combinar a estrutura de dados com os diversos modelos dos componentes de uma rede elétrica. Assim, o programa é organizado em vários objetos separados que incorporam tanto a estrutura de dados quanto o comportamento destes.

Desse modo apresenta-se neste capítulo o desenvolvimento de uma plataforma computacional para análise de redes elétricas utilizando modelagem orientado a objetos. Nessa plataforma implementou-se o Fluxo de Potência Trifásico em Coordenadas Polares apresentada no capítulo anterior assim como modelos para a representação gráfica e numérica dos diversos componentes do sistema elétrico.

A base computacional proposta é de simples entendimento, fácil de programar e alto desempenho computacional, para que diversas ferramentas possam ser implementadas por equipes distintas com a mínima dependência entre elas.

Outros tipos de modelos, sistemas hexafásicos como exemplo, podem ser facilmente implementados. Esta plataforma também permite a representação dos diversos componentes (barras, seccionadores, geradores, linhas, transformadores, equipamentos de controle, etc.) de tal forma que diversas ferramentas implementadas possam interagir entre si e o banco de dados. Além disso, foram implementados também nessa plataforma modelos para a representação gráfica dos diversos componentes do sistema elétrico bem como sua interligação com as demais metodologias.

3.2 Classes Auxiliares

São as classes que não pertencem à rede elétrica em si. Tem como objetivo a transferência ou armazenamento de dados de forma genérica, a realização de operações matemáticas, solução de sistemas lineares e a compatibilidade de transferência de informações entre os diversos aplicativos.

3.2.1 Classe CMatriz

As formulações implementadas no programa computacional desenvolvido utilizam sub-matrizes e estes blocos são matrizes *m* x *n*, por isto tornou-se necessário o desenvolvimento de uma classe que possibilita o tratamento dessas matrizes. Por conseguinte foi criada uma classe chamada CMatriz, que permite a criação e manipulação de matrizes *m* x *n*.

Esta classe não será apresentada em detalhes, pois as operações matriciais são básicas. O único detalhe relevante é que os elementos da matriz são armazenados dinamicamente usando m_ptrMatriz, que é um ponteiro de ponteiros do tipo *template*. A Figura 3.1 mostra o diagrama da classe CMatriz.

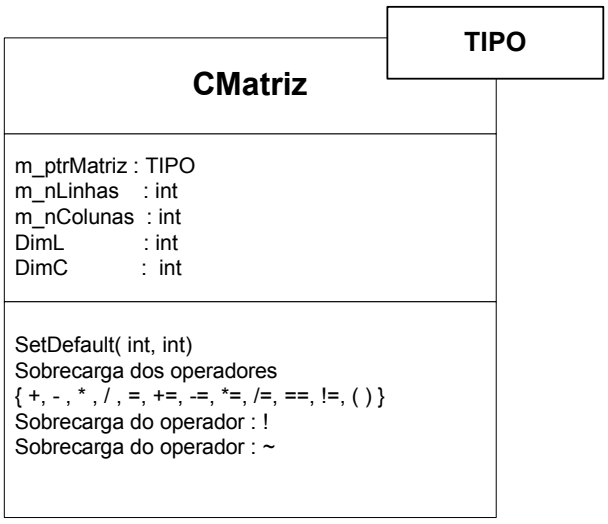

Figura 3.1 - Classe CMatriz

Tabela 3.1– Dados membros de CMatriz

Dados Membros	Descrição
TIPO** m_ptrMatriz	Armazena os valores numéricos da matriz da matriz.
int m_nLinhas	Indica o numero de linhas da matriz.
int m_nColunas	Indica o numero de colunas da matriz.
static int DimL, DimC	Variáveis utilizadas para definir as dimensões da matriz quando não forem explicitadas no construtor.

Tabela 3.2– Funções membros de CMatriz

Funções Membros	Descrição
CMatriz( )	O construtor da classe que utiliza as variáveis DimL e DimC para dimensionar a matriz.
CMatriz(int linha, int coluna)	Construtor da classe que possibilita o usuário dimensionar a matriz.
~ CMatriz( )	O destrutor da classe
static void SetDefault(int linha, int coluna)	Utilizada para modificar os valores de DimL e DimC.
Sobrecarga dos operadores {+, -, /, *, =, +=, -=, *=, /=, ==, !=}	Executa as operações matriciais básicas, definidas matematicamente pelos os operadores.
Sobrecarga do operador !	Inverte a matriz.
Sobrecarga do operador ~	Transpõe a matriz.

Duas ressalvas devem ser feitas:

1 – A utilização de alocações dinâmicas não é aconselhável para a solução de sistemas lineares (várias alocações e desalocações), e como os blocos de dados possuem dimensões fixas dependendo da metodologia utilizada, esta classe permite que m_ptrMatriz seja uma matriz fixa [*m*][*n*], aumentando a eficiência computacional.

2 – Esta classe possui um tratamento para operações com matrizes com grande número de zeros.

3.2.2 Classe CTransferencia

Esta classe é responsável pela transferência de dados entre aplicações de forma genérica. Por exemplo, durante o processo de solução do Fluxo de Potência, é necessária a transferência de elementos entre classes. Estes elementos na verdade são

blocos do tipo CMatriz (estruturas blocadas) com dimensão variável. Na montagem da matriz Jacobiana os blocos são de dimensões 6x6, já no vetor independente estes blocos possuem dimensão 6x1.

Um outro problema ocorre quando é necessário montar alguma estrutura (Jacobiana, Hessiana, etc.) onde um componente deve fornecer mais de um bloco, como as contribuições das linhas para a montagem da matriz um $\mathbf{Y}_{barra}$. Esta classe transfere quantos blocos de um componente forem necessários de uma só vez, aumentando a velocidade da metodologia implementada.

Apenas um bloco ou um conjunto de blocos não significa nada, por isto cada bloco deve possuir uma identificação de como o dado deve ser tratado. Na Figura 3.2 é apresentada o diagrama em UML.

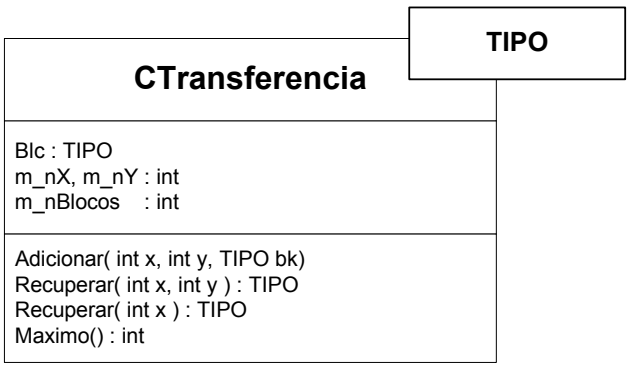

Figura 3.2 - Classe CTransferencia

Tabela 3.3 – Dados membros de CTransferencia

Dados Membros	Descrição
TIPO* Blc	Vetor para armazenamento de dados *templates*.
int* m_nX, m_nY	Identifica o conteúdo dos dados armazenados em *Blc.
int m_nMaximo	Quantidade de dados armazenados em CTransferencia.

Tabela 3.4 – Funções membros de CTransferencia

Funções Membros	Descrição
CTransferencia( )	O construtor da classe.
~ CTransferencia ( )	O destrutor da classe
void Adicionar(int x, int y, TIPO bk)	Adicionar um dado no objeto da classe Ctransferencia.
TIPO Recuperar(int x, int y)	Recuperar o dados com os identificadores x e y.

TIPO Recuperar(int n)	Recuperar um dado na enésima posição de Blc.
int Maximo()	Retorna o valor da variável m_nMaximo.

3.2.3 Classe CSistemaLinear

Considerando que os sistemas lineares representativos de sistemas elétricos de grande porte possuem a maioria dos elementos nulos, formas alternativas de armazenamento vêm sendo apresentadas na literatura (TINNEY, 1972), buscando a otimização do uso da memória, a redução do tempo computacional e a melhoria da robustez do processo numérico.

Em ARAUJO (2000) foi desenvolvida uma ferramenta matemática orientada a objetos para solução de sistemas lineares esparsos de grande porte, onde os elementos podem ser reais, complexos, matrizes blocadas ou qualquer outro tipo. Este ferramenta manipula sistemas simétricos e assimétricos de formas distintas visando um melhor desempenho computacional, ela também possui rotinas internas na qual o usuário não precisa se preocupar com a ordenação e nem com problemas de pivoteamento.

A Figura 3.3 apresenta esta ferramenta matemática encapsulada como uma classe denominada CSistemaLinear. A classe CEsparsa gerencia os processos de esparsidade, detalhes podem ser encontrados em ARAUJO (2000).

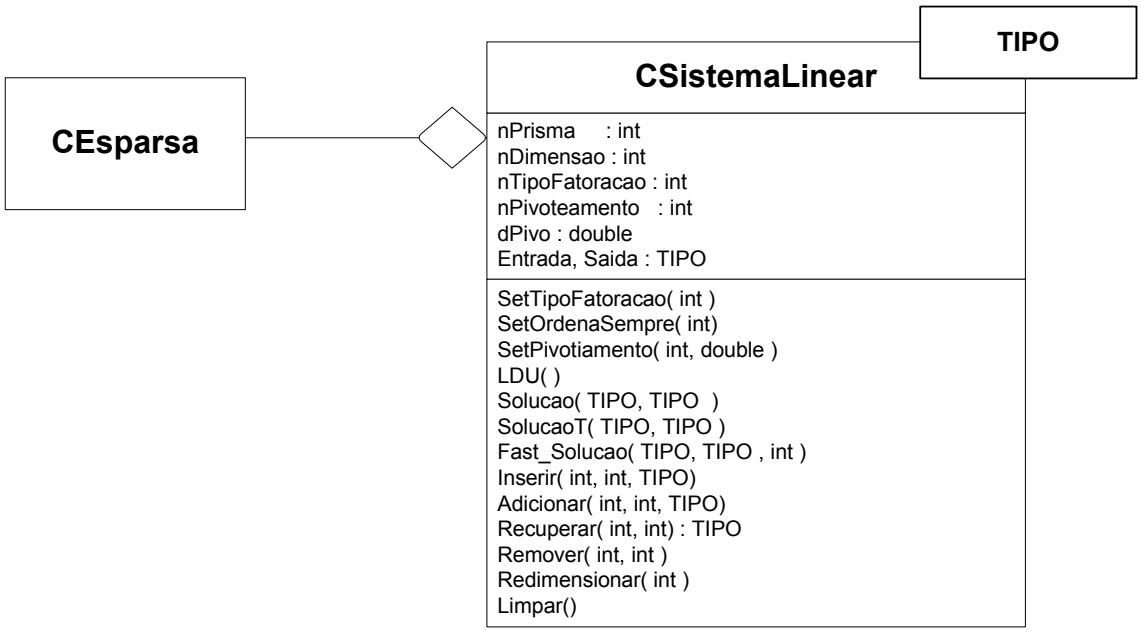

Figura 3.3 - Classe CSistema Linear

A Tabela 3.5 mostra os dados membros da classe CSistemaLinear e na Tabela 3.6 uma descrição detalhada do que realiza cada função. A classe agregada CEsparsa permite o tratamento de matrizes esparsas, esta classe encontra-se detalhada em ARAUJO (2000).

Tabela 3.5 – Dados membros de CSistemaLinear

Dados Membros	Descrição
int* nPrisma	Armazena a ordem de eliminação tornando o processo de ordenação invisível para o usuário.
int nDimensao	Armazena a dimensão do sistema linear.
int nTipoFatoracao	Indica o tipo de ordenação.
int nPivoteamento	Indica se será necessário o pivoteamento para evitar problemas numéricos.
doublé dPivo	Se for necessário o pivoteamento, dPivo será o menor valor admitido para pivotear.
TIPO* Entrada	Armazena o vetor independente.
TIPO* Saída	Armazena o vetor solução.

Tabela 3.6 – Funções membros de CSistemaLinear

Funções Membros	Descrição
CSistemaLinear( )	O construtor da classe, responsável pela inicialização de todas as variáveis.
~ CSistemaLinear ( )	O destrutor da classe
void SetTipoFatoracao(int tipo)	Ativa o tipo de fatoração a ser utilizada.
void SetOrdenaSempre(int ordena)	Força a reordenação a cada fatoração.
void SetPivoteamento(int tipo, double pivo)	Ordena e fatora o sistema linear simultaneamente para evitar problemas numéricos realizando testes nos pivôs para garantir estabilidade numérica.
void LDU( )	Fatora a sistema linear utilizando a fatoração LDU
void Solucao(TIPO* X, TIPO* B)	Faz a solução do sistema usando as rotinas de substituição direta e inversa. O vetor independente e solução são passados pelo argumento da função.
void SolucaoT(TIPO* X, TIPO* B)	Faz a solução do sistema linear transposto.
void Fast_Solucao(TIPO* X, TIPO* B, int pos)	Similar a função acima, sendo utilizada quando apenas um valor do vetor independente é não nulo, a posição em que se encontra o elemento não nulo é determinada pelo terceiro parâmetro da função. Este método é descrito em TINNEY (1985).
void Inserir(int x, int y, TIPO valor)	Insere um valor na posição (x,y) na matriz de coeficientes, desconsiderando o anterior.

void Adicionar(int x, int y, TIPO valor)	Adiciona um valor na posição (x,y) na matriz de coeficientes, somando com o anterior.
TIPO Recuperar(int x, int y)	Retorna o valor (x,y) da matriz de coeficientes.
void Remover(int x, int y)	Remove a posição (x,y) da matriz de coeficientes.
void Redimensionar(int n)	Redimensiona o sistema de equações lineares.
void Limpar()	Desaloca a memória e dimensiona o sistema linear para zero.

3.3 Modelagem dos Componentes do Sistema Elétrico

Para uma correta modelagem do componente do sistema elétrico, deve-se observar o comportamento físico e a conectividade. A Figura 3.4 mostra o modelo de objetos (estrutura de classes) proposta neste trabalho. Esta estrutura apresenta três níveis hierárquicos com funções e objetivos bem definidos.

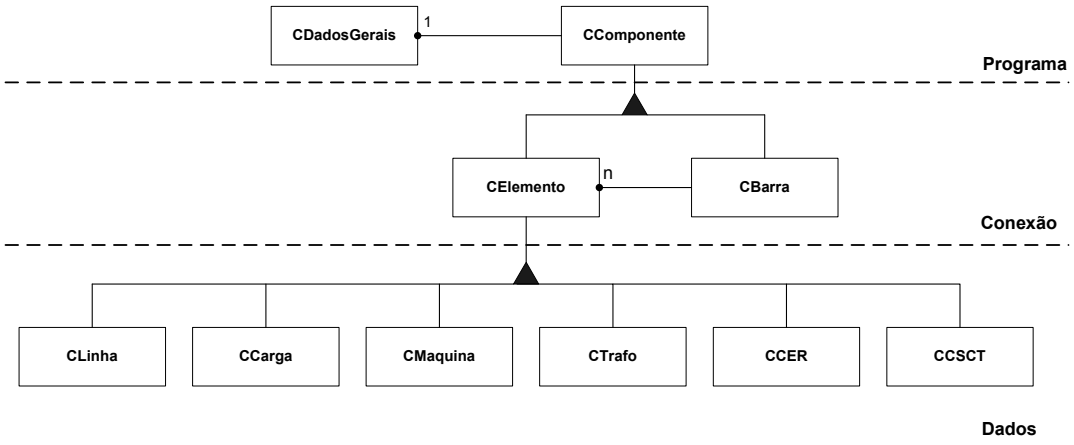

Figura 3.4 - Estrutura de classes

3.3.1 Nível Programa

No primeiro nível da Figura 3.4 encontra-se somente a classe CComponente. Ela é uma classe abstrata definida ao nível de programa e tem como único objetivo facilitar a implementação computacional através dos mecanismos de herança e funções virtuais.

Esta classe possui uma ligação com a classe CDadosGerais que é responsável pelo armazenamento de dados vitais necessários para uma correta modelagem matemática de ferramentas de análise de SEP.

3.3.1.1 Classe CDadosGerais

A classe CDadosGerais serve para armazenar dados e funções que devem estar disponíveis a todos equipamentos da rede, mas não pertencem ao equipamento ou a rede. Na Figura 3.5 é apresentado o diagrama de classe e nas Tabela 3.7 e Tabela 3.8 a descrição dos dados membros e métodos respectivamente.

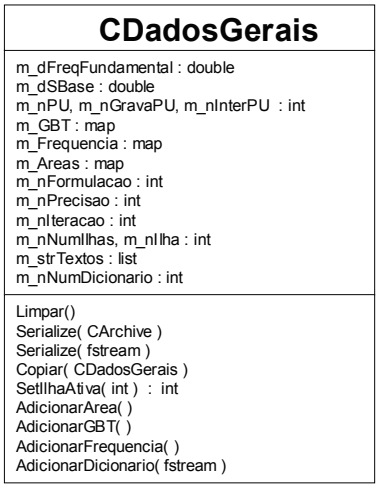

Figura 3.5 - Classe CDadosGerais

Tabela 3.7 – Dados membros de CDadosGerais

Dados Membros	Descrição
double m_dFreqFundamental	Armazena a freqüência do sistema.
double m_dSBase	Armazena a potência base.
int m_nPU	Indica se os dados estão em PU ou em unidades elétricas.
int m_nGravaPU	Indica se os dados serão salvo em PU ou em unidades elétricas.
int m_nInterPU	Indica se os dados apresentados na interface estarão em PU ou em unidades elétricas.
int m_nPrecisao	Indica a precisão que os números serão apresentados.
int m_nFormulacao	Indica se os dados são referentes a sistemas monofásicos, trifásicos ou quatro condutores.

int m_nIteracao	Armazena o número da iteração atual.
int m_nNumIlhas	Número de ilhas do sistema, ou seja, partes desconexas do sistema.
int m_nIlha	Indica a sub-rede ou ilha ativa.
map<double, double> m_GBT	Armazena os grupos base de tensão.
map<int, string> m_Area	Armazena as áreas do sistema.
map<double, complex> m_Frequencias	Armazena freqüências para análise harmônica.
map<string, CDicionario> m_Dic	Armazena dicionários de componentes.
int m_nNumDicionario	Número de dicionários armazenados.
list<string> m_strTextos	Armazena textos.

Tabela 3.8 – Funções membros de CDadosGerais

Funções Membros	Descrição
CDadosGerais( )	O construtor da classe, responsável pela inicialização de todas as variáveis.
~CDadosGerais( )	O destrutor da classe.
void Limpar( )	Desaloca memória das listas e inicializa as variáveis.
void Serialize(CArchive& ar)	São as funções de salvar e ler arquivos.
void Serialize(fstream& fio)	Sobrecarga das funções de arquivo
void Copiar(CDadosGerais &pDG)	Duplica os dados da classe.
void AdicionarGBT(double id, double valor)	Adiciona valor de um grupo base de tensão.
void AdicionarAreas(int id, string nome)	Adiciona uma área.
void AdicionarFrequencia(double freq, complex fq)	Adiciona uma freqüência.
void AdicionarDicionario(string id, string path)	Armazena um dicionário.
void AdicionarFrequencia(int id, double fq)	Adiciona uma freqüência.
int SetIlhaAtiva(int ilha)	Seleciona uma ilha para se tornar ativa, retorna a antiga.

3.3.1.2 Classe CComponente

Tem como objetivo compatibilizar a passagem de dados entre funções. O modelo gráfico de acordo com a UML para a representação da classe CComponente é mostrado na Figura 3.6, seus dados membros na Tabela 3.9 e métodos na Tabela 3.10.

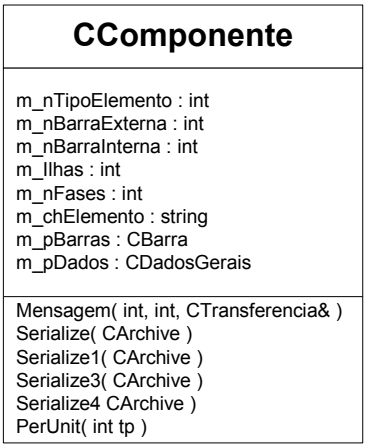

Figura 3.6 - Classe CComponente

Tabela 3.9 – Dados membros de CComponente

Dados Membros	Descrição
int m_nTipoElemento	Identifica o tipo do componente armazenando genericamente.
int m_nBarraExterna[n]	Número das barras externas conectadas ao componente, cada elementos pode conectar-se genericamente a *n* barras do sistema. Onde *n* é um valor definido em tempo de compilação.
int m_nBarraInterna[n]	Número interno das barras conectadas ao componente.
int m_nIlhas[5]	Indica a que ilha pertence cada fase do componente, ligações elétricas.
int m_nFases[5]	Indica se a fases esta ativa ou não para uma determinada ilha.
int m_nCircuito	Indica o número do circuito.
char m_chElemento[n][20]	Identifica se o elemento está ligado a uma chave e qual é a chave.
CBarra* m_pBarra[n]	Apontador para barra em que cada terminal está conectado, mesmo através de chaves.
CDadosGerais* pDados	Ponteiro para CdadosGerais, todos os elementos possuem um ponteiro que aponta para ela.

Tabela 3.10 – Funções membros de CComponente

Funções Membros	Descrição
CComponente( )	O construtor da classe, responsável pela inicialização de todas as variáveis.
~ CComponente( )	O destrutor da classe.
void Mensagem(int param1, int param2,	Esta é uma função genérica com objetivo

CTransferencia& tb)	de comunicação entre os objetos, os dois primeiros parâmetros são responsáveis pela identificação da mensagem e o terceiro o seu conteúdo.
void Serialize(fstream& fio)	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Serialize(Carchieve& ar)	Sobrecarga da função de ler arquivos utilizando mecanismo do VC++.

3.3.2 Nível Conexão

No segundo nível encontram-se as classes CElemento, CChave e CBarra. Este nível é utilizado para definir as associações entre os componentes de uma rede elétrica, ou seja, determina como os objetos poderão se relacionar.

3.3.2.1 Classe CElemento

A classe CElemento representa de forma genérica as conexões dos diversos equipamentos de um sistema elétrico, independentemente do número de conexões e da natureza do equipamento.

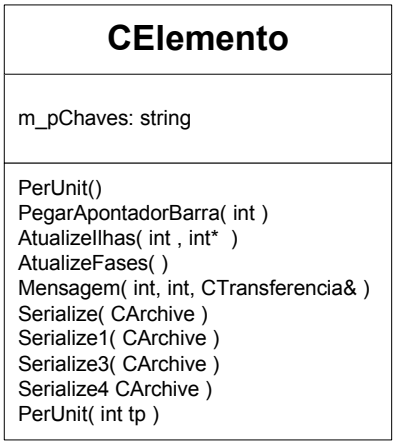

Figura 3.7 - Classe CElemento

A Figura 3.7 apresenta uma descrição dos dados membros e a Tabela 3.11 a funções membros.

Tabela 3.11 – Funções membros de CElemento

Funções Membros	Descrição
CElemento( )	O construtor da classe, responsável pela inicialização de todas as variáveis.
~ CElemento( )	O destrutor da classe.
void PerUnit(int tipo)	Transforma os dados para PU ou para unidades elétricas dependendo do parâmetro da função. Tipo=0 => Unidades Elétricas => p.u. tipo=1 => p.u. => Unidades Elétricas
CBarra* PegarApontadorBarra(int bar)	Retorna o endereço de memória de um objeto barra.
void AtualizeIlhas(int fase, int* Ilhas)	Aloca o equipamento em uma ilha dependendo da topologia do sistema.
void AtualizeFases()	Atualiza a variável da m_nFases da classe Ccomponente está de acordo com a topologia do sistema para determinada ilha.
void Mensagem(int metodologia, int estrutura, CTransferencia& tb)	Mensagem entre objetos
void Serialize(fstream& fio)	São as funções de salvar e ler arquivos utilizando mecanismo do C++..
void Serialize(CArchieve& ar)	Sobrecarga da função de ler arquivos utilizando mecanismo do VC++.

3.3.2.2 Classe CBarra

A classe CBarra armazena todas as informações relativas às barras e todas as posições de memória dos componentes conectados a ela. O armazenamento das posições de memória é realizado através de listas encadeadas de ponteiros. Uma característica importante é que uma barra pode estar associada a vários elementos ou chaves. Na Figura 3.8 é mostrado o diagrama da classe Cbarra, na Tabela 3.12 os dados membros e as funções membros na Tabela 3.13.

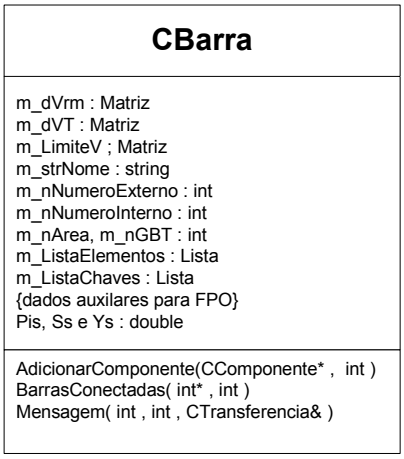

Figura 3.8 - Classe CBarra

Tabela 3.12 – Dados membros de CBarra

Dados Membros	Descrição
Matriz m_dVT	Armazena os valores da tensão em coordenadas polares.
char m_strNome[]	Armazena o nome da barra.
int m_nNumeroExterno	Armazena a designação externa da barra.
int m_nNumeroInterno	Armazena a designação interna da barra.
int m_nArea, m_nGBT	Indica a que área e GBT a barra pertence.
lista m_ListaElementos	Lista dos elementos conectados a barra.

Tabela 3.13 – Dados membros de CBarra

Funções Membros	Descrição
CBarra( )	O construtor da classe, responsável pela inicialização de todas as variáveis.
~CBarra( )	O destrutor da classe.
void AdicionarComponente(Ccomponente* pCmp, int tipo)	Adiciona um componente ou chave nas listas do objeto.
void BarrasConectadas(int* Barras, int fase)	Retorna todas a barras eletricamente conectadas em determinada fase.
void Mensagem(int metodologia, int estrutura, Ctransferencia& tb)	Mensagem entre objetos
void Serialize(fstream& fio)	São as funções de salvar e ler arquivos utilizando mecanismo do C++.

3.3.3 Nível Dados

As classes relativas aos elementos propriamente ditos (CLinha, CCarga, CTrafo2, etc.) encontram-se no terceiro nível. O objetivo destas classes é ler, escrever e armazenar os dados, nenhuma metodologia é manipulada por estas classes.

Nas classes desta subseção, as funções Serialize(), PerUnit(), Limite() e Mensagem() são reescritas para manipular corretamente os dados. Como as especificações destas funções são descritas, de forma genérica, na Tabela 3.11 referente a classe base CElemento, elas não serão reespecificadas em tabelas, mas serão apresentadas nos diagramas. Os métodos não definidos na classe bases serão apresentados.

3.3.3.1 CLinha

Esta classe armazena todos os dados relevantes de uma linha de transmissão CA para as diversas metodologias implementadas na aplicação. O diagrama é apresentado na Figura 3.9.

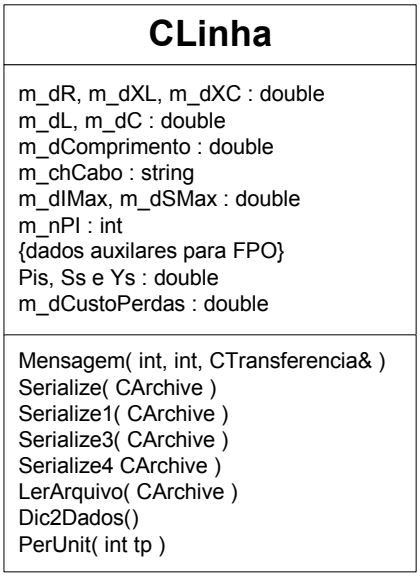

Figura 3.9 - Classe CLinha

Uma detalhada descrição dos dados membros é apresentada na Tabela 3.14 e na Tabela 3.15 os novos métodos.

Tabela 3.14 – Dados membros de CLinha

Dados Membros	Descrição
Matriz m_dR	Armazena os valores de resistência das fases e as mútuas.
Matriz m_dXL	Armazena os valores de reatância das fases e as mútuas.
Matriz m_dXC	Armazena os valores de susceptância em derivação das fases e as mútuas.
Matriz m_dL	Armazena os valores de indutância fases e as mútuas.
Matriz m_dC	Armazena os valores das capacitâncias em derivação das fases e as mútuas.
doublé m_dComprimento	Comprimento do cabo em Km.
string m_chCabo[]	Armazena informações do cabo dos condutores.
double m_dIMax	Valor máximo de corrente permitida na fase.
double m_dSMax	Valor máximo de potência permitida na fase.

Tabela 3.15 – Funções membros de CLinha

Funções Membros	Descrição
void LerDados(Carchive)	Utilizado quando está definido no arquivo o tipo do cabo invés dos dados de impedância, para utilizar está função o dicionário de elementos deve ser carregado.

3.3.3.2 CCarga

Representa as cargas do sistema, pode armazenar os dados em forma de potência para um fluxo de potência ou fluxo de potência ótimo, como também em valores de resistência e impedância corrigidos pela tensão para a analise harmônica. O diagrama padrão é apresentado na Figura 3.10, os dados membros podem ser observados em detalhes na Tabela 3.16.

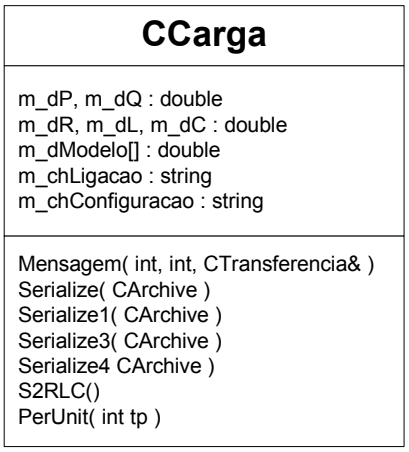

Figura 3.10 - Classe CCarga

Tabela 3.16 – Dados membros de CCarga

Dados Membros	Descrição
Matriz m_dP	Armazena os valores de potência ativa das fases.
Matriz m_dQ	Armazena os valores de potência reativa das fases.
double m_dModelo[]	Modelo ZIP da carga.
char m_chligacao[2]	Tipo de carga: [0] => Indica se a carga está conectada em estrela (y) ou delta (d). [1] => Indica se a carga está conectada no solo (t), no cabo neutro (n) ou flutuando (f). Está opção só pode ser utilizada em sistemas com 4 fios.
Matriz m_dR, m_dL, m_dC	Transforma os valores da carga PQ em valores de RLC para um determinado nível de tensão.
char m_chConfiguracao	Identifica se os componentes RLC da carga estão conectados em série ou paralelo.

A função membro S2RLC() tem como objetivo transformar a carga S em dados de RLC, utilizando a tensão atual da barra onde está conectada e de acordo com o modelo referido em m_chConfiguracao.

3.3.3.3 CMaquina

Representa as máquinas do sistema. Subestações, barras Vθ, barras PV e co-geração são modeladas nesta classe. O diagrama da máquina é ilustrado na Figura 3.11 e os dados membros na Tabela 3.17.

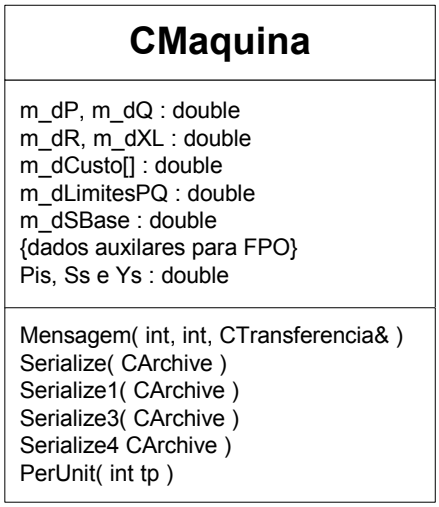

Figura 3.11 - Classe CMaquina

Tabela 3.17 – Dados membros de CMaquina

Dados Membros	Descrição
Matriz m_dP	Armazena os valores de potência ativa gerada da máquina.
Matriz m_dQ	Armazena os valores de potência reativa gerada da máquina.
Matriz m_dR	Armazena os valores de resistência da máquina.
Matriz m_dXL	Armazena os valores de reatância da máquina.
Matriz m_dLimitesPQ	Limites de geração da máquina.
double m_dSBase	Potência base da máquina.

3.3.3.4 CTrafo2

A classe CTrafo2 (Figura 3.12) representada os transformadores de 2 e 3 enrolamentos. Esta classe também pode representar as diversas conexões entre primário e secundário, estrela-delta, por exemplo. Os dados membros são apresentados Tabela 3.18.

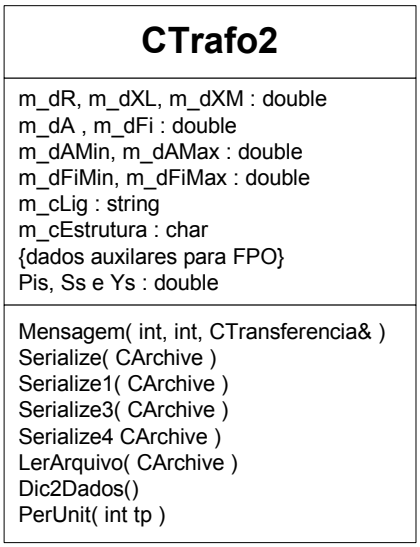

Figura 3.12 - Classe CTrafo

Tabela 3.18 – Dados membros de CTrafo

Dados Membros	Descrição
Matriz m_dR	Armazena os valores de resistência do enrolamento.
Matriz m_dXL	Armazena os valores de reatância do enrolamento.
Matriz m_dXM	Armazena os valores de reatância de magnetização.
char m_cLig[4]	[0] e [1] tipo da conexão do primário. [2] e [3] tipo da conexão do secundário. O primeiro valor informa se a conexão e estrela ou delta, o segundo se aterrada ou não.
char m_cEstrutura	Informa se é um transformador trifásico ou bancos de transformadores monofásicos.

As funções membros extras possuem a mesma definição que os métodos da classe CLinha (Tabela 3.15).

3.3.3.5 CCER

Os compensadores estáticos de reativos em derivação (CER) são representados por esta classe. O diagrama pode ser observado na Figura 3.13 e os dados membros na Tabela 3.19.

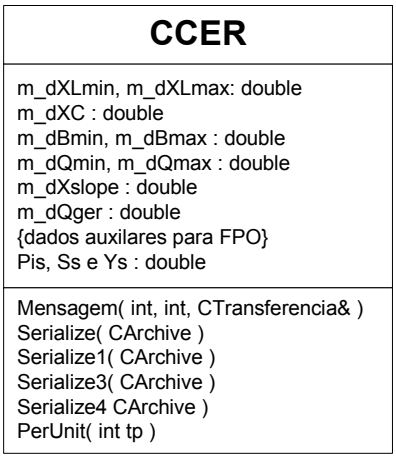

Figura 3.13 - Classe CCER

Tabela 3.19 – Dados membros de CCER

Dados Membros	Descrição
Matriz m_dXLmin, m_dXLmax	Dependendo do ângulo de disparo o valor de indutância pode variar entre o valor 0 e o valor nominal.
Matriz m_dXC	Valor da reatância capacitiva..
Matriz m_dBmin, m_dBmax	Susceptância mínima e máxima do CER.
Matriz m_dQmin, m_dQmax	Potência reativa mínima e máxima gerada pelo CER.
Matriz m_dXslope	Inclinação da reta relativa à região controlável do CER.
Matriz m_dQger	Geração atual.

3.3.3.6 CCSCT

Armazena os dados de Compensadores Série Controlado a Tiristor (CSCT), o modelo pode ser conferido na Tabela 3.20 e os dados membros na Figura 3.14.

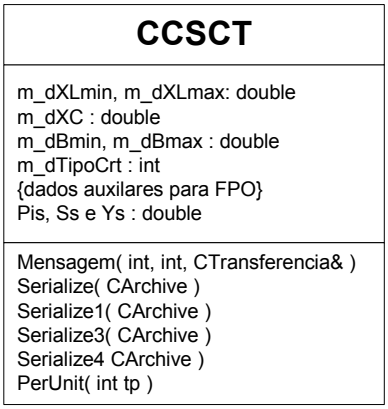

Figura 3.14 - Classe CCSCT

Tabela 3.20 – Dados membros de CCSCT

Dados Membros	Descrição
Matriz m_dXLmin, m_dXLmax	Dependendo do ângulo de disparo o valor de indutância pode variar entre o valor 0 e o valor nominal.
Matriz m_dXC	Valor da reatância capacitiva.
Matriz m_dBmin, m_dBmax	Susceptância mínima e máxima do CSCT.
int m_dTipoCrt	Indica o tipo de controle, controle de tensão ou de fluxo.

3.4 Modelagem do Sistema Elétrico

Na seção anterior apresentou-se a estrutura de classes em nível de dados dos componentes de um sistema de potência. Contudo, para realizar qualquer tipo de cálculo deve-se ainda instanciar e conectar os objetos de tal forma que uma rede seja totalmente caracterizada. Depois se deve definir os modelos de componentes para cada aplicação

Para isso, criou-se uma nova classe, denominada CRede. Esta classe possui como objetivo a montagem topológica da rede assim como a manutenção desta (abertura de chaves, retirada de equipamentos, etc.). É importante ressaltar que nenhuma tarefa relacionada às aplicações é realizada por esta classe.

O diagrama de CRede é apresentado na Figura 3.15 e os dados nas Tabela 3.21 e Tabela 3.22.

Tabela 3.21 – Dados membros de CRede

Dados Membros	Descrição
CDadosGerais m_Dados	Classe agregada para armazenar dados gerais.
list<CBarra> m_ListaBarras	Armazena todas os objetos barras em listas encadeadas.
list<CElemento> m_ListaElementos	Armazena todas os objetos elementos e derivados em listas encadeadas.
list<CChave> m_ListaChaves	Armazena todas os objetos chaves em listas encadeadas.
list<CBarra> m_ListIlhas[]	É um vetor de listas encadeadas, onde em cada posição estão os objetos barras pertencentes a cada ilha.
int m_nRadial[]	Indica se a rede é radial ou não.
char m_strIdentificacao[]	Pode armazenar diversos dados relevantes para a rede, nome, estado, etc.
map<string, int> m_ListIntExt	Mapeia identificadores de barras externo em interno.
int m_nConfiavel	Indica se a rede está configurada e pronta para manipulações.

Tabela 3.22 – Funções membros de CRede

Funções Membros	Descrição
CRede( )	O construtor da classe, responsável pela inicialização de todas as variáveis.
~CRede( )	O destrutor da classe.
void MontarApontadores()	Inicializa os apontadores das classes, os equipamentos apontam para as barras e chaves a que estão conectados. As barras para os equipamentos.
void NumeracaoInternaTMP()	Criar uma numeração interna para as barras em números seqüenciais [1..n^{o}Barras].
void MostreIlhas(int* Ilhas, int NumIlhas, ofstream& fout)	Imprime as barras pertencentes a cada ilha.
void SetIlhaAtiva(int ilha)	Marca qual ilha estará ativa para execução do aplicativo.
int GetIlhaAtiva( )	Retorna a ilha ativa.
void ApagueTodos(int tipo)	Apaga todos os objetos do tipo passado pelo parâmetro da função.
CCompoente* CriarElemento(int tipo)	Cria um determinado tipo de elemento.
void Limpar( )	Apaga todos os objetos da rede e a inicializa.
void Serialize(CArchive& ar)	Lê e salva em disco utilizando o mecanismo CArchive da MFC
void Serialize(char strPath[])	Lê e salva quando é especificado o caminho do arquivo.
void CopiarRede(Crede* pRd)	Cria uma cópia da rede.
Funções de Procura	
CComponente* AchePrimeiro(int tipo)	Retorna o primeiro componente da lista de um determinado tipo.
CComponente* AcheProximo(int tipo)	Retorna o próximo componente da lista.
CComponente* AchePrimeiroIlha(int tipo)	Retorna o primeiro componente pertencente à ilha ativa.
CComponente* AcheProximoIlha(int tipo)	Retorna o próximo componente da ilha ativa.
CBarra* AcheBarra(int bar, int tipo)	Retorna o endereço de um objeto barra segundo a numeração interna (tipo = 0) ou a numeração externa (tipo = 1).
int Numero (int tipo)	Retorna o número de componentes de um determinado tipo.
int NumeroBarrasIlha(int ilha)	Retorna o número de barras de uma determinada ilha.
Configuração	
void ConFigurarRede()	Inicia processo de configuração de rede, monta apontadores, detecta ilhamentos, cargas não atendidas, etc
void Configurando(Cbarra* pBar, int fase,	Função recursiva para detectar ilhamentos.

int* Ilhas, int& NumIlhas, int& removidas)	
void ConfigurandoElementos(int fase, int* Ilhas)	Atualiza o estado dos elementos, se estão ativos em determinada ilha, se estão energizados.
int TesteConectividade(int* Ilhas)	Testa se o sistema está todo conexão.
int AindaFalta(int* Ilhas)	Testa se algum elemento ainda não foi designado a uma ilha.
void AdicioneNaIlha(Cbarra* pBar, int ilha, int fase)	Inclui um elemento para uma ilha.

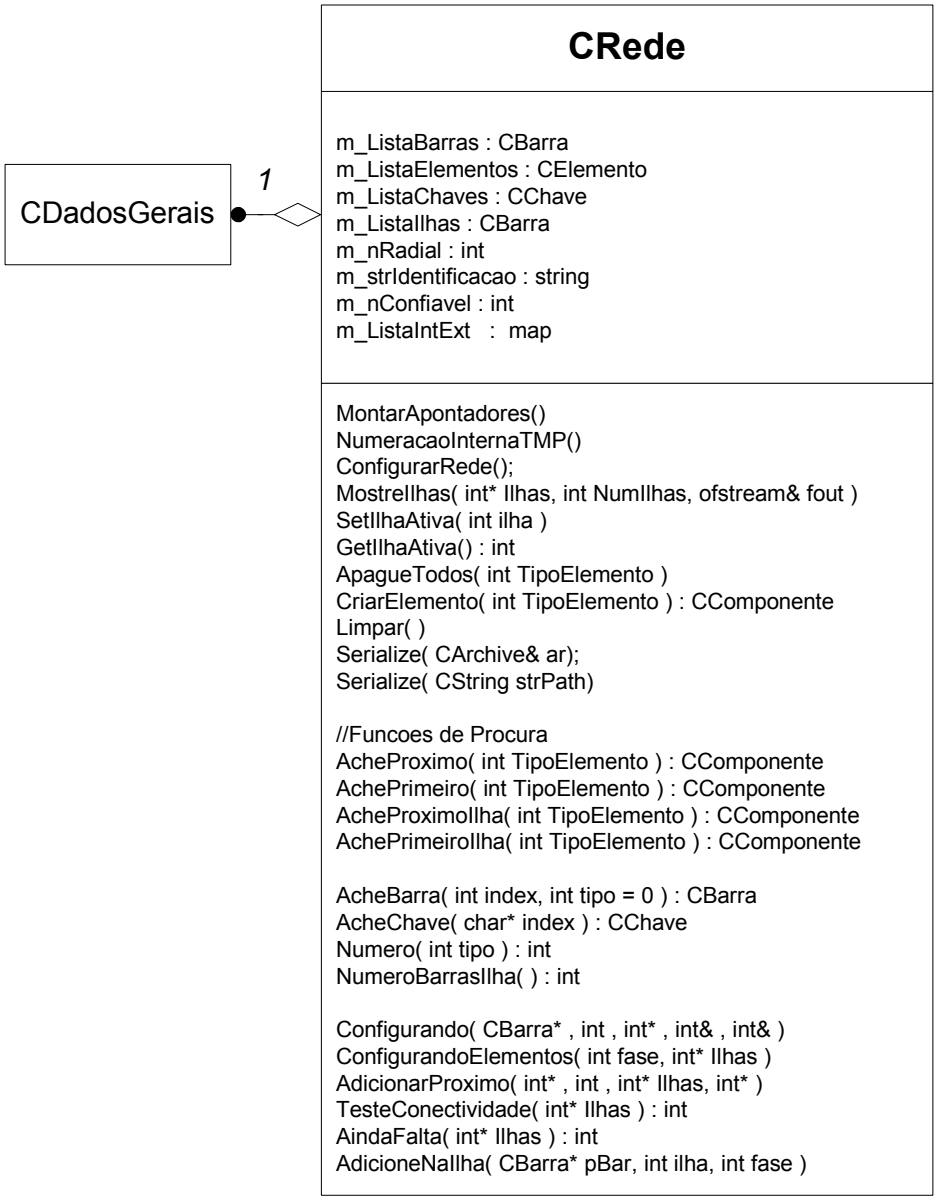

Figura 3.15 - Classe CRede

A Figura 3.16 ilustra a associação entre as classes CRede e CComponente. Como pode ser visto, são classes agregadas e a condição mínima para criação de uma rede é a existência de pelo menos três objetos.

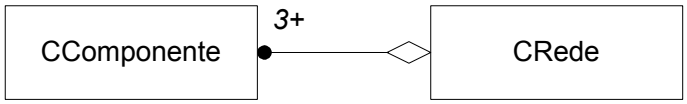

Figura 3.16 - Associação das classes CComponente e CRede

3.4.1 Funcionamento do Modelo Proposto

Considere o sistema mostrado na Figura 3.17. Este sistema é composto por três barras, um gerador, uma linha, um transformador e duas cargas.

Primeiramente objetos são armazenados em listas encadeadas segundo suas propriedades, ou seja, segundo as classes CBarra e CElemento. A Figura 3.18 ilustra o armazenamento dos objetos.

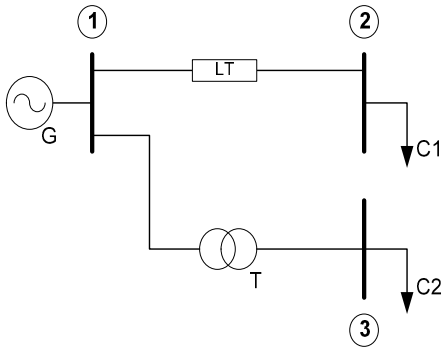

Figura 3.17 - Sistema exemplo três barras

Posteriormente a classe CRede conecta os objetos conforme a topologia do sistema e obedecendo as associações definidas na Figura 3.17. A Figura 3.19 ilustra este procedimento. É importante ressaltar que as ligações mostradas nas Figura 3.18 e Figura 3.19 existem simultaneamente. As ligações da Figura 3.19 determinam a conectividade do sistema, incluindo a conectividade por fases. Por exemplo, caso a carga da barra três seja monofásica conectada na *fase a* e o transformador entre as barras 1 e 3 monofásico na *fase c*, a classe CRede detectará esta desconectividade, emitindo uma mensagem ao usuário pedindo confirmação dos dados.

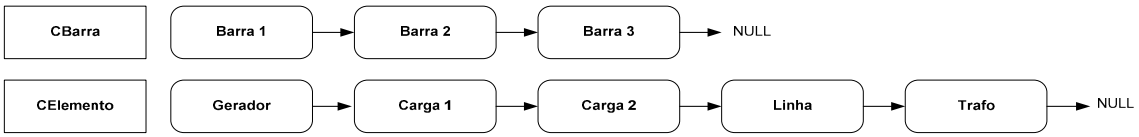

Figura 3.18 - Listas de objetos

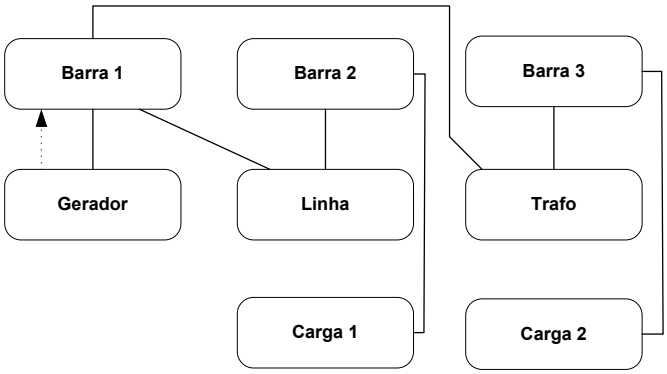

Figura 3.19 - Estrutura topológica

Percebe-se que na classe CRede e nas classes agregadas não existem referências aos aplicativos. Os métodos matemáticos relativos a cada aplicação são acessados através de ponteiros. Assim sendo, um componente é instanciado uma única vez sendo a modelagem acessada pelos aplicativos através de parâmetros de funções.

3.5 Metodologias Implementadas

Até a presente seção foi apresentada a estrutura de classes relativa à montagem e ao armazenamento de uma rede elétrica. Esta estrutura deve atender a qualquer aplicativo sem que nenhuma mudança estrutural no código seja realizada.

No presente trabalho incorporou-se a metodologia de Fluxo de Potência Trifásico em Coordenadas Polares(FPTP). A Figura 3.20 mostra a estrutura de classe após a incorporação do aplicativos. As classes em cinza são metodologias que podem ser implementadas na plataforma sem prejudicar a estrutura criada.

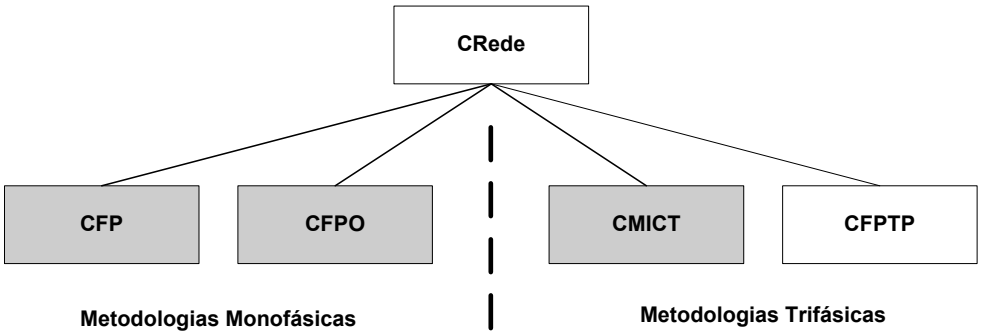

Figura 3.20 - Metodologias implementadas

Como pode ser visto, as classes relativas aos métodos são agregadas à classe CRede. Dessa forma, qualquer aplicativo pode acessar uma rede obtendo a modelagem matemática pertinente ao mesmo.

É importante ressaltar que qualquer aplicativo poderá interagir com a classe CRede através da inclusão/retirada de equipamentos, mudança de estado dos equipamentos de controle, etc. Com isso, cada modificação realizada por um aplicativo é novamente validada e disponibilizada imediatamente para os demais métodos.

3.5.1 Classe CFPT

Na classe CFPT (Figura 3.21) o fluxo de potência trifásico é resolvido utilizando a formulação polar trifásica como mostrado no Capítulo 2.

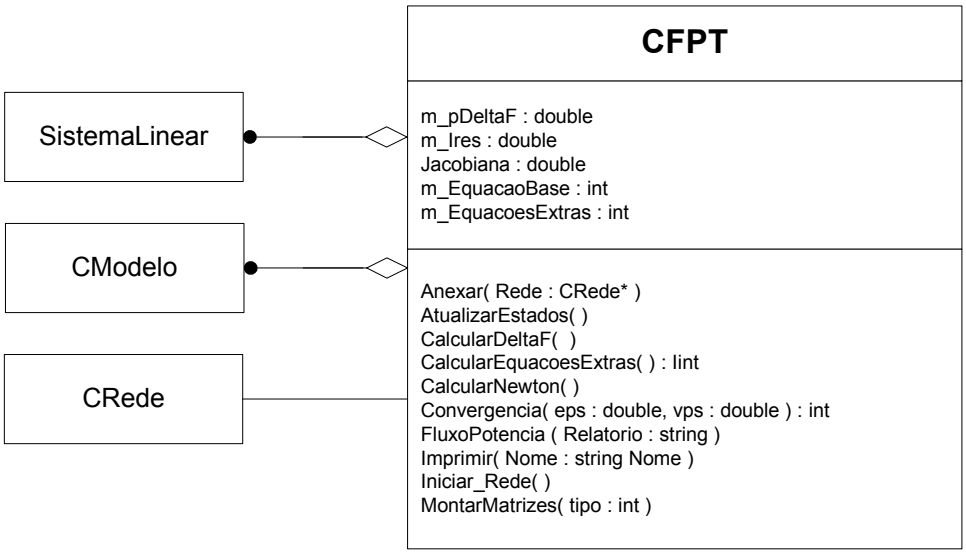

Figura 3.21 - Classe CFPT

Os dados membros das classes CFPTP podem ser vistos na Tabela 3.23. Da mesma forma, as funções membros podem ser vistas na Tabela 3.24

Tabela 3.23 – Dados membros da classe CFPTP

Dados Membros	Descrição
Matriz<double>* m_pDeltaF	Armazena o vetor independente, em forma blocada.
Matriz<double>* m_pIres	Auxiliar no cálculo da geração de potência das máquinas.
CSistemaLinear Jacobiana	Armazena a matriz Jacobiana.
int* m_EquacaoBase	Apontador para as barras de folga.
int m_nEquacaoExtras	Equações extras devido a controles.

Tabela 3.24 – Funções membros da classe CFPTP

Dados Membros	Descrição
void AnexarRede(CRede* rede)	Determina para qual rede será calculado o fluxo de potência.
void AtualizarEstados()	Atualiza, se necessário, o estado das variáveis.
void CalcularDeltaF()	Monta o vetor independente.
int CalcularEquacoesExtras()	Calcula o número de equações de controles, para dimensionar o sistema linear.
void CalcularNewton()	Calcula a solução do sistema linear utilizando fatoração LDU.
int Convergência(double eps, double vps)	Testa se os valores do vetor independente estão dentro da tolerância especificada.
int FluxoPotência(string Relatorio)	Executa o fluxo de potência e escreve os resultados em um arquivo em disco.
int Imprimir(string Relatorio)	Imprime o relatório do fluxo de potência convergido.
void IniciarRede()	Transforma, se necessário, os dados para pu e testa a rede.
void MontarMatrizes(int tipo)	Para o estudo de fluxo de potência, monta a matriz Jacobiana.

3.6 Modelos de Componentes

Os modelos matemáticos dos componentes para cada metodologia de análise dos sistemas elétricos são armazenados a parte. Sendo que todos derivam de apenas uma

única classe base como por ser observado na Figura 3.22, o motivo de todos derivarem de uma classe base é a padronização de chamadas.

Para acessar um modelo, deve utilizar a função PegarModelo(Ccomponente *pCmp, int estrutura, Ctransferencia<Matriz>& Bloco) da classe de modelos requerida.

O primeiro parâmetro o elemento que será modelado, o segundo o tipo do modelo (Hessiana, Jacobiana, etc.) e terceiro os dados serão armazenados.

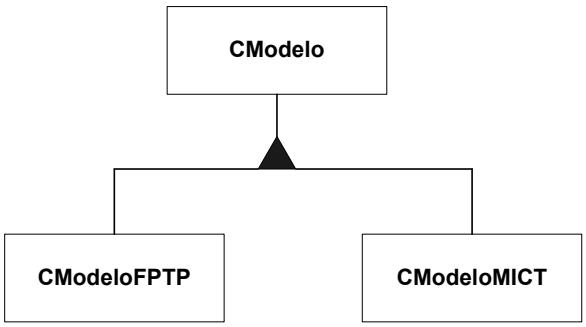

Figura 3.22 - Diagramas de modelos

As classes de modelos são muito parecidas, o que muda são as formulações matemáticas. Na Figura 3.23 é apresentado o diagrama genérico da classe.

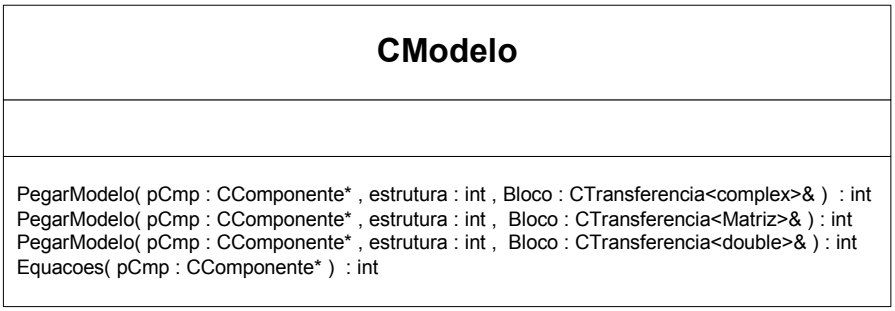

Figura 3.23 - Classe de CModelo

3.7 Estrutura de Classes da Interface Gráfica

Para a representação gráfica dos componentes da rede elétrica é necessária a criação de uma plataforma para representação desses elementos. Neste tópico será apresentada essa estrutura que é baseada nas estruturas desenvolvidas nos itens anteriores desse capítulo.

A interface gráfica foi concebida utilizando-se a tecnologia da biblioteca gráfica OpenGL (NEIDER e DAVIS,1996, WRIGHT e SWEET,1996) o que possibilitou grande robustez gráfica da aplicação.

3.7.1 Classes Auxiliares

São as classes que não pertencem a modelagem gráfica da rede elétrica. Tem como objetivo auxiliar a interface a referenciar os pontos no programa e escrever textos pertinentes a aplicação.

3.7.1.1 Classe CGLPonto

Esta classe tem como objetivo gerenciar e armazenar a posição, nas três dimensões, dos vários elementos pertencentes ao sistema elétrico. Na aplicação desenvolvida, utiliza-se somente as coordenadas x e y, pois ainda não desenha nenhum gráfico ou equipamento em três dimensões Todos os componentes da rede possuem um objeto desta classe agregado. A Figura 3.24 mostra o diagrama da classe CGLPonto e a Tabela 3.25 mostra os dados membros.

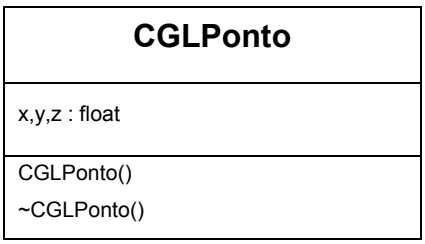

Figura 3.24– Classe CGLPonto

Tabela 3.25 – Dados membros de CGLPonto

Dados Membros	Descrição
float x,y,z	Posição dos elementos nas três dimensões.

Tabela 3.26 – Funções membros de CGLPonto

Funções Membros	Descrição
CGLPonto()	Construtor da classe CGLPonto
~CGLPonto()	Destrutor da classe CGLPonto.

3.7.1.2 Classe CGLTexto

Esta classe é responsável por escrever qualquer tipo de informação de texto na interface do programa, tais como nome de barra, corrente em linha, tap de transformador, etc.

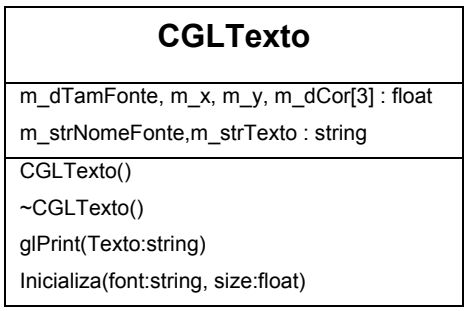

Figura 3.25– Classe CGLPonto

A Figura 3.25 mostra o diagrama da classe apresentada, a Tabela 3.27 os dados membros e a Tabela 3.28 as funções membros

Tabela 3.27 – Dados membros de CGLPonto

Dados Membros	Descrição
float m_dTamFonte	Armazena o tamanho da fonte
float m_strNomeFonte	Armazena o nome da fonte
CString m_strTexto	Armazena o texto a ser exibido na interface
float m_x, m_y	Armazena a posição x e y do texto
float m_dCor[3]	Vetor responsável por armazenar as 3 componentes de cor do texto (Vermelho, Verde, Azul)

Tabela 3.28 – Funções membros de CGLPonto

Funções Membros	Descrição
CGLTexto()	Construtor da classe CGLTexto
~CGLTexto()	Destrutor da classe CGLTexto.
void glPrint(CString texto)	Escreve o texto definido no parâmetro na tela.
void Inicializa(CString font, float size)	Inicializa as variáveis da classe com os parâmetros da função

3.7.2 Modelagem dos Componentes Gráficos do Sistema Elétrico

Da mesma forma que os componentes da rede anteriormente desenvolvida no item 3.3, a modelagem dos equipamentos deve observar o comportamento físico e a conectividade. Além disso, esses componentes devem promover conexão com a rede e

com os elementos previamente desenvolvidos. A estrutura apresenta três níveis hierárquicos com funções e objetivos bem definidos como pode ser visto na Figura 3.26.

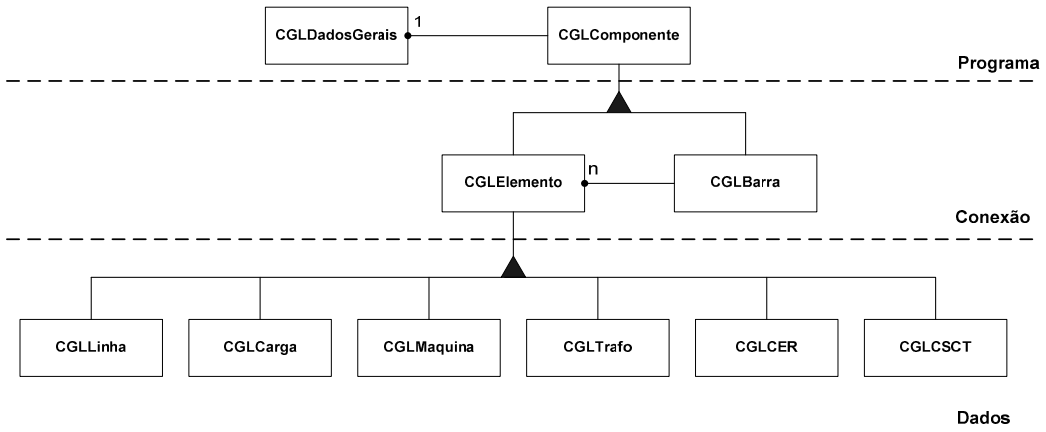

Figura 3.26 – Estrutura de classes da Interface

3.7.2.1 Nível Programa

No primeiro nível encontra-se a classe *CGLComponente*. Esta classe é base para todas as demais classes dos elementos da rede e possui dados comuns a todos eles. Esta é uma classe abstrata definida ao nível de programa e tem como único objetivo facilitar a implementação computacional através dos mecanismos de herança e funções virtuais. Agregada a esta classe, a *CGLDadosGerais* é responsável pelo armazenamento de dados importantes que devem ser acessados por todos os componentes, como por exemplo a cor padrão das barras.

3.7.2.1.1 Classe CGLDadosGerais

A classe *CGLDadosGerais* é responsável por armazenar dados e funções que devem estar disponíveis a todos os equipamentos da rede, porém não pertencem a nenhum componente da rede, como por exemplo nível de zoom, tamanho da tela, ângulo de rotação da tela, etc.

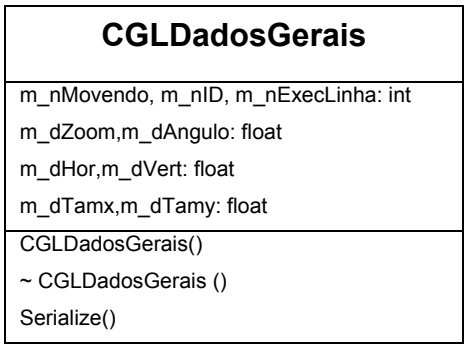

Figura 3.27– Classe CGLDadosGerais

Tabela 3.29 – Dados membros de CGLDadosGerais

Dados Membros	Descrição
int m_nMovendo	Indica se existe algum elemento sendo movido
int m_nID	Armazena o ID do elemento selecionado
int m_nExecLinha	Indica se existe algum elemento série sendo criado
float m_dZoom	Armazena o zoom que é exibido na interface
float m_dAngulo	Armazena o ângulo de rotação da tela
float m_dHort, m_dVert	Armazena o deslocamento horizontal e vertical da tela
float m_dTamx, m_dTamy	Armazena o tamanho da tela na horizontal e na vertical

Tabela 3.30 – Funções membros de CGLDadosGerais

Funções Membros	Descrição
CGLDadosGerais()	Construtor da classe CGLDadosGerais
~ CGLDadosGerais ()	Destrutor da classe CGLDadosGerais.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.

A Figura 3.27 mostra o diagrama da classe apresentada, a Tabela 3.29 os dados membros e a Tabela 3.30 as funções membros.

3.7.2.1.2 Classe CGLComponente

Esta classe é base para todas as demais classes dos elementos da rede. Além disso, ela é responsável por conectar cada elemento da rede desenvolvida para o cálculo de

fluxo de potência com os elementos da classe gráfica. O modelo gráfico de acordo com a UML para a representação da classe é mostrado na Figura 3.28

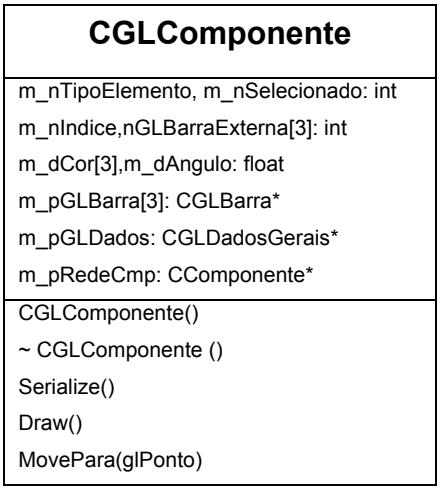

Figura 3.28– Classe CGLComponente

A Tabela 3.31 apresenta os dados membros da classe CGLDadosGerais e a Tabela 3.32 as funções membro.

Tabela 3.31 – Dados membros de CGLDadosGerais

Dados Membros	Descrição
int m_nTipoElemento	Armazena o tipo de elemento
int , m_nSelecionado	Indica se o elemento está selecionado
int m_nIndice	Armazena o índice do elemento que é usado para identificá-lo para a OpenGL
int nGLBarraExterna[3]	Armazena os números das barras conectadas a um elemento
float m_dCor[3]	Armazena as componentes de cor de um elemento
float m_dAngulo	Armazena a rotação de um elemento
CGLBarra* m_pGLBarra[3]	Armazena os ponteiros para as barras conectadas a um elemento
CGLDadosGerais* m_pGLDados	Armazena o ponteiro para a classe CGLDadosGerais
CComponente* m_pRedeCmp	Armazena o ponteiro para a classe CComponente que é responsável pela conexão da classe gráfica a classe numérica

Tabela 3.32 – Funções membros de CGLDadosGerais

Funções Membros	Descrição
CGLComponente()	Construtor da classe CGLComponente
~ CGLComponente ()	Destrutor da classe CGLComponente.
void Draw()	Desenha o elemento na interface

void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
Void MovePara(glPonto ponto)	Move o elemento para a posição indicada no parâmetro *ponto*.

3.7.2.2 Nível Conexão

No segundo nível encontram-se as classes *CGLElemento* e *CGLBarra,* que são classes derivadas diretamente da classe *CGLComponente*. Estas classes foram criadas para definir as associações entre os componentes de uma rede elétrica, ou seja, determina como os objetos poderão se relacionar.

3.7.2.2.1 Classe CGLElemento

A classe *CGLElemento* apresenta de forma genérica as conexões com os diversos equipamentos de um sistema elétrico, independente do equipamento ou do numero de conexões. O modelo desta classe segundo a UML pode ser visto na Figura 3.29.

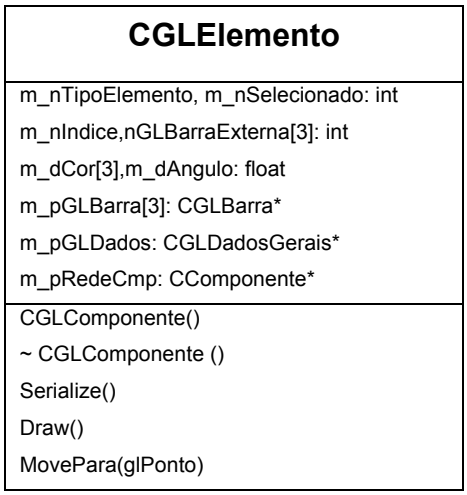

Figura 3.29– Classe CGLElemento

A Tabela 3.33 apresenta os dados membros da classe CGLElemento e a Tabela 3.34 as funções membro.

Tabela 3.33 – Dados membros de CGLElemento

Dados Membros	Descrição
CGLBarra* m_pControle	Aponta para barra no qual o elemento controla alguma grandeza

Tabela 3.34 – Funções membros de CGLElemento

Funções Membros	Descrição
CGLElemento()	Construtor da classe CGLComponente
~ CGLComponente ()	Destrutor da classe CGLComponente.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento na tela
CGLBarra* PegarApontadorBarra(int bar)	Retorna o ponteiro para a barra cujo número é especificado pelo parâmetro bar

3.7.2.2.2 Classe CGLBarra

A classe *CGLBarra* apresenta os barramentos de forma gráfica os barramentos onde os diversos equipamentos de um sistema elétrico (cargas, linhas, geradores) devem se conectar . O modelo desta classe segundo a UML pode ser visto na Figura 3.30.

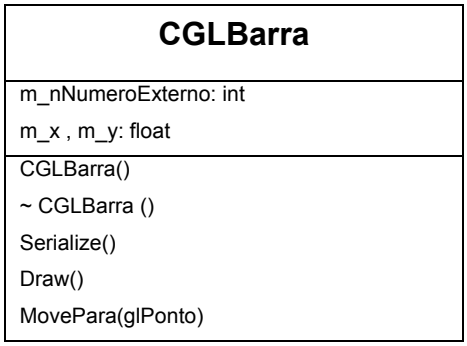

Figura 3.30– Classe CGLBarra

A Tabela 3.35 apresenta os dados membros da classe CGLElemento e a Tabela 3.36 as funções membro.

Tabela 3.35 – Dados membros de CGLBarra

Dados Membros	Descrição
int m_nNumeroExterno	Armazena o número da barra
float m_x	Armazena a posição x do elemento barra na tela

float m_y	Armazena a posição y do elemento barra na tela

Tabela 3.36 – Funções membros de CGLBarra

Funções Membros	Descrição
CGLBarra()	Construtor da classe CGLBarra
~ CGLBarra ()	Destrutor da classe CGLBarra.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento barra para uma determinada posição especificada pelo parâmetro glPonto

3.7.2.3 Nível Dados

As classes relativas aos elementos propriamente ditos (CGLLinha, CGLCarga, CGLTrafo2, etc.) encontram-se no terceiro nível. O objetivo destas classes é ler, escrever, armazenar os dados e desenhar os elementos na interface gráfica. Nenhuma metodologia ou dado elétrico são manipulados por estas classes.

Vale resaltar que todas as classes desse nível não possuem dados membro, pois toda informação necessária a essas entidades são providas pelas classes das quais elas foram derivadas.

3.7.2.3.1 CGLLinha

Esta classe armazena todos os dados gráficos relevantes de uma linha de transmissão CA. O diagrama é apresentado na Figura 3.31.

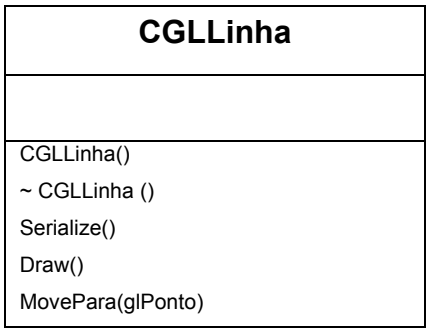

Figura 3.31 - Classe CGLLinha

Uma detalhada descrição das funções membros é apresentada na Tabela 3.37.

Tabela 3.37 – Funções membros de CGLLinha

Funções Membros	Descrição
CGLLinha()	Construtor da classe CGLLinha
~ CGLLinha ()	Destrutor da classe CGLLinha.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento linha para uma determinada posição especificada pelo parâmetro glPonto

3.7.2.3.2 CGLCarga

Esta classe armazena todos os dados gráficos relevantes as cargas do sistema. O diagrama é apresentado na Figura 3.32.

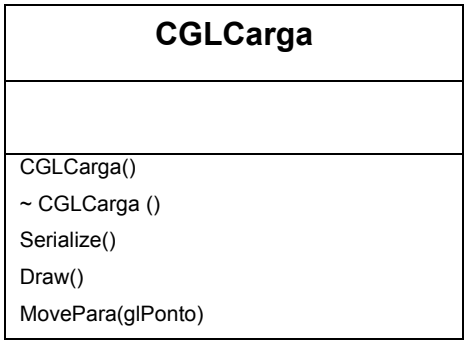

Figura 3.32 - Classe CGLCarga

Uma detalhada descrição das funções membros é apresentada na Tabela 3.38.

Tabela 3.38 – Funções membros de CGLCarga

Funções Membros	Descrição
CGLCarga()	Construtor da classe CGLCarga
~ CGLCarga ()	Destrutor da classe CGLCarga.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento carga para uma determinada posição especificada pelo parâmetro glPonto

3.7.2.3.3 CGLMaquina

Esta classe armazena todos os dados gráficos relevantes as máquinas do sistema. O diagrama é apresentado na Figura 3.33.

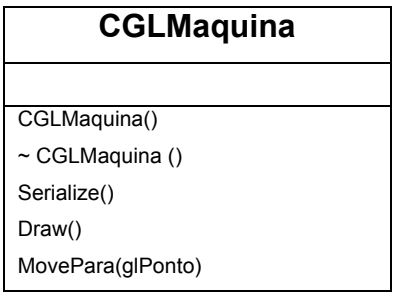

Figura 3.33 - Classe CGLMaquina

Uma detalhada descrição das funções membros é apresentada na Tabela 3.39.

Tabela 3.39 – Funções membros de CGLMaquina

Funções Membros	Descrição
CGLMaquina()	Construtor da classe CGLMaquina
~ CGLMaquina ()	Destrutor da classe CGLMaquina.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento máquina para uma posição especificada pelo parâmetro glPonto

3.7.2.3.4 CGLTrafo

Esta classe armazena todos os dados gráficos relevantes aos transformadores do sistema. O diagrama é apresentado na Figura 3.34.

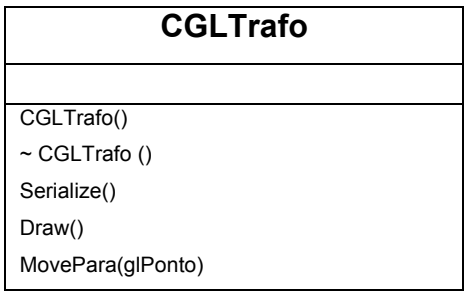

Figura 3.34 - Classe CGLTrafo

Uma detalhada descrição das funções membros é apresentada na Tabela 3.40.

Tabela 3.40 – Funções membros de CGLTrafo

Funções Membros	Descrição
CGLTrafo()	Construtor da classe CGLTrafo
~ CGLTrafo ()	Destrutor da classe CGLTrafo.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento transformador para uma determinada posição especificada pelo parâmetro glPonto

3.7.2.3.5 CGLCER

Esta classe armazena todos os dados gráficos relevantes aos Compensadores Estáticos de Reativos do sistema. O diagrama é apresentado na Figura 3.35.

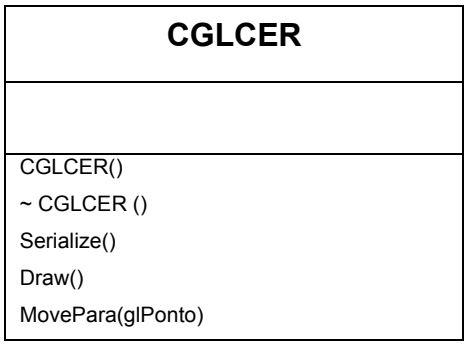

Figura 3.35 - Classe CGLCER

Uma detalhada descrição das funções membros é apresentada na Tabela 3.41.

Tabela 3.41 – Funções membros de CGLCER

Funções Membros	Descrição
CGLCER()	Construtor da classe CGLCER
~ CGLCER ()	Destrutor da classe CGLCER.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento compensador para uma posição especificada pelo parâmetro glPonto

3.7.2.3.6 CGLCSCT

Esta classe armazena todos os dados gráficos relevantes a Compensação Série Controlada à Tiristores. O diagrama é apresentado na Figura 3.36.

Uma detalhada descrição das funções membros é apresentada na Tabela 3.42

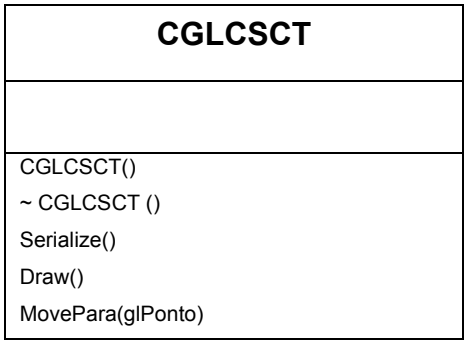

Figura 3.36 - Classe CGLCSCT

Tabela 3.42 – Funções membros de CGLCSCT

Funções Membros	Descrição
CGLCSCT()	Construtor da classe CGCSCT
~ CGLCSCT ()	Destrutor da classe CGLCSCT.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Draw()	Desenha o elemento barra na tela
void MovePara(glPonto)	Move o elemento compensação série para uma determinada posição especificada pelo parâmetro glPonto

3.7.3 Modelagem da Rede Gráfica

Após a modelagem dos objetos da rede, fez necessário criar uma classe para o gerenciamento, montagem, conexão e manutenção dos elementos gráficos rede. Foi então criada a classe *CGLRede* que possui funções de desenho e manipulação de coordenadas de tela, mouse, etc. As classes já criadas dos componentes ficam agregadas a classe *CGLRede* como mostra a Figura 3.37

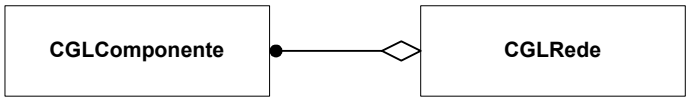

Figura 3.37 - Agregação da classe CGLRede

O diagrama de CGLRede é apresentado na Figura 3.38 e os dados nas Tabela 3.43 e Tabela 3.44.

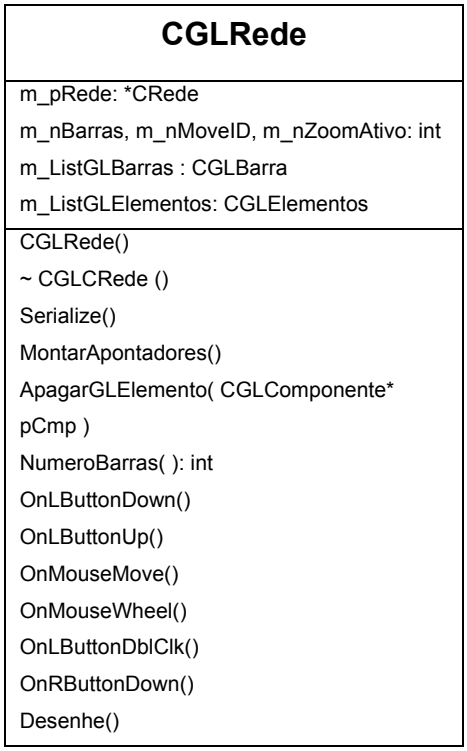

Figura 3.38– Classe CGLRede

Tabela 3.43 – Dados membros de CGLRede

Dados Membros	Descrição
CRede* m_pRede	Aponta para a classe CRede
int m_nBarras	Armazena o número de barras
int m_nMoveID	Armazena o ID do objeto que está sendo movimentado
int m_nZoomAtivo	Indica se o zoom está ativo
CGLBarra m_ListGLBarras	Lista encadeada das barras gráficas do sistema
CGLElementos m_ListGLElementos	Lista encadeada dos elementos gráficos do sistema

Tabela 3.44 – Funções membros de CGLRede

Funções Membros	Descrição
CGLRede()	Construtor da classe CGLRede
~ CGLRede ()	Destrutor da classe CGLRede.
void Serialize()	São as funções de salvar e ler arquivos utilizando mecanismo do C++.
void Desenha()	Desenha todos os elementos na tela do programa
void MontarApontadores()	Carrega os ponteiros dos elementos, fazendo as conexões entre eles
void ApagarGLElemento(CGLComponente* pCmp)	Apaga o elemento especificado pelo parâmetro pCmp
int NumeroBarras()	Retorna o número de barras do sistema
void OnLButtonDown()	Função chamada quando botão esquerdo do mouse é pressionado
void OnLButtonUp()	Função chamada quando botão esquerdo do mouse é liberado
void OnMouseMove()	Função chamada quando o mouse é movimentado
void OnMouseWheel()	Função chamada quando o *"scroll"* mouse é movimentado
void OnLButtonDblClk()	Função chamada quando botão esquerdo do mouse é pressionado duplamente
void OnRButtonDown()	Função chamada quando botão direito do mouse é pressionado

3.8 Estrutura de Classes Completa da Aplicação

Após a modelagem das classes para representação dos elementos da rede, resolução de fluxo de potência e representação gráfica, a estrutura final de classes da aplicação desenvolvida é dada na Figura 3.39

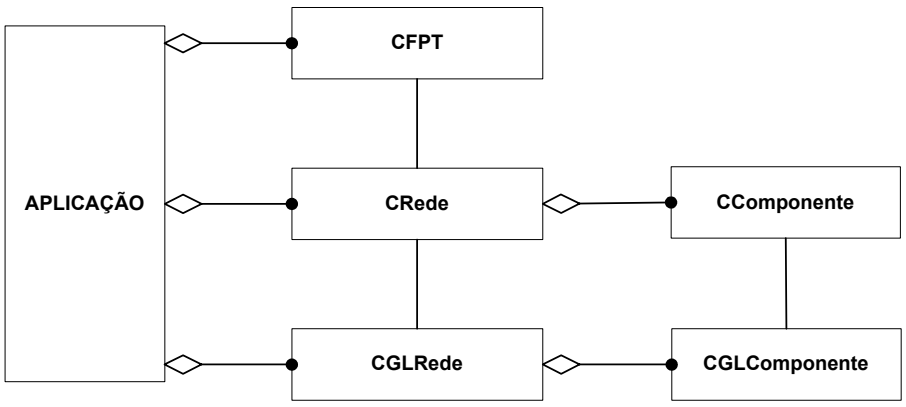

Figura 3.39 - Estrutura completa de classes

Como pode ser visto, a estrutura é bastante modular, pois cada módulo (CGLRede, CRede, CFPT) foi desenvolvido em um bloco separadamente. Esta modularidade permite que vários grupos trabalhem na mesma aplicação independentemente, permitindo um ganho de tempo no desenvolvimento do aplicativo. Além disso, a estrutura permite a inclusão de novas metodologias de análise de rede (Fluxo de Potência Harmônico, Estabilidade Transitória, Cálculo de Perdas, etc) sem que seja necessário modificar a base desenvolvida anteriormente. Para tanto basta agregarmos na aplicação as novas metodologias e comunicá-las com a classe *CRede*, como pode ser visto na Figura 3.40 .

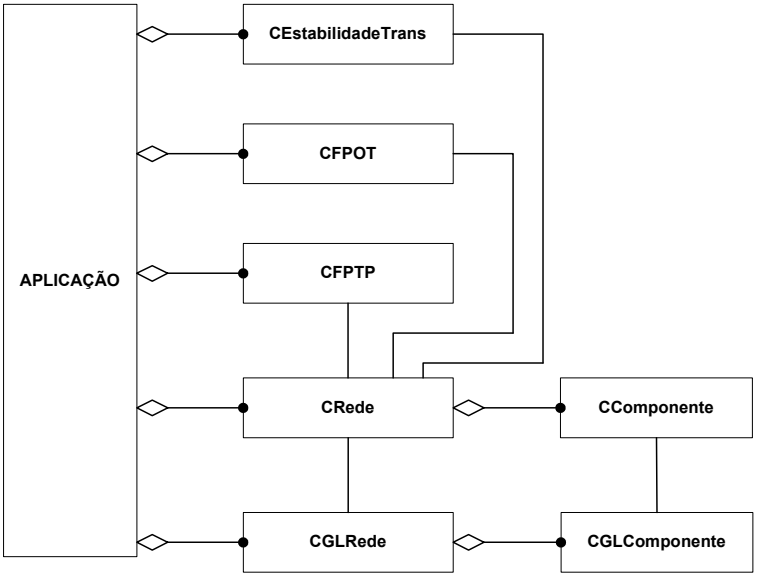

Figura 3.40 - Estrutura de classes com novas metodologias agregadas

3.9 Exemplo da Estrutura de Classes

Na Figura 3.41 é apresentado um sistema de três barras a fim de demonstrar a estrutura de classes proposta nesse capítulo.

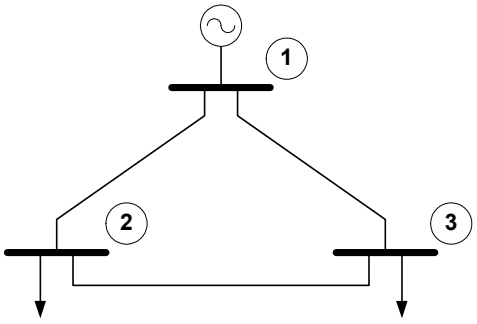

Figura 3.41-Sistema Exemplo 3 barras

Seguindo as premissas definidas no capítulo, monta-se a estrutura de classes mostrada na Figura 3.42. Na estrutura mostra com clareza a ligação entre os diversos componentes do sistema elétrico (Ex: linha 1-2 conectada as barra1 e barra2)

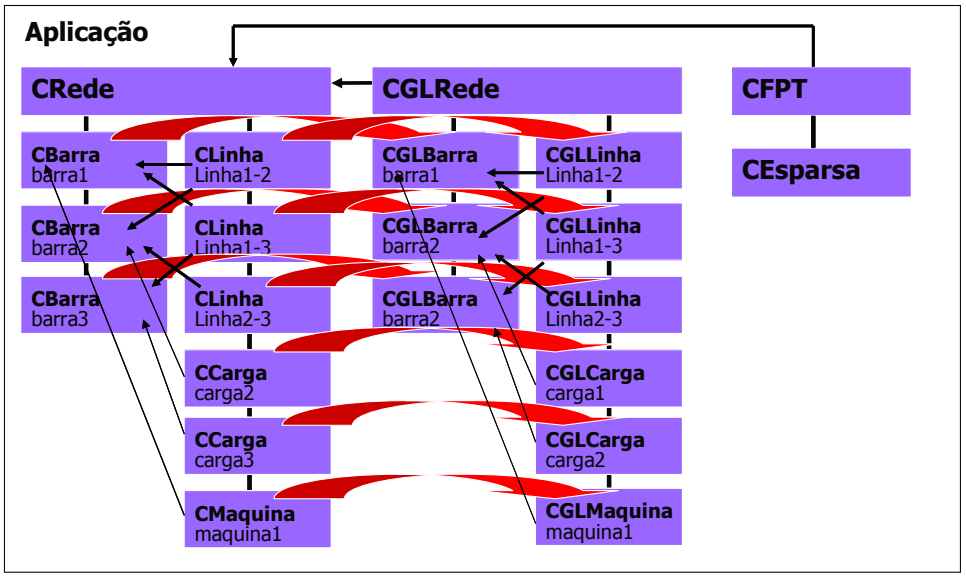

Figura 3.42-Estrutura de Classes do Exemplo

Como pode ser visto na Figura 3.42 a classe da interface gráfica (CGLRede) não interfere no fluxo de dados entre as classes da rede (CRede) e a classe de cálculo (CFPT). Devido a essa modularidade das classes, a aplicação final melhora seu desempenho computacional.

3.10 Implementação da Aplicação Computacional

A partir da estrutura de classes apresentada anteriormente, a aplicação proposta neste trabalho foi concebida utilizando o Microsoft Visual C++ 6.0 (HORTON, 1998), uma poderosa ferramenta no desenvolvimento de aplicações orientadas a objeto de grande robustez. O sistema operacional escolhido foi o Microsoft Windows por ser o sistema mais usual nos computadores pessoais.

Num primeiro momento, foi desenvolvida somente a aplicação numérica, sem a interface gráfica, para a validação da metodologia matemática proposta no Capítulo 2. Esta aplicação funcionou sem problemas, externando os resultados esperados para diversos sistemas testados.

Para a implementação da interface gráfica foram pesquisadas diversas tecnologias (GDI, DIRECTX, etc). Após vários testes, concluiu-se que o OpenGL (NEIDER e DAVIS,1996, WRIGHT e SWEET,1996) seria a linguagem gráfica mais viável para o desenvolvimento do projeto. Podemos destacar entre suas vantagens:

- Grande desempenho gráfico;
- Fácil implementação;
- É de domínio público;
- Supera todos os problemas apresentados pela GDI.

Maiores detalhes sobre essa tecnologia gráfica pode ser visto no Apêndice C.

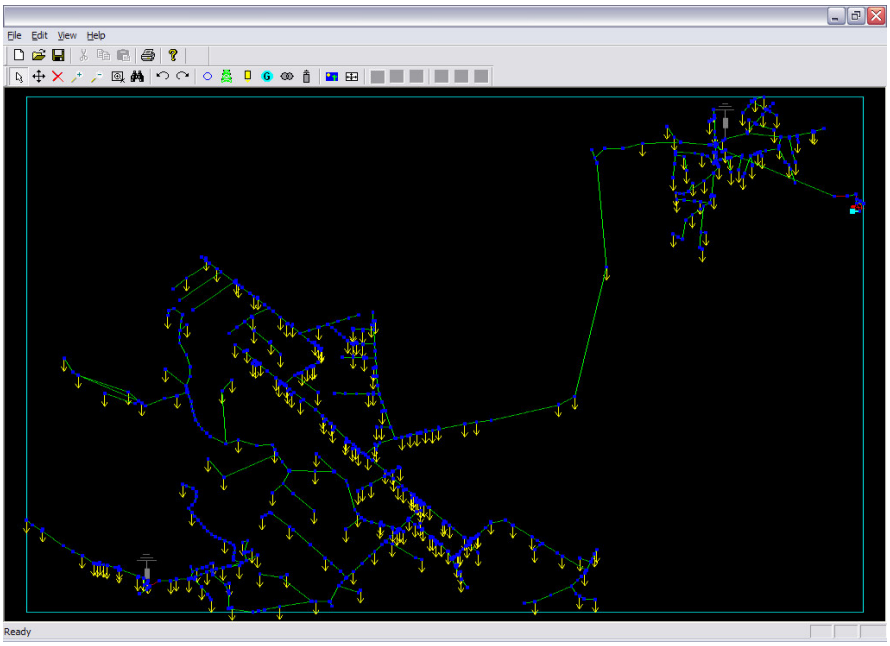

Figura 3.43 - Aplicação desenvolvida utilizando Visual C++ e OpenGL

A aplicação desenvolvida se apresenta na Figura 3.43. Ela foi concebida baseando-se nos consagrados programas de simulação de circuitos PSPICE e ATP, a fim de facilitar sua utilização e tornar sua interface mais amigável.

3.11 - Sumário do Capítulo

Neste capítulo foram apresentadas em detalhes todo o desenvolvimento das classes que compõe a aplicação, tanto da parte de cálculo como da interface gráfica.

A plataforma proposta apresentou como grande vantagem, a criação de um ambiente integrado para análise de sistemas elétricos onde novos métodos e modelos de componentes podem ser facilmente incorporados ou modificados. Adicionalmente, a possibilidade de interação entre diversos aplicativos e a base de dados e a fácil visualização e interpretação dos resultados também podem ser citadas como grandes vantagens do modelo proposto.

A base computacional e de objetos é a mesma para qualquer metodologia. Para utilizar uma ou outra basta atribuir valor a uma variável que é utilizada para dimensionar os vetores, escolher as funções de ler/escrever dados, etc. Logicamente o programador deverá escrever os modelos necessários, mas a escolha destes também será automática.

Vale resaltar que devido a modularidade das classes da interface gráfica, as mesmas foram utilizadas em outras aplicações desenvolvidas pelo Grupo de Sistemas de Potência da Universidade Federal de Juiz de Fora. São elas:

- Programa para o cálculo de curto-circuito em sistemas de distribuição – Projeto de P&D desenvolvido com a empresa Light.
- Programa de alocação ótima de capacitores em sistemas de distribuição – Projeto de P&D desenvolvido com a empresa Ampla.

Capítulo 4 Resultados

4.1 Considerações Gerais

Neste capítulo são apresentados os resultados obtidos na implementação da metodologia de Fluxo de Potência (Fluxo de Potência Trifásico Utilizando Coordenadas Polares) juntamente com a interface gráfica que foi desenvolvida utilizando os preceitos da orientação a objetos e da biblioteca gráfica OpenGL.

Os testes realizados tem por objetivo mostrar as características e funcionalidades do método proposto no que diz respeito ao estudo dos diversos aspectos dos sistemas trifásicos que seriam impossíveis de se analisar utilizando apenas ferramentas monofásicas.

Os sistemas testados e suas particularidades estão apresentados na Tabela 4.1. Seus dados e variantes podem ser encontrados nas referencias citadas.

Tabela 4.1 - Referência para os casos estudados

Sistema	Barras	Referência
IEEE – 4	4	KERSTING (2000)
IEEE – 13	13	KERSTING (2000)
IEEE – 14	14	UWEE (2005)
IEEE – 37	37	KERSTING (2000)

Além dos sistemas acima citados, foi testado um caso real de uma concessionária de energia elétrica. Esse sistema tem como objetivo mostrar a capacidade da aplicação em simular sistemas de grande porte.

4.2 Caso 1 – IEEE 4 barras

O primeiro caso estudado foi o IEEE 4 barras. O unifilar deste sistema se encontra na Figura 4.1.

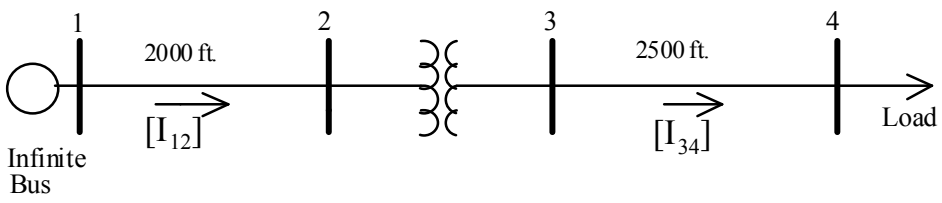

Figura 4.1 - Diagrama unifilar do sistema IEEE4

O principal propósito deste sistema é simular a metodologia apresentada em sistemas com linhas e cargas desequilibradas, bem como o modelo proposto de transformador. A Figura 4.2 apresenta o sistema IEEE4 montado na aplicação desenvolvida neste presente trabalho.

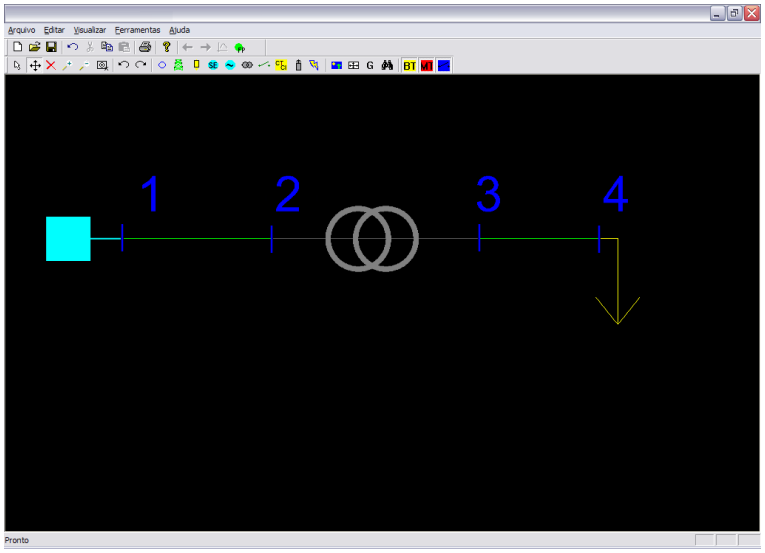

Figura 4.2 - Sistema IEEE4 na aplicação desenvolvida

A tensão base das barras 1 e 2 são 7,2 kV *fase-terra*, e das barras 3 e 4 é 2,4 kV *fase-terra*. O transformador considerado é um transformador Yaterrado-Yaterrado abaixador. Os resultados da simulação podem ser vistos na Tabela 4.2.

Tabela 4.2 - Resultados do caso IEEE4

Barra	Va (kV)	∠a (graus)	Vb(kV)	∠b (graus)	Vc(kV)	∠c (graus)
1	7200	0	7200	-120	7200	120
2	7115.76	-0.3135	7136.64	-120.3417	7125.84	119.5671
3	2253.36	-3.5772	2265.36	-123.4558	2259.6	116.3235
4	1940.16	-9.0046	2068.56	-128.4123	1998.48	110.6656

Estes resultados apresentados com a metodologia proposta são iguais aos obtidos em KERSTING (2000) e aos obtidos utilizando-se o MICT (Garcia, 2001), o que permite validar a metodologia e a implementação realizadas no âmbito deste trabalho.

4.3 Caso 2 – IEEE 13 barras

O diagrama unifilar do sistema IEEE13 encontra-se ilustrado na Figura 4.3. Este é um sistema pequeno e altamente carregado. Possui cabos aéreos e subterrâneos, bancos de capacitores, diversos tipos de cargas, um regulador de tensão e ramais monofásicos e

bifásicos. A Figura 4.4 mostra o diagrama unifilar do sistema IEEE13 barras montado na plataforma de análise de redes desenvolvida.

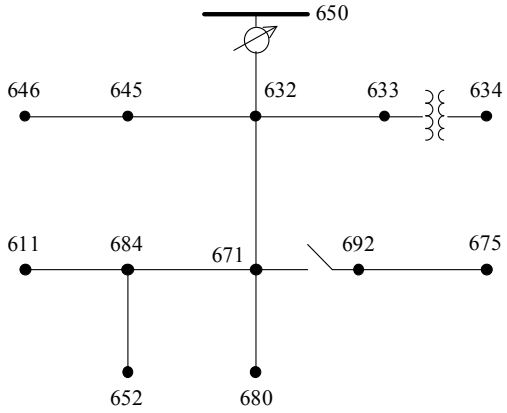

Figura 4.3 - Diagrama unifilar do sistema IEEE13

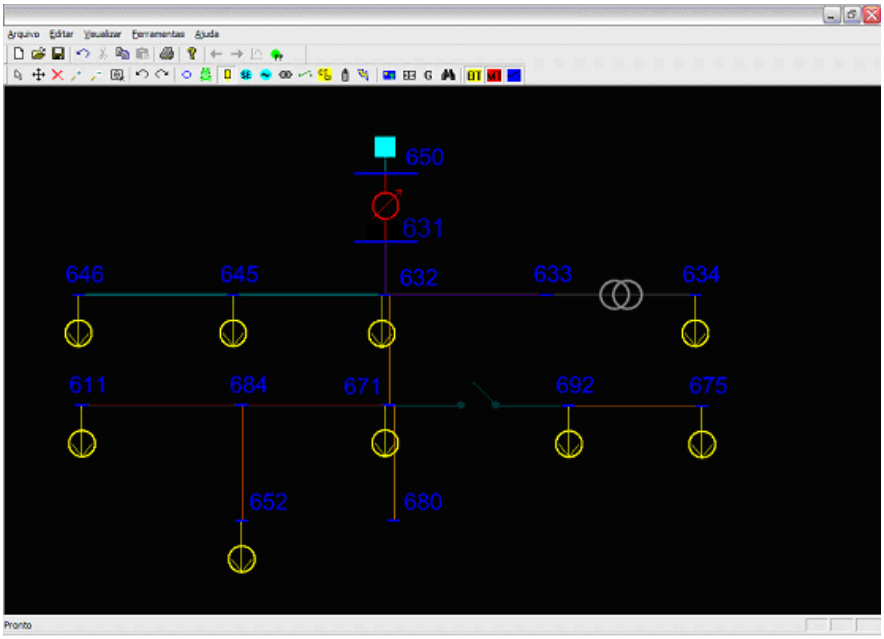

Figura 4.4 - Sistema IEEE13 na aplicação desenvolvida

A tensão base para o sistema é 2,4 kV para todas barras, exceto para barra 637 que é de 277 V. O regulador de tensão foi configurado para controlar a tensão na barra 631 em Va = 2.5511 kV, Vb = 2.5211 kV e Vc = 2,5660 kV

Tabela 4.3 - Resultados do caso IEEE13

Barra	Va (kV)	∠a (graus)	Vb(kV)	∠b (graus)	Vc(kV)	∠c (graus)
650	2.401	0	2.401	-120	2.401	120
631	2.5511	0.0064	2.5211	120.0051	2.5660	119.9932
632	2.4515	-2.4964	2.5019	-121.726	2.4428	117.8213
645	2.4515	-2.4964	2.4799	-121.906	2.4381	117.8485
646	2.4515	-2.4964	2.4757	-121.981	2.4332	117.8938

671	2.377	-5.3067	2.5281	-122.346	2.3474	116.0155
684	2.3724	-5.3296	2.5281	-122.346	2.3426	115.9142
652	2.3572	-5.2569	2.5281	-122.346	2.3426	115.9142
611	2.3724	-5.3296	2.5281	-122.346	2.3378	115.7683
692	2.377	-5.3073	2.5282	-122.346	2.3474	116.015
675	2.3614	-5.5572	2.5339	-122.522	2.3429	116.0296
680	2.377	-5.3067	2.5281	-122.346	2.3474	116.0155
633	2.4442	-2.561	2.4973	-121.771	2.4366	117.8172
634	0.2753	-3.2371	0.283	-122.227	0.2759	117.3376

Como no caso anterior os resultados obtidos com a metodologia proposta são iguais aos obtidos em KERSTING (2000) e aos obtidos utilizando-se o MICT (Garcia, 2001), o que permite validar a metodologia e a implementação realizadas no âmbito deste trabalho.

4.4 Caso 3 – IEEE 14 barras

O sistema teste IEEE14 é de pequeno porte e bastante difundido na literatura, por isto tornou-se um ótimo candidato para testar e validar resultados da metodologia proposta (trifásica). Neste caso utilizou-se uma variação trifásica do caso IEEE14 que pode ser encontrada em UWEE (2005).

Na Figura 4.5 apresenta-se o diagrama unifilar do caso. O caso possui quatro máquinas, três transformadores, sendo um deles de três enrolamentos e vinte linhas, todas trifásicas e equilibradas. A representação do sistema na aplicação desenvolvida pode ser vista na Figura 4.6.

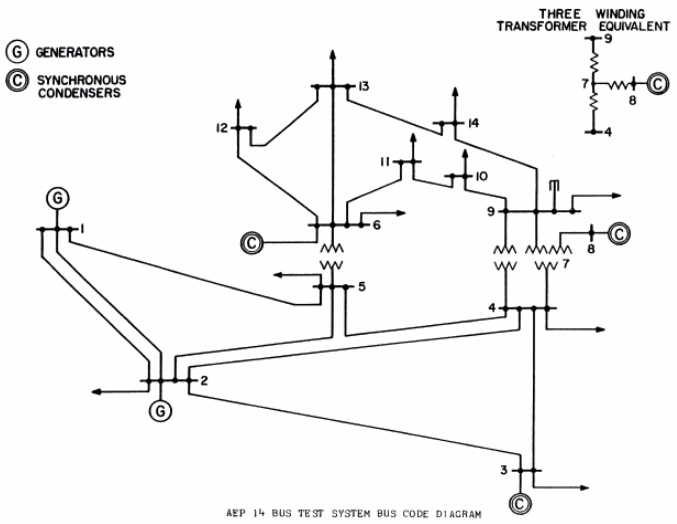

Figura 4.5 - Diagrama unifilar do sistema IEEE14

Neste sistema apresentado, todos os dados estão em pu, não necessitando transformação adicional para a utilização do método matemático.

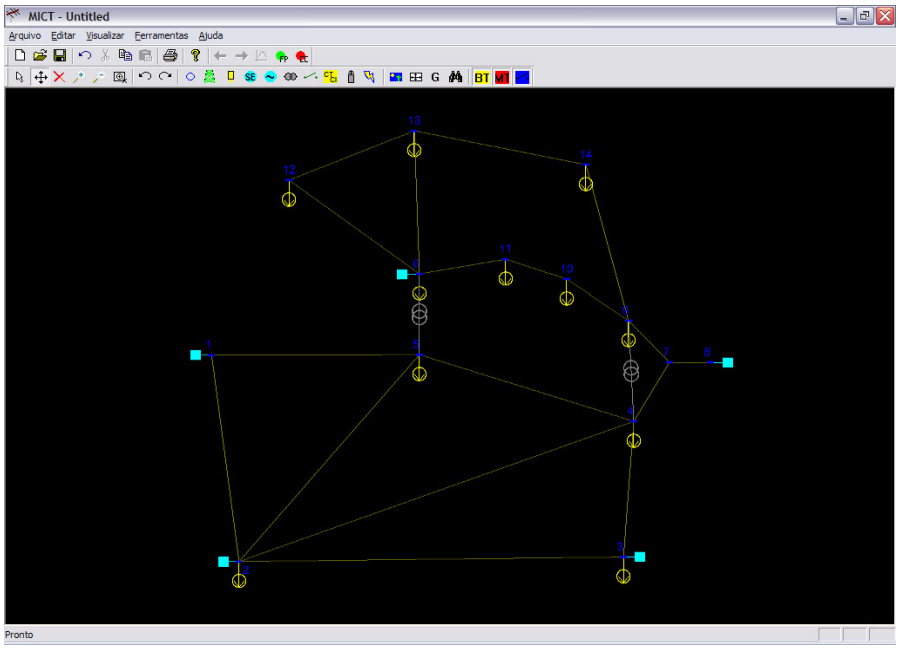

Figura 4.6 - Sistema IEEE14 na aplicação desenvolvida

Tabela 4.4 - Resultado do sistema IEEE14

Barra	Va (pu)	∠a (graus)	Vb(pu)	∠b (graus)	Vc(pu)	∠c (graus)
1	1.06	0	1.06	-120	1.06	120
2	1.045	-6.1318	1.045	-126.1318	1.045	113.8682
3	1.01	-13.6967	1.01	-133.6967	1.01	106.3033
4	1.0239	-11.2715	1.0239	-131.2715	1.0239	108.7285
5	1.0299	-9.7621	1.0299	-129.7621	1.0299	110.2379
6	1.07	-15.7459	1.07	-135.7459	1.07	104.2541
7	1.0438	-14.3469	1.0438	-134.3469	1.0438	105.6531
8	1.09	-14.3469	1.09	-134.3469	1.09	105.6531
9	1.0266	-15.9599	1.0266	-135.9599	1.0266	104.0401
10	1.0267	-16.2039	1.0267	-136.2039	1.0267	103.7961
11	1.0445	-16.0891	1.0445	-136.0891	1.0445	103.9109
12	1.0529	-16.5885	1.0529	-136.5885	1.0529	103.4115
13	1.0461	-16.6079	1.0461	-136.6079	1.0461	103.3921
14	1.0168	-17.2763	1.0168	-137.2763	1.0168	102.7237

Estes resultados apresentados com a metodologia proposta são iguais aos obtidos em UWEE (2005).e aos obtidos utilizando-se o MICT (Garcia, 2001)

4.5 Caso 4 – IEEE 37 barras

O sistema IEEE37, apresentado na Figura 4.7, é um sistema atípico, pois é totalmente conectado em delta, sendo todos os cabos subterrâneos. O sistema é bastante desequilibrado e não apresenta problemas de tensão. Segundo KERSTING (2000) algumas metodologias podem encontrar problemas para tratar sistemas deste tipo. A metodologia proposta mostrou-se robusta e eficiente, não apresentando nenhum problema de convergência.

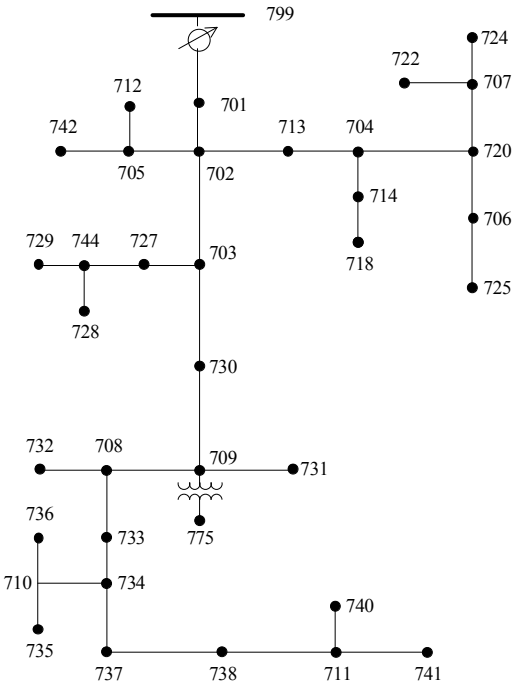

Figura 4.7 - Diagrama unifilar do sistema IEEE37

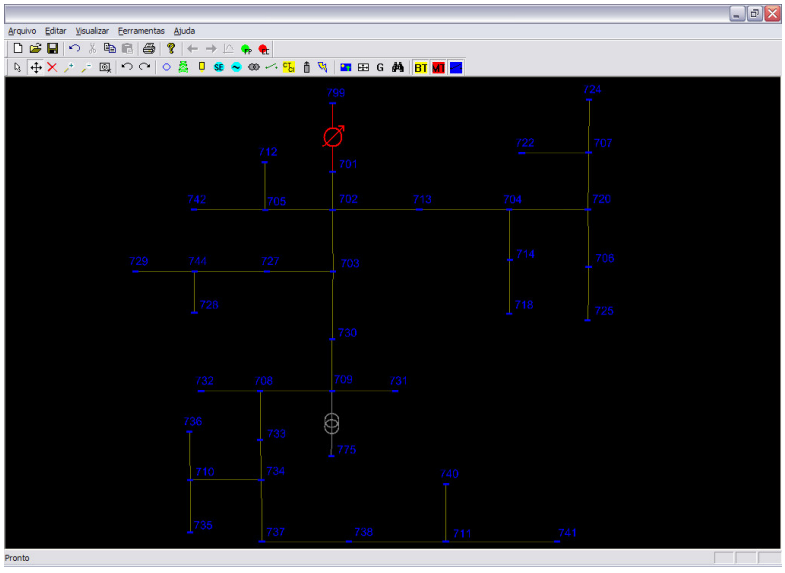

Figura 4.8 - Sistema IEEE37 na aplicação desenvolvida

A tensão base do sistema para todas as barras é de 2,7 kV.

Tabela 4.5 - Resultado do caso IEEE37

Barra	Va (kV)	∠a (graus)	Vb(kV)	∠b (graus)	Vc(kV)	∠c (graus)
799	2.9095	0	2.8403	-120	2.8755	120
701	2.8678	-0.1151	2.8133	-120.205	2.8372	119.5712
702	2.8523	-0.1755	2.8021	-120.29	2.8229	119.4165
703	2.8374	-0.263	2.7921	-120.316	2.813	119.268
727	2.8355	-0.2688	2.7907	-120.308	2.812	119.26
744	2.8339	-0.2797	2.7891	-120.299	2.8112	119.2567
728	2.8332	-0.28	2.7885	-120.299	2.8106	119.2552
729	2.8335	-0.285	2.7887	-120.294	2.8112	119.2562
730	2.8211	-0.2746	2.782	-120.266	2.8026	119.1661
709	2.8163	-0.2844	2.7787	-120.249	2.7998	119.1411
708	2.8088	-0.3015	2.7743	-120.209	2.7962	119.0914
732	2.8082	-0.2965	2.7743	-120.209	2.7957	119.0844
733	2.8017	-0.3236	2.7698	-120.17	2.7931	119.0485
734	2.791	-0.3401	2.7637	-120.119	2.7876	118.9754
710	2.7893	-0.3236	2.763	-120.129	2.7853	118.96
735	2.7886	-0.3176	2.763	-120.129	2.7847	118.951
736	2.7892	-0.3211	2.7611	-120.152	2.7833	118.9801
737	2.782	-0.3895	2.7577	-120.049	2.7851	118.9245
738	2.7783	-0.3946	2.756	-120.029	2.7834	118.8953
711	2.7763	-0.3761	2.756	-120.029	2.7817	118.8684
740	2.7756	-0.3701	2.756	-120.029	2.7811	118.8594
741	2.7756	-0.3698	2.756	-120.029	2.7811	118.8597
731	2.8162	-0.2821	2.7769	-120.271	2.7979	119.1591
705	2.8508	-0.164	2.8008	-120.303	2.8205	119.4109
712	2.85	-0.1572	2.8008	-120.303	2.8198	119.4004
742	2.8506	-0.164	2.7998	-120.314	2.8194	119.4205
713	2.8482	-0.1655	2.7977	-120.313	2.8174	119.4118
704	2.844	-0.1659	2.7914	-120.346	2.811	119.4279
714	2.8438	-0.1696	2.7911	-120.344	2.8109	119.4281
718	2.8423	-0.1902	2.7895	-120.326	2.811	119.4261
720	2.8404	-0.1293	2.7852	-120.423	2.8016	119.4507
706	2.8404	-0.1282	2.7843	-120.434	2.8006	119.4601
725	2.8404	-0.1276	2.7839	-120.439	2.8002	119.4645
707	2.8394	-0.115	2.7794	-120.495	2.7948	119.5036
722	2.8393	-0.1133	2.7788	-120.502	2.7941	119.5086
724	2.8393	-0.1135	2.7783	-120.508	2.7936	119.5157

Estes resultados apresentados com a metodologia proposta são iguais aos obtidos em KERSTING (2000) e aos obtidos utilizando-se o MICT (Garcia, 2001)

4.6 Caso 5 – Concessionária de Energia Elétrica

O sistema apresentado na Figura 4.9, é um sistema de distribuição que possui 186 barras e uma subestação. Os alimentadores primários são radiais e o secundário ligado de forma reticulada. Todos os transformadores possuem a ligação delta/Yaterrado, sendo o delta do lado de alta tensão o Yaterrado do lado de baixa tensão.

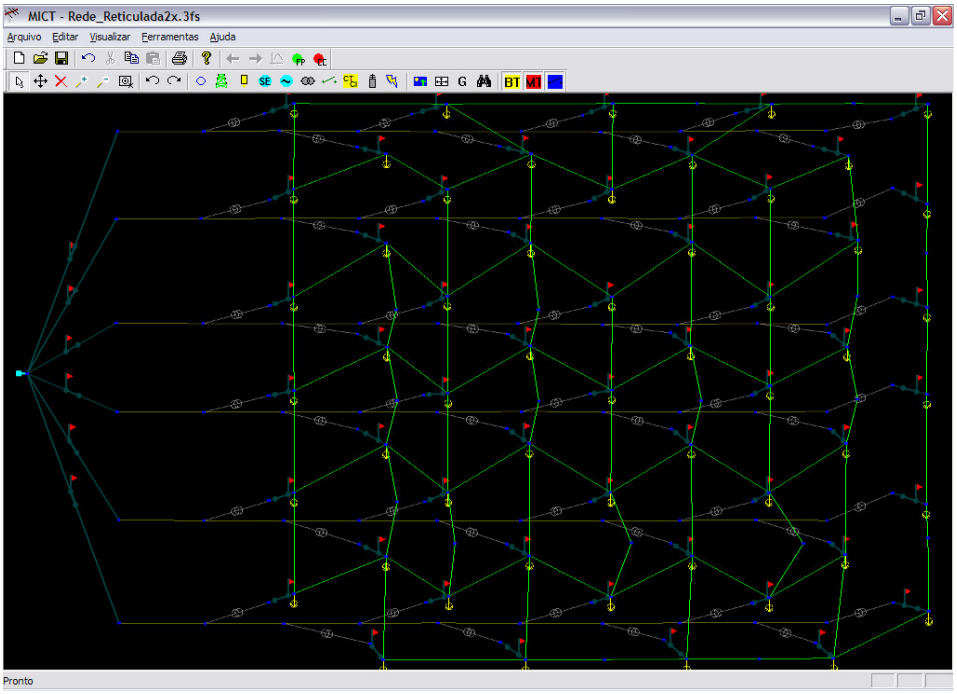

Figura 4.9 - Sistema de distribuição na aplicação desenvolvida

Os resultados dos sistema podem ser vistos na Tabela 4.6 .

Tabela 4.6 - Resultados do caso sistema de distribuição

Barra	Va (kV)	∠a (graus)	Vb(kV)	∠b (graus)	Vc(kV)	∠c (graus)
1000	7.6210	0.0000	7.6210	-119.9999	7.6210	119.9999
114	7.6210	-0.0008	7.6210	-120.0007	7.6210	119.9991
113	7.6210	-0.0006	7.6210	-120.0005	7.6210	119.9993
112	7.6210	-0.0006	7.6210	-120.0005	7.6210	119.9993
111	7.6210	-0.0006	7.6210	-120.0005	7.6210	119.9993
110	7.6210	-0.0006	7.6210	-120.0005	7.6210	119.9993
109	7.6210	-0.0006	7.6210	-120.0005	7.6210	119.9993
1	7.6037	-0.0267	7.6037	-120.0266	7.6037	119.9732
3	7.5883	-0.0499	7.5883	-120.0498	7.5883	119.9500
5	7.5749	-0.0704	7.5749	-120.0703	7.5749	119.9295
7	7.5634	-0.0879	7.5634	-120.0878	7.5634	119.9120
9	7.5538	-0.1026	7.5538	-120.1025	7.5538	119.8973
11	7.5461	-0.1143	7.5461	-120.1142	7.5461	119.8856
13	7.5404	-0.1231	7.5404	-120.1230	7.5404	119.8768
15	7.5365	-0.1290	7.5365	-120.1289	7.5365	119.8709
19	7.6037	-0.0267	7.6037	-120.0266	7.6037	119.9732
21	7.5884	-0.0500	7.5884	-120.0499	7.5884	119.9499

23	7.5749	-0.0704	7.5749	-120.0703	7.5749	119.9295
25	7.5634	-0.0879	7.5634	-120.0878	7.5634	119.9120
27	7.5538	-0.1026	7.5538	-120.1025	7.5538	119.8973
29	7.5461	-0.1143	7.5461	-120.1142	7.5461	119.8855
31	7.5404	-0.1232	7.5404	-120.1231	7.5404	119.8767
33	7.5365	-0.1291	7.5365	-120.1290	7.5365	119.8708
17	7.5346	-0.1320	7.5346	-120.1319	7.5346	119.8679
35	7.5346	-0.1320	7.5346	-120.1319	7.5346	119.8679
37	7.6037	-0.0268	7.6037	-120.0267	7.6037	119.9731
39	7.5884	-0.0501	7.5884	-120.0500	7.5884	119.9498
41	7.5749	-0.0705	7.5749	-120.0704	7.5749	119.9294
43	7.5634	-0.0881	7.5634	-120.0880	7.5634	119.9118
45	7.5538	-0.1028	7.5538	-120.1027	7.5538	119.8971
47	7.5461	-0.1146	7.5461	-120.1145	7.5461	119.8853
49	7.5403	-0.1234	7.5403	-120.1233	7.5403	119.8765
51	7.5365	-0.1293	7.5365	-120.1292	7.5365	119.8706
53	7.5346	-0.1323	7.5346	-120.1322	7.5346	119.8676
55	7.6037	-0.0279	7.6037	-120.0278	7.6037	119.9720
57	7.5883	-0.0512	7.5883	-120.0511	7.5883	119.9487
59	7.5749	-0.0717	7.5749	-120.0716	7.5749	119.9282
61	7.5634	-0.0893	7.5634	-120.0892	7.5634	119.9106
63	7.5538	-0.1040	7.5538	-120.1039	7.5538	119.8959
65	7.5461	-0.1158	7.5461	-120.1157	7.5461	119.8841
67	7.5403	-0.1247	7.5403	-120.1246	7.5403	119.8752
69	7.5364	-0.1306	7.5364	-120.1305	7.5364	119.8693
71	7.5345	-0.1336	7.5345	-120.1335	7.5345	119.8663
73	7.6028	-0.0315	7.6028	-120.0314	7.6028	119.9684
75	7.5875	-0.0555	7.5875	-120.0554	7.5875	119.9444
77	7.5740	-0.0765	7.5740	-120.0764	7.5740	119.9234
79	7.5625	-0.0943	7.5625	-120.0942	7.5625	119.9056
81	7.5528	-0.1092	7.5528	-120.1091	7.5528	119.8907
83	7.5451	-0.1209	7.5451	-120.1208	7.5451	119.8790
85	7.5393	-0.1297	7.5393	-120.1296	7.5393	119.8702
87	7.5355	-0.1355	7.5355	-120.1354	7.5355	119.8644
89	7.5335	-0.1385	7.5335	-120.1384	7.5335	119.8614
91	7.5959	-0.0249	7.5959	-120.0248	7.5959	119.9750
93	7.5797	-0.0559	7.5797	-120.0558	7.5797	119.9440
95	7.5654	-0.0828	7.5654	-120.0827	7.5654	119.9171
97	7.5539	-0.1028	7.5539	-120.1027	7.5539	119.8971
99	7.5443	-0.1188	7.5443	-120.1187	7.5443	119.8811
101	7.5367	-0.1306	7.5367	-120.1305	7.5367	119.8693
103	7.5310	-0.1395	7.5310	-120.1394	7.5310	119.8604
105	7.5272	-0.1454	7.5272	-120.1453	7.5272	119.8545
107	7.5253	-0.1484	7.5253	-120.1483	7.5253	119.8515
4	0.1226	-31.9648	0.1226	-151.9647	0.1226	88.0352
8	0.1223	-31.9748	0.1223	-151.9747	0.1223	88.0252
116	0.1221	-31.9960	0.1221	-151.9959	0.1221	88.0041
12	0.1219	-32.0172	0.1219	-152.0171	0.1219	87.9828
117	0.1219	-32.0280	0.1219	-152.0279	0.1219	87.9720
16	0.1218	-32.0389	0.1218	-152.0388	0.1218	87.9612
2	0.1229	-31.9282	0.1229	-151.9281	0.1229	88.0718
6	0.1224	-31.9626	0.1224	-151.9625	0.1224	88.0375
10	0.1221	-32.0050	0.1221	-152.0049	0.1221	87.9950
14	0.1218	-32.0323	0.1218	-152.0322	0.1218	87.9678
22	0.1227	-31.9433	0.1227	-151.9431	0.1227	88.0568
26	0.1222	-31.9836	0.1222	-151.9835	0.1222	88.0165
30	0.1219	-32.0204	0.1219	-152.0203	0.1219	87.9796
34	0.1218	-32.0381	0.1218	-152.0380	0.1218	87.9620

20	0.1229	-31.9261	0.1229	-151.9260	0.1229	88.0740
24	0.1224	-31.9611	0.1224	-151.9610	0.1224	88.0389
28	0.1221	-32.0062	0.1221	-152.0061	0.1221	87.9938
32	0.1218	-32.0326	0.1218	-152.0325	0.1218	87.9675
40	0.1227	-31.9381	0.1227	-151.9380	0.1227	88.0619
44	0.1222	-31.9854	0.1222	-151.9853	0.1222	88.0146
48	0.1219	-32.0213	0.1219	-152.0212	0.1219	87.9788
52	0.1218	-32.0382	0.1218	-152.0381	0.1218	87.9618
119	0.1224	-31.9618	0.1224	-151.9617	0.1224	88.0382
120	0.1221	-32.0056	0.1221	-152.0055	0.1221	87.9944
121	0.1218	-32.0324	0.1218	-152.0323	0.1218	87.9676
38	0.1229	-31.9311	0.1229	-151.9310	0.1229	88.0689
42	0.1224	-31.9554	0.1224	-151.9553	0.1224	88.0446
46	0.1221	-32.0107	0.1221	-152.0106	0.1221	87.9894
50	0.1218	-32.0348	0.1218	-152.0347	0.1218	87.9653
58	0.1227	-31.9353	0.1227	-151.9352	0.1227	88.0647
62	0.1222	-31.9854	0.1222	-151.9853	0.1222	88.0146
66	0.1219	-32.0236	0.1219	-152.0235	0.1219	87.9765
70	0.1218	-32.0407	0.1218	-152.0406	0.1218	87.9594
123	0.1227	-31.9367	0.1227	-151.9366	0.1227	88.0633
118	0.1227	-31.9407	0.1227	-151.9406	0.1227	88.0594
124	0.1222	-31.9854	0.1222	-151.9853	0.1222	88.0146
125	0.1219	-32.0224	0.1219	-152.0223	0.1219	87.9776
126	0.1218	-32.0395	0.1218	-152.0394	0.1218	87.9606
56	0.1231	-32.0573	0.1231	-152.0572	0.1231	87.9428
60	0.1224	-31.9600	0.1224	-151.9599	0.1224	88.0401
64	0.1221	-32.0006	0.1221	-152.0005	0.1221	87.9995
68	0.1218	-32.0445	0.1218	-152.0444	0.1218	87.9556
129	0.1218	-32.0454	0.1218	-152.0453	0.1218	87.9547
76	0.1227	-31.9424	0.1227	-151.9423	0.1227	88.0576
80	0.1223	-31.9937	0.1223	-151.9936	0.1223	88.0064
84	0.1219	-32.0309	0.1219	-152.0308	0.1219	87.9692
88	0.1218	-32.0501	0.1218	-152.0500	0.1218	87.9499
127	0.1227	-31.9389	0.1227	-151.9388	0.1227	88.0612
128	0.1222	-31.9896	0.1222	-151.9895	0.1222	88.0105
74	0.1227	-33.0560	0.1227	-153.0559	0.1227	86.9440
78	0.1225	-31.9773	0.1225	-151.9772	0.1225	88.0227
82	0.1221	-32.0301	0.1221	-152.0300	0.1221	87.9699
86	0.1218	-32.0496	0.1218	-152.0495	0.1218	87.9505
94	0.1228	-32.0626	0.1228	-152.0625	0.1228	87.9375
98	0.1223	-32.0959	0.1223	-152.0958	0.1223	87.9042
102	0.1218	-32.0284	0.1218	-152.0283	0.1218	87.9716
106	0.1217	-32.0437	0.1217	-152.0436	0.1217	87.9564
92	0.1119	-37.7932	0.1119	-157.7931	0.1119	82.2068
96	0.1221	-33.0988	0.1221	-153.0987	0.1221	86.9013
100	0.1222	-32.1319	0.1222	-152.1318	0.1222	87.8682
104	0.1217	-32.0379	0.1217	-152.0378	0.1217	87.9622
133	0.1220	-32.0849	0.1220	-152.0848	0.1220	87.9151
108	0.1216	-32.0539	0.1216	-152.0538	0.1216	87.9461
134	0.1217	-32.0459	0.1217	-152.0458	0.1217	87.9542
90	0.1217	-32.0618	0.1217	-152.0617	0.1217	87.9382
72	0.1217	-32.0508	0.1217	-152.0507	0.1217	87.9493
54	0.1217	-32.0483	0.1217	-152.0482	0.1217	87.9517
132	0.1217	-32.0563	0.1217	-152.0562	0.1217	87.9438
1008	0.1223	-31.9748	0.1223	-151.9747	0.1223	88.0253
1016	0.1218	-32.0388	0.1218	-152.0387	0.1218	87.9612
1002	0.1229	-31.9282	0.1229	-151.9281	0.1229	88.0719
1006	0.1224	-31.9625	0.1224	-151.9624	0.1224	88.0375

1010	0.1221	-32.0049	0.1221	-152.0048	0.1221	87.9951
1014	0.1218	-32.0322	0.1218	-152.0321	0.1218	87.9678
1022	0.1227	-31.9432	0.1227	-151.9431	0.1227	88.0569
1026	0.1222	-31.9835	0.1222	-151.9834	0.1222	88.0165
1030	0.1219	-32.0204	0.1219	-152.0203	0.1219	87.9797
1034	0.1218	-32.0380	0.1218	-152.0379	0.1218	87.9621
1020	0.1229	-31.9260	0.1229	-151.9259	0.1229	88.0740
1024	0.1224	-31.9611	0.1224	-151.9610	0.1224	88.0390
1028	0.1221	-32.0062	0.1221	-152.0061	0.1221	87.9939
1032	0.1218	-32.0325	0.1218	-152.0324	0.1218	87.9675
1040	0.1227	-31.9381	0.1227	-151.9379	0.1227	88.0620
1044	0.1222	-31.9854	0.1222	-151.9853	0.1222	88.0147
1048	0.1219	-32.0212	0.1219	-152.0211	0.1219	87.9788
1052	0.1218	-32.0382	0.1218	-152.0381	0.1218	87.9619
1038	0.1229	-31.9311	0.1229	-151.9310	0.1229	88.0690
1042	0.1224	-31.9554	0.1224	-151.9553	0.1224	88.0447
1046	0.1221	-32.0106	0.1221	-152.0105	0.1221	87.9894
1050	0.1218	-32.0347	0.1218	-152.0346	0.1218	87.9654
1058	0.1227	-31.9353	0.1227	-151.9352	0.1227	88.0648
1062	0.1222	-31.9854	0.1222	-151.9853	0.1222	88.0147
1066	0.1219	-32.0235	0.1219	-152.0234	0.1219	87.9766
1070	0.1218	-32.0406	0.1218	-152.0405	0.1218	87.9594
1056	0.1231	-32.0572	0.1231	-152.0571	0.1231	87.9428
1060	0.1224	-31.9599	0.1224	-151.9598	0.1224	88.0401
1064	0.1221	-32.0005	0.1221	-152.0004	0.1221	87.9996
1068	0.1218	-32.0444	0.1218	-152.0443	0.1218	87.9556
1076	0.1227	-31.9424	0.1227	-151.9423	0.1227	88.0577
1080	0.1223	-31.9936	0.1223	-151.9935	0.1223	88.0065
1084	0.1219	-32.0308	0.1219	-152.0307	0.1219	87.9692
1088	0.1218	-32.0501	0.1218	-152.0500	0.1218	87.9500
1074	0.1227	-33.0559	0.1227	-153.0558	0.1227	86.9441
1078	0.1225	-31.9773	0.1225	-151.9772	0.1225	88.0228
1082	0.1221	-32.0301	0.1221	-152.0300	0.1221	87.9700
1086	0.1218	-32.0495	0.1218	-152.0494	0.1218	87.9505
1094	0.1228	-32.0625	0.1228	-152.0624	0.1228	87.9376
1098	0.1223	-32.0958	0.1223	-152.0957	0.1223	87.9042
1102	0.1218	-32.0284	0.1218	-152.0283	0.1218	87.9717
1106	0.1217	-32.0436	0.1217	-152.0435	0.1217	87.9564
1092	0.1119	-37.7929	0.1119	-157.7928	0.1119	82.2071
1096	0.1221	-33.0987	0.1221	-153.0986	0.1221	86.9014
1100	0.1222	-32.1318	0.1222	-152.1317	0.1222	87.8683
1104	0.1217	-32.0378	0.1217	-152.0377	0.1217	87.9623
1108	0.1216	-32.0539	0.1216	-152.0538	0.1216	87.9462
36	0.1217	-32.0476	0.1217	-152.0475	0.1217	87.9525
122	0.1218	-32.0460	0.1218	-152.0459	0.1218	87.9540
18	0.1218	-32.0445	0.1218	-152.0444	0.1218	87.9556
1090	0.1217	-32.0617	0.1217	-152.0616	0.1217	87.9383
1072	0.1217	-32.0507	0.1217	-152.0506	0.1217	87.9493
1054	0.1217	-32.0483	0.1217	-152.0482	0.1217	87.9518
1036	0.1217	-32.0475	0.1217	-152.0474	0.1217	87.9525
1018	0.1218	-32.0444	0.1218	-152.0443	0.1218	87.9556
1012	0.1219	-32.0172	0.1219	-152.0170	0.1219	87.9829
1004	0.1226	-31.9648	0.1226	-151.9647	0.1226	88.0353

Estes resultados apresentados com a metodologia proposta são iguais aos obtidos utilizando-se o MICT (Garcia, 2001), o que permite validar a metodologia e a implementação realizadas.

4.7 Considerações Finais

Neste capítulo foram apresentadas diversas simulações utilizando-se sistemas teste consagrados na literatura com o objetivo de validar a metodologia de fluxo de potência proposta, bem como demonstrar a aplicação desenvolvida. Foram feitas comparações com os resultados fornecidos nas referências citadas e com a metodologia MICT (Garcia, 2001).

Todos os resultados mostraram grande eficiência do método proposto, para sistemas elétricos radiais, malhados, equilibrados e desequilibrados.

Capítulo 5 Conclusões

5.1 Considerações Finais

Neste trabalho apresentou-se uma inovadora plataforma para a análise de Sistemas Elétricos de Potência utilizando os conceitos da Modelagem Orientada a Objetos (MOO).

A estrutura de classes proposta foi desenvolvida observando-se a estrutura física dos sistemas elétricos. Assim sendo, os diversos componentes do sistema, tanto da parte gráfica quanto da parte numérica, são representados em classes distintas que se relacionam conforme a conectividade entre eles.

Neste trabalho foi desenvolvido o Fluxo de Potência Trifásico em Coordenadas Polares, um método para o cálculo de fluxo de potência em sistemas trifásicos, que se mostrou robusto nos mais diversos sistemas testados, tornando-se uma alternativa aos métodos de cálculo tradicionais. Esta metodologia pode ser utilizada para análise de sistemas equilibrados ou desequilibrados, redes radiais ou reticuladas, com cargas ou linhas monofásicas, bifásicas ou trifásicas, tanto para sistemas de transmissão, subtransmissão ou distribuição. A grande vantagem da metodologia trifásica proposta é possibilitar uma modelagem mais completa do sistema, com inclusão de modelagens mais complexas dos equipamentos, tornando assim os resultados obtidos nas simulações mais próximos da realidade. Além disso, todos os casos analisados obtiveram resultados corretos, confirmando assim a validade da metodologia apresentada.

Para transmissão o método em coordenadas polares possui a vantagem de trabalhar diretamente nas grandezas de tensão e ângulo das barras e Potência Ativa e Reativa, o que permite obter matrizes de sensibilidade que são muito utilizadas.

Além da metodologia matemática, foi desenvolvida uma interface gráfica para a visualização dos elementos dos Sistemas Elétricos. Uma contribuição extremamente importante deste trabalho é a utilização da biblioteca gráfica OpenGL para a confecção de interface gráfica de bom desempenho e robustez. Esta interface permitiu uma rápida e fácil análise dos dados e resultados por parte do usuário uma vez que representa a topologia da rede como ela realmente é e possui um código de cores para mostrar quais elementos estão fora dos limites especificados. Um aspecto que merece destaque é o

fato de que a interface gráfica não interfere no desempenho do método numérico, pois foram desenvolvidas em classes distintas.

Vale ainda ressaltar que as duas metodologias apresentadas neste trabalho, tanto a gráfica quanto numérica, foram desenvolvidas sobre um mesma plataforma, de forma modular e independente. Desta forma permite-se que qualquer novo equipamento ou método numérico seja a ela agregado, sem grandes mudanças no código do programa.

5.2 Trabalhos Futuros

Este pacote de análise de redes pode ser considerado o início dos estudos para desenvolvimento de uma grande plataforma para análise, envolvendo operação e planejamento de Sistemas Elétricos de Potência de forma geral. Seria interessante que outras aplicações fossem desenvolvidas e implementadas sobre essa mesma plataforma de forma a se obter um único ambiente para os mais diversos tipos de análises (curto-circuito, transitórios eletromagnéticos, fluxo de potência continuado, etc).

Sugestões de trabalhos a serem realizados futuramente são:

- Implementação das novas metodologias numéricas desenvolvidas sobre uma mesma plataforma;
- Implementação da metodologia a quatro fios utilizando a metodologia desenvolvida.
- Implementação de um Fluxo de Potencia Ótimo a partir do Fluxo de Potencia Trifásico em Coordenadas Polares.
- Implementação dos modelos de HVDC, retificadores e inversores;
- Implementação de modelos de motores de indução;
- Inserção de um de Banco de Dados permitindo a atualização dos dados remotamente.
- Melhoria dos gráficos e de algumas funcionalidades da interface gráfica.

Apêndice A Modelagem Orientada a Objetos

A.1 Introdução

Com o aperfeiçoamento contínuo dos dispositivos computacionais, os modelos matemáticos implementados apresentam mais conceitos e nuances do mundo real, tornando assim os programas maiores e mais complexos. Com isso, as técnicas de modelagem estruturada começaram a apresentar falta de recursos tanto para o desenvolvimento do projeto quanto para a implementação e a manutenção. Desta forma surgiu um novo conceito de modelagem; "A Modelagem Baseada em Objetos".

Este novo conceito apresenta um modo de estudar os problemas reais, no qual, combina-se a estrutura (dados) do projeto com o comportamento dos dados (funções) em uma única entidade, o objeto. Isto quer dizer que o software é organizado como vários objetos separados que incorporam tanto a estrutura quanto o comportamento de dados, diferenciando da modelagem estruturada onde existe pouca vinculação entre estrutura e o comportamento dos dados. Essa abordagem possui várias características, que serão detalhadas neste capítulo.

Identidade: São os objetos em si. No mundo real o objeto pode ser uma mesa, uma sala, um prédio ou um planeta, dependendo do que estamos estudando e como se encaixa esta entidade no conceito estudado. Por exemplo, considere que um decorador utilizará a mesa como um objeto dentro de um universo (sistema) sala, porém um engenheiro poderá utilizar o objeto sala dentro do universo prédio, que por sua vez será como um objeto dentro do universo cidade, ou seja, os objetos são relativos ao sistema estudado.

No mundo real um objeto limita-se apenas em existir, mas, em linguagem de programação este objeto deve ser identificado por um identificador ou um endereço de memória ou um atributo exclusivo, sendo estas referências uniformes e independentes do conteúdo dos mesmos. Estes objetos em linguagem de programação podem ser tanto a representação de um objeto real, como um transformador que possui dados como material, dimensão, cor e outros dados que forem julgados pertinentes ao projeto em estudo, como também podem ser uma entidade puramente computacional, listas encadeadas, árvores binárias, etc.

Classificação: É o conceito de classe, significa que todos os objetos que possuem mesma estrutura e mesmo comportamento são agrupados em uma classe, então podemos ter vários objetos transformadores distintos entre si, mas todos eles pertencem a uma mesma classe. A classe é uma abstração que descreve as implicações relevantes ao estudo de um caso.

Uma determinada classe possui vários objetos individuais, por isto dizemos que cada objeto é uma instância de sua classe, mas cada um deles possui seu próprio valor para cada atributo e um identificador único, compartilhando apenas o mesmo nome de atributos e operações.

Polimorfismo: São métodos (funções) que possuem o mesmo nome, mas tem implementações completamente diferentes, dependendo do objeto que sofre a ação. Assim classes podem ser melhoradas ou criadas sem que seja necessário reescrever todo o código já existente, mas apenas criando novas funções.

Herança: É o refinamento de uma determinada classe em um relacionamento hierárquico. Uma classe *CElemento* é criada, possuindo atributos e métodos próprios. Desta classe é derivada uma outra classe que possui todos os atributos e métodos da superclasse (Classe Pai), com as características exclusivas adicionais. Pode-se, por exemplo, derivar a classe *CElemento* para uma classe *CLinha*, que possui todas as características da classe *CElemento*, mais outros dados, métodos ou polimorfismo de funções julgadas relevantes.

A.2 Características da Tecnologia Baseada em Objetos

Um sistema baseado em objetos deve possuir características que possam ser utilizadas de modo eficiente e abrangente visando uma melhor reutilização do código e facilidade de projetos. Abaixo são descritas as características mais importantes.

Abstração: Consiste em focalizar os aspectos de interesse de um objeto e ignorar as propriedades irrelevantes ao problema estudado, ou seja, tratar um objeto pelas características essenciais ao problema estudado. A abstração é utilizada no projeto, não na implementação. Desta forma evita-se o comprometimento prematuro com detalhes

que dificultaria a análise do projeto como um todo, e não dando assim, importância a fatos ainda não compreendidos. O projeto deve ser independente da linguagem de programação.

Encapsulamento: Chamado de ocultamento de informação, onde cada objeto deve separar seus dados e funções de forma que as informações relativas ao funcionamento interno estejam ocultas e as externas sejam acessíveis por outros objetos. Isto impede que um programa se torne dependente de pequenas alterações de modo que cause grandes efeitos de propagação. Como exemplo tem-se objeto Caixa Eletrônico, onde várias características estão encapsuladas como o cofre interno e circuitos, e apresentam interfaces externas como o teclado e monitor para se relacionar com o objeto Pessoa.

Combinação de Dados e Comportamento: Esta característica, também conhecida como polimorfismo, permite a adoção de nomes iguais para funções de mesma natureza, porém de classes distintas. Por exemplo, ao se desenvolver um software para desenhos geométricos não utilizando MOO, precisa-se de funções com nomes distintos para cada figura que for desenhada na tela (Desenhe_Circulo, Desenhe_Quadrado, etc..). Utilizando o polimorfismo, todo objeto possui a seu próprio método de desenho, onde este nome é comum a todos (Desenhe) e basta convocar este método para que o próprio compilador, com base nas classes, tome implicitamente a decisão que qual método irá chamar. O uso de polimorfismo propicia que manutenção, reaproveitamento e implementação de novos recursos tornam-se mais simples.

Compartilhamento: Consiste na reutilização do código por herança. Evita a redundância de informações, a redução de trabalho de codificação e apresenta uma maior clareza no desenvolvimento, pois mostra que diferentes operações são na realidade a mesma. O desenvolvimento baseado em objetos não somente permite que as informações sejam compartilhadas em uma aplicação, como também oferece a possibilidade da reutilização de modelos e códigos em projetos futuros.

Estrutura de Objetos: Especifica o que um objeto é, e não como ele é utilizado, pois sua estrutura permanece mais estável durante o desenvolver de um projeto, enquanto, seu uso é altamente dependente dos detalhes de implementação.

A.3 A Representação dos Modelos Utilizando UML

A construção de um determinado processo sem que antes seja realizado um estudo detalhado pode trazer grandes prejuízos econômicos e financeiros. Desta forma deve-se estudar e modelar o projeto para que os riscos sejam avaliados. No caso de modelos computacionais, o projeto consiste em uma abstração do que se deseja construir, ou seja, a partir daí pode-se compreender melhor o fenômeno, ressaltando seus pontos relevantes e abstraindo as irrelevâncias.

Para descrever uma aplicação computacional, existem metodologias que padronizam processos de modo simples, diagramas. Sendo a *Unified Modeling Language* (UML) a metodologia mais aceita.

Na UML cada diagrama representa uma perspectiva do modelo (aplicação), mostrando aspectos particulares do sistema e dando enfoque a ângulos e níveis de abstrações diferentes para que uma figura completa do sistema seja construída. Os principais diagramas da UML são:

- Diagramas de Classe:
- Diagramas de Interação entre Classes
- Diagramas de Estado
- Diagramas de Interface e Arquitetura

A.3.1 Diagramas de Classe

O diagrama de classe oferece modelos sobre o aspecto chave em orientação a objetos, a representação da classe/Objeto.

Representação de Classe (Objeto)

A representação de uma classe na UML é um retângulo dividido em três partes: O nome da classe, os atributos (dados membros) e operações ou métodos (funções membros), como pode ser observado na Figura A.1. A sintaxe utilizada nestes compartimentos é independente da linguagem a ser utilizada para a programação. A UML aceita uma forma simplificada de representação de classes, onde apenas o retângulo com o nome da classe é representado (Figura A.2).

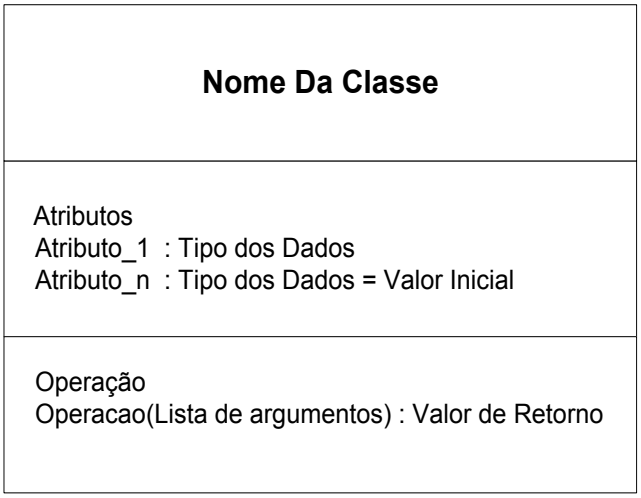

Figura A.1 – Diagrama completo de uma classe

Figura A.2 – Diagrama simplificado de uma classe

A.3.2 Diagramas de Interação entre Classes

São os relacionamentos que ligam classes/objetos entre si criando relações lógicas entre estas entidades. A seguir serão apresentados os relacionamentos e suas representações gráficas.

Associação: São as ligações físicas ou conceituais entre os objetos, uma associação normalmente determina interações entre objetos. Numa implementação computacional, uma associação é referida como um ponteiro de um objeto para o outro. A Figura A.3 ilustra os diagramas das associações simples e recursiva.

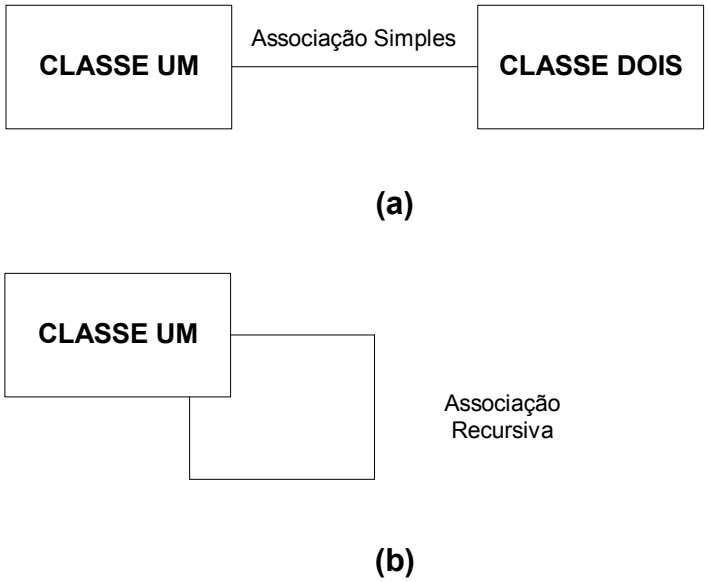

Figura A.3 – Diagrama de Associações entre Classes: (a) Simples; (b) Recursiva

Dependência: Indica um relacionamento, onde uma classe cliente é dependente de outra, mas não existem ligações físicas ou estruturais entre os objetos. O diagrama desta associação é apresentado na Figura A.4.

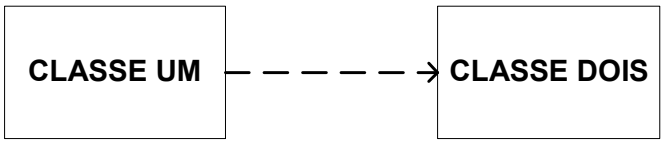

Figura A.4 – Diagrama de dependências

Herança: São abstrações para o compartilhamento de semelhanças entre as classes, ao mesmo tempo em que suas diferenças são preservadas. O relacionamento de uma classe (denominada Superclasse) com uma ou mais refinações dela (denominadas Subclasses) é denominado generalização ou herança. A herança/generalização pode ser transmitida a um número arbitrário de níveis. A instância de uma subclasse é simultaneamente uma instância de todas as superclasses a ela, ou seja, herda todas as características e operações de suas superclasses além de acrescentar atributos e operações próprias. Um diagrama de herança é apresentado na Figura A.5.

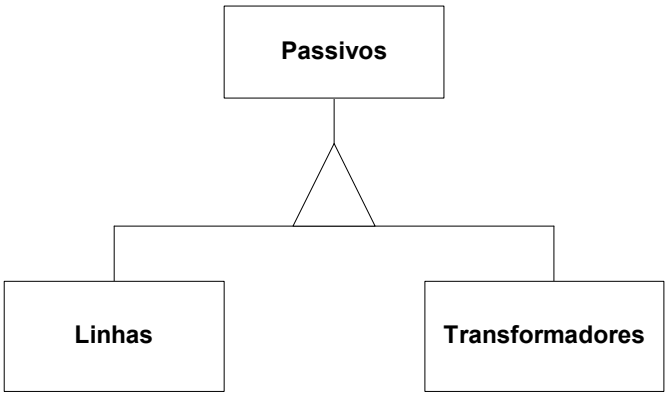

Figura A.5 – Diagramas de herança

Agregação: É uma relação na qual um objeto (componente) faz parte de um outro objeto (agregado), ou seja, utiliza-se um objeto para criar um novo objeto, onde seus dados membros são outros objetos. Um relacionamento de agregação é definido como o relacionamento de uma classe com uma outra classe (componente), logo uma estrutura com vários tipos de componentes equivale a muitos relacionamentos de agregação. Uma agregação pode ser observada na Figura A.6.

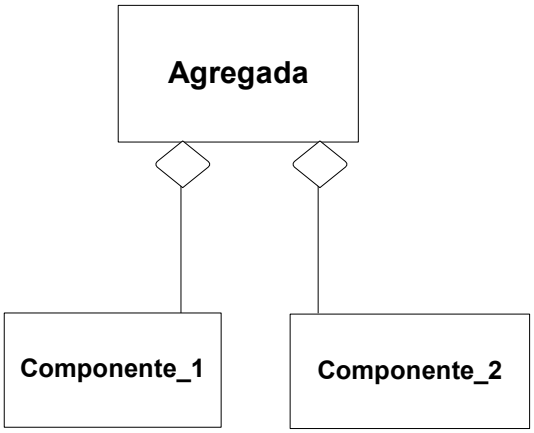

Figura A.6 – Diagrama de agregação

Pacotes

São macroentidades que agrupam vários objetos e seus relacionamentos definindo um conceito mais amplo e geral. A Figura A.7 mostra um diagrama de pacotes.

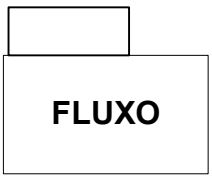

Figura A.7 – Diagrama de pacotes

Templates

São definidas como um conjunto de operações que podem ser aplicadas para vários tipos de dados. Um objeto *template* (classe, função ou estrutura) tem seu tipo de dados definido através de parâmetros. Quando feito isto, o compilador gera automaticamente o código correto para o tipo de dados utilizado, ou seja, quando criada uma "*template*" esta pode automaticamente sobrecarregar a si mesma. Uma *template* é definida conforme a Figura A.8.

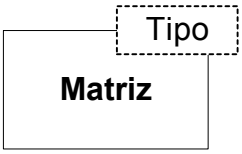

Figura A.8 – Diagrama de template

Apêndice B Derivadas do Método de Fluxo de Potência Trifásico em Coordenadas Polares (FPTP)

As derivadas apresentadas nas Equações de (B.1) a (B.36) são referentes à Equação **(2.9)**.

$$\frac{\partial P_{km}^{a}}{\partial \theta_{k}^{a}} = -G_{km}^{aa}V_{k}^{a}V_{k}^{a}0.\sin(\theta_{k}^{a}-\theta_{k}^{a}) - B_{km}^{aa}V_{k}^{a}V_{k}^{a}0.\cos(\theta_{k}^{a}-\theta_{k}^{a}) + G_{km}^{aa}V_{k}^{a}V_{m}^{a}\sin(\theta_{k}^{a}-\theta_{m}^{a}) \quad \textbf{(B.1)}$$
$$- B_{km}^{aa}V_{k}^{a}V_{m}^{a}\cos(\theta_{k}^{a}-\theta_{m}^{a}) - G_{km}^{ab}V_{k}^{a}V_{k}^{b}.\sin(\theta_{k}^{a}-\theta_{k}^{b}) + B_{km}^{ab}V_{k}^{a}V_{k}^{b}.\cos(\theta_{k}^{a}-\theta_{k}^{b})$$
$$+ G_{km}^{ab}V_{k}^{a}V_{m}^{b}\sin(\theta_{k}^{a}-\theta_{m}^{b}) - B_{km}^{ab}V_{k}^{a}V_{m}^{b}\cos(\theta_{k}^{a}-\theta_{m}^{b}) - G_{km}^{ac}V_{k}^{a}V_{k}^{c}.\sin(\theta_{k}^{a}-\theta_{k}^{c})$$
$$+ B_{km}^{ac}V_{k}^{a}V_{k}^{c}\cos(\theta_{k}^{a}-\theta_{k}^{c}) + G_{km}^{ac}V_{k}^{a}V_{m}^{c}\sin(\theta_{k}^{a}-\theta_{m}^{c}) - B_{km}^{ac}V_{k}^{a}V_{m}^{c}\cos(\theta_{k}^{a}-\theta_{m}^{c})$$

$$\frac{\partial P_{km}^{a}}{\partial \theta_{k}^{b}} = +G_{km}^{ab}V_{k}^{a}V_{k}^{b}\sin(\theta_{k}^{a}-\theta_{k}^{b}) + B_{km}^{ab}V_{k}^{a}V_{k}^{b}\cos(\theta_{k}^{a}-\theta_{k}^{b}) \quad \textbf{(B.2)}$$

$$\frac{\partial P_{km}^{a}}{\partial \theta_{k}^{c}} = +G_{km}^{ac}V_{k}^{a}V_{k}^{c}\sin(\theta_{k}^{a}-\theta_{k}^{c}) + B_{km}^{ac}V_{k}^{a}V_{k}^{c}\cos(\theta_{k}^{a}-\theta_{k}^{c}) \quad \textbf{(B.3)}$$

$$\frac{\partial P_{km}^{a}}{\partial V_{k}^{a}} = 2.V_{k}^{a}G_{km}^{aa}0.\cos(\theta_{k}^{a}-\theta_{k}^{a}) - 2.V_{k}^{a}B_{km}^{aa}0.\sin(\theta_{k}^{a}-\theta_{k}^{a}) + G_{km}^{aa}V_{m}^{a}\cos(\theta_{k}^{a}-\theta_{m}^{a}) \quad \textbf{(B.4)}$$
$$- B_{km}^{aa}V_{m}^{a}\sin(\theta_{k}^{a}-\theta_{m}^{a}) + G_{km}^{ab}V_{k}^{b}\cos(\theta_{k}^{a}-\theta_{k}^{b}) + B_{km}^{ab}V_{k}^{b}\sin(\theta_{k}^{a}-\theta_{k}^{b}) - G_{km}^{ab}V_{m}^{b}\cos(\theta_{k}^{a}-\theta_{m}^{b})$$
$$- B_{km}^{ab}V_{m}^{b}\sin(\theta_{k}^{a}-\theta_{m}^{b}) + G_{km}^{ac}V_{k}^{c}\cos(\theta_{k}^{a}-\theta_{k}^{c}) + B_{km}^{ac}V_{k}^{c}\sin(\theta_{k}^{a}-\theta_{k}^{c}) - G_{km}^{ac}V_{m}^{c}\cos(\theta_{k}^{a}-\theta_{m}^{c})$$
$$- B_{km}^{ac}V_{m}^{c}\sin(\theta_{k}^{a}-\theta_{m}^{c})$$

$$\frac{\partial P_{km}^{a}}{\partial V_{k}^{b}} = G_{km}^{ab}V_{k}^{a}\cos(\theta_{k}^{a}-\theta_{k}^{b}) + B_{km}^{ab}V_{k}^{a}\sin(\theta_{k}^{a}-\theta_{k}^{b}) \quad \textbf{(B.5)}$$

$$\frac{\partial P_{km}^{a}}{\partial V_{k}^{c}} = G_{km}^{ac}V_{k}^{a}\cos(\theta_{k}^{a}-\theta_{k}^{c}) + B_{km}^{ac}V_{k}^{a}\sin(\theta_{k}^{a}-\theta_{k}^{c}) \quad \textbf{(B.6)}$$

$$\frac{\partial P_{km}^{a}}{\partial \theta_{m}^{a}} = -G_{km}^{aa}V_{k}^{a}V_{m}^{a}\sin(\theta_{k}^{a}-\theta_{m}^{a}) + B_{km}^{aa}V_{k}^{a}V_{m}^{a}\cos(\theta_{k}^{a}-\theta_{m}^{a}) \quad \textbf{(B.7)}$$

$$\frac{\partial P_{km}^{a}}{\partial \theta_{m}^{b}} = -G_{km}^{ab} V_{k}^{a} V_{m}^{b} \sin(\theta_{k}^{a} - \theta_{m}^{b}) + B_{km}^{ab} V_{k}^{a} V_{m}^{b} \cos(\theta_{k}^{a} - \theta_{m}^{b}) \quad \textbf{(B.8)}$$

$$\frac{\partial P_{km}^{a}}{\partial \theta_{m}^{c}} = -G_{km}^{ac} V_{k}^{a} V_{m}^{c} \sin(\theta_{k}^{a} - \theta_{m}^{c}) + B_{km}^{ac} V_{k}^{a} V_{m}^{c} \cos(\theta_{k}^{a} - \theta_{m}^{c}) \quad \textbf{(B.9)}$$

$$\frac{\partial P_{km}^{a}}{\partial V_{m}^{a}} = -G_{km}^{aa} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{a}) - B_{km}^{aa} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{a}) \quad \textbf{(B.10)}$$

$$\frac{\partial P_{km}^{a}}{\partial V_{m}^{b}} = -G_{km}^{ab} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{b}) - B_{km}^{ab} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{b}) \quad \textbf{(B.11)}$$

$$\frac{\partial P_{km}^{a}}{\partial V_{m}^{c}} = -G_{km}^{ac} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{c}) - B_{km}^{ac} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{c}) \quad \textbf{(B.12)}$$

$$\frac{\partial P_{km}^{b}}{\partial \theta_{k}^{a}} = +G_{km}^{ba} V_{k}^{b} V_{k}^{a} \sin(\theta_{k}^{b} - \theta_{k}^{a}) + B_{km}^{ba} V_{k}^{b} V_{k}^{a} \cos(\theta_{k}^{b} - \theta_{k}^{a}) \quad \textbf{(B.13)}$$

$$\begin{aligned}\frac{\partial P_{km}^{b}}{\partial \theta_{k}^{b}} &= -G_{km}^{bb} V_{k}^{b} V_{k}^{b} 0.\sin(\theta_{k}^{b} - \theta_{k}^{b}) - B_{km}^{bb} V_{k}^{b} V_{k}^{b} 0.\cos(\theta_{k}^{b} - \theta_{k}^{b}) + G_{km}^{bb} V_{k}^{b} V_{m}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{b}) \\ &- B_{km}^{bb} V_{k}^{b} V_{m}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{b}) - G_{km}^{ba} V_{k}^{b} V_{k}^{a}.\sin(\theta_{k}^{b} - \theta_{k}^{a}) + B_{km}^{ba} V_{k}^{b} V_{k}^{a}.\cos(\theta_{k}^{b} - \theta_{k}^{a}) \\ &+ G_{km}^{ba} V_{k}^{b} V_{m}^{a} \sin(\theta_{k}^{b} - \theta_{m}^{a}) - B_{km}^{ba} V_{k}^{b} V_{m}^{a} \cos(\theta_{k}^{b} - \theta_{m}^{a}) - G_{km}^{bc} V_{k}^{b} V_{k}^{c}.\sin(\theta_{k}^{b} - \theta_{k}^{c}) \\ &+ B_{km}^{bc} V_{k}^{b} V_{k}^{c} \cos(\theta_{k}^{b} - \theta_{k}^{c}) + G_{km}^{bc} V_{k}^{b} V_{m}^{c} \sin(\theta_{k}^{b} - \theta_{m}^{c}) - B_{km}^{bc} V_{k}^{b} V_{m}^{c} \cos(\theta_{k}^{b} - \theta_{m}^{c})\end{aligned} \quad \textbf{(B.14)}$$

$$\frac{\partial P_{km}^{b}}{\partial \theta_{k}^{c}} = +G_{km}^{bc} V_{k}^{b} V_{k}^{c} \sin(\theta_{k}^{b} - \theta_{k}^{c}) + B_{km}^{bc} V_{k}^{b} V_{k}^{c} \cos(\theta_{k}^{b} - \theta_{k}^{c}) \quad \textbf{(B.15)}$$

$$\frac{\partial P_{km}^{b}}{\partial V_{k}^{a}} = G_{km}^{ba} V_{k}^{b} \cos(\theta_{k}^{b} - \theta_{k}^{a}) + B_{km}^{ba} V_{k}^{b} \sin(\theta_{k}^{b} - \theta_{k}^{a}) \quad \textbf{(B.16)}$$

$$\frac{\partial P_{km}^{b}}{\partial V_{k}^{b}} = 2.V_{k}^{b}G_{km}^{bb}0.\cos(\theta_{k}^{b}-\theta_{k}^{b}) - 2.V_{k}^{b}B_{km}^{bb}0.\sin(\theta_{k}^{b}-\theta_{k}^{b}) + G_{km}^{bb}V_{m}^{b}\cos(\theta_{k}^{b}-\theta_{m}^{b}) \quad \textbf{(B.17)}$$
$$-B_{km}^{bb}V_{m}^{b}\sin(\theta_{k}^{b}-\theta_{m}^{b}) + G_{km}^{ba}V_{k}^{a}\cos(\theta_{k}^{b}-\theta_{k}^{a}) + B_{km}^{ba}V_{k}^{a}\sin(\theta_{k}^{b}-\theta_{k}^{a}) - G_{km}^{ba}V_{m}^{a}\cos(\theta_{k}^{b}-\theta_{m}^{a})$$
$$-B_{km}^{ba}V_{m}^{a}\sin(\theta_{k}^{b}-\theta_{m}^{a}) + G_{km}^{bc}V_{k}^{c}\cos(\theta_{k}^{b}-\theta_{k}^{c}) + B_{km}^{bc}V_{k}^{c}\sin(\theta_{k}^{b}-\theta_{k}^{c}) - G_{km}^{bc}V_{m}^{c}\cos(\theta_{k}^{b}-\theta_{m}^{c})$$
$$-B_{km}^{bc}V_{m}^{c}\sin(\theta_{k}^{b}-\theta_{m}^{c})$$

$$\frac{\partial P_{km}^{b}}{\partial V_{k}^{c}} = G_{km}^{cb}V_{k}^{b}\cos(\theta_{k}^{b}-\theta_{k}^{c}) + B_{km}^{bc}V_{k}^{b}\sin(\theta_{k}^{b}-\theta_{k}^{c}) \quad \textbf{(B.18)}$$

$$\frac{\partial P_{km}^{b}}{\partial \theta_{m}^{a}} = -G_{km}^{ba}V_{k}^{b}V_{m}^{a}\sin(\theta_{k}^{b}-\theta_{m}^{a}) + B_{km}^{ba}V_{k}^{b}V_{m}^{a}\cos(\theta_{k}^{b}-\theta_{m}^{a}) \quad \textbf{(B.19)}$$

$$\frac{\partial P_{km}^{b}}{\partial \theta_{m}^{b}} = -G_{km}^{bb}V_{k}^{b}V_{m}^{b}\sin(\theta_{k}^{b}-\theta_{m}^{b}) + B_{km}^{bb}V_{k}^{b}V_{m}^{b}\cos(\theta_{k}^{b}-\theta_{m}^{b}) \quad \textbf{(B.20)}$$

$$\frac{\partial P_{km}^{b}}{\partial \theta_{m}^{c}} = -G_{km}^{bc}V_{k}^{b}V_{m}^{c}\sin(\theta_{k}^{b}-\theta_{m}^{c}) + B_{km}^{bc}V_{k}^{b}V_{m}^{c}\cos(\theta_{k}^{b}-\theta_{m}^{c}) \quad \textbf{(B.21)}$$

$$\frac{\partial P_{km}^{b}}{\partial V_{m}^{a}} = -G_{km}^{ba}V_{k}^{b}\cos(\theta_{k}^{b}-\theta_{m}^{a}) - B_{km}^{ba}V_{k}^{b}\sin(\theta_{k}^{b}-\theta_{m}^{a}) \quad \textbf{(B.22)}$$

$$\frac{\partial P_{km}^{b}}{\partial V_{m}^{b}} = -G_{km}^{bb}V_{k}^{b}\cos(\theta_{k}^{b}-\theta_{m}^{b}) - B_{km}^{bb}V_{k}^{b}\sin(\theta_{k}^{b}-\theta_{m}^{b}) \quad \textbf{(B.23)}$$

$$\frac{\partial P_{km}^{b}}{\partial V_{m}^{c}} = -G_{km}^{bc}V_{k}^{b}\cos(\theta_{k}^{b}-\theta_{m}^{c}) - B_{km}^{bc}V_{k}^{b}\sin(\theta_{k}^{b}-\theta_{m}^{c}) \quad \textbf{(B.24)}$$

$$\frac{\partial P_{km}^{c}}{\partial \theta_{k}^{a}} = +G_{km}^{ca}V_{k}^{c}V_{k}^{a}\sin(\theta_{k}^{c}-\theta_{k}^{a}) + B_{km}^{ca}V_{k}^{c}V_{k}^{a}\cos(\theta_{k}^{c}-\theta_{k}^{a}) \quad \textbf{(B.25)}$$

$$\frac{\partial P_{km}^{c}}{\partial \theta_{k}^{b}} = +G_{km}^{cb}V_{k}^{c}V_{k}^{b}\sin(\theta_{k}^{c}-\theta_{k}^{b}) + B_{km}^{cb}V_{k}^{c}V_{k}^{b}\cos(\theta_{k}^{c}-\theta_{k}^{b}) \quad \textbf{(B.26)}$$

$$\frac{\partial P_{km}^{c}}{\partial \theta_{k}^{c}} = -G_{km}^{cc}V_{k}^{c}V_{k}^{c}0.\sin(\theta_{k}^{c}-\theta_{k}^{c}) - B_{km}^{cc}V_{k}^{c}V_{k}^{c}0.\cos(\theta_{k}^{c}-\theta_{k}^{c}) + G_{km}^{cc}V_{k}^{c}V_{m}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{c}) \quad \textbf{(B.27)}$$
$$-B_{km}^{cc}V_{k}^{c}V_{m}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{c}) - G_{km}^{ca}V_{k}^{c}V_{k}^{a}.\sin(\theta_{k}^{c}-\theta_{k}^{a}) + B_{km}^{ca}V_{k}^{c}V_{k}^{a}.\cos(\theta_{k}^{c}-\theta_{k}^{a})$$
$$+G_{km}^{ca}V_{k}^{c}V_{m}^{a}\sin(\theta_{k}^{c}-\theta_{m}^{a}) - B_{km}^{ca}V_{k}^{c}V_{m}^{a}\cos(\theta_{k}^{c}-\theta_{m}^{a}) - G_{km}^{cb}V_{k}^{c}V_{k}^{b}.\sin(\theta_{k}^{c}-\theta_{k}^{b})$$
$$+B_{km}^{cb}V_{k}^{c}V_{k}^{b}\cos(\theta_{k}^{c}-\theta_{k}^{b}) + G_{km}^{cb}V_{k}^{c}V_{m}^{b}\sin(\theta_{k}^{c}-\theta_{m}^{b}) - B_{km}^{cb}V_{k}^{c}V_{m}^{b}\cos(\theta_{k}^{c}-\theta_{m}^{b})$$

$$\frac{\partial P_{km}^{c}}{\partial V_{k}^{a}} = G_{km}^{ca}V_{k}^{c}\cos(\theta_{k}^{c}-\theta_{k}^{a}) + B_{km}^{ca}V_{k}^{c}\sin(\theta_{k}^{c}-\theta_{k}^{a}) \quad \textbf{(B.28)}$$

$$\frac{\partial P_{km}^{c}}{\partial V_{k}^{b}} = G_{km}^{cb}V_{k}^{b}\cos(\theta_{k}^{c}-\theta_{k}^{b}) + B_{km}^{cb}V_{k}^{b}\sin(\theta_{k}^{c}-\theta_{k}^{b}) \quad \textbf{(B.29)}$$

$$\frac{\partial P_{km}^{c}}{\partial V_{k}^{c}} = 2.V_{k}^{c}G_{km}^{cc}0.\cos(\theta_{k}^{c}-\theta_{k}^{c}) - 2.V_{k}^{c}B_{km}^{cc}0.\sin(\theta_{k}^{c}-\theta_{k}^{c}) + G_{km}^{cc}V_{m}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{c}) \quad \textbf{(B.30)}$$
$$-B_{km}^{cc}V_{m}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{c}) + G_{km}^{ca}V_{k}^{a}\cos(\theta_{k}^{c}-\theta_{k}^{a}) + B_{km}^{ca}V_{k}^{a}\sin(\theta_{k}^{c}-\theta_{k}^{a}) - G_{km}^{ca}V_{m}^{a}\cos(\theta_{k}^{c}-\theta_{m}^{a})$$
$$-B_{km}^{ca}V_{m}^{a}\sin(\theta_{k}^{c}-\theta_{m}^{a}) + G_{km}^{cb}V_{k}^{b}\cos(\theta_{k}^{c}-\theta_{k}^{b}) + B_{km}^{cb}V_{k}^{b}\sin(\theta_{k}^{c}-\theta_{k}^{b}) - G_{km}^{cb}V_{m}^{b}\cos(\theta_{k}^{c}-\theta_{m}^{b})$$
$$-B_{km}^{cb}V_{m}^{b}\sin(\theta_{k}^{c}-\theta_{m}^{b})$$

$$\frac{\partial P_{km}^{c}}{\partial \theta_{m}^{a}} = -G_{km}^{ca}V_{k}^{c}V_{m}^{a}\sin(\theta_{k}^{c}-\theta_{m}^{a}) + B_{km}^{ca}V_{k}^{c}V_{m}^{a}\cos(\theta_{k}^{c}-\theta_{m}^{a}) \quad \textbf{(B.31)}$$

$$\frac{\partial P_{km}^{c}}{\partial \theta_{m}^{b}} = -G_{km}^{cb}V_{k}^{c}V_{m}^{b}\sin(\theta_{k}^{c}-\theta_{m}^{b}) + B_{km}^{cb}V_{k}^{c}V_{m}^{b}\cos(\theta_{k}^{c}-\theta_{m}^{b}) \quad \textbf{(B.32)}$$

$$\frac{\partial P_{km}^{c}}{\partial \theta_{m}^{c}} = -G_{km}^{cc}V_{k}^{c}V_{m}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{c}) + B_{km}^{cc}V_{k}^{c}V_{m}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{c}) \quad \textbf{(B.33)}$$

$$\frac{\partial P_{km}^{c}}{\partial V_{m}^{a}} = -G_{km}^{ca}V_{k}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{a}) - B_{km}^{ca}V_{k}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{a}) \quad \textbf{(B.34)}$$

$$\frac{\partial P_{km}^{c}}{\partial V_{m}^{b}} = -G_{km}^{cb}V_{k}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{b}) - B_{km}^{cb}V_{k}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{b}) \quad \textbf{(B.35)}$$

$$\frac{\partial P_{km}^{c}}{\partial V_{m}^{c}} = -G_{km}^{cc} V_{k}^{c} \cos(\theta_{k}^{c} - \theta_{m}^{c}) - B_{km}^{cc} V_{k}^{c} \sin(\theta_{k}^{c} - \theta_{m}^{c}) \quad \textbf{(B.36)}$$

As derivadas apresentadas nas Equações de (B.37) a (B.72) são referentes à Equação **(2.10)**

$$\begin{aligned}\frac{\partial Q_{km}^{a}}{\partial \theta_{k}^{a}} &= G_{km}^{aa} V_{k}^{a} V_{k}^{a} 0.\cos(\theta_{k}^{a} - \theta_{k}^{a}) + B_{km}^{aa} V_{k}^{a} V_{k}^{a} 0.\sin(\theta_{k}^{a} - \theta_{k}^{a}) - G_{km}^{aa} V_{k}^{a} V_{m}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{a}) \\ &- B_{km}^{aa} V_{k}^{a} V_{m}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{a}) + G_{km}^{ab} V_{k}^{a} V_{k}^{b}.\cos(\theta_{k}^{a} - \theta_{k}^{b}) + B_{km}^{ab} V_{k}^{a} V_{k}^{b}.\sin(\theta_{k}^{a} - \theta_{k}^{b}) \\ &- G_{km}^{ab} V_{k}^{a} V_{m}^{b} \cos(\theta_{k}^{a} - \theta_{m}^{b}) - B_{km}^{ab} V_{k}^{a} V_{m}^{b} \sin(\theta_{k}^{a} - \theta_{m}^{b}) + G_{km}^{ac} V_{k}^{a} V_{k}^{c}.\cos(\theta_{k}^{a} - \theta_{k}^{c}) \\ &+ B_{km}^{ac} V_{k}^{a} V_{k}^{c} \sin(\theta_{k}^{a} - \theta_{k}^{c}) - G_{km}^{ac} V_{k}^{a} V_{m}^{c} \cos(\theta_{k}^{a} - \theta_{m}^{c}) - B_{km}^{ac} V_{k}^{a} V_{m}^{c} \sin(\theta_{k}^{a} - \theta_{m}^{c})\end{aligned} \quad \textbf{(B.37)}$$

$$\frac{\partial Q_{km}^{a}}{\partial \theta_{k}^{b}} = -G_{km}^{ab} V_{k}^{a} V_{k}^{b} \cos(\theta_{k}^{a} - \theta_{k}^{b}) - B_{km}^{ab} V_{k}^{a} V_{k}^{b} \sin(\theta_{k}^{a} - \theta_{k}^{b}) \quad \textbf{(B.38)}$$

$$\frac{\partial Q_{km}^{a}}{\partial \theta_{k}^{c}} = -G_{km}^{ac} V_{k}^{a} V_{k}^{c} \cos(\theta_{k}^{a} - \theta_{k}^{c}) - B_{km}^{ac} V_{k}^{a} V_{k}^{c} \sin(\theta_{k}^{a} - \theta_{k}^{c}) \quad \textbf{(B.39)}$$

$$\begin{aligned}\frac{\partial Q_{km}^{a}}{\partial V_{k}^{a}} &= 2.V_{k}^{a} G_{km}^{aa} 0.\sin(\theta_{k}^{a} - \theta_{k}^{a}) - 2.V_{k}^{a} B_{km}^{aa} 0.\cos(\theta_{k}^{a} - \theta_{k}^{a}) - G_{km}^{aa} V_{m}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{a}) \\ &+ B_{km}^{aa} V_{m}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{a}) + G_{km}^{ab} V_{k}^{b} \sin(\theta_{k}^{a} - \theta_{k}^{b}) - B_{km}^{ab} V_{k}^{b} \cos(\theta_{k}^{a} - \theta_{k}^{b}) - G_{km}^{ab} V_{m}^{b} \sin(\theta_{k}^{a} - \theta_{m}^{b}) \\ &+ B_{km}^{ab} V_{m}^{b} \cos(\theta_{k}^{a} - \theta_{m}^{b}) + G_{km}^{ac} V_{k}^{c} \sin(\theta_{k}^{a} - \theta_{k}^{c}) - B_{km}^{ac} V_{k}^{c} \cos(\theta_{k}^{a} - \theta_{k}^{c}) - G_{km}^{ac} V_{m}^{c} \sin(\theta_{k}^{a} - \theta_{m}^{c}) \\ &+ B_{km}^{ac} V_{m}^{c} \cos(\theta_{k}^{a} - \theta_{m}^{c}) - 2Bshunt V_{k}^{a}\end{aligned} \quad \textbf{(B.40)}$$

$$\frac{\partial Q_{km}^{a}}{\partial V_{k}^{b}} = G_{km}^{ab} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{k}^{b}) - B_{km}^{ab} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{k}^{b}) \quad \textbf{(B.41)}$$

$$\frac{\partial Q_{km}^{a}}{\partial V_{k}^{c}} = G_{km}^{ac} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{k}^{c}) - B_{km}^{ac} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{k}^{c}) \quad \textbf{(B.42)}$$

$$\frac{\partial Q_{km}^{a}}{\partial \theta_{m}^{a}} = G_{km}^{aa} V_{k}^{a} V_{m}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{a}) + B_{km}^{aa} V_{k}^{a} V_{m}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{a}) \quad \textbf{(B.43)}$$

$$\frac{\partial Q_{km}^{a}}{\partial \theta_{m}^{b}} = G_{km}^{ab} V_{k}^{a} V_{m}^{b} \cos(\theta_{k}^{a} - \theta_{m}^{b}) + B_{km}^{ab} V_{k}^{a} V_{m}^{b} \sin(\theta_{k}^{a} - \theta_{m}^{b}) \quad \textbf{(B.44)}$$

$$\frac{\partial Q_{km}^{a}}{\partial \theta_{m}^{c}} = G_{km}^{ac} V_{k}^{a} V_{m}^{c} \cos(\theta_{k}^{a} - \theta_{m}^{c}) + B_{km}^{ac} V_{k}^{a} V_{m}^{c} \sin(\theta_{k}^{a} - \theta_{m}^{c}) \quad \textbf{(B.45)}$$

$$\frac{\partial Q_{km}^{a}}{\partial V_{m}^{a}} = -G_{km}^{aa} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{a}) + B_{km}^{aa} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{a}) \quad \textbf{(B.46)}$$

$$\frac{\partial Q_{km}^{a}}{\partial V_{m}^{b}} = -G_{km}^{ab} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{b}) + B_{km}^{ab} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{b}) \quad \textbf{(B.47)}$$

$$\frac{\partial Q_{km}^{a}}{\partial V_{m}^{c}} = -G_{km}^{ac} V_{k}^{a} \sin(\theta_{k}^{a} - \theta_{m}^{c}) + B_{km}^{ac} V_{k}^{a} \cos(\theta_{k}^{a} - \theta_{m}^{c}) \quad \textbf{(B.48)}$$

$$\frac{\partial Q_{km}^{b}}{\partial \theta_{k}^{a}} = -G_{km}^{ba} V_{k}^{b} V_{k}^{a} \cos(\theta_{k}^{b} - \theta_{k}^{a}) - B_{km}^{ba} V_{k}^{b} V_{k}^{a} \sin(\theta_{k}^{b} - \theta_{k}^{a}) \quad \textbf{(B.49)}$$

$$\begin{aligned}\frac{\partial Q_{km}^{b}}{\partial \theta_{k}^{b}} &= G_{km}^{bb} V_{k}^{b} V_{k}^{b} 0.\cos(\theta_{k}^{b} - \theta_{k}^{b}) + B_{km}^{bb} V_{k}^{b} V_{k}^{b} 0.\sin(\theta_{k}^{b} - \theta_{k}^{b}) - G_{km}^{bb} V_{k}^{b} V_{m}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{b}) \\ &- B_{km}^{bb} V_{k}^{b} V_{m}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{b}) + G_{km}^{ba} V_{k}^{b} V_{k}^{a}.\cos(\theta_{k}^{b} - \theta_{k}^{a}) + B_{km}^{ba} V_{k}^{b} V_{k}^{a}.\sin(\theta_{k}^{b} - \theta_{k}^{a}) \\ &- G_{km}^{ba} V_{k}^{b} V_{m}^{a} \cos(\theta_{k}^{b} - \theta_{m}^{a}) - B_{km}^{ba} V_{k}^{b} V_{m}^{a} \sin(\theta_{k}^{b} - \theta_{m}^{a}) + G_{km}^{bc} V_{k}^{b} V_{k}^{c}.\cos(\theta_{k}^{b} - \theta_{k}^{c}) \\ &+ B_{km}^{bc} V_{k}^{b} V_{k}^{c} \sin(\theta_{k}^{b} - \theta_{k}^{c}) - G_{km}^{bc} V_{k}^{b} V_{m}^{c} \cos(\theta_{k}^{b} - \theta_{m}^{c}) - B_{km}^{bc} V_{k}^{b} V_{m}^{c} \sin(\theta_{k}^{b} - \theta_{m}^{c})\end{aligned} \quad \textbf{(B.50)}$$

$$\frac{\partial Q_{km}^{b}}{\partial \theta_{k}^{c}} = -G_{km}^{bc} V_{k}^{b} V_{k}^{c} \cos(\theta_{k}^{b} - \theta_{k}^{c}) - B_{km}^{bc} V_{k}^{b} V_{k}^{c} \sin(\theta_{k}^{b} - \theta_{k}^{c}) \quad \textbf{(B.51)}$$

$$\frac{\partial Q_{km}^{b}}{\partial V_{k}^{a}} = G_{km}^{ba} V_{k}^{b} \sin(\theta_{k}^{b} - \theta_{k}^{a}) - B_{km}^{ba} V_{k}^{b} \cos(\theta_{k}^{b} - \theta_{k}^{a}) \quad \textbf{(B.52)}$$

$$\begin{aligned}\frac{\partial Q_{km}^{b}}{\partial V_{k}^{b}} &= 2.V_{k}^{b} G_{km}^{bb} 0.\sin(\theta_{k}^{b} - \theta_{k}^{b}) - 2.V_{k}^{b} B_{km}^{bb} 0.\cos(\theta_{k}^{b} - \theta_{k}^{b}) - G_{km}^{bb} V_{m}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{b}) \\ &+ B_{km}^{bb} V_{m}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{b}) + G_{km}^{ba} V_{k}^{a} \sin(\theta_{k}^{b} - \theta_{k}^{a}) - B_{km}^{ba} V_{k}^{a} \cos(\theta_{k}^{b} - \theta_{k}^{a}) - G_{km}^{ba} V_{m}^{a} \sin(\theta_{k}^{b} - \theta_{m}^{a}) \\ &+ B_{km}^{ba} V_{m}^{a} \cos(\theta_{k}^{b} - \theta_{m}^{a}) + G_{km}^{bc} V_{k}^{c} \sin(\theta_{k}^{b} - \theta_{k}^{c}) - B_{km}^{bc} V_{k}^{c} \cos(\theta_{k}^{b} - \theta_{k}^{c}) - G_{km}^{bc} V_{m}^{c} \sin(\theta_{k}^{b} - \theta_{m}^{c}) \\ &+ B_{km}^{bc} V_{m}^{c} \cos(\theta_{k}^{b} - \theta_{m}^{c}) - 2Bshunt V_{k}^{b}\end{aligned} \quad \textbf{(B.53)}$$

$$\frac{\partial Q_{km}^{b}}{\partial V_{k}^{c}} = G_{km}^{bc} V_{k}^{b} \sin(\theta_{k}^{b} - \theta_{k}^{c}) - B_{km}^{bc} V_{k}^{b} \cos(\theta_{k}^{b} - \theta_{k}^{c}) \quad \textbf{(B.54)}$$

$$\frac{\partial Q_{km}^{b}}{\partial \theta_{m}^{a}} = G_{km}^{ba} V_{k}^{b} V_{m}^{a} \cos(\theta_{k}^{b} - \theta_{m}^{a}) + B_{km}^{ba} V_{k}^{b} V_{m}^{a} \sin(\theta_{k}^{b} - \theta_{m}^{a}) \quad \textbf{(B.55)}$$

$$\frac{\partial Q_{km}^{b}}{\partial \theta_{m}^{b}} = G_{km}^{bb} V_{k}^{b} V_{m}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{b}) + B_{km}^{bb} V_{k}^{b} V_{m}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{b}) \quad \textbf{(B.56)}$$

$$\frac{\partial Q_{km}^{b}}{\partial \theta_{m}^{c}} = G_{km}^{bc} V_{k}^{b} V_{m}^{c} \cos(\theta_{k}^{b} - \theta_{m}^{c}) + B_{km}^{bc} V_{k}^{b} V_{m}^{c} \sin(\theta_{k}^{b} - \theta_{m}^{c}) \quad \textbf{(B.57)}$$

$$\frac{\partial Q_{km}^{b}}{\partial V_{m}^{a}} = -G_{km}^{ba} V_{k}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{a}) + B_{km}^{ba} V_{k}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{a}) \quad \textbf{(B.58)}$$

$$\frac{\partial Q_{km}^{b}}{\partial V_{m}^{b}} = -G_{km}^{bb} V_{k}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{b}) + B_{km}^{bb} V_{k}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{b}) \quad \textbf{(B.59)}$$

$$\frac{\partial Q_{km}^{b}}{\partial V_{m}^{c}} = -G_{km}^{bc} V_{k}^{b} \sin(\theta_{k}^{b} - \theta_{m}^{c}) + B_{km}^{bc} V_{k}^{b} \cos(\theta_{k}^{b} - \theta_{m}^{c}) \quad \textbf{(B.60)}$$

$$\frac{\partial Q_{km}^{c}}{\partial \theta_{k}^{a}} = -G_{km}^{ca} V_{k}^{c} V_{k}^{a} \cos(\theta_{k}^{c} - \theta_{k}^{a}) - B_{km}^{ca} V_{k}^{c} V_{k}^{a} \sin(\theta_{k}^{c} - \theta_{k}^{a}) \quad \textbf{(B.61)}$$

$$\frac{\partial Q_{km}^{c}}{\partial \theta_{k}^{b}} = -G_{km}^{cb} V_{k}^{c} V_{k}^{b} \cos(\theta_{k}^{c} - \theta_{k}^{b}) - B_{km}^{cb} V_{k}^{c} V_{k}^{b} \sin(\theta_{k}^{c} - \theta_{k}^{b}) \quad \textbf{(B.62)}$$

$$\begin{aligned}\frac{\partial Q_{km}^{c}}{\partial \theta_{k}^{c}} &= G_{km}^{cc} V_{k}^{c} V_{k}^{c} 0.\cos(\theta_{k}^{c} - \theta_{k}^{c}) + B_{km}^{cc} V_{k}^{c} V_{k}^{c} 0.\sin(\theta_{k}^{c} - \theta_{k}^{c}) - G_{km}^{cc} V_{k}^{c} V_{m}^{c} \cos(\theta_{k}^{c} - \theta_{m}^{c}) \quad \textbf{(B.63)}\\ &- B_{km}^{cc} V_{k}^{c} V_{m}^{c} \sin(\theta_{k}^{c} - \theta_{m}^{c}) + G_{km}^{ca} V_{k}^{c} V_{k}^{a}.\cos(\theta_{k}^{c} - \theta_{k}^{a}) + B_{km}^{ca} V_{k}^{c} V_{k}^{a}.\sin(\theta_{k}^{c} - \theta_{k}^{a})\\ &- G_{km}^{ca} V_{k}^{c} V_{m}^{a} \cos(\theta_{k}^{c} - \theta_{m}^{a}) - B_{km}^{ca} V_{k}^{c} V_{m}^{a} \sin(\theta_{k}^{c} - \theta_{m}^{a}) + G_{km}^{cb} V_{k}^{c} V_{k}^{b}.\cos(\theta_{k}^{c} - \theta_{k}^{b})\\ &+ B_{km}^{cb} V_{k}^{c} V_{k}^{b} \sin(\theta_{k}^{c} - \theta_{k}^{b}) - G_{km}^{cb} V_{k}^{c} V_{m}^{b} \cos(\theta_{k}^{c} - \theta_{m}^{b}) - B_{km}^{cb} V_{k}^{c} V_{m}^{b} \sin(\theta_{k}^{c} - \theta_{m}^{b})\end{aligned}$$

$$\frac{\partial Q_{km}^{c}}{\partial V_{k}^{a}} = G_{km}^{ca} V_{k}^{c} \sin(\theta_{k}^{c} - \theta_{k}^{a}) - B_{km}^{ca} V_{k}^{c} \cos(\theta_{k}^{c} - \theta_{k}^{a}) \quad \textbf{(B.64)}$$

$$\frac{\partial Q_{km}^{c}}{\partial V_{k}^{b}} = G_{km}^{cb} V_{k}^{c} \sin(\theta_{k}^{c} - \theta_{k}^{b}) - B_{km}^{cb} V_{k}^{c} \cos(\theta_{k}^{c} - \theta_{k}^{b}) \quad \textbf{(B.65)}$$

$$\frac{\partial Q_{km}^{c}}{\partial V_{k}^{c}} = 2.V_{k}^{c}G_{km}^{cc}0.\sin(\theta_{k}^{c}-\theta_{k}^{c}) - 2.V_{k}^{c}B_{km}^{cc}0.\cos(\theta_{k}^{c}-\theta_{k}^{c}) - G_{km}^{cc}V_{m}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{c}) \quad \textbf{(B.66)}$$
$$+B_{km}^{cc}V_{m}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{c}) + G_{km}^{ca}V_{k}^{a}\sin(\theta_{k}^{c}-\theta_{k}^{a}) - B_{km}^{ca}V_{k}^{a}\cos(\theta_{k}^{c}-\theta_{k}^{a}) - G_{km}^{ca}V_{m}^{a}\sin(\theta_{k}^{c}-\theta_{m}^{a})$$
$$+B_{km}^{ca}V_{m}^{a}\cos(\theta_{k}^{c}-\theta_{m}^{a}) + G_{km}^{cb}V_{k}^{b}\sin(\theta_{k}^{c}-\theta_{k}^{b}) - B_{km}^{cb}V_{k}^{b}\cos(\theta_{k}^{c}-\theta_{k}^{b}) - G_{km}^{cb}V_{m}^{b}\sin(\theta_{k}^{c}-\theta_{m}^{b})$$
$$+B_{km}^{cb}V_{m}^{b}\cos(\theta_{k}^{c}-\theta_{m}^{b}) - 2BshuntV_{k}^{c}$$

$$\frac{\partial Q_{km}^{c}}{\partial \theta_{m}^{a}} = G_{km}^{ca}V_{k}^{c}V_{m}^{a}\cos(\theta_{k}^{c}-\theta_{m}^{a}) + B_{km}^{ca}V_{k}^{c}V_{m}^{a}\sin(\theta_{k}^{c}-\theta_{m}^{a}) \quad \textbf{(B.67)}$$

$$\frac{\partial Q_{km}^{c}}{\partial \theta_{m}^{b}} = G_{km}^{cb}V_{k}^{c}V_{m}^{b}\cos(\theta_{k}^{c}-\theta_{m}^{b}) + B_{km}^{cb}V_{k}^{c}V_{m}^{b}\sin(\theta_{k}^{c}-\theta_{m}^{b}) \quad \textbf{(B.68)}$$

$$\frac{\partial Q_{km}^{c}}{\partial \theta_{m}^{c}} = G_{km}^{cc}V_{k}^{c}V_{m}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{c}) + B_{km}^{cc}V_{k}^{c}V_{m}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{c}) \quad \textbf{(B.69)}$$

$$\frac{\partial Q_{km}^{c}}{\partial V_{m}^{a}} = -G_{km}^{ca}V_{k}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{a}) + B_{km}^{ca}V_{k}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{a}) \quad \textbf{(B.70)}$$

$$\frac{\partial Q_{km}^{c}}{\partial V_{m}^{b}} = -G_{km}^{cb}V_{k}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{b}) + B_{km}^{cb}V_{k}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{b}) \quad \textbf{(B.71)}$$

$$\frac{\partial Q_{km}^{c}}{\partial V_{m}^{c}} = -G_{km}^{cc}V_{k}^{c}\sin(\theta_{k}^{c}-\theta_{m}^{c}) + B_{km}^{c}V_{k}^{c}\cos(\theta_{k}^{c}-\theta_{m}^{c}) \quad \textbf{(B.72)}$$

Apêndice C Biblioteca Gráfica OpenGL

C.1 Introdução

A biblioteca OpenGL (Open Graphic Library) é um conjunto de funções gráficas 3D de alto desempenho e de domínio público. Esta biblioteca comunica-se diretamente com o hardware da placa de vídeo, evitando que o sistema operacional interfira nessa tarefa. Com a diminuição dos custos destas placas, a utilização do OpenGL pelos desenvolvedores de aplicações gráficas vem se tornando cada vez mais comum.

O precursor do OpenGL foi o GL da Silicon Graphics Inc. (SGI), uma renomada empresa líder mundial em computação gráfica e animação. A "IRIS GL" foi a linguagem de programação 3D das estações gráficas IRIS. Estes computadores eram especialmente otimizados para exibição e criação de sofisticados gráficos, onde o próprio hardware provinha matrizes de transformação muito rápidas (o que é pré-requisito para gráficos tridimensionais), suporte de hardware para buffer de profundidade e outras características. Mas quando tentaram passar "IRIS GL" para outras plataformas de hardware, ocorreram vários problemas de compatibilidade.(NEIDER e DAVIS,1996)

A primeira tentativa de implementar uma tecnologia semelhante a OpenGL em PCs, foi feita pela Microsoft. A empresa criou a GDI (Graphics Device Interface), com a qual era possível escrever gráficos independente do hardware utilizado, porém isso levava um certo tempo. Então os fabricantes de placas começaram a escrever drivers otimizados para a GDI. A partir daí, a Microsoft introduziu o WinG que consistia em algumas poucas funções que exibiam bitmaps no monitor, porém ainda muito lento. Após isso, a Microsoft criou o Direct Draw API para acesso ao hardware em baixo nível. Este se tornou parte das APIs do DirectX para acesso direto ao hardware, fazendo com que, por exemplo, jogos fossem facilmente implementados e melhorando muito seu desempenho, porém apresentavam alguma lentidão.(WRIGHT e SWEET, 1996)

O OpenGL é o resultado do esforço da SGI em melhorar a portabilidade do "IRIS GL". Esta nova linguagem deveria ser poderosa como o GL e ser "OPEN", permitindo assim portabilidade para outras plataformas de hardware. Porém, um padrão aberto não é realmente aberto se somente um fabricante o controla. Assim todas as características do OpenGL são decididas pelo associação OpenGL Architeture Review

Board (ARB), que foi fundada pelos membros da SGI, Digital Equipment Corporation, IBM, Intel e Microsoft. O OpenGL ARB reune duas vezes por ano. Essas reuniões são abertas ao público e as companhias que não fazem parte da associação podem participar, dar sugestões, apesar de não terem direito a voto.(NEIDER e DAVIS,1996)

A OpenGL é definida como "uma interface de software para um hardware gráfico". Em essência é uma biblioteca de gráficos tridimensionais, modelos extremamente portáveis e muito rápida. Usando o OpenGL, pode-se criar gráficos tridimensionais com grande qualidade visual como pode ser visto na Figura C.1.

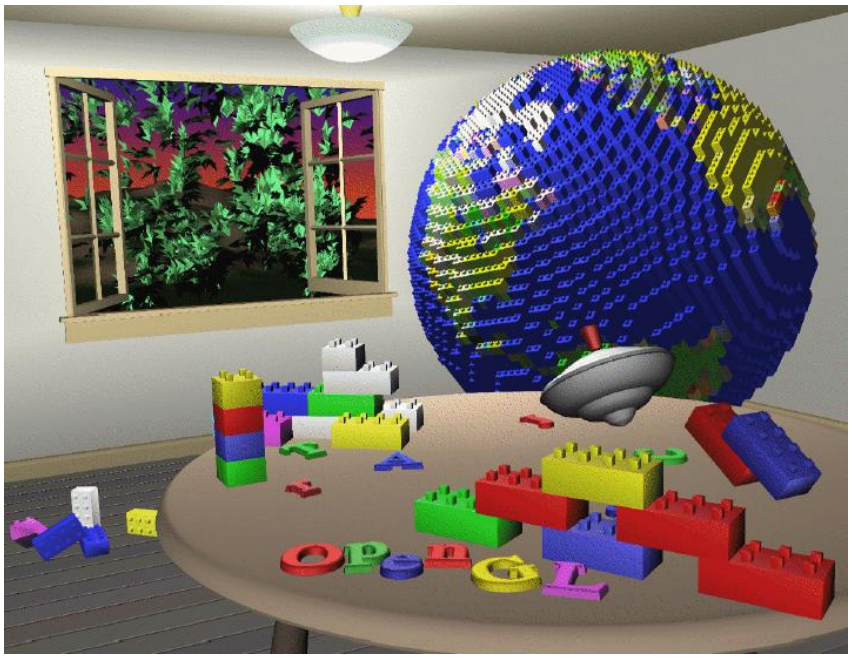

Figura C.1- Exemplo de gráfico produzido em OpenGL

A biblioteca foi implementada para ser usado em computadores com hardware gráfico desenvolvido e otimizado para a exibição e manipulação de gráficos 3D. Implementações somente de software, sem a utilização de hardware específico são possíveis, porém o desempenho computacional diminui consideravelmente. As implementações do Microsoft Windows caem nessa categoria. Com a queda de preço das placas de vídeo com aceleração 3D, elas vêm se tornando um equipamento comum nos PCs No entanto, as aplicações OpenGL não distinguem entre a implementação por hardware acelerado ou por software. O usuário nota um ganho de desempenho incrível quando a aceleração de hardware está presente.

O OpenGL é uma linguagem de procedimentos e não descritiva. Ao invés de descrever a cena e como ela deve aparecer, o programador descreve os passos

necessários para executar um objeto ou um efeito. Estes passos envolvem a chamada de mais de 120 funções de alto nível. Estas funções são usadas para desenhar gráficos primitivos, como pontos, linhas, polígonos em três dimensões. Além disso, o OpenGL suporta efeitos de luz, sombra, uso de texturas, animação e outros efeitos especiais. Porém, o OpenGL não inclui nenhuma função para o gerenciamento de janelas, interação com o usuário, entrada ou saída de dados. Cada plataforma (como por exemplo o Windows) tem suas próprias funções para esse fim.

Entre os recursos gráficos disponíveis no OpenGL, podem ser destacados os seguintes:

- Modos de desenho de pontos;
- Ajuste de largura de linhas;
- Aplicação de transparência;
- Ativação/desativação de serrilhamento (aliasing);
- Mapeamento de superfícies com textura;
- Seleção de janela de desenho;
- Manipulação de fontes/tipos de iluminação e sombreamento;
- Transformação de sistemas de coordenadas;
- Transformações em perspectiva;
- Combinação de imagens (blending).

C.2 Operação da OpenGL

A Figura C.2 descreve as operações gráficas que a OpenGL usa para desenhar uma imagem na tela.

Apesar de executar gráficos de grande complexidade, a estrutura básica do programa é simples. Deve-se inicializar certos controles de como o OpenGL desenha uma imagem e especificar os objetos a serem desenhados. A complexidade consiste nas formulações matemáticas dos efeitos apresentados na tela.

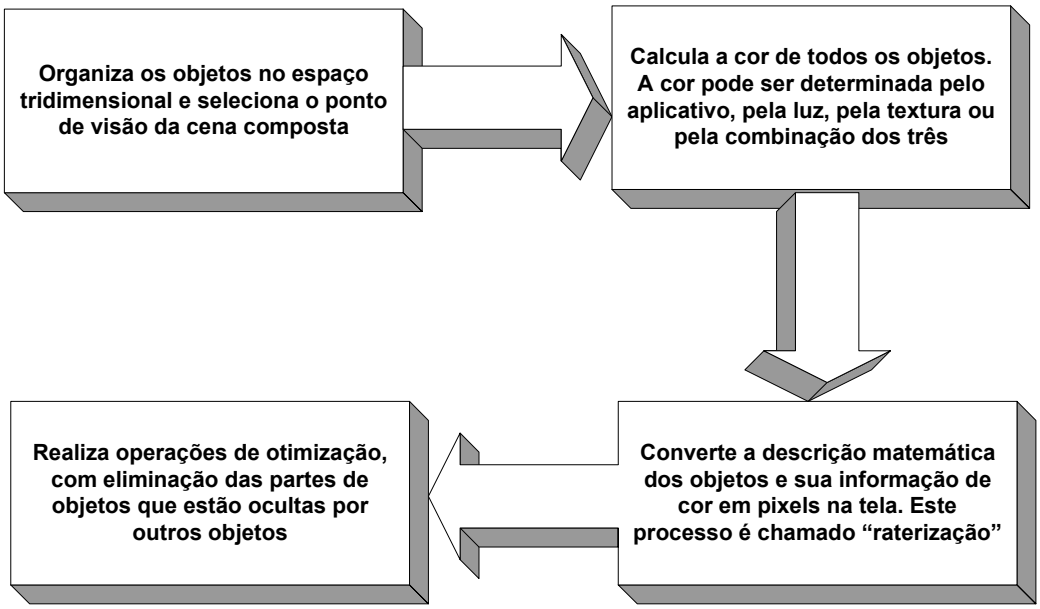

Figura C.2 - Operações gráficas da OpenGL

Abaixo, pode-se ver o código para desenhar um retângulo branco com o fundo preto.

```
#include  OpenGL.h  //inclui a OpenGL no projeto
main( )
{
        InicializaJanela( );   //Inicializa a janela
        glClearColor( 0.0, 0.0, 0.0, 0.0);  //Seleciona a cor de fundo
        glClear(GL_COLOR_BUFFER_BIT);  //limpa a tela
        glColor3f(1.0 ,1.0 , 1.0); //seleciona a cor atual
        glOrtho(0.0 , 1.0 , 0.0 , 1.0 , -1.0 , 1.0 ); //escolhe o método de projeção ortogonal
        glBegin(GL_POLYGON); //comando para desenhar um polígono
                glVertex3f (0.25, 0.25, 0.0); // 1º vértice
                glVertex3f (0.75, 0.25, 0.0); // 2º vértice
                glVertex3f (0.75, 0.75, 0.0); //3º vértice
                glVertex3f (0.25, 0.75, 0.0); //4º vértice
        glEnd(); //termina o comando de desenho
        AtualizaJanela( ); //atualiza os dados da janela
}
```

C.3 Conclusão

Diante das funcionalidades providas pelo OpenGL, a facilidade de aprendizado, a estabilidade das rotinas e pelos resultados visuais consistentes para qualquer sistema de exibição concordante com este padrão, tal biblioteca tem se tornado um padrão amplamente utilizado na indústria de desenvolvimento de aplicações. Diversos jogos, aplicações científicas e comerciais têm utilizado OpenGL como ferramenta de apresentação de recursos visuais, principalmente com a adoção deste padrão por parte dos fabricantes de placas de vídeo destinadas aos consumidores domésticos.

Apêndice D Aplicação Desenvolvida

A aplicação desenvolvida se apresenta na Figura D.1. Ela foi concebida baseando-se no consagrado programa de simulação de circuitos PSPICE, um software de grande familiaridade para estudantes e engenheiros, a fim de facilitar sua utilização e tornar sua interface mais amigável.

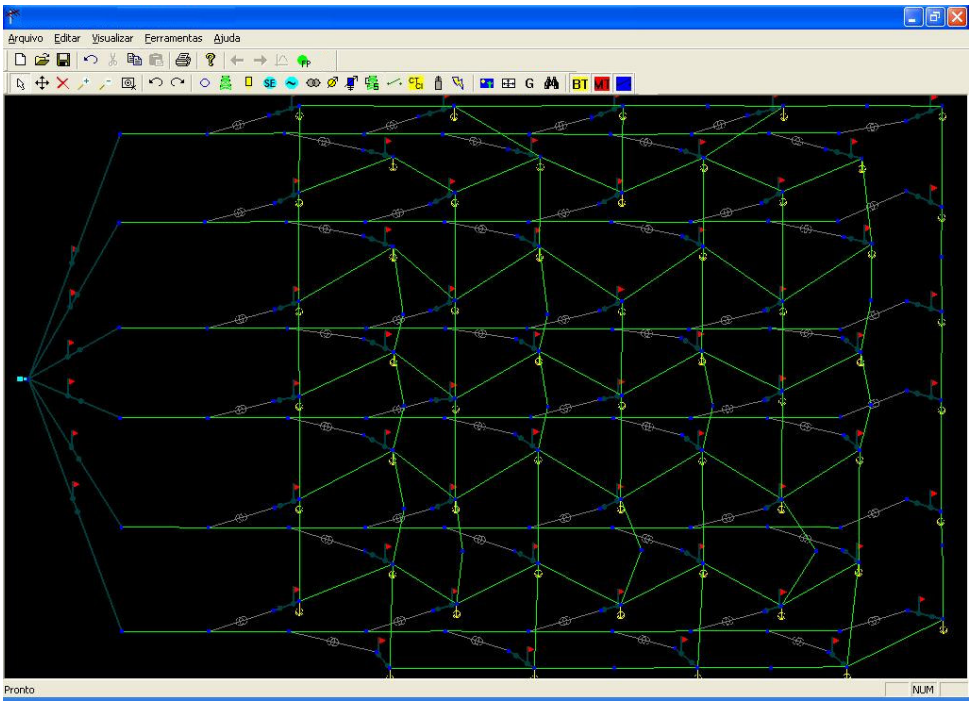

Figura D.1 - Aplicação desenvolvida utilizando OpenGL

Na Figura D.2 e na Figura D.3 ficam caracterizadas as barras de ferramentas do aplicativo com suas devidas funções.

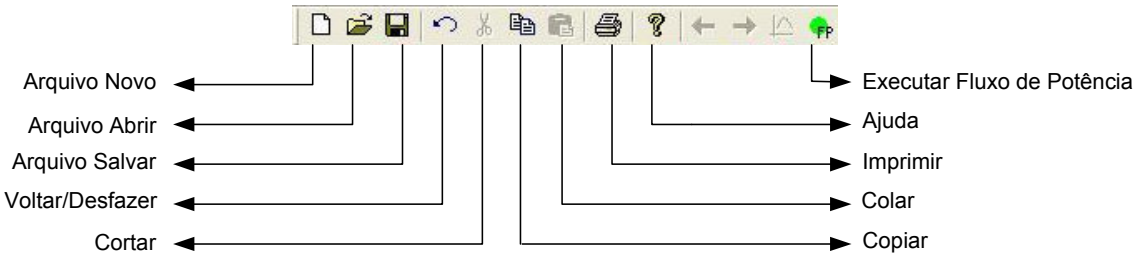

Figura D.2 - Barra de ferramentas superior do aplicativo

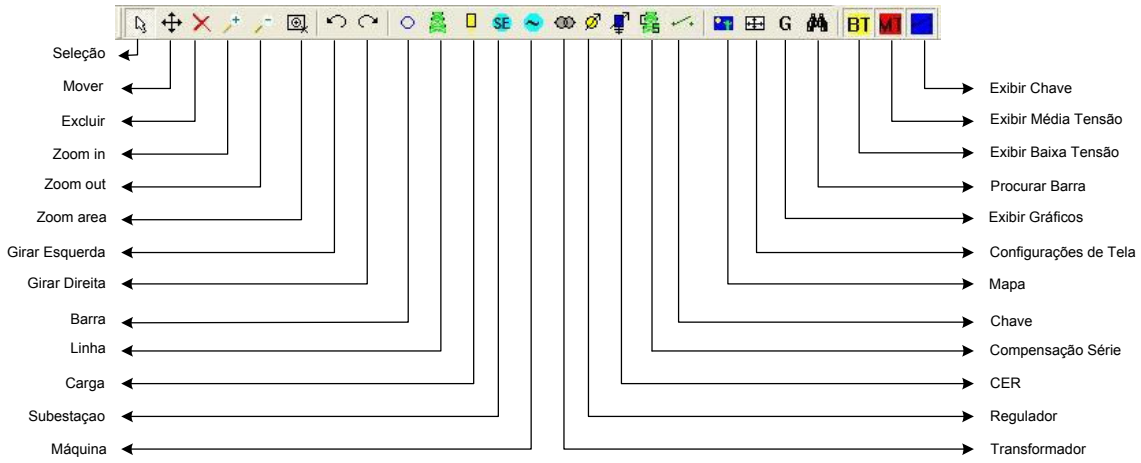

Figura D.3 - Barra de ferramentas inferior do aplicativo

Ao iniciar o programa, deve-se clicar no menu de configurações de tela e configurar o tamanho da tela e as coordenadas UTM dos extremos desta área de trabalho (Figura D.4). A partir desse ponto, o aplicativo está pronto para ser utilizado. Todos elementos que serão inseridos no aplicativo já tem suas coordenadas referenciadas as coordenadas UTM definidas anteriormente.

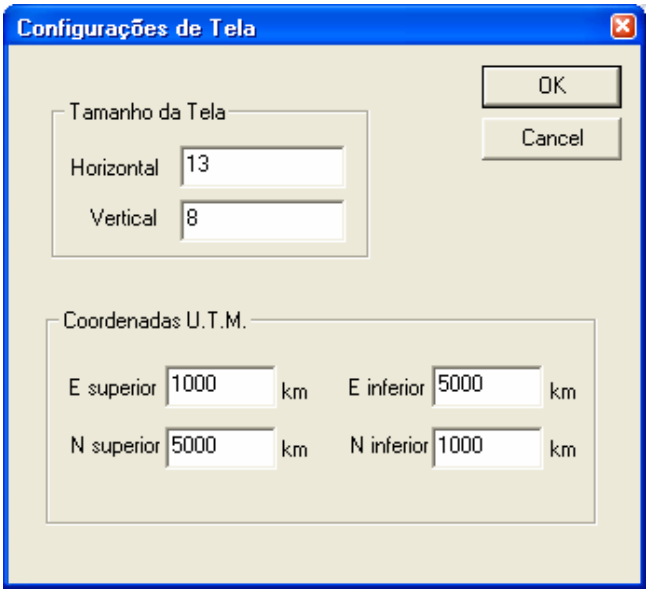

Figura D.4 - Caixa de diálogo de configuração da tela do programa

Para se inserir um elemento do sistema elétrico de potência basta selecionar o mesmo na barra de ferramentas e clicar na posição da tela onde se deseja inserí-lo. No caso das linhas, geradores e cargas, eles só podem ser inseridos nos locais onde existam barras. Os elementos se apresentam no aplicativo como mostra a Figura D.5

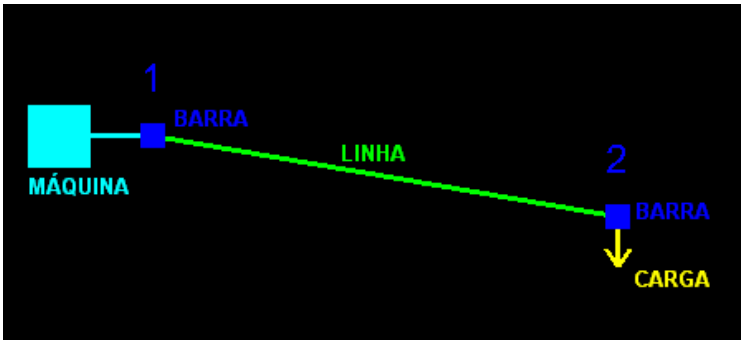

Figura D.5 - Representação dos elementos no aplicativo

Para a inserção de plantas, deve-se clicar no botão de mapas. A caixa de diálogo da Figura D.6 irá aparecer. Deve entrar com o caminho do arquivo de imagem contendo o mapa, no formato BMP (Bitmap) e as coordenadas dos extremos do mapa a ser inserido. Clicando em OK o mesmo é inserido na posição especificada pelas coordenadas UTM como pode ser visto na Figura D.7.

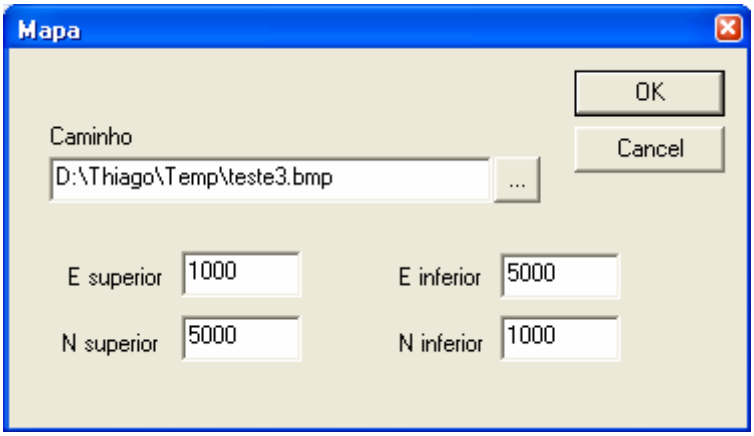

Figura D.6 - Caixa de diálogo de inserção de mapas

Vale ressaltar que se entrarmos nas configurações de tela e mudarmos a referencia das coordenadas UTM todos os elementos inseridos no aplicativo, inclusive o mapa, mudarão de posição pois estão referenciados no sistema de coordenadas UTM

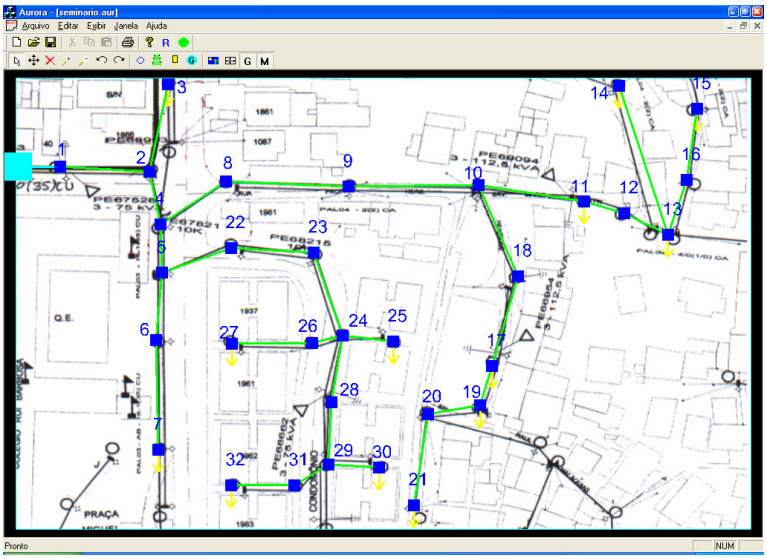

Figura D.7 – Mapa Inserido na Aplicação

Para a edição dos dados elétricos da rede, deve selecionar na barra de ferramentas a opção *Ferramenta de Seleção* e logo após dar um duplo clique em cima do elemento que se deseja editar.

No caso das barras, deve-se entrar com o valor inicial das tensões e ângulos em todas as fases e os limites superior e inferior de tensão como pode ser visto na Figura D.8

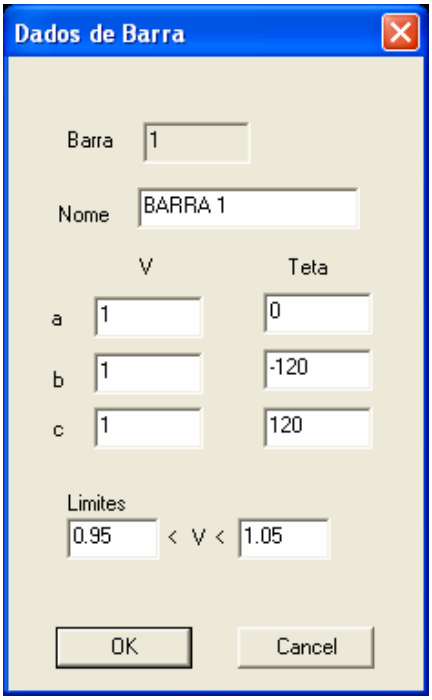

Figura D.8 - Caixa de diálogo de edição dos dados de barra

Na Figura D.9 pode-se ver a caixa de diálogo de edição dos dados de linha. Para esse tipo de elemento deve-se entrar com a matriz de reatância de linha, o valor dos shunts e o limite superior de potência ativa na linha. Na parte de fluxos, o programa externa o valor do fluxo de potência na linha depois de executado o fluxo de carga.

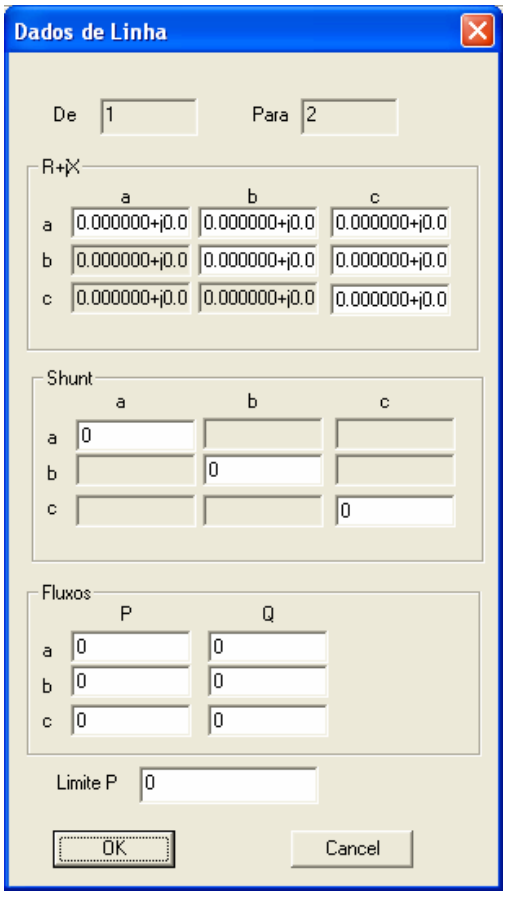

Figura D.9 - Caixa de diálogo de edição dos dados de linha

No caso das cargas, caixa de diálogo da Figura D.10 se apresenta onde pode-se editar os dados de potência ativa e reativa de cada fase, as porcentagens de cada componente no modelo ZIP e o tipo de conexão da carga (Estrela ou Delta)

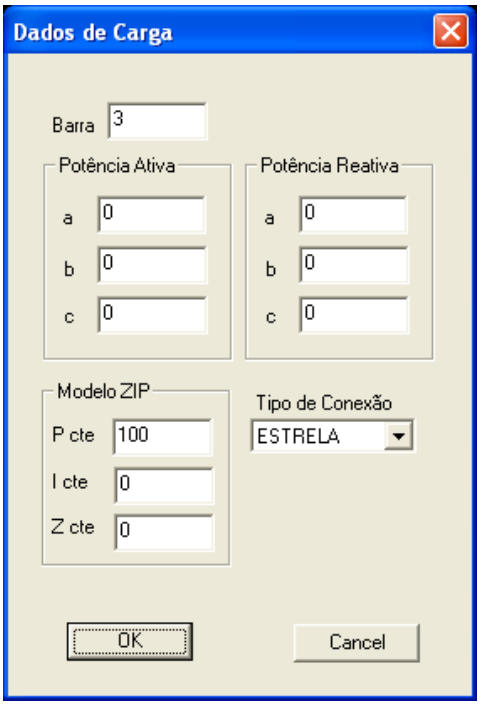

Figura D.10 - Caixa de diálogo de edição dos dados de carga

Na Figura D.11 pode-se ver a caixa de diálogo para edição dos dados dos geradores. Neste caixa, deve-se configurar o tipo do gerador (Referência, Fixo ou Controle), a potência ativa e reativa fornecida pelo gerador, os limites de potência ativa e reativa, a barra a ser controlada e o valor da tensão controlada (no caso do gerador ser de controle).

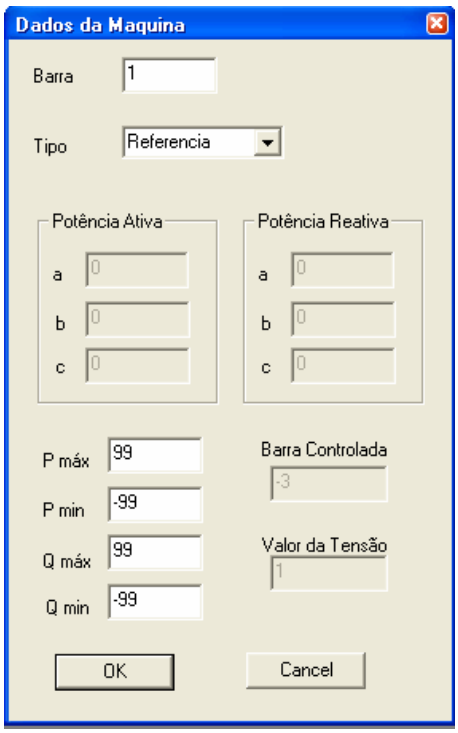

Figura D.11 - Caixa de diálogo de edição dos dados das máquinas

Depois de montada a rede e editado todos os dados elétrico, pode-se executar o fluxo de potência. Para executar esta tarefa basta clicarmos no botão de execução do fluxo. Após este procedimento, o aplicativo retorna uma caixa de diálogo dizendo se o fluxo convergiu ou não e número de iterações, como pode ser visto na Figura D.12.

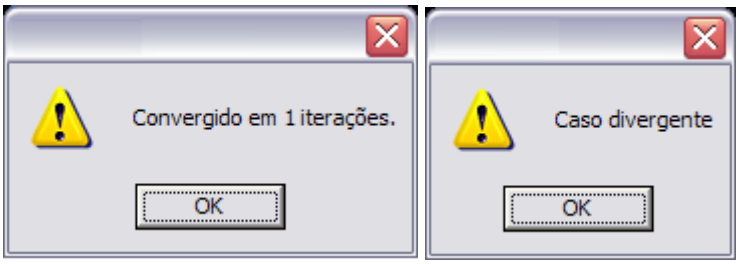

Figura D.12 - Caixas de diálogo de fluxo convergido de Caso Divergente

Os resultados podem ser vistos dando um duplo clique em cada elemento ou através do botão relatório na barra de ferramentas. Os resultados também podem ser visto através da mudança de cores dos elementos. Se um elemento exceder o limite especificado nos seus dados, sua cor muda para vermelho e se uma determinada grandeza fica com um valor abaixo de um limite especificado, sua cor muda para laranja como pode ser visto na Figura D.13

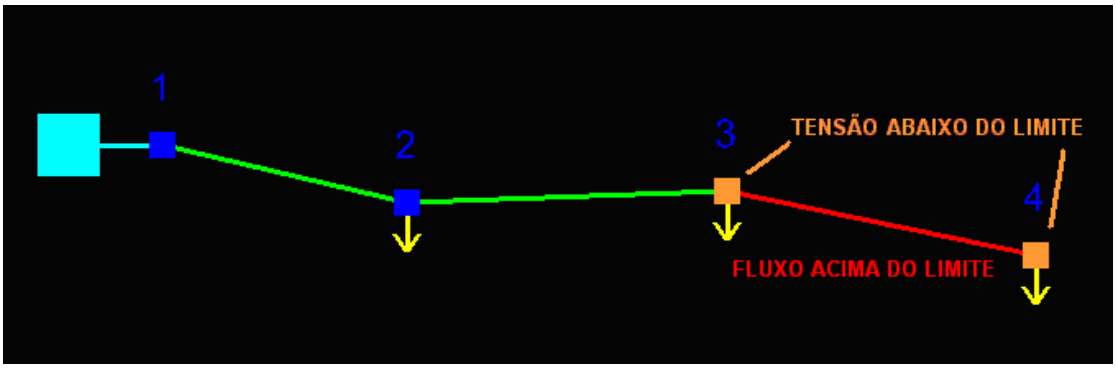

Figura D.13 - Visualização dos resultados na tela

Essa mudança de cor dos elementos permite uma rápida visualização dos elementos da rede que estão com algum problema, permitindo ao usuário do software uma rápida análise e interpretação dos resultados.

Bibliografia

ABORESHALD, S., BASOUDAN, O., 1999, "Graphical User Interface for Teaching Power System Reliability", Canadian Conference on Electrical and Computer Engineering, Alberta, Canada, 1999

AGOSTINI M. N., DECKER I. C., SILVA A. S., 2000, "Desenvolvimento e Implementação de uma Base Computacional Orientada a objetos para aplicações em sistemas de energia elétrica", XIII Congresso Brasileiro de Automática, Florianópolis, Brasil, 2000.

ANDERSON P. M., 1995, "Analysis of Faulted Power Systems", IEEE Press Power Systems Engineering Series.

ARAUJO L. R., 2000, "Técnicas de Programação Esparsa Utilizando Modelagem Orientada a Objetos", Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil

ARAUJO, L. R., GARCIA, P. A. N., PEREIRA J. L. R., CARNEIRO, S. e VINAGRE M. P., 2002, "Modelagem Orientada a Objetos Aplicada Na Solução de Programas de Distribuição", XIV Congresso Brasileiro de Automática, Natal, Brasil, 2002.

ARAUJO, L. R, 2005, "Aplicação De Fluxo De Potência Ótimo A Sistemas Trifásicos Usando O Método Dos Pontos Interiores", Tese de D.Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil

ARRILLAGA, J., 1983, "Computer Modeling of Electrical Power Systems" John Wiley and Sons Ltd , West Sussex, England

BIJWE, P. R., KELAPURE, S. M., 2003, "Nondivergent Fast Power Flow Methods", IEEE Transactions on Power Systems, v. 18, n. 2, pp. 633 – 638, May 2003.

BIRT, K. A., GRAFFY, J. J., McDONALD, J. D., 1976, "Three phase load flow program", IEEE Transactions on Power Apparatus and Systems, PAS-95, January 1976.

CÉSPEDES, G., 1990, "New Method for the Analysis of Distribution Networks", IEEE Transactions on Power Delivery, v. 5, n. 1, pp. 391 – 396, January 1990.

CHEN, T. H., et al., 1991, "Distribution System Power Flow Analysis – A Rigid Approach*", IEEE Transactions on Power Delivery*, v. 6, n. 3, pp. 1146 – 1152, July 1991.

CHEN, T., YANG, W., 2001, "Analysis of Multi-Grounded Four-Wire Distribution Systems Considering the Neutral Grounding", IEEE Transactions on Power Delivery, v. 16, n. 4, pp. 710 – 717, October 2001.

CHIANG, H. D., 1991, "A Decoupled Load Flow Method for Distribution Power Networks", Electrical Power and Energy Systems, v. 13, n. 3, pp. 130 – 138, June 1991.

CIRIC, R. M., FELTRIN, A. P., OCHOA, L. F., 2003, "Power Flow in Four-Wire Distribution Networks – General Approach", *IEEE Transactions on Power Systems,* v. 18, n. 4, pp. 1283 – 1290, November 2003.

COSTA V. M., MARTINS N., PEREIRA J. L. R., 1999, "Developments in the Newton Raphson Power Flow Formulation Based on Current Injections", IEEE Transactions on Power Systems, v.14, n. 4, pp. 1320-1336, November. 1999

DAS, D., KOTHARI, D. P., KALAM, A., 1995, "A Simple and Efficient Method for Load Flow Solution of Radial Distribution Networks", Electrical Power and Energy Systems, v. 17, n. 5, pp. 335 – 346, 1995.

ESQUIVEL C. F., et al., 1998, "Efficient Object Oriented Power Systems Software for the Analysis of Large-scale Networks Containing FACTS-Controlled Branches", IEEE Transactions on Power Apparatus and Systems, v. 13, n. 2, pp. 464–472, May 1998.

Gomez Esposito, A., Romero Ramos, E. , 1999, "Reliable Load Flow Technique for Radial Distribution Networks", IEEE Transactions on Power Systems, v. 14, n. 3, pp. 1063 – 1069, August 1999.

Flinn, D. G., 1995, "Engineering Analysis Takes an Interacive View", IEEE Computer Applications in Power, pp. 39 – 42, October 1995.

FOLEY,M., Bose, A., Mitchell, A., Faustini, A., 1993, "An Object Based Graphical User Interface for Power Systems", IEEE Transactions on Power Systems, v. 8, n. 1, pp. 97 – 104, February 1993.

GARCIA, P. A. N, 2001, "Cálculo do Fluxo de Potência Trifásico em Sistemas de Distribuição Incluindo a Representação de Dispositivos de Controle", Tese de D.Sc., Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil

GARCIA P. A. N., ARAUJO, L. R., PEREIRA J. L. R., CARNEIRO Jr. S., PINTO, H. C. P., 2000b, "Fluxo de Potência Trifásico Para Sistemas de Transmissão e Distribuição", XIV SENDI, 19 e 23 de Novembro de 2000, Foz do Iguaçu

GARCIA P. A. N., PEREIRA J. L. R., CARNEIRO Jr. S., COSTA V. M., MARTINS N., 2000, "Three-phase Power Flow Calculations Using the Current Injection Method", IEEE Transactions on Power Systems, v.15, n. 2, pp. 508-514, May. 2000

GARCIA, A. V., ZAGO, M. G., 1996, "Three-Phase Fast Decoupled Power Flow for Distribution Networks", IEEE Proceedings Generation, Transmission and Distribution, v. 142, n. 2, pp. 188 – 192, March 1996.

GOMES, F. V., 2000, "Fluxo de Potência em Sistemas que Apresentam Linhas de Potência Natural Elevada", Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil.

HAKAVIK B., HOLEN A. T., (1994). Power System Modeling and Sparse Matrix Operations Using Object-Oriented Programming, IEEE Transactions on Power Apparatus and Systems, Vol. 9, No. 2, May 1994, pp. 1045−1051.

HATZIARGYRIOU, N., JASZCZYNSKI, J., ATSAVES, G., AGOTIS, D., 1998, "Building GUIs for Interactive Network Analysis", IEEE Computer Applications in Power, pp. 46 – 51, October 1998.

HORTON, I., 1998, "Beginning Visual C++ 6", Wrox Press, First Edition

JONES, M. P., 2000, "Fundamentals of Object-Oriented Design in UML", New York, Addison-Wesley object technology series, 2000

KERSTING, W. H., 2000, "Radial Distribution Test Feeders", Distribution System Analysis Subcommittee Report, http://ewh.ieee.org/soc/dsacom/testfeeders.html

KERSTING, W. H., MENDIVE, D. L., 1976, "An Application of Ladder Network Theory to the Solution of Three-Phase Radial Load Flow Problems", IEEE PES – Winter Meeting, New York, January 1976.

KOBER, F., MANZONI, A., 2003, "An Objected-Oriented Approach to Development and Integration of Graphical User Interface and Power System Framework", IEEE Bologna Power Tech Conference, Bologna, Italy, June 2003

LI, S., SHAHIDEHPOUR, 1993, "An Object Oriented Power System Graphics Package for Personal Computer Environment", IEEE Transactions on Power Systems, v. 8, n. 3, pp. 1054 – 1060, August 1993.

LIN, W. M., et al., 1999, "Three-Phase Unbalanced Distribution Power Flow Solutions With Minimum Data Preparation", IEEE Transactions on Power Systems, v. 14, n. 3, pp. 1178 – 1183, August 1999.

LUO, G. X., SEMLYEN, A., 1990, "Efficient Load Flow for Large Weakly Meshed Networks", IEEE Transactions on Power Systems, v. 5, n. 4, pp. 1309 – 1316, November 1990.

MANZONI A., SILVA A. S., DECKER I. C., 1999, "Power Systems Dynamic Simulation Using Object-Oriented Programming", IEEE Transactions on Power Systems, v. 14, n. 1, pp. 249–255, February. 1999.

MENEZES S., 2003, "Curto Circuito Trifásico", Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil

MIU, K. N., WANG, J. C., CHIANG, H. D., 1997, "Explicit Loss Formula, Voltage Formula and Current Flow Formula for Large Scale Unbalanced Distribution Systems", IEEE Transactions on Power Systems, v. 12, n. 3, pp. 1061 – 1067, August 1997.

MONTICELLI A. J., 1983, Fluxo de Carga em Redes de Energia Elétrica Editora Edgard Blucher Ltda, São Paulo, SP – 1983.

MONTICELLI, A., GARCIA, A., SAAVEDRA, O. R., 1990, "Fast Decoupled Load Flow: Hypothesis, Derivations and Testing", IEEE Transactions on Power Systems, pp. 1425 – 1431, November 1990.

MOON, Y. H., et al., 1999, "Fast and Reliable Distribution System Load Flow Algorithm Based on the Y_{bus} Formulation", Proceedings of IEEE PES Summer Meeting – 1999, v. 1, pp. 238 – 242, July 1999.

NEIDER J., DAVIS T., "OpenGL Programming Guide", Addison-Wesley Publishing Company), Second Edition, (1997)

NEYER F. A., WU F. F., 1990, "Object-Oriented Programming for Flexible Software: Example of Load Flow", IEEE Transactions on Power Systems, v. 5, n. 3, pp. 689–696, August. 1990,

NGUYEN, H. L., 1997, "Newton-Raphson Method in Complex Form", IEEE Transactions on Power Systems, v. 12, n. 3, pp. 1355 – 1359, August 1997.

OLIVEIRA, E. J. de ; LIMA, J. W. M. ; ALMEIDA, K. C., 2000, "Allocation of FACTS Devices in Hydrothermal Systems". IEEE Transactions on Power Systems, Estados Unidos, v. 15, n. 1, p. 276-282, 2000.

OLIVEIRA, L. W. de, 2005, "Ajuste Ótimo e Coordenado dos Parâmetros do Compensador Estático de Reativos", Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil.

PASSOS FILHO, J. A., 2000, "Modelagem e Incorporação de Dispositivos de Controle no Problema de Fluxo de Potência" , Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil.

PENIDO, D. R. R., 2004, "Fluxo de Potência a Quatro Condutores", Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil.

RAJICIC, D., BOSE A., 1988, "A Modification to the Fast Decoupled Power Flow for Network with High Ratios", IEEE Transactions on Power Systems, v. 3, n. 2, May 1988.

RUMBAUGH J., ET AL., (1994). "Modelagem e Projetos Baseados em Objetos", Rio de Janeiro, Editora Campus, 9a Tiragem, 1994

SHIRMOHAMMADI, D., et al., 1988, "A Compensation-based Power Flow Method for Weakly Meshed Distribution and Transmission Networks", IEEE Transactions on Power Systems, v. 3, pp. 753 – 762, May 1988.

SHORT, T. A., et al., 2002, "Five-Wire Distribution System Demonstration Project", IEEE Transactions on Power Delivery, v. 17, n. 2, pp. 649 – 654, April 2002.

STEVENSON, D., 1986, "Elementos de Análise de Sistemas de Potência" , São Paulo: Mc Graw-Hill, 1986

SILVA, F., 2004, "Modelagem De Transformadores Trifásicos de Distribuição Para Estudos De Fluxo De Potência", Tese de M.Sc., Universidade Federal de Juiz de Fora, Juiz de Fora, MG, Brasil.

STOTT, B., ALSAC, 1974, "Fast Decoupled Load Flow", IEEE Transactions on Power Apparatus and Systems, v. PAS-93, May-June 1974.

TENG, J., 2003, "A Direct Approach for Distribution System Load Flow Solutions", IEEE Transactions on Power Delivery, v. 18, n. 3, pp. 882 – 887, July 2003.

TINNEY W. F., (1972). "Compensation methods for network solution by optimally ordered triangular factorizations", IEEE Transactions on Power Apparatus and Systems, No. 1, January/February 1972, pp. 123−127.

TINNEY W. F., SUN D. I, 1987, "Optimal Power Flow: Research and Code", EPRI Research Project No. 1724-1, Final Report, February 1987

TINNEY, W. F., HART, C. E., 1967, "Power Flow Solution by Newton's Method", IEEE Transactions on Power Apparatus and Systems, v. PAS-86, n. 3, 1967.

UWEE, 2005, "Power Systems Test Case Archive", University of Washington Electrical Engineering, http://www.ee.washington.edu/research/pstca

WARD, D. J., et al., 2003, "An Analysis of the Five-Wire Distribution System", IEEE Transactions on Power Delivery, v. 18, n. 1, pp. 295 – 299, January 2003.

WASLEY, R. G., *SHLASH, M. A., 1974,* "Newton-Raphson Algorithm for 3-Phase Load Flow", Proceedings of IEEE, Panel Session: Distribution Systems Analysis Methods, July 1974.

WRIGHT R. S. Jr., SWEET M., "OpenGL Super Bible!", Waite Group Press , Second Edition, (1996)

ZHANG, F., CHENG, C. S, 1997, "A Modified Newton Method for Radial Distribution System Power Flow Analysis", IEEE Transactions on Power System, v. 12, n. 1, pp. 389 – 397, February 1997.

ZHOU E. Z., 1996, "Object-Oriented Programming, C++ and Power System Simulation", IEEE Transactions on Power Systems, v. 11, n. 1, pp. 206–215, February 1996.

ZIMMERMAN, R. D., 1995, "Fast Decoupled Power Flow for Unbalanced Radial Distribution Systems", IEEE Transactions on Power Systems, v. 10, n. 2, pp. 2045 – 2052, November 1995.