

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

JHONISSON RICARDO DE PAULA

***SOFTWARE* PARA DESENVOLVIMENTO DE TRANSFORMADORES
PARA EQUIPAMENTOS DE ELETROCIRURGIA**

DISSERTAÇÃO

CURITIBA
2012

JHONISSON RICARDO DE PAULA

**SOFTWARE PARA DESENVOLVIMENTO DE TRANSFORMADORES
PARA EQUIPAMENTOS DE ELETROCIRURGIA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do grau de “Mestre em Ciências” - Área de Concentração: Engenharia Biomédica.

Orientador: Prof. Dr. Bertoldo Schneider Jr.

CURITIBA
2012

Dados Internacionais de Catalogação na Publicação

- P324 Paula, Jhonisson Ricardo de
Software para desenvolvimento de transformadores para equipamentos de eletrocirurgia /
Jhonisson Ricardo de Paula. – 2012.
89 f. : il. ; 30 cm
- Orientador: Bertoldo Schneider Junior.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-
graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2012.
Bibliografia: f. 61-64.
1. Instrumentos e aparelhos cirúrgicos. 2. Transformadores elétricos. 3. Software –
Desenvolvimento. 4. Simulação (computadores). 5. Engenharia biomédica. 6. Engenharia elétrica
– Dissertações. I. Schneider Junior, Bertoldo, orient. II. Universidade Tecnológica Federal do
Paraná. Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD (22. ed.) 621.3

Título da Dissertação Nº.610

“Software para Desenvolvimento de Transformadores para Equipamentos de Eletrocirurgia”

por

Jhonisson Ricardo de Paula

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM CIÊNCIAS – Área de Concentração: Engenharia Biomédica, pelo Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial – CPGEI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Curitiba, às 14:30h do dia 20 de setembro de 2012. O trabalho foi aprovado pela Banca Examinadora, composta pelos doutores:

Prof. Bertoldo Schneider Junior, Dr.
(Presidente – UTFPR)

Prof. Alessandro Zimmer, Dr.
(UFPR)

Prof. Sérgio Francisco Pichorim, Dr.
(UTFPR)

Visto da coordenação:

Prof. Ricardo Lüders, Dr.
(Coordenador do CPGEI)

À minha mãe, Marinês Mello de Paula, mulher guerreira e meu maior exemplo de vida, que me deu a oportunidade de hoje me tornar um mestre graças ao seu esforço e dedicação, trabalhou muito e ainda trabalha pra manter a nossa casa e dar formação acadêmica aos seus quatro filhos, graças a ela também me considero uma pessoa digna e honesta, pois recebi a educação necessária para isto desde muito pequeno.

AGRADECIMENTOS

Ao professor Bertoldo Schneider Jr., pelas orientações, indicações e todos os conselhos passados durante o mestrado.

Aos colegas de laboratório e de curso que prestaram grande auxílio durante o desenvolvimento da dissertação e os estudos das disciplinas.

Aos meus amigos da época da graduação, Eduardo, Greg, Ronaldo, Janeallyson e especialmente ao meu amigo Jesse, colega de trabalho de conclusão de curso e grande incentivador durante o mestrado, com auxílio e colaboração durante este período.

Aos meus irmãos, meu pai, Benedito de Paula, por todo incentivo, força e apoio, não me deixando desanimar durante essa trajetória, e minha namorada, Francelize, que sempre me deu forças e não me deixou desistir dos meus objetivos e soube ainda entender quando não estive presente devido aos estudos.

Faço um agradecimento especial ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pela bolsa de estudos fornecida durante o mestrado, pois sem este auxílio financeiro eu dificilmente poderia realizar este projeto.

RESUMO

DE PAULA, Jhonisson R. *Software* para desenvolvimento de transformadores para equipamentos de eletrocirurgia. 2012. 89 f. Dissertação (Mestrado em Engenharia Biomédica) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

Este trabalho propõe e implementa um *software* que realiza o cálculo de vários modelos de transformadores de instrumentação e orienta o usuário em como efetuar medições e correções importantes nos mesmos. O *software* engloba transformadores de baixa e média potência, até 500 W, e com frequências de até alguns megahertz. O *software* cobre 3 modelos: o clássico quadripolo de baixa potência e baixa frequência; o quadripolo T com capacitor de saída para baixa potência e média frequência; e o modelo delta-estrela completo, para transformadores eletrônicos de alta frequência. O trabalho teve como objetivo criar uma ferramenta para o projeto de transformadores, especialmente transformadores de saída de instrumentação biomédica, como por exemplo, o transformador de saída do bisturi eletrônico. O *software* indica o modelo de acordo com os dados de entrada ou por escolha do usuário. O *software* foi criado para uso gratuito e foi desenvolvido em código Java, de modo que qualquer pessoa interessada possa usá-lo. Por suas características, o *software* pode trabalhar em todos os computadores, *desktops* ou celulares, e em quase todos os sistemas operacionais, basta que o usuário tenha a máquina virtual Java instalada em seu *hardware*. O usuário deve inserir os coeficientes necessários para realizar os cálculos, obtendo os resultados numéricos e gráficos do módulo da impedância *versus* frequência e de fase *versus* frequência na tela ou em um arquivo de dados no formato CSV (*comma-separated values*). O módulo de saída gráfica apresenta o módulo e a fase da impedância de entrada (Z_{in}) em função da frequência. Isto permite ao usuário ver as ressonâncias e comportamentos característicos do transformador e tomar providências para executar um projeto conforme suas necessidades.

Palavras-chave: Instrumentação Biomédica. *Software*. Transformadores de Alta Frequência.

ABSTRACT

DE PAULA, Jhonisson R. Development software for electrosurgical equipments transformers. 2012. 87 f. Dissertação (Mestrado em Engenharia Biomédica) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2012.

This work describes a software that performs calculation and gives some tips about building and measuring transformers parameters up to 500 W till a few megahertz. The software emulates 3 models: the classical T model for low power and low frequency; the T model with output capacitance; and the complete delta-star model, for high frequency electronic transformers. This work aimed to create a tool for design of transformers, especially biomedical instrumentation output transformers, like the output electrosurgical device transformer. Tips like, for example, how to wrap the secondary coil to minimize intrinsic capacitance are presented to the software user. The software indicates the model according to input data or by user choice. The software is for free use and is programmed in Java language, so that, any person interested can use it. It will be available on the internet for free download. This software can work on all computers, desktops or mobiles, and on almost all operating systems, as long as the user has the virtual machine in his hardware. The user must insert the coefficients required to perform the calculations. After this insertion he must click on the calculation button, so the results and the impedance (modulus and phase) versus frequency graphs are seen on the screen. The user can additionally export an XY CSV (comma-separated values) data file. The software graphical output modulus and phase parts shows behavior of the input impedance (Z_{in}) versus frequency. This makes possible for the user see the resonances and characteristic behavior of the transformer.

Keywords: Biomedical Instrumentation. Software. High Frequency Transformers.

LISTA DE FIGURAS

Figura 2. 1 – Representação de um Quadripolo.....	19
Figura 2. 2 - Transformador ideal.....	20
Figura 2. 3 - Modelo T Clássico.....	22
Figura 2. 4 - Modelo T com capacitor de saída	24
Figura 2. 5 - Modelo delta-estrela de transformador	25
Figura 3. 1 – Diagrama de caso de uso.	33
Figura 3. 2 – Diagrama de sequência.	34
Figura 3. 3 – Diagrama de estado.....	35
Figura 3. 4 – Diagrama de Classes do software.....	36
Figura 4. 1 – Tela principal do software.	39
Figura 4. 2 – Tela principal após clicar no botão “Complete Model”.....	40
Figura 4. 3 – Tela auxiliar apresentada ao clicar no botão “Complete Model”.....	41
Figura 4. 4 – Tela principal após clicar no botão “Quadripole T”.....	42
Figura 4. 5 – Tela auxiliar apresentada ao clicar no botão “Quadripole T”.....	43
Figura 4. 6 – Tela principal após clicar no botão “Quadripole T plus C2”.....	44
Figura 4. 7 – Tela auxiliar apresentada ao clicar no botão “Quadripole T plus C2”...	44
Figura 4. 8 – Seleção do botão “Variation”	45
Figura 4. 9 – Tela auxiliar apresentada ao clicar no botão “Variation”	46
Figura 4. 10 – Seleção do botão “Calculate k”	46
Figura 4. 11 – Tela auxiliar apresentada ao clicar no botão “Calculate k”.....	47
Figura 4. 12 – Seleção do botão “Tips”.....	47
Figura 4. 13 – Primeira tela auxiliar apresentada após clicar no botão “Tips”.....	48
Figura 4. 14 - Segunda tela auxiliar apresentada após clicar no botão “Tips”.....	49
Figura 4. 15 – Seleção do botão “About”.....	50
Figura 4. 16 – Tela apresentada ao clicar no botão “About”.....	50
Figura 4. 17 – Tela apresentada ao clicar no botão “Export CSV format file”.....	51
Figura 4. 18 – Arquivo CSV gerado pelo software e importado pelo MS Word®.	52
Figura 4. 19 – Arquivo CSV gerado pelo software e importado pelo MS Excel®, resultado utilizando o recurso gráfico deste software.....	52
Figura 4. 20 – Sobreposição das curvas de impedância de entrada dos três modelos	53
Figura 4. 21 – Resultados obtidos através do software Eletronics Workbench®.	54
Figura 4. 22 – Resultado gráfico da impedância de entrada (normalizada) plotada no Excel, comparando o resultado do software e do EWB...	55
Figura 4. 23 – Resultado gráfico comparando o resultado obtido pelo software e pelo EWB para o módulo da impedância de entrada normalizada.	55

LISTA DE TABELAS

- Tabela 4. 1 – Cálculo do erro percentual relativo, variando apenas o valor de R_L56
Tabela 4. 2 – Cálculo do erro percentual relativo, variando todos os coeficientes....56

LISTA DE ABREVIATURAS E SIGLAS

CSV	<i>Comma-separated values</i> , formato de arquivo que armazena dados separando-os por delimitadores, como a quebra de linha e o ponto e vírgula.
EWB	<i>Eletronics Worbench</i> ®, <i>software</i> para modelagem de circuitos elétricos.
IDE	<i>Integated Development Environment</i> , ambiente de desenvolvimento integrado para desenvolvedores de <i>software</i> .
NBR	Norma Brasileira, segundo a Associação Brasileira de Normas Técnicas.
UML	<i>Unified Modeling Language</i> , linguagem para modelagem de sistemas orientados a objeto.
Vrms	<i>root mean square</i> , Tensão eficaz ou média quadrática. É o valor de tensão contínua que entregaria a mesma potência ao sistema sob ação de uma forma de onda periódica qualquer de tensão.

SUMÁRIO

1 INTRODUÇÃO	12
1.1 CONSIDERAÇÕES INICIAIS	12
1.2 OBJETIVO.....	12
1.3 JUSTIFICATIVA	13
1.4 ORGANIZAÇÃO DA DISSERTAÇÃO	14
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 COMPONENTES BÁSICOS	16
2.1.1 Indutor	16
2.1.2 Capacitor	17
2.1.3 Impedância.....	18
2.1.4 Quadripolos	19
2.2 TRANSFORMADORES.....	19
2.2.1 Transformador Ideal	19
2.2.2 Transformador Real.....	21
2.3 MODELOS DE TRANSFORMADORES.....	22
2.3.1 Modelo T Clássico.....	22
2.3.2 Modelo T Com Capacitor De Saída.....	24
2.3.3 Modelo Delta-Estrela	24
2.3.4 Modelo T De Magnetização E Perdas.....	27
2.3.5 Modelo Gama Do Transformador.....	28
3 DESENVOLVIMENTO DO SOFTWARE	29
3.1 JAVA	29
3.1.1 Bytecode	30
3.1.2 Máquina Virtual	31
3.2 NETBEANS	31
3.3 AMBIENTE DE DESENVOLVIMENTO	32
3.4 DIAGRAMAS UML	33
3.4.1 Diagrama De Caso De Uso.....	33
3.4.2 Diagrama De Sequência	34
3.4.3 Diagrama De Estado	35
3.4.4 Diagrama De Classes.....	36
3.5 VERSÃO EM INGLÊS.....	36
3.6 DISPONIBILIDADE PARA <i>DOWNLOAD</i>	37
3.7 MANUAL DE INSTRUÇÕES	37
3.8 NOME E VERSÃO	37
4 RESULTADOS	38
4.1 ESCOLHA DOS MODELOS.....	38
4.2 TELA PRINCIPAL	38
4.3 MODELO DELTA-ESTRELA OU MODELO COMPLETO	39
4.4 MODELO T CLÁSSICO.....	41
4.5 MODELO T COM CAPACITOR DE SAÍDA.....	43
4.6 VARIAÇÃO DE ACORDO COM OS PARÂMETROS ELÉTRICOS	45
4.7 CÁLCULO DO VALOR DE k	46
4.8 INSTRUÇÕES ÚTEIS PARA PROJETAR UM TRANSFORMADOR	47
4.9 INFORMAÇÕES SOBRE O <i>SOFTWARE</i>	50
4.10 SALVANDO DADOS	51

4.11 COMPARAÇÃO DE RESULTADOS ENTRE MODELOS	53
4.12 COMPARAÇÕES COM EWB.....	53
5 CONCLUSÃO	58
5.1 DISCUSSÕES.....	59
5.2 TRABALHOS FUTUROS	59
REFERÊNCIAS	61
ANEXO A	65
ANEXO B	72
ANEXO C	88

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

A utilização de transformadores em equipamentos biomédicos é de grande importância, devido as suas especificidades e aplicações (FILCHEV & CARASTRO, 2010). Os transformadores de instrumentação vão desde ordinários transformadores de alimentação com entrada de 127 ou 220 Vrms e frequências de 50 ou 60 Hz, cuja modelagem e construção estão absolutamente dominadas, até transformadores tão específicos como os que têm a função de isolar o paciente do equipamento ou de entregar dezenas ou centenas de watts em frequências que podem ir até alguns megahertz, como no caso dos transformadores de saída de equipamentos eletrocirúrgicos.

Transformadores eletrocirúrgicos têm as seguintes peculiaridades:

- 1- Deve ser isolado, i. e., deve ter o enrolamento que está em contato com o circuito do paciente eletricamente separado do resto do equipamento, conforme a norma NBR IEC 60601-1 (1994).
- 2- Deve fornecer tipicamente dezenas a poucas centenas de watts e, para isto, deve ser projetado e confeccionado de modo a minimizar a potência reativa do circuito, por motivos de economia de espaço.
- 3- Deve funcionar entre 300 kHz e 5 MHz conforme norma NBR IEC 60601-2-2:2001 e deve ser capaz de gerar altas tensões na saída, tipicamente entre 200 volts e poucos milhares de volts.

Estas características colocam o transformador de saída dos equipamentos eletrocirúrgicos numa região tensão *versus* corrente *versus* frequência muito pouco explorada pela engenharia.

Outra característica importante é que este trabalho aborda transformadores ordinários (de potência ou tensão) capazes de ter um bom desempenho com sinais senoidais, ao contrário dos transformadores de pulsos comumente utilizados nestas aplicações, por ser uma solução mais fácil de ser desenvolvida e mais fácil de ser comercializada.

1.2 OBJETIVO

O objetivo do trabalho é desenvolver um *software* em linguagem livre, com código aberto e que pode ser executado em praticamente qualquer computador, que auxilie projetos de transformadores para equipamentos biomédicos, simplificando a realização de cálculos e demonstrando os resultados obtidos numérica e graficamente. Os equipamentos eletrocirúrgicos necessitam trabalhar em uma determinada faixa de frequência (300 kHz a 5 MHz), valores abaixo de 300 kHz podem causar eletroestimulação e queimaduras em pacientes (SCHNEIDER, DIAS e ABATTI, 2010; SCHNEIDER e ABATTI, 2008), e valores acima de 5 MHz tornam o seu funcionamento muito complicado, devido a capacitâncias e indutâncias parasitas. O desenvolvimento das equações e a realização dos cálculos necessários para o projeto de um transformador demandam muito tempo e esforço, além de que, há sempre uma grande chance de ocorrerem erros ou falhas, devido ao grande número de coeficientes presentes nas mesmas. O *software* desenvolvido simplifica todo este processo, pois evita que haja erros nas equações e nos desenvolvimentos destes cálculos. Os resultados gerados do módulo e da fase da impedância de entrada do transformador possuem a opção de exportação no formato CSV (*Comma-separated values*), reconhecível em qualquer *software* gerador de gráficos (MS Excel®, Sigma Plot® e outros). Além disso, instruções de construção e medição de parâmetros são previstos. Para uma maior universalização do uso, optou-se pela língua inglesa.

1.3 JUSTIFICATIVA

Através do avanço da tecnologia, faz-se necessário cada dia mais a utilização de ferramentas técnicas e tecnológicas para a realização de tarefas que tendem a ser demoradas e complexas. Algo como efetuar cálculos matemáticos utilizando grandes equações pode ser auxiliado por uma ferramenta simples que realize estes mesmos cálculos.

Para a modelagem de transformadores ou de outros circuitos elétricos já existem *softwares*, como o EWB (Eletronics Workbench®), capazes de simular o

funcionamento de um circuito elétrico e apresentar os resultados obtidos, como por exemplo, a tensão ou a corrente em determinado ponto. Estes *softwares* não mostram facilmente suas características de modelagem, tendo herméticos modelos *spice* de componentes (nome genérico da maioria dos modelos utilizados para simular componentes eletrônicos em ambientes de simulação), difíceis de lidar na prática. Além disso, tais modelos são geralmente numéricos ou polinomiais e dificilmente apresentam vantagens analíticas para projetos. Uma limitação importante deste tipo de modelo é o fato de uma indutância não poder receber valor negativo, impossibilitando que os modelos aqui tratados possam ser facilmente simulados (*L11-M* pode ter facilmente valores negativos). O EWB, por exemplo, não processa tais valores negativos e uma intervenção ou modificação nos parâmetros *spice* de seus componentes seria necessária, ou uma adaptação massiva para outro modelo. Outro problema é que o EWB é um *software* pago e, portanto, não pode ser utilizado por qualquer pessoa ou em qualquer lugar. *Softwares* de planilhamento e geração de gráficos como o MS Excel® ou o Sigma Plot® também executam estes cálculos, mas o usuário deve dominar todo o equacionamento, além do risco sempre existente do lançamento errado das equações e o conseqüente mau funcionamento do processo.

Pensando nessas barreiras apresentadas pelos *softwares* já existentes surgiu a ideia de desenvolver um *software* gratuito, onde os modelos de transformadores já estivessem incorporados e prontos e o usuário pudesse escolher qual modelo de transformador atendesse melhor suas necessidades ou sua aplicação.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

No Capítulo presente, apresentou-se a introdução geral, os objetivos e as justificativas deste trabalho.

Apresenta-se no Capítulo 2 a base teórica dos transformadores utilizados, bem como a descrição sucinta de alguns parâmetros como a impedância, a indutância, as reatâncias e a capacitância.

O Capítulo 3 apresenta o desenvolvimento do *software*, descrevendo os seguintes pontos: a linguagem utilizada; a versão em inglês; o diagrama de caso de

uso, o diagrama de sequência, o diagrama de estado, o diagrama de classes e a disponibilidade para *download*.

Já no Capítulo 4 estão dispostos os resultados obtidos pelo *software* desenvolvido e seus comparativos com *softwares* já existentes.

O capítulo 5 aborda as discussões e a conclusão obtida através do desenvolvimento do *software*.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve sucintamente os modelos de transformadores e comenta suas aplicações e limitações, bem como seus elementos constituintes e suas equações.

Os transformadores estudados e aqui citados têm como objetivo sua utilização como transformadores de saída de equipamentos de eletrocirurgia (faixa de operação entre 300 kHz e 5 MHz). Dificilmente estes transformadores podem ser utilizados com o intuito de eletroestimulação, pois acima dos 100 kHz os nervos e músculos apresentam pouca resposta aos estímulos (WEBSTER, 1988), podendo ainda ocorrer se a energia for suficiente para esta estimulação em frequências de até 1 MHz (LaCOURSE et al, 1988).

2.1 COMPONENTES BÁSICOS

A fim de facilitar o entendimento de alguns parâmetros que serão bastante abordados neste trabalho, alguns elementos elétricos básicos são aqui descritos, como o indutor, o capacitor, as reatâncias e as impedâncias.

2.1.1 INDUTOR

Indutores e capacitores são elementos passivos, ambos são capazes de fornecer e armazenar quantidades finitas de energia. As características elétricas associadas a eles são a indutância e a capacitância, respectivamente (LOPES, 2008).

A equação que descreve a indutância (L) é (HAYT & KEMMERLY, 1975):

$$v = L \frac{di}{dt} \quad [V] \quad (2.1)$$

onde v é a tensão (volts), di/dt é a derivada da corrente (ampères) pelo tempo (segundos) e L é a indutância (henrys).

Um indutor pode ser criado enrolando-se um pedaço de fio condutor na forma de bobina.

A reatância indutiva (X_L) pode ser entendida como a quantidade de oposição (resistência) oferecida à passagem de corrente alternada pelo indutor (BRAGA, 2012), é representada pela equação 2.2:

$$X_L = 2\pi fL \quad (2.2) \quad [\Omega] \quad (2.2)$$

onde f é a frequência em hertz e L é a indutância em henrys.

2.1.2 CAPACITOR

O capacitor é um elemento passivo e a capacitância é simbolizada pela letra C , sendo definida pela equação 2.3, tendo como unidade de medida o farad (F) (HAYT & KEMMERLY, 1975).

$$i = C \frac{dv}{dt} \quad [A] \quad (2.3)$$

onde i é a corrente em ampères e dv/dt é a derivada da tensão em volts pelo tempo em segundos.

O capacitor pode ser entendido fisicamente como duas superfícies condutoras que podem armazenar energia na forma de campo elétrico, sendo estas superfícies separadas por um material dielétrico.

A reatância capacitiva (X_C) pode ser entendida como a quantidade de oposição (resistência) oferecida à passagem de corrente alternada pelo capacitor (BRAGA, 2012), é representada pela equação 2.4:

$$X_C = \frac{1}{2\pi fC} \quad [\Omega] \quad (2.4)$$

onde f é a frequência em hertz e C é a capacitância em farad.

2.1.3 IMPEDÂNCIA

Impedância (Z) é a oposição que um componente elétrico oferece à passagem de corrente alternada e é sempre descrita através de um número complexo. As formas utilizadas neste trabalho são a polar, onde um módulo (valor absoluto de Z) é associado a uma fase θ (este ângulo é sempre o simétrico do ângulo de defasagem da corrente em relação à tensão) num duplete do tipo:

$$(M, \theta) \quad [\Omega] \quad (2.5)$$

e a forma cartesiana (ou retangular) cuja representação é:

$$Z = a + bj \quad [\Omega] \quad (2.6)$$

onde a é a parte real em ohms, é a resistência pura, e b a parte imaginária de Z , correspondente às reatâncias capacitivas e/ou indutivas.

As transformações entre as duas formas são:

$$M = \sqrt{a^2 + b^2} \quad [\Omega] \quad (2.7)$$

$$\theta = \arctg \frac{b}{a} + \left(1 - \frac{a}{|a|}\right) \cdot 90^\circ \quad [^\circ] \quad (2.8)$$

$$a = M \cos \theta \quad [] \quad (2.9)$$

e

$$b = M \sin \theta \quad [] \quad (2.10)$$

A admitância Y é a recíproca da impedância ($1/Z$):

$$Y = gc + bmj \quad [S] \quad (2.11)$$

onde gc é a condutância, e representa a parte real da admitância, e bm é a susceptância, que representa a parte imaginária da admitância, sendo as duas medidas em siemens [S].

A parte real gera potência real, i. e., aquela capaz de realizar trabalho no mundo físico e a parte imaginária está relacionada com a potência reativa (armazenamento de energia na forma de campo elétrico ou magnético).

2.1.4 QUADRIPOLOS

Para realizar o estudo de alguns fenômenos elétricos, como a indutância mútua, consideramos um fenômeno associado a dois pares de terminais. Um elemento fundamental para análise de sistemas eletrônicos é o quadripolo, ele possui um “terminal de entrada” e outro “terminal de saída”. O sinal que passa por este elemento normalmente sofre algum tipo de alteração entre estes dois pares de terminais. A Figura 2.1 exemplifica um caso de quadripolo (HAYT, 1975).

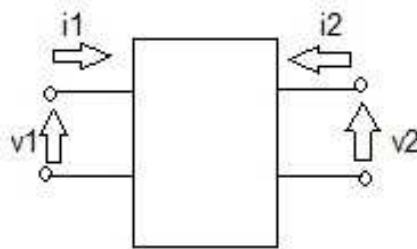


Figura 2. 1 – Representação de um Quadripolo. Fonte: Adaptado de Hayt (1975).

No quadripolo, v_1 e i_1 são tensão e corrente de entrada, respectivamente, v_2 e i_2 representam a tensão e a corrente de saída, respectivamente. Convencionalmente os sentidos das mesmas são como mostrados na Figura 2.1

2.2 TRANSFORMADORES

2.2.1 TRANSFORMADOR IDEAL

Um transformador ideal é um dispositivo eletromagnético sem perdas do fluxo magnético entre as bobinas de um enrolamento primário e o secundário, Figura 2.2. Considera-se que a energia é totalmente conservada (GUSSOW, 1985). A descrição da permeabilidade real e complexa encontra-se no Anexo C.

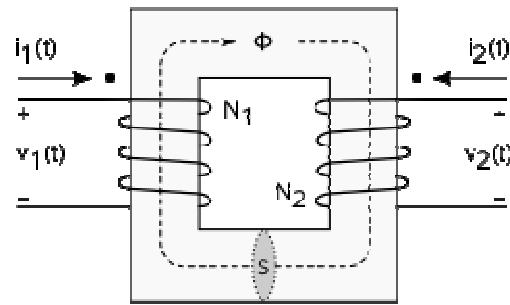


Figura 2. 2 - Transformador ideal - i_1 e v_1 são a corrente e a tensão de excitação do enrolamento primário e i_2 e v_2 são a corrente e a tensão induzidas no secundário. Φ é o fluxo magnético causada pela excitação do primário. Fonte: Gussow (1985).

Através da corrente $i_1(t)$ variante no tempo, ocorre a excitação da indutância própria do enrolamento L_{11} , gerando um fluxo magnético Φ no sentido indicado. Por meio deste fluxo, segundo a Lei de Lenz, é criada uma tensão $v_2(t)$ no outro enrolamento, de indutância própria L_{22} , gerando assim uma corrente $i_2(t)$ e um fluxo magnético que se opõe ao fluxo produzido por $i_1(t)$.

No transformador ideal o fluxo gerado por $i_1(t)$ é anulado pelo fluxo gerado por $i_2(t)$, considerando-se assim que nenhuma energia é armazenada no núcleo, não havendo perdas, desta forma $v_2(t).i_2(t) = v_1(t).i_1(t)$. A equação do ganho de tensão (a) de um transformador ideal é dada por:

$$a = \sqrt{\frac{L_{22}}{L_{11}}} \quad [] \quad (2.12)$$

Existem outras formas de expressar este ganho, sendo considerada a mais popular a equação 2.13:

$$a = \frac{N_2}{N_1} = \frac{V_2}{V_1} = \frac{i_1}{i_2} \quad [] \quad (2.13)$$

onde N_2 representa o número de espiras do enrolamento secundário e N_1 representa o número de espiras do enrolamento primário, V_2 é a tensão no secundário e V_1 a tensão no primário, i_1 é a corrente no primário e i_2 a corrente no secundário.

Algumas propriedades de um transformador ideal (SCHNEIDER, 2004):

- 1 – Quando o secundário está em curto, a impedância de entrada é zero.
- 2 – Quando o secundário está em aberto, a impedância de entrada é função de L_{11} .
- 3 – A potência é perfeitamente conservada.

- 4 – O deslocamento de fase através do transformador é zero.
- 5 – A potência reativa é conservada.
- 6 – Quando a frequência de excitação não é zero, todos os itens anteriores são independentes da frequência.
- 7 – Formas de ondas periódicas são preservadas.

Todas estas propriedades não podem ser aplicadas a um transformador real, pois existe perda de energia nos enrolamentos e no núcleo, além das impedâncias parasitas que surgem no caso real.

2.2.2 TRANSFORMADOR REAL

Normalmente o cobre esmaltado é utilizado nos condutores de transformadores, enrolado em dois conjuntos de espiras sobre um carretel de papel ou plástico, possibilitando o encaixe com um núcleo de ferro ou ferrite. Por se tratar de um condutor metálico o cobre possui uma resistência intrínseca distribuída não desprezível, podendo ser relacionada com a resistividade (SEARS & ZEMANSKI, 1980). Esta resistência é o principal motivo para o aquecimento dos transformadores. Estas resistências intrínsecas e as indutâncias próprias dos enrolamentos representam as perdas geradas nos enrolamentos (SCHNEIDER, 2004).

Outra característica encontrada no transformador real é a ressonância devido aos seus componentes reativos intrínsecos. Para que sejam facilitados os cálculos, o transformador deve ser utilizado na região que precede a primeira frequência de ressonância (f_{01}), para que se obtenha apenas o seu período com comportamento indutivo para o circuito que o excita.

Existem alguns problemas ocasionados quando se utiliza o transformador acima da sua f_{01} . Além disso, sua primeira frequência de ressonância pode ser alterada devido às capacitâncias parasitas, fazendo com que o transformador deixe de apresentar um comportamento indutivo e tenha um comportamento capacitivo, isso é especialmente problemático para transformadores de tensão com valores de carga de saída (R_L) muito altos (SCHNEIDER, 2004).

Para um bom funcionamento, um transformador deve ainda apresentar uma impedância que possibilite a transferência de potência para a carga. A impedância

gerada pelo circuito de saída de potência deve estar equilibrada com a do enrolamento primário do transformador, para que não haja desperdício de potência na forma de potência reativa ou por descasamento de impedâncias.

A região entre a primeira e a segunda frequência de ressonância possui um comportamento capacitivo e as regiões localizadas próximas as ressonâncias são instáveis, podendo, sob certas condições, apresentar comportamento indutivo, capacitivo ou resistivo, desta forma provocando oscilações prejudiciais ao estágio de saída de potência (SCHNEIDER, 2004).

2.3 MODELOS DE TRANSFORMADORES

Os modelos de transformadores estudados e implementados no *software* são apresentados em seguida. Cada modelo apresenta suas peculiaridades e a utilização de cada um depende de sua aplicação. Os modelos foram tratados para cargas capacitivas e com R_L puramente resistivos. A justificativa é que o corpo humano (a carga dos transformadores de saída de instrumentação biomédica) é essencialmente resistivo e a reatância espúria essencialmente capacitiva.

2.3.1 MODELO T CLÁSSICO

O modelo T clássico é usualmente utilizado para baixas frequências. É considerado um modelo padrão, pois apresenta uma forma simplificada, Figura 2.3.

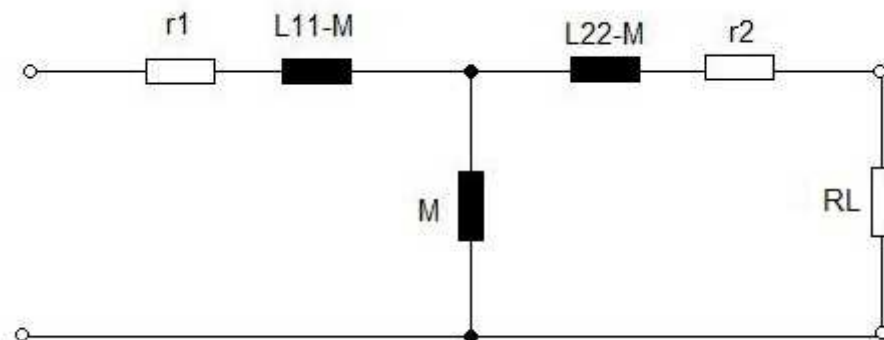


Figura 2. 3 - Modelo T Clássico. Fonte: Adaptado de Schneider (2004).

Os valores r_1 e r_2 representam as perdas dos enrolamentos através da resistência intrínseca do enrolamento primário e do secundário, respectivamente. A indutância mútua é representada por M e L_{11} e L_{22} são as indutâncias próprias do primário e do secundário, respectivamente, e R_L é carga de saída, podendo neste caso ser representada apenas pela parte resistiva, pois o corpo humano é considerado essencialmente uma carga resistiva, segundo a NBR-IEC 601-1.

Sua impedância de entrada pode ser equacionada genericamente, como:

$$Re[Zin] = r_1 + \frac{(r_2 + R_L) \cdot M^2}{(r_2 + R_L)^2 + (XL_{22} + M)^2} \quad [\Omega] \quad (2.14)$$

$$Im[Zi] = XL_{11} + \frac{M \cdot (r_2 + R_L)^2 + M \cdot XL_{22}^2 + XL_{22} \cdot M^2}{(r_2 + R_L)^2 + (XL_{22} + M)^2} \quad [\Omega] \quad (2.15)$$

$$Zin = Re[Zin] + jIm[Zin] \quad [\Omega] \quad (2.16)$$

onde XL_{11} e XL_{22} são as reatâncias indutivas do enrolamento primário e secundário, respectivamente e M é a indutância mútua entre os enrolamentos. Lembrando que este modelo de transformador não considera perdas ocorridas no núcleo.

A indutância mútua (M) pode ser representada por (EDIMINISTER, 1991):

$$M = k \cdot \sqrt{L_{11} \cdot L_{22}} \quad [H] \quad (2.17)$$

onde k é o coeficiente de acoplamento magnético e pode ser determinado pela equação 2.18 (BLACHE, KEDAREC & COGITORE, 1994; SCHNEIDER, 2004):

$$k \cong \sqrt{1 - \frac{L_{11SC}}{L_{11}}} \quad [] \quad (2.18)$$

onde L_{11SC} é a medida da indutância de entrada em farad com o secundário em curto e L_{11} a indutância de entrada com o secundário em aberto.

2.3.2 MODELO T COM CAPACITOR DE SAÍDA

Este modelo de transformador, Figura 2.4, é o que mais se aproxima do modelo T clássico, e apresenta bons resultados para aplicações de baixa potência e média frequência.

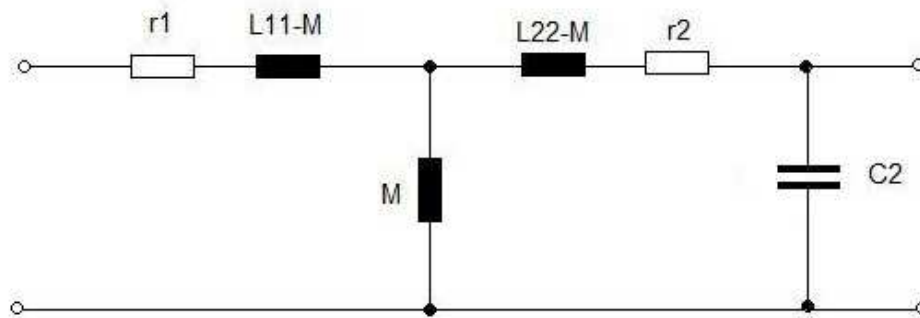


Figura 2. 4 - Modelo T com capacitor de saída. Fonte: Adaptado de Schneider (2004).

Para este modelo de transformador a impedância de entrada pode ser dada por (SCHNEIDER, 2004):

$$Z_{in} = \left[r1 + \frac{r2 \cdot XM^2}{r2^2 + (XL_{22} - XC2)^2} \right] + j \cdot \left[XL_{11} - XM + \frac{r2^2 \cdot XM + XM \cdot (XL_{22} - XC2) \cdot (XL_{22} - XM - XC2)}{r2^2 + (XL_{22} - XC2)^2} \right] [\Omega] \quad (2.19)$$

2.3.3 MODELO DELTA-ESTRELA

O modelo Delta-Estrela usa as auto-indutâncias dos enrolamentos primário e secundário e a indutância mútua entre estas, Figura 2.5 (SCHNEIDER, 2004).

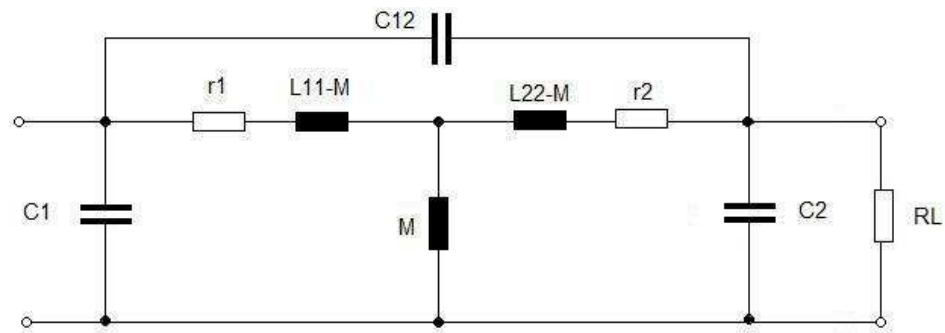


Figura 2. 5 - Modelo delta-estrela de transformador. L_{11} e L_{22} são as auto-indutâncias dos enrolamentos primário e secundário, respectivamente e a indutância mútua entre estes, M . $C1$ e $C2$, as capacitâncias intrínsecas dos enrolamentos primário e secundário, respectivamente, e $C12$ a capacitância intrínseca entre os enrolamentos. Fonte: Adaptado de Schneider (2004).

Quando em altas frequências as capacitâncias intrínsecas dos enrolamentos primário e secundário, $C1$ e $C2$, são colocados em paralelo com a entrada e saída do transformador.

A capacitância entre os enrolamentos é representada por $C12$. Para aumentar ou diminuir esta capacitância são utilizadas diversas técnicas, por exemplo, o espaçamento das espiras dos enrolamentos através do uso de fios de nylon, polímeros ou algodão para diminuí-la (HARA, 2003).

A impedância de entrada do modelo Delta-Estrela pode ser representada pela seguinte equação:

$$Z_{in} = [Ra.\Lambda + Ima.\Theta] + j.[Ima.\Lambda - Ra.\Theta], \quad [\Omega] \quad (2.20)$$

por se tratar de uma equação muito extensa e complexa, ela está desmembrada nas seguintes equações:

$$\Lambda = \left(1 - \frac{\alpha.\gamma + \beta.\delta}{\gamma^2 + \delta^2}\right), \quad [] \quad (2.21)$$

$$\Theta = \left(\frac{(\beta.\gamma - \alpha.\delta)}{\gamma^2 + \delta^2}\right), \quad [] \quad (2.22)$$

$$\alpha = Ra.(Rc + RL) - Ima.Imc, \quad [] \quad (2.23)$$

$$\beta = (Rc + RL).Ima + Ra.Imc, \quad [] \quad (2.24)$$

$$\gamma = (Ra + Rb + Rc).(Rc + RL) - Imc.(Ima + Imb + Imc) - Rc^2 + Imc^2, \quad [] \quad (2.25)$$

$$\delta = Imc.(Ra + Rb + Rc) + (Rc + RL).(Ima + Imb + Imc) - 2.Rc.Imc, [] \quad (2.26)$$

$$Ra = \frac{ReZa.XC1^2}{ReZa^2 + (ImZa - XC1)^2}, \quad [\Omega] \quad (2.27)$$

$$Ima = \frac{ImZa.XC1^2 - XC1.ReZa^2 - XC1.ImZa^2}{ReZa^2 + (ImZa - XC1)^2}, \quad [\Omega] \quad (2.28)$$

$$Rb = \frac{ReZb.XC12^2}{ReZb^2 + (ImZb - XC12)^2}, \quad [\Omega] \quad (2.29)$$

$$Imb = \frac{ImZb.XC12^2 - XC12.ReZb^2 - XC12.ImZb^2}{ReZb^2 + (ImZb - XC12)^2}, \quad [\Omega] \quad (2.30)$$

$$Rc = \frac{ReZc.XC2^2}{ReZc^2 + (ImZc - XC2)^2}, \quad [\Omega] \quad (2.31)$$

$$Imc = \frac{ImZc.XC2^2 - XC2.ReZc^2 - XC2.ImZc^2}{ReZc^2 + (ImZc - XC2)^2}, \quad [\Omega] \quad (2.32)$$

$$ReZa = r1 + XM.\left(\frac{r2.(XM - XL_{11}) + r1.(XL_{22} - XM)}{r2^2 + (XL_{22} - XM)^2}\right), \quad [\Omega] \quad (2.33)$$

$$ImZa = XL_{11} + XM.\left(\frac{r1.r2 + (XL_{22} - XM).(XL_{11} - XM)}{r2^2 + (XL_{22} - XM)^2}\right), \quad [\Omega] \quad (2.34)$$

$$ReZb = r1 + r2 + \frac{r1.(XL_{22} - XM) + r2.(XL_{11} - XM)}{XM}, \quad [\Omega] \quad (2.35)$$

$$ImZb = XL_{11} + XL_{22} - 2.XM + \frac{(XL_{22} - XM).(XL_{11} - XM) - r1.r2}{XM}, \quad [\Omega] \quad (2.36)$$

$$ReZc = r2 + XM.\left(\frac{r1.(XM - XL_{22}) + r2.(XL_{11} - XM)}{r1^2 + (XL_{11} - XM)^2}\right), \quad [\Omega] \quad (2.37)$$

e

$$ImZc = XL_{22} + XM.\left(\frac{r2.r1 + (XL_{11} - XM).(XL_{22} - XM)}{r1^2 + (XL_{11} - XM)^2}\right), \quad [\Omega] \quad (2.38)$$

Uma equação dessa envergadura é de difícil manipulação. A mecanização da mesma através de um *software* é algo desejável e muito útil.

2.3.4 MODELO T DE MAGNETIZAÇÃO E PERDAS

Segundo Smith e Alley (1992) este modelo é apropriado para transformadores com grandes perdas sendo utilizado normalmente com transformadores de potência, Figura 2.6. Não será abordado no *software*, mas é aqui apresentado para ilustrar casos onde existam perdas significativas.

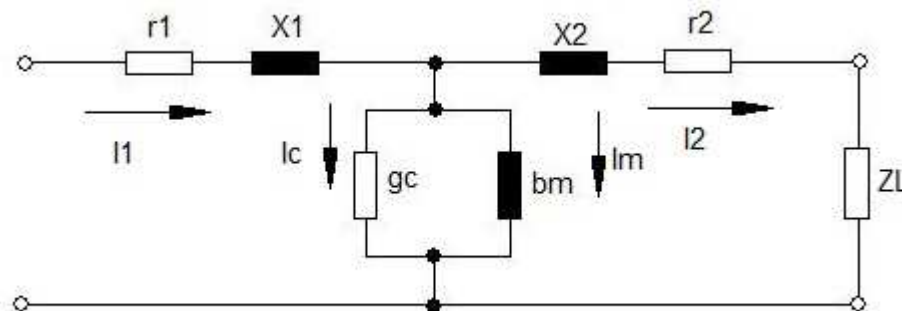


Figura 2.6 - Modelo de magnetização e perdas, utilizados para estudo de transformadores. g_c representa a condutância do núcleo e b_m a susceptância; I_1 e I_2 são as indutâncias de fuga e I_m a indutância de magnetização. Fonte: Adaptado de Smith & Alley (1992).

Para o melhor entendimento de alguns coeficientes deste modelo faz-se necessário que se explique que a impedância é formada pela resistência em sua parte real e pela reatância em sua parte imaginária, e é medida em ohms. Já a admitância é formada pela condutância (g_c) em sua parte real e pela susceptância (b_m) em sua parte imaginária, e é medida em siemens. A condutância (g_c) dá conta das perdas reais de energia no núcleo e a susceptância (b_m) das perdas reativas.

2.3.5 MODELO GAMA (Γ) DO TRANSFORMADOR

Este modelo é utilizado em grande escala para o estudo de transformadores de potência em frequências de distribuição de energia, Figura 2.7, com a diferença de que os parâmetros usados para compô-lo são de auto e mútua indutância.

O modelo Gama é recomendável nas situações de alta frequência com banda estreita de funcionamento, altas potências e/ou grande faixa de variação da carga do transformador (SCHNEIDER, 2004).

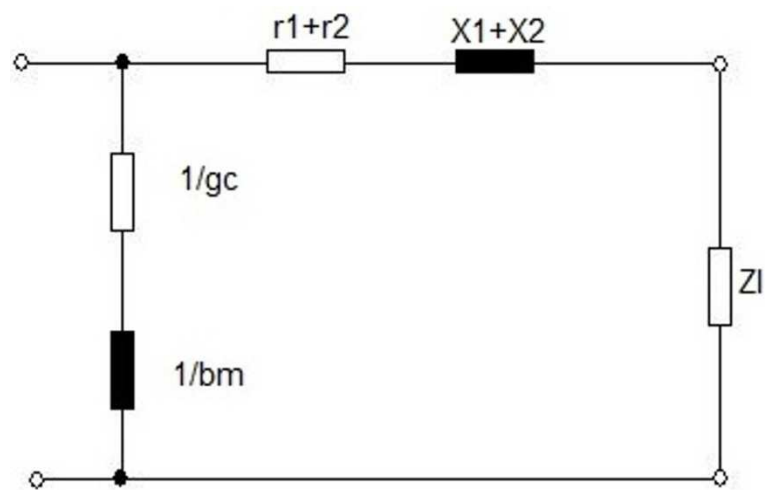


Figura 2. 7 - Modelo simplificado (GAMA, Γ), com o laço de entrada transposto para o laço de saída "saída". Fonte: Adaptado de Smith & Alley (1992).

Dos modelos apresentados neste capítulo, somente aqueles sem grandes perdas no núcleo serão utilizados no *software*. Os modelos desprezados são mais utilizados em sistemas de distribuição de energia, enquanto os modelos escolhidos são apropriados para os transformadores de instrumentação.

3 DESENVOLVIMENTO DO SOFTWARE

O objetivo do trabalho é desenvolver um *software* gratuito, capaz de ser executado em qualquer plataforma e com o intuito de auxiliar em projetos de transformadores usados em equipamentos biomédicos, especialmente os transformadores de saída eletrocirúrgica. Para tanto, faz-se necessária a exploração de tópicos sobre linguagem de programação e sobre engenharia de *software*.

3.1 JAVA

Java é uma linguagem de programação orientada a objeto. Linguagens de programação orientada a objeto apresentam algumas características específicas, sendo elas:

- Classe: define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos. Representa um conjunto de objetos com características específicas;
- Objeto: possui a capacidade de armazenar estados através de seus atributos e reage a mensagens enviadas a ele, e também pode possuir a capacidade de se relacionar e enviar mensagens a outros objetos;
- Atributo: são características de um objeto. É a estrutura de dados que representa a classe. Os atributos possuem valores. Os valores dos atributos de um determinado objeto são chamados de estado;
- Método: definem as habilidades dos objetos. A ação provocada pelo método só ocorre quando o método é evocado através do objeto (SICA, 2006; DALL'OGGIO; 2007).

Para facilitar a compreensão destes conceitos podemos utilizar o seguinte exemplo, digamos que um cachorro seja uma classe, esta classe pode possuir atributos como sexo, altura, peso, idade e raça. Além disso, ela também pode possuir métodos, sendo declarados como: latir(), correr(), brincar(), dormir() e etc. Podemos agora declarar alguns objetos como sendo cachorros individuais, utilizando o seguinte modo: “cachorro: Dick, Bidu e Lilica”. Todos estes objetos declarados

podem latir, correr, brincar e dormir. Os métodos que executam estas ações atuam sobre as variáveis internas de cada objeto: altura, peso e etc.

O Java foi desenvolvida pela Sun Microsystems® na década de 1990 e hoje mantida pela Oracle®. Apresenta uma grande diferença das demais linguagens, normalmente compiladas por código nativo, pois é compilada por um *bytecode* sendo executado por uma máquina virtual, desta forma podendo ser independente da plataforma utilizada (SILVA, 2012), por exemplo, pode ser executada em sistemas operacionais Windows®, Mac OS®, Linux® e outros.

Por haver esta possibilidade de ser independente da plataforma, o Java se tornou uma linguagem bastante empregada, sendo hoje em dia utilizado nos mais diversos dispositivos tecnológicos como *desktops*, *notebooks*, *tablets*, *palms* e celulares.

Além das vantagens já citadas sobre o Java pode-se citar ainda: desalocação de memória automática por processo de coletor de lixo; facilidade de internacionalização, suporta nativamente caracteres Unicode, Unicode é um padrão utilizado em computadores para representar qualquer tipo de caractere, seja ele uma letra, um acento ou um símbolo (SIQUEIRA, 2012); distribuição com um vasto conjunto de bibliotecas; simplicidade na especificação, tanto da linguagem como do ambiente de execução; e por ser distribuída gratuitamente, em sua maioria. Desde 2007 a Sun Microsystems® tornou praticamente todo o código Java um *software* de código aberto sob os termos da GNU *General Public License* (GPL) (GNU, 2012).

3.1.1 BYTECODE

Linguagens de programação como o C e o C++ utilizam um compilador diferente para cada sistema operacional ou plataforma em que irá ser executado. O código fonte é facilmente portátil, capaz de funcionar em diferentes versões de sistema operacional, mas ainda sim é necessário recompilar o código fonte, sendo gerado cada vez um binário diferente para cada caso.

Linguagens de programação mais recentes como o Java utilizam um novo conceito chamado de Bytecode, onde um único binário pode ser executado em qualquer plataforma, dentro de uma máquina virtual.

Desta forma criam-se aplicativos que são escritos apenas uma vez e executados em qualquer plataforma, reduzindo os custos de desenvolvimento. O que se altera é o *software* da máquina virtual para cada plataforma, mas este trabalho é realizado pelos desenvolvedores da linguagem e não pelos programadores (MORIMOTO, 2005).

3.1.2 MÁQUINA VIRTUAL

A máquina virtual Java (*Java Virtual Machine JVM*) é o componente (*software*) capaz de tornar independente de plataforma o código Java, ela carrega e executa os aplicativos, convertendo os *bytecodes* em código executável de máquina. Da mesma forma que um processador real, ela possui um conjunto de instruções e manipula várias áreas de memória durante a execução.

Este conceito de máquina virtual já é antigo, e uma das mais conhecidas é P-Code do UCSD Pascal de 1980.

A máquina virtual Java não tem referência direta à linguagem de programação Java, ela reconhece um tipo de formato binário, o formato do arquivo *class*, este arquivo contém instruções (*bytecodes*) da máquina virtual Java e a respectiva tabela de símbolos, além de outras informações relacionadas ao programa que será executado (LINDHOLM & YELLIN, 1999; VENNERS, 1998).

3.2 NETBEANS

O NetBeans é uma IDE (*integrated development environment*), ambiente de desenvolvimento integrado, gratuito e de código aberto para desenvolvedores de *software*. Pode ser executado nas mais diversas plataformas, como Windows®, Linux® e Mac OS®. É um ambiente que permite que o desenvolvedor crie aplicativos profissionais de *desktop*, empresariais, *web* e móveis multiplataforma.

Desde 2000 a Sun Microsystems® disponibilizou o código fonte do IDE tornando-se uma plataforma de código aberto, a partir de então a comunidade de desenvolvedores que utiliza e contribui com este projeto não parou de crescer, tornando-se uma das IDEs mais populares da atualidade (NETBEANS, 2000).

A versão utilizada para desenvolver este *software* é NetBeans IDE 7.1, mas atualmente já existe uma nova versão o NetBeans IDE 7.2.

3.3 AMBIENTE DE DESENVOLVIMENTO

O *hardware* utilizado para desenvolver o *software* é bem simples e bem fácil de ser encontrado no mercado local, possuindo as seguintes especificações:

- Modelo: Notebook Acer 4553;
- CPU: Amd Turion X2 processor P540;
- RAM: 4.0 GB;
- GPU: ATI Mobility Radeon HD 540;
- HD: 500 GB;
- SO: Windows 7 Ultimate 32 Bits.

Os requerimentos mínimos para o funcionamento do sistema são: processador 1,2 GHz, 1024 MB de RAM e espaço mínimo em disco de 10 MB para manipulação de dados, estes requerimentos são necessários para que se tenha a melhor execução possível do *software* desenvolvido, por se tratar de um *software* de geração de resultados gráficos e por realizar milhares de operações matemáticas a cada cálculo de impedância de entrada.

As ferramentas utilizadas para o desenvolvimento do *software* são:

- IDE: NetBeans 7.1;
- Linguagem de programação: Java 1.5;
- Biblioteca para gráficos: JfreeChart.

Lembrando apenas que as ferramentas mencionadas não necessitam estarem presentes nos computadores ou dispositivos dos usuários, sendo necessárias apenas para o desenvolvimento do *software*, mas para executar o *software* é necessário que esteja instalado na máquina o JRE (Java Runtime Environment) a partir da versão 6, que contem o Java Virtual Machine, as classes centrais e bibliotecas de suporte da plataforma Java, o JRE representa a parte de tempo de execução do *software* Java.

3.4 DIAGRAMAS UML

UML é uma linguagem padrão para a elaboração de estrutura de modelagem de projetos de *software*. Podendo ser utilizada para visualização, especificação, construção e documentação de itens que façam uso de sistemas complexos do *software*. A UML é considerada uma linguagem que abrange todas as visões necessárias ao desenvolvimento e implementação desses sistemas.

Deve-se lembrar que, ela é apenas uma linguagem, sendo somente uma parte de um método para desenvolvimento de *software* (BOENTE, 2003).

3.4.1 DIAGRAMA DE CASO DE USO

Segundo Ribeiro (2012) esse diagrama UML descreve as principais funcionalidades e a interação dos usuários do sistema com essas funcionalidades, sendo assim, este diagrama, Figura 3.1, documenta as possibilidades do sistema de acordo com o ponto de vista do usuário.

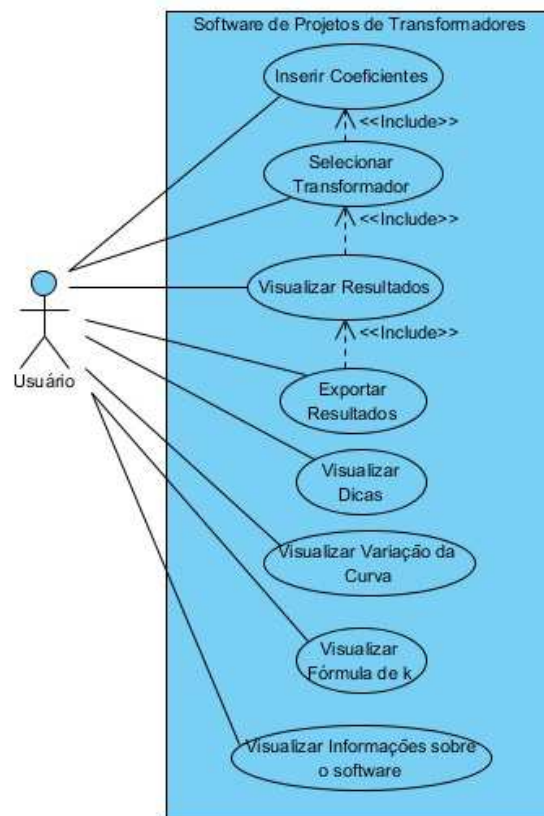


Figura 3. 1 – Diagrama de caso de uso.

3.4.2 DIAGRAMA DE SEQUÊNCIA

Esse é um diagrama UML que tem como objetivo mostrar como os objetos trocam mensagens para realização de uma operação, no decorrer do tempo (ROCHA NETO, 2008). Seu formato pode ser visualizado na Figura 3.2.

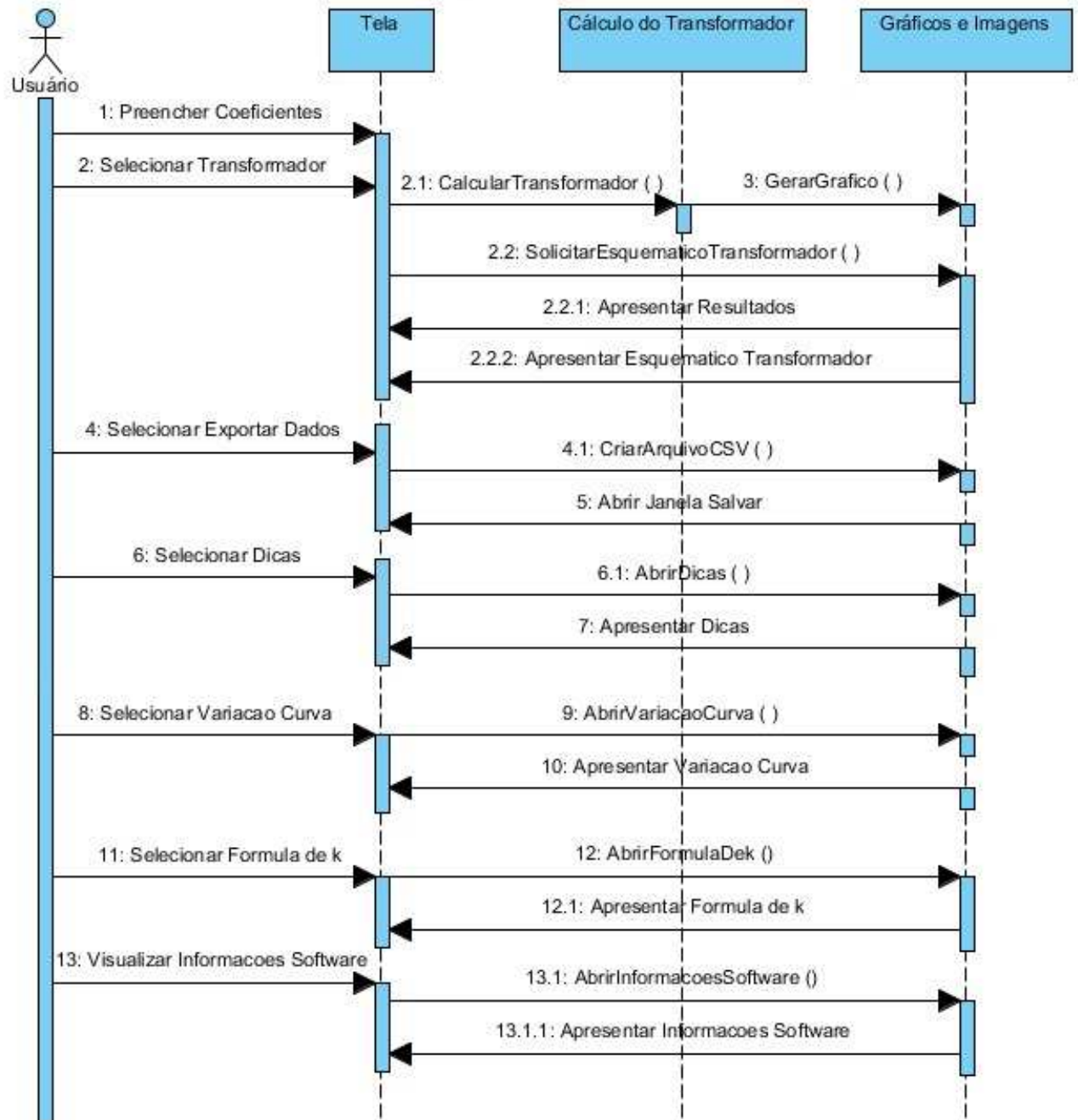


Figura 3. 2 – Diagrama de sequência.

3.4.3 DIAGRAMA DE ESTADO

Este diagrama é responsável por mostrar qual o estado em que o objeto está em determinado momento. E deve ser utilizado por objetos que tem seus estados definidos e onde o comportamento do objeto pode mudar por causa de um determinado estado (SOUZA, 2011). O diagrama de estado utilizado no *software* desenvolvidos neste trabalho é representado pela Figura 3.3.

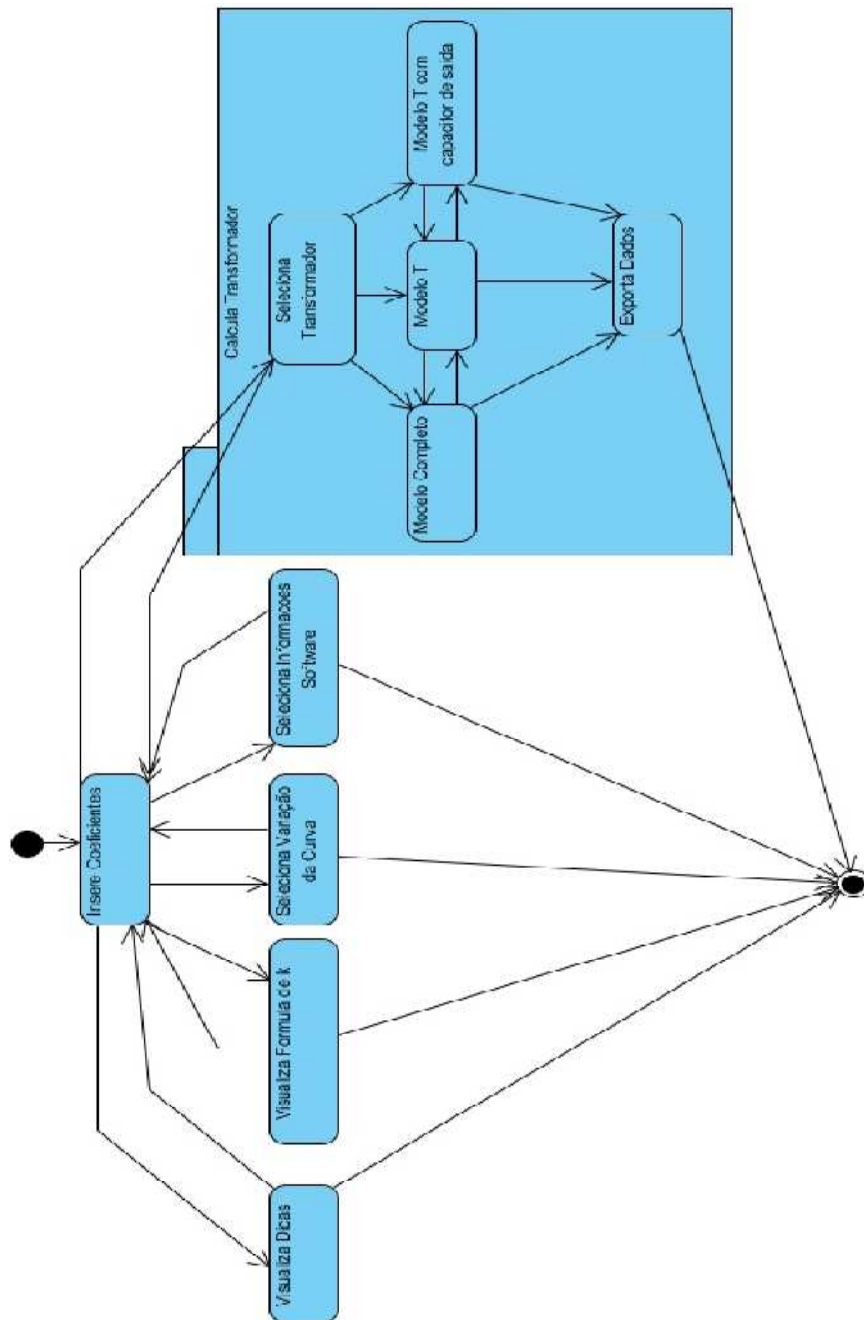


Figura 3.3 – Diagrama de estado.

3.4.4 DIAGRAMA DE CLASSES

Um diagrama de classes tem como objetivo descrever os tipos de objetos de um sistema e os vários tipos de relacionamentos estáticos que existem entre eles (FOWLER & SCOTT, 2000).

Nele estão contidos os nomes das classes, seus atributos e métodos, o diagrama de classes do *software* desenvolvido está representado pela Figura 3.4.

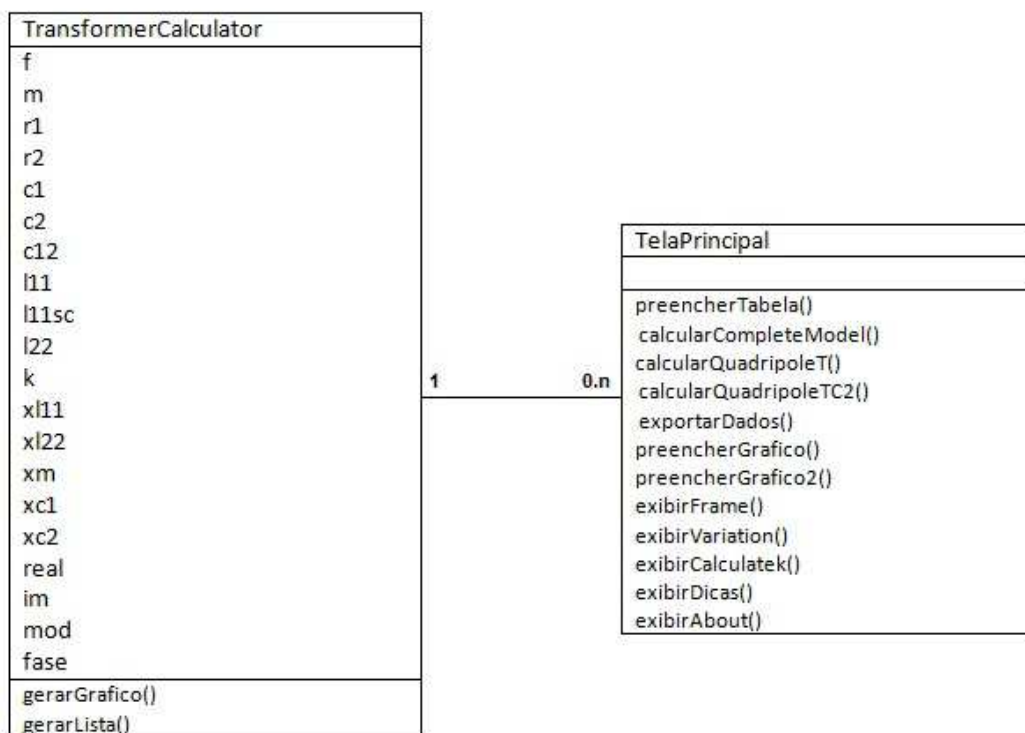


Figura 3. 4 – Diagrama de Classes do *software*.

3.5 VERSÃO EM INGLÊS

Como o objetivo deste *software* é que ele possa ser distribuído de forma gratuita e universalizada, ele está todo apresentado na língua inglesa, por se tratar de um língua considerada global, utilizada na maioria dos casos quando se espera que o maior número de pessoas possa utilizar e compartilhar um *software*.

3.6 DISPONIBILIDADE PARA *DOWNLOAD*

A versão final do *software* pode ser baixada gratuitamente através do *site*: www.pessoal.utfpr.edu.br/bertoldo/downloads_CEFET.htm, todos os arquivos necessários estão juntos em um arquivo compactado, e as instruções para sua utilização estão em um arquivo formato texto com o nome de “ReadMe”. Neste arquivo estão algumas instruções importantes, como a informação de que todos os arquivos presentes neste arquivo devem ser mantidos sempre dentro do mesmo diretório após a descompactação, para que não haja erros na execução do programa, além de informar que o arquivo a ser executado é o arquivo “DesignTransformer.jar” e algumas informações sobre a instituição onde o *software* foi desenvolvido, o nome do autor e o ano em que foi criado.

3.7 MANUAL DE INSTRUÇÕES

A fim de facilitar a utilização do *software* desenvolvido criou-se um manual de instruções, que descreve o objetivo principal do *software* e cada uma das suas funções, este manual se encontra no anexo B deste trabalho.

3.8 NOME E VERSÃO

O *software* desenvolvido neste trabalho recebe o nome de TRANSFORMER DESGINER e encontra-se na versão 1.1.

4 RESULTADOS

As telas geradas pelo *software* podem ser visualizadas neste capítulo, bem como todos os resultados numéricos e gráficos e seus comparativos com outros resultados.

4.1 ESCOLHA DOS MODELOS

O *software* desenvolvido é capaz de plotar as impedâncias de entrada (Z_{in}) em função da frequência para 3 modelos de transformadores: o modelo Delta Estrela, o modelo T clássico e o modelo T com capacitor de saída. Além disso, o usuário pode acessar instruções de como utilizar o *software*, como efetuar medidas de parâmetros e o que representa cada um dos coeficientes presentes na tela principal.

4.2 TELA PRINCIPAL

Na tela principal do *software*, Figura 4.1, em sua parte superior encontra-se a área denominada “Coeficiente values” onde existem alguns campos a serem preenchidos pelo usuário. Esses campos representam os coeficientes que o usuário deve utilizar para calcular a impedância de entrada do transformador. Inicialmente eles já recebem determinados valores a fim de facilitar a utilização do *software*. Todos os valores estão preenchidos de forma a projetar um transformador modelo Delta Estrela, pois dentre as 3 opções de transformadores este é o mais completo. À direita encontram-se os botões sobre os quais o usuário deve clicar para calcular o modelo de transformador desejado, além dos botões para visualizar as dicas e informações sobre o *software*. Na área central estão os dois espaços intitulados “CHART PANEL” onde são apresentados os resultados gráficos do módulo da impedância de entrada e da fase do transformador. À esquerda encontra-se uma tabela intitulada “CHART VALUES” onde são apresentados os valores numéricos dos pontos calculados e utilizados para gerar os resultados gráficos. Finalmente, na

parte inferior à esquerda encontra-se o botão “Export CSV format file” que é responsável por exportar os valores apresentados na tabela “CHART VALUES”.

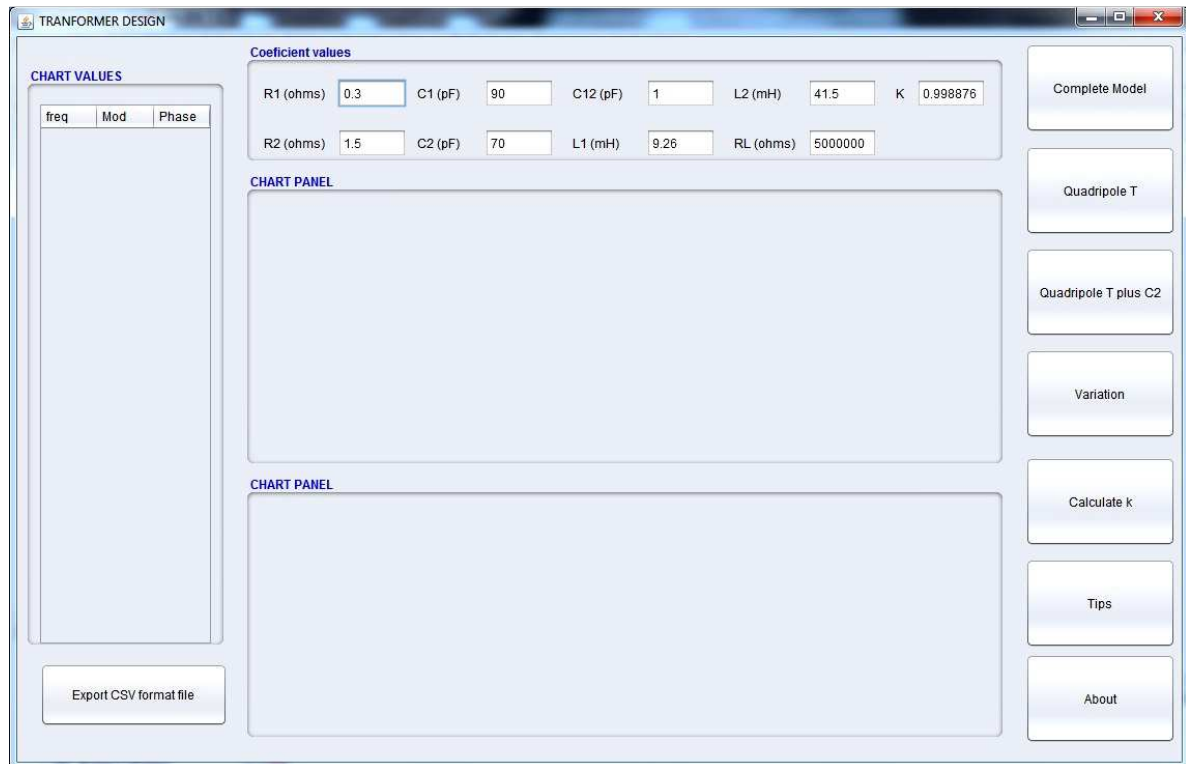


Figura 4. 1 – Tela principal do *software*.

4.3 MODELO DELTA-ESTRELA OU MODELO COMPLETO

Ao clicar no botão “Complete Model”, o *software* realiza o cálculo do valor da impedância de entrada a partir dos valores inseridos nos campos da tela principal, e automaticamente projeta na tela o gráfico do módulo de Z_{in} , em ohms pela frequência, em hertz, de 10 Hz a 100 MHz, e o gráfico da fase, em graus, pela frequência. Também cria uma tabela à esquerda da tela principal com todos os valores dos pontos X e Y utilizados para criar o gráfico. Para este exemplo os valores utilizados para os coeficientes são $r_1 = 0,3$ ohms, $r_2 = 1,5$ ohms, $C_1 = 90$ pF, $C_2 = 70$ pF, $C_{12} = 1$ pF, $L_{11} = 9,26$ mH, $L_{22} = 41,5$ mH, $R_L = 5000000$ ohms e $k = 0,998876$, valores típicos usados no trabalho de Schneider (2004) e aqui utilizados para comparação.

As equações utilizadas para realizar o cálculo de Z_{in} deste modelo são as mesmas 2.20 a 2.38. Para calcular a fase, em graus, utiliza-se a equação 2.8.

A tela gerada pelo *software* após gerar os resultados está apresentada na Figura 4.2.

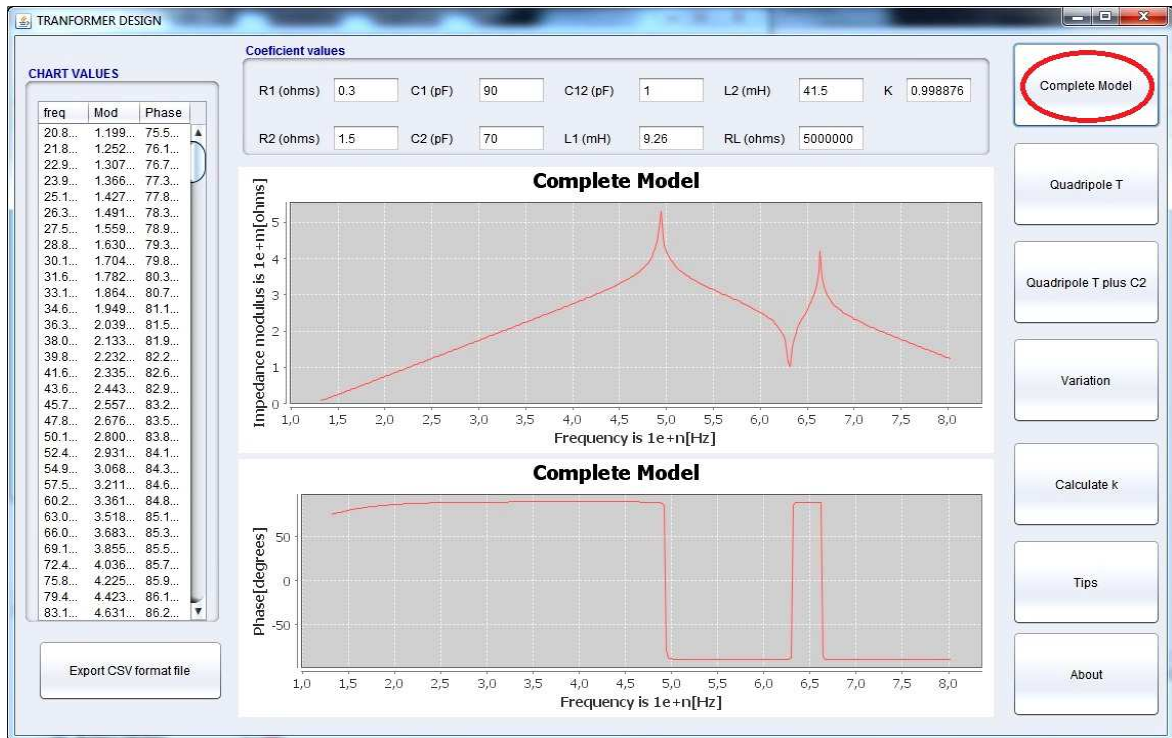


Figura 4. 2 – Tela principal após clicar no botão “Complete Model”.

Além de realizar os comandos já informados, o *software* também exibe uma nova tela ao usuário, Figura 4.3, onde estão descritos todos os coeficientes da tela inicial e o que cada um representa, neste caso todos os coeficientes são utilizados para realizar a plotagem espectral da impedância de entrada.

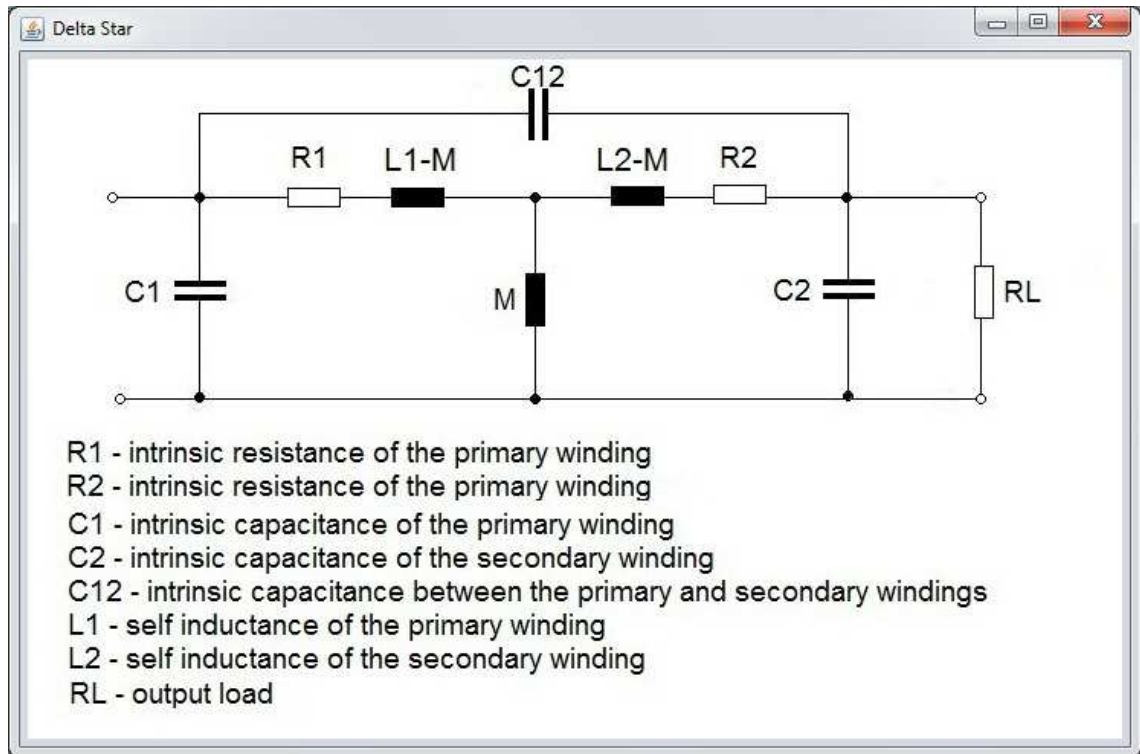


Figura 4. 3 – Tela auxiliar apresentada ao clicar no botão “Complete Model”.

4.4 MODELO T CLÁSSICO

A segunda opção disponível ao usuário é o transformador modelo T clássico, para utilizá-lo o usuário deve clicar sobre o botão “Quadripole T”, Figura 4.4, automaticamente o *software* apresenta o gráfico do módulo de Z_{in} pela frequência, e o gráfico da fase pela frequência e preenche a tabela com todos os valores utilizados para gerar o gráfico. Para este exemplo os valores utilizados para os coeficientes são $r_1 = 0,3$ ohms, $r_2 = 1,5$ ohms, $C_1 = 0$ pF, $C_2 = 0$ pF, $C_{12} = 0$ pF, $L_{11} = 9,26$ mH, $L_{22} = 41,5$ mH, $R_L = 5000000$ ohms e $k = 0,998876$.

Após calcular os resultados numéricos e gráficos a tela da Figura 4.4 é apresentada ao usuário.

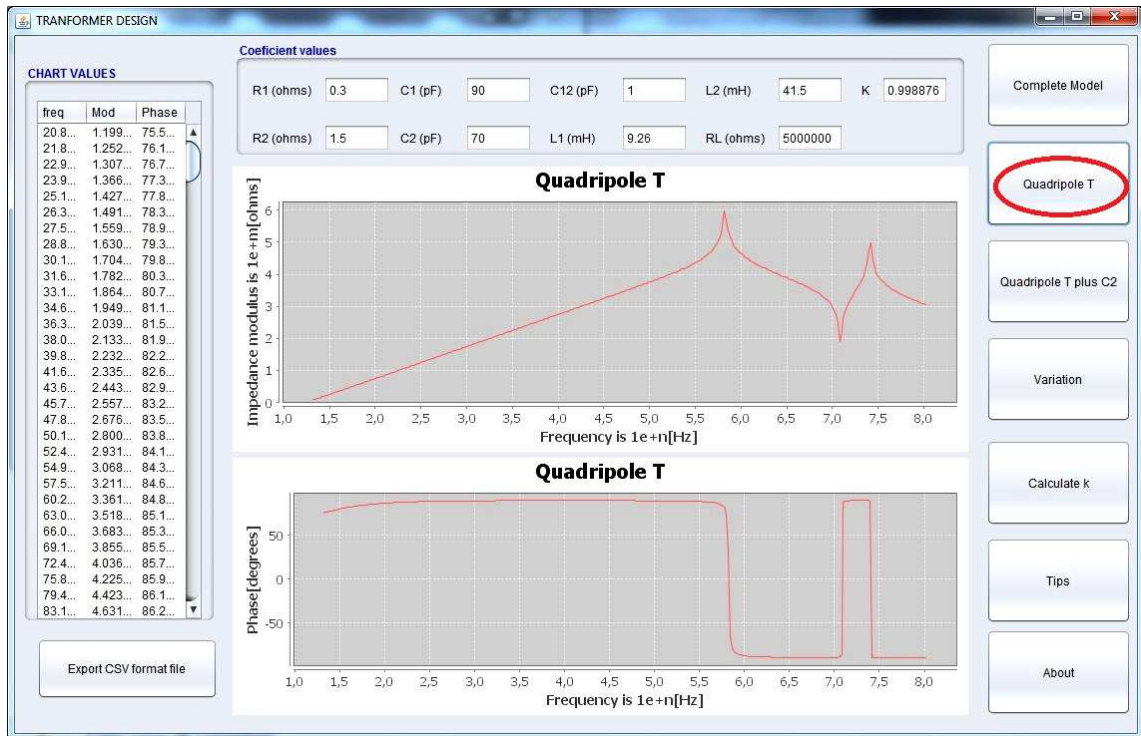


Figura 4. 4 – Tela principal após clicar no botão “Quadrupole T”.

As equações utilizadas para realizar estes cálculos são 2.14, 2.15 e 2.16. E para calcular a fase utiliza-se a equação 2.8.

Uma nova tela é apresentada ao usuário, Figura 4.5, descrevendo todos os coeficientes utilizados para calcular o valor de Z_{in} para o modelo T clássico, neste caso os valores de $C1$ e $C2$ da tela inicial são substituídos por valores muito próximos a zero (1×10^{-12}) para realizar o equacionamento deste transformador.

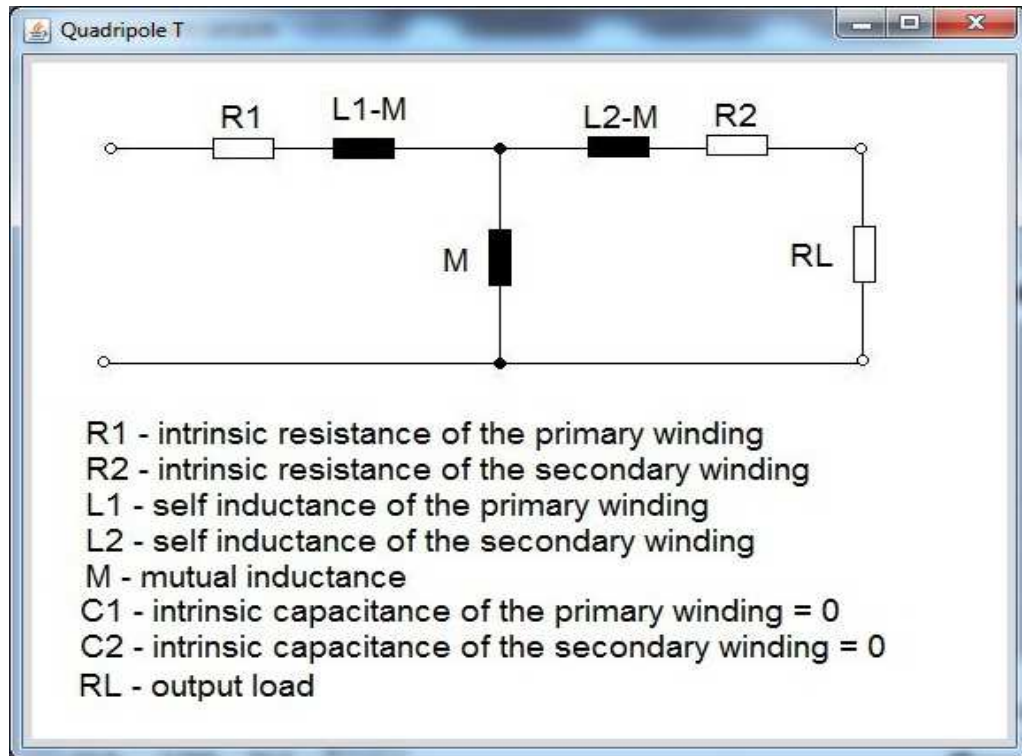


Figura 4. 5 – Tela auxiliar apresentada ao clicar no botão “Quadripole T”.

4.5 MODELO T COM CAPACITOR DE SAÍDA

A terceira opção de cálculo de transformadores é o modelo T com capacitor de saída, que é executado ao clicar no botão “Quadripole T plus C2”, Figura 4.6. Os gráficos recebem os pontos calculados, além de incluir na tabela estes mesmos valores de X e Y para cada ponto. Para este exemplo os valores utilizados para os coeficientes são $r1 = 0,3$ ohms, $r2 = 1,5$ ohms, $C1 = 0$ pF, $C2 = 70$ pF, $C12 = 0$ pF, $L_{11} = 9,26$ mH, $L_{22} = 41,5$ mH, $R_L = 0$ ohms e $k = 0,998876$.

A equação utilizada para calcular a impedância de entrada do transformador é a equação 2.19. E para calcular a fase utiliza-se a equação 2.8.

Sendo apresentado ao usuário o seguinte resultado:

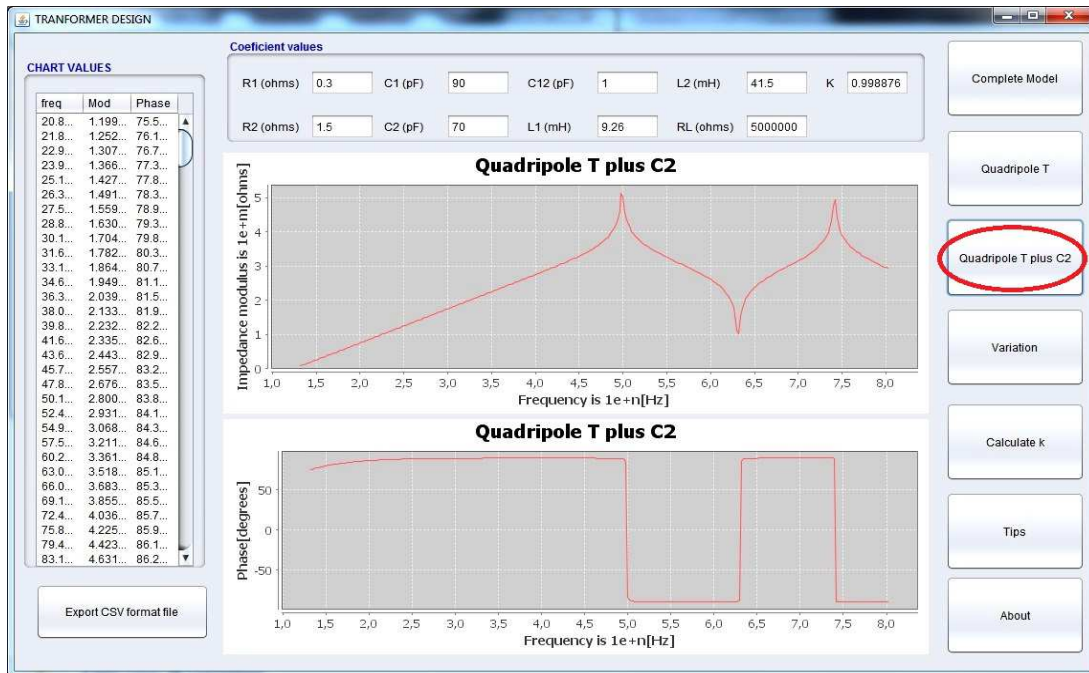


Figura 4. 6 – Tela principal após clicar no botão “Quadrupole T plus C2”.

O *software* exibe uma tela auxiliar, Figura 4.7, onde são descritos os coeficientes utilizados e também informa que para este transformador o valor de C1 é zero.

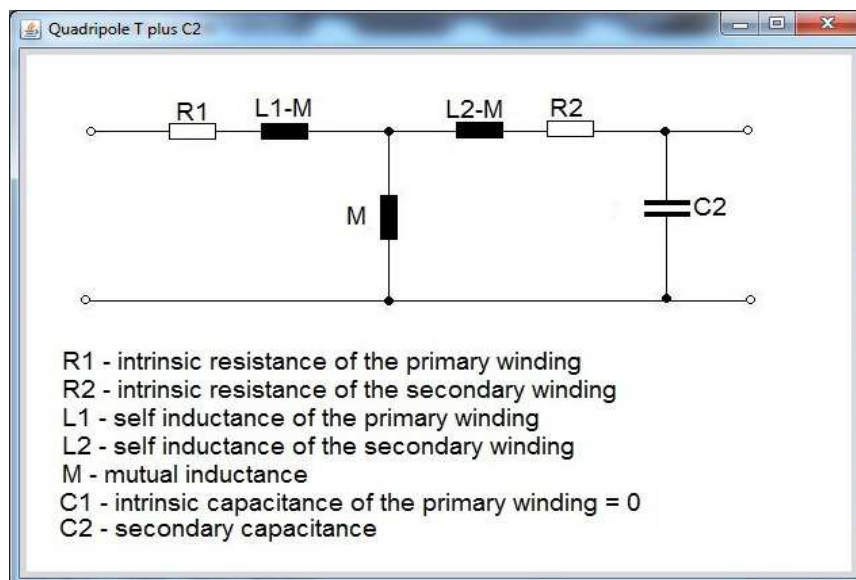


Figura 4. 7 – Tela auxiliar apresentada ao clicar no botão “Quadrupole T plus C2”.

4.6 VARIAÇÃO DE ACORDO COM OS PARÂMETROS ELÉTRICOS

Uma opção bastante interessante para analisar os possíveis resultados obtidos pelo *software* pode ser visualizada ao clicar no botão “Variation”, Figura 4.8.

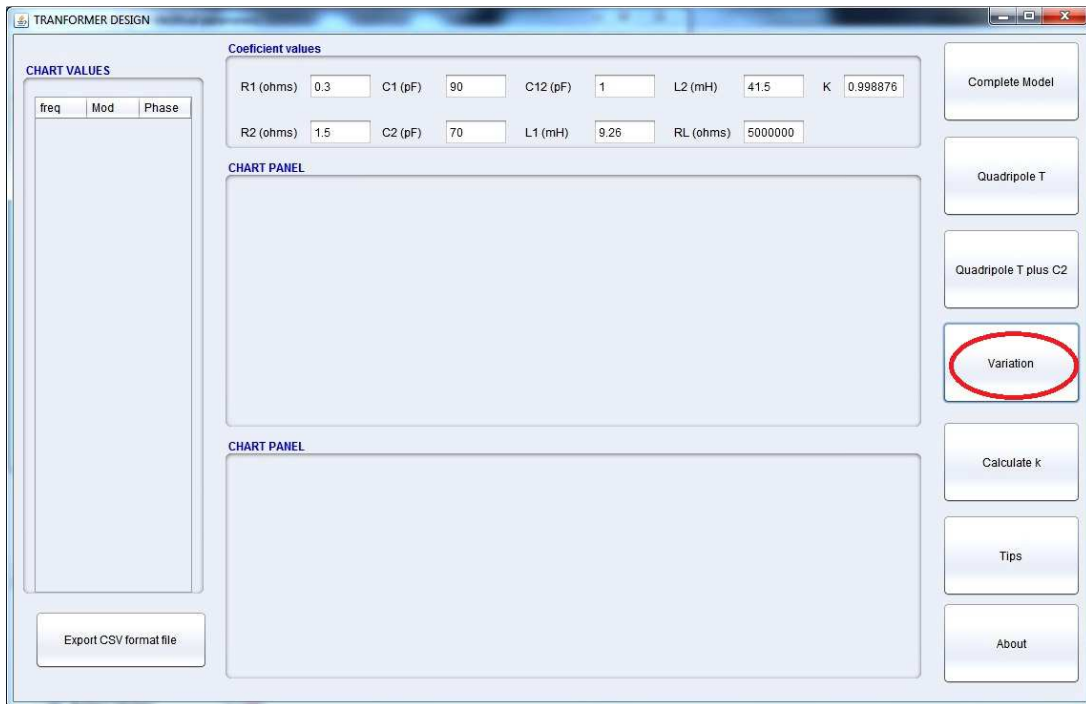


Figura 4. 8 – Seleção do botão “Variation”

Após clicar sobre o botão o *software* apresenta uma nova tela, Figura 4.9, nesta figura estão presentes todos os coeficientes utilizados para realizar os cálculos dos transformadores, e representa a modificação prevista nas curvas do gráfico, para cada um dos coeficientes, do módulo de Z_{in} (ohms) pela frequência (Hz).

Para se entender melhor a Figura 4.9, pode-se dar o seguinte exemplo, caso o valor de R_L seja aumentado. Os ápices e vales da primeira frequência de ressonância (f_{01}) e da segunda frequência de ressonância (f_{02}), respectivamente, são intensificados, mas as frequências não sofrem deslocamento no espectro. Se $C2$ aumenta, a primeira frequência de ressonância é deslocada para a esquerda, ou seja, as setas indicam em qual direção as curvas vão caso o valor de determinado coeficiente seja aumentado.

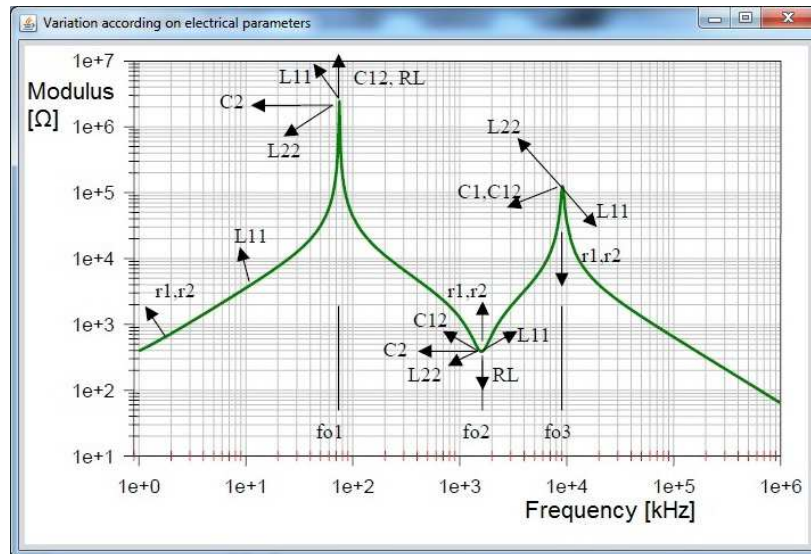


Figura 4. 9 – Tela auxiliar apresentada ao clicar no botão “Variation”, segundo gráfico de Schneider (2004)

4.7 CÁLCULO DO VALOR DE k

Outra opção presente no *software* é a descrição do método para calcular o parâmetro k, para isso, o usuário deve clicar no botão “Calculate k”, Figura 4.10.

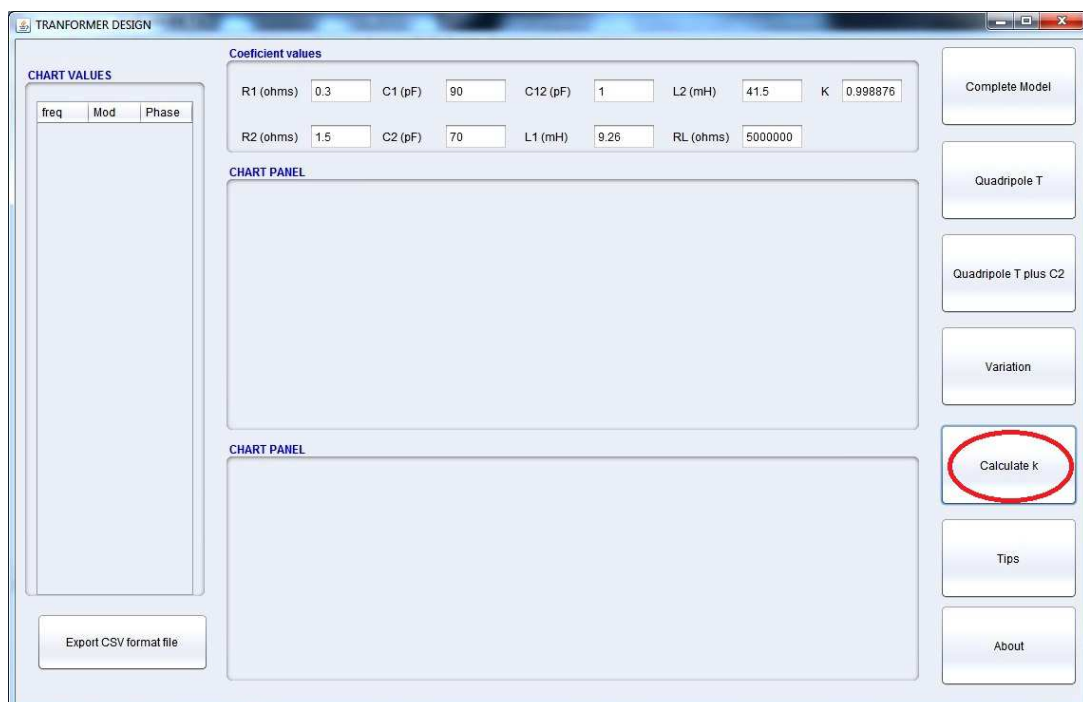


Figura 4. 10 – Seleção do botão “Calculate k”

Após clicar no botão o usuário visualiza uma pequena tela, Figura 4.11, com a fórmula desta variável, onde L_{1sc} é a indutância medida do primário com o secundário em curto e L_1 a indutância própria do enrolamento primário com o secundário em aberto, medidos preferencialmente na frequência de operação do transformador.

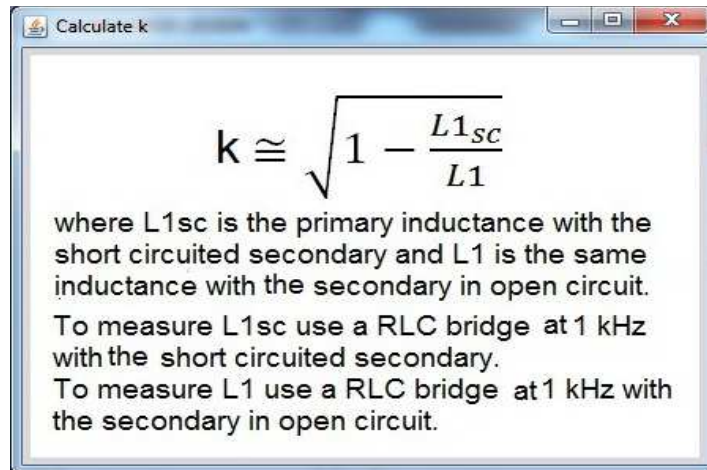


Figura 4. 11 – Tela auxiliar apresentada ao clicar no botão “Calculate k”.

4.8 INSTRUÇÕES ÚTEIS PARA PROJETAR UM TRANSFORMADOR

A fim de ajudar o usuário no projeto de um transformador algumas instruções são disponibilizadas para o usuário, para visualizar essas instruções o usuário deve clicar sobre o botão “Tips”, Figura 4.12.



Figura 4. 12 – Seleção do botão “Tips”.

Após clicar sobre o botão “Tips” duas telas são apresentadas ao usuário, elas estão representadas pelas Figura 4.13 e Figura 4.14, nesta primeira tela estão informações importantes sobre ferrites.

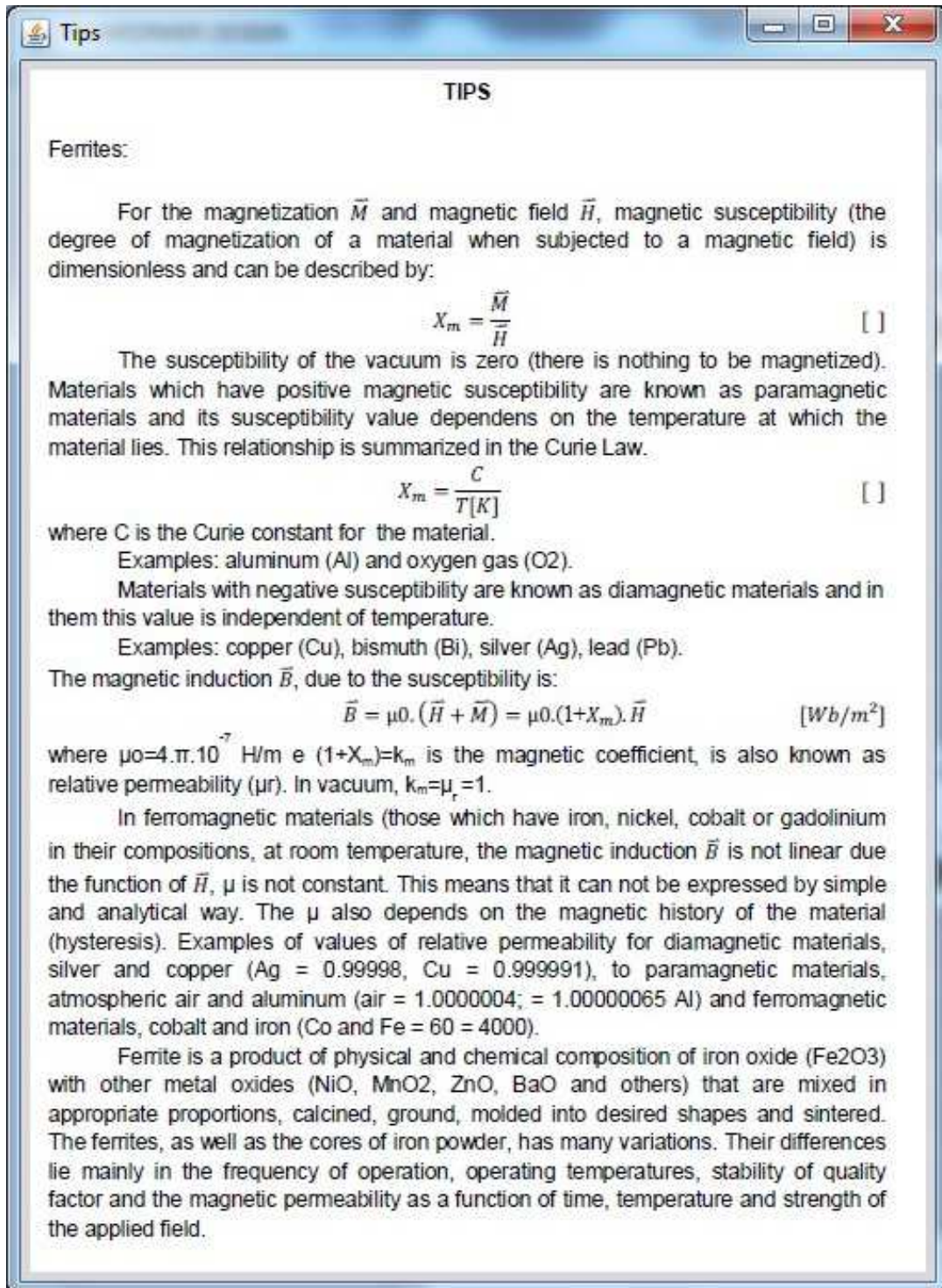


Figura 4. 13 – Primeira tela auxiliar apresentada após clicar no botão “Tips”.

Na segunda tela estão apresentadas informações sobre como medir os coeficientes do transformador, e algumas dicas sobre enrolamentos, por exemplo, como fazer para diminuir os valores das capacitâncias do transformador, Figura 4.14.

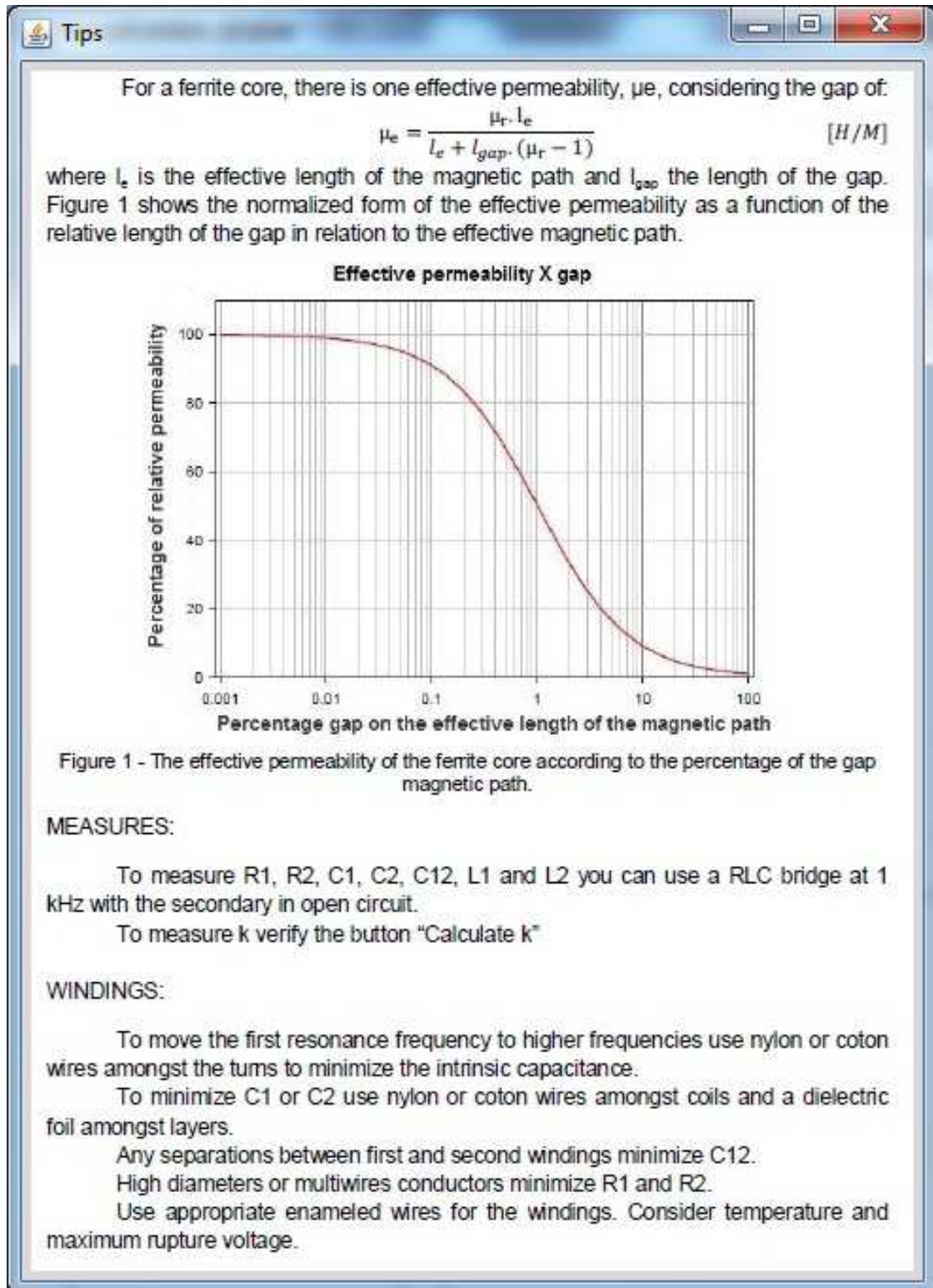


Figura 4. 14 - Segunda tela auxiliar apresentada após clicar no botão "Tips".

4.9 INFORMAÇÕES SOBRE O SOFTWARE

Para saber informações sobre os autores, a instituição e o programa de pós-graduação onde foi desenvolvido o *software* o usuário pode clicar sobre o botão “About”, Figura 4.15.

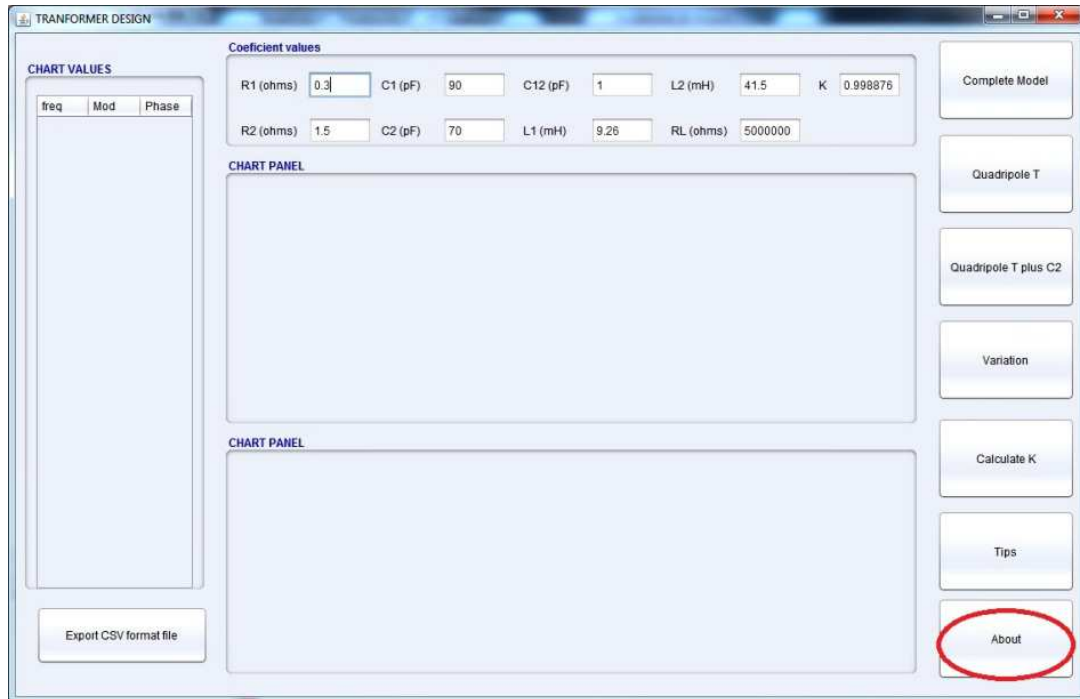


Figura 4. 15 – Seleção do botão “About”.

O *software* apresenta a tela mostrada na Figura 4.16.



Figura 4. 16 – Tela apresentada ao clicar no botão “About”.

4.10 SALVANDO DADOS

Após verificar todas as possibilidades de modelos transformadores e escolher qual atende melhor suas necessidades, o usuário pode exportar os pontos calculados pelo *software*, do módulo da impedância de entrada e da fase pela frequência, clicando no botão “Export CSV format file”. O arquivo gerado está no formato CSV e pode ser aberto pelo MS Word®, MS Excel® e outros *softwares*. O usuário pode escolher em qual diretório deseja salvar o arquivo e deve atribuir um nome para este arquivo, Figura 4.17.

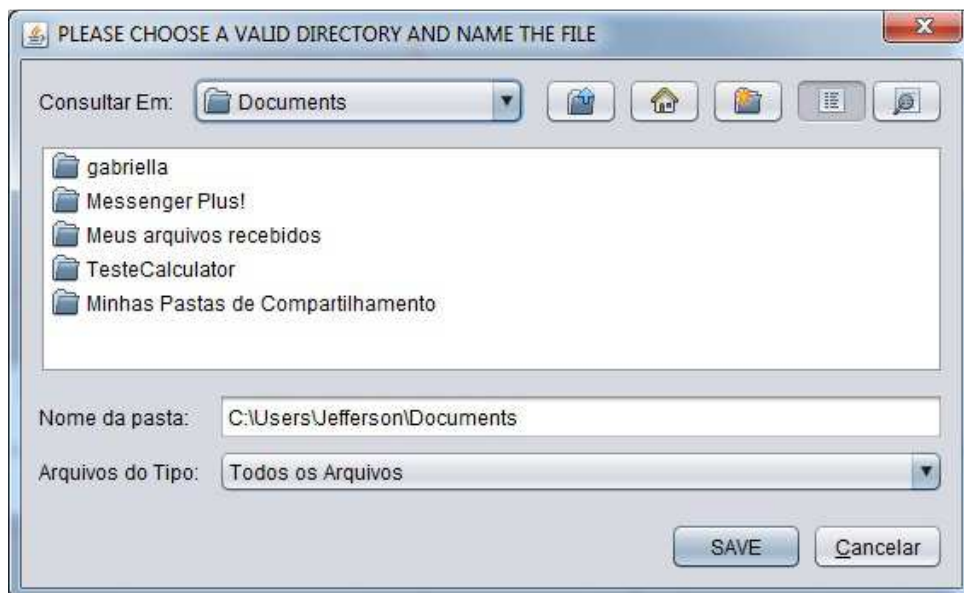


Figura 4. 17 – Tela apresentada ao clicar no botão “Export CSV format file”.

A Figura 4.18 exemplifica um trecho dos resultados obtidos pelo modelo Delta-Estrela com os seguintes valores de coeficientes: $r_1 = 0,3$ ohms, $r_2 = 1,5$ ohms, $C_1 = 90$ pF, $C_2 = 70$ pF, $C_{12} = 1$ pF, $L_{11} = 9,26$ mH, $L_{22} = 41,5$ mH, $R_L = 5000000$ ohms e $k = 0,998876$. O *software* utilizado para abrir o arquivo gerado foi o MS Word®.

```

Frequency:Modulus:Phase
20.892961308540407;1.1990267474001801;75.51041953788435
21.87761623949554;1.2520724657793865;76.13684205344948
22.90867652767775;1.307765234029564;76.73824506112135
23.988329190194925;1.3662242568277407;77.315373203464
25.11886431509582;1.4275744795003267;77.86898135229005
26.302679918953842;1.4919468672358216;78.39982954260915
27.54228703338169;1.559478695650874;78.90867853752155
28.840315031266087;1.6303138534338086;79.39628597734071
30.199517204020193;1.7046031578143137;79.86340306518461
31.622776601683825;1.7825046836324274;80.31077174147568
33.11311214825915;1.8641841068044087;80.73912230093913
34.6736850452532;1.9498150630084765;81.14917140753926
36.30780547701018;2.0395795224399036;81.54162046513736
38.01893963205617;2.1336681815127587;81.91715430431424
39.81071705534978;2.2322808724153367;82.27644014864947
41.68693834703359;2.33562699145788;82.62012682666624
43.651583224016655;2.443925947185206;82.9488441985709
45.708818961487566;2.5574076292626597;83.2632027697453
47.863009232263906;2.6763128991826335;83.563793465692
50.1187233627273;2.800894103880114;83.85118754570033
52.480746024977336;2.931415613389259;84.12593663492675
54.954087385762534;3.0681543837201826;84.38857285681756
57.54399373371578;3.2114005461841857;84.63960904986459
60.25595860743587;3.3614580244482912;84.87953905456637

```

Figura 4. 18 – Arquivo CSV gerado pelo *software* e importado pelo MS Word®.

Este mesmo arquivo foi importado pelo MS Excel® gerando os seguintes gráficos, Figura 4.19, o gráfico de cima representa a impedância de entrada (coluna 2 da Figura 4.18) pela frequência (coluna 1 da Figura 4.18) e o gráfico de baixo representa a fase (coluna 3 da Figura 4.18) pela frequência:

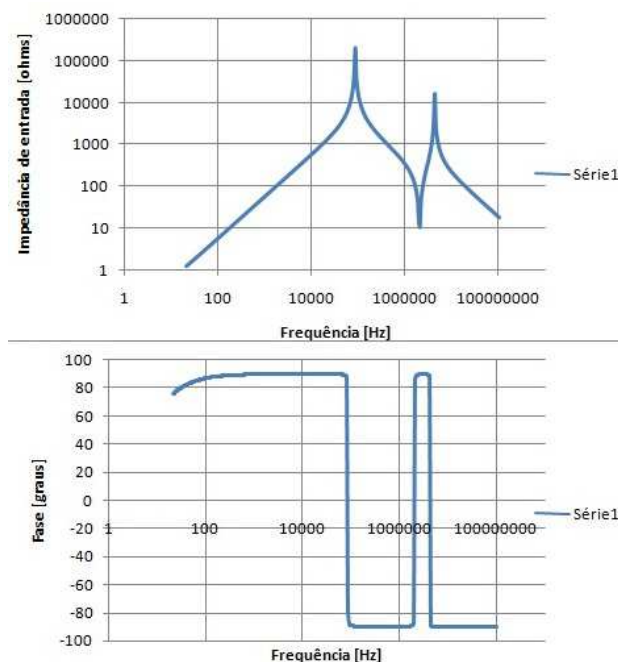


Figura 4. 19 – Arquivo CSV gerado pelo *software* e importado pelo MS Excel®, resultado utilizando o recurso gráfico deste *software*.

4.11 COMPARAÇÃO DE RESULTADOS ENTRE MODELOS

Para se ter uma ideia melhor de quais são as diferenças das curvas de resposta dos três modelos de transformadores cobertos pelo *software* desenvolvido, gerou-se um único gráfico com as curvas sobrepostas, Figura 4.20. Para os três modelos utilizaram-se os mesmos valores de coeficientes: $r_1 = 0,3$ ohms, $r_2 = 1,5$ ohms, $C_1 = 90$ pF, $C_2 = 7$ pF, $C_{12} = 1$ pF, $L_{11} = 9,26$ mH, $L_{22} = 41,5$ mH, $R_L = 5000000$ ohms e $k = 0,998876$. Lembrando que para o transformador modelo T os valores de C_1 , C_2 e C_{12} são zero e que para o transformador modelo T com capacitor de saída os valores de C_1 e C_{12} são zero. Mais uma vez a ferramenta utilizada para importar o arquivo CSV e gerar o gráfico foi o MS Excel®.

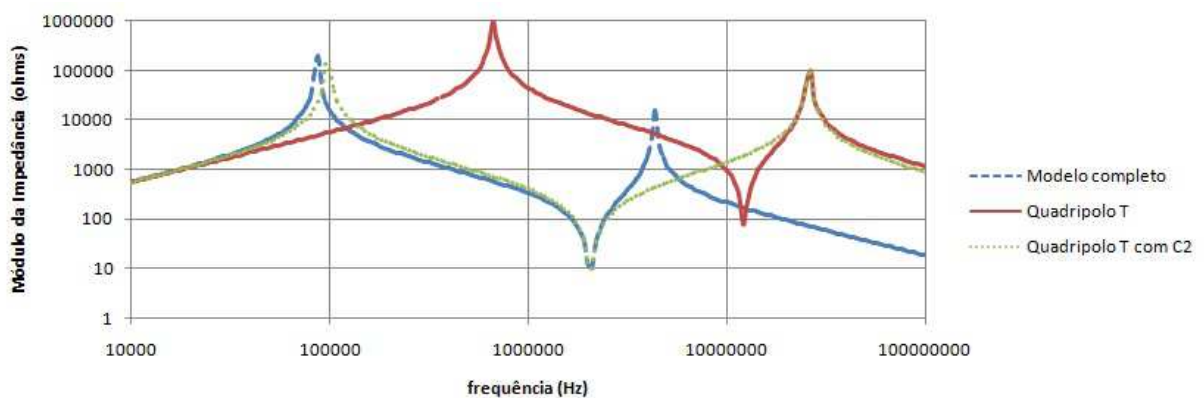


Figura 4. 20 – Sobreposição das curvas de impedância de entrada dos três modelos. A linha tracejada representa o modelo completo, a linha cheia representa o quadripolo T e a linha formada por pontos representa o quadripolo T com capacitor de saída.

4.12 COMPARAÇÕES COM EWB

Para realizar a comparação dos resultados obtidos, utilizou-se o *software* EWB (Electronics Workbench®), onde foi criado um circuito de um transformador modelo Delta-Estrela, Figura 4.21, foi utilizado no EWB os seguintes valores: $r_1 = 0,3$ ohms, $r_2 = 1,5$ ohms, $C_1 = 90$ pF, $C_2 = 70$ pF, $C_{12} = 1$ pF, $L_{11} = 30$ mH, $L_{22} = 40$ mH, $R_L = 5000000$ ohms e $M = 20$ mH.

Através da utilização da ferramenta “Bode Ploter” analisou-se a curva do ganho de V_o (tensão de saída) por V_i (tensão de entrada), criou-se uma grande carga na entrada do transformador (100 M Ω) ligada a uma fonte de tensão, desta

forma o resultado gerado apresenta a morfologia da impedância deste transformador.

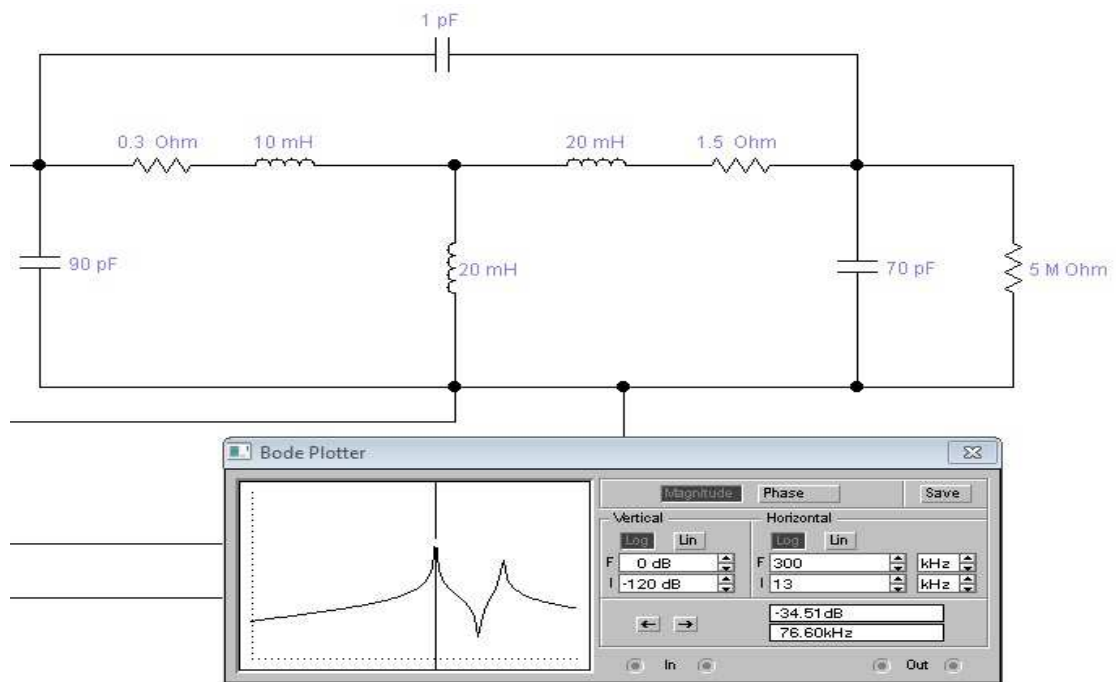


Figura 4. 21 – Resultados obtidos através do software Eletronics Workbench®.

O resultado gerado pelo EWB para os valores de coeficientes anteriormente citados e apresentados na Figura 4.21 foram importados pelo Excel para comparação com o resultado gerado pelo *software* desenvolvido neste trabalho. Vale lembrar que no *software* não é inserido o valor para a indutância mútua (M), mas sim o coeficiente de acoplamento (k), tendo sido neste caso utilizado o valor 0,5774, sendo então possível realizar esta comparação, Figura 4.22, além disso, ambas as impedâncias estão normalizadas para facilitar a comparação entre ambos resultados.

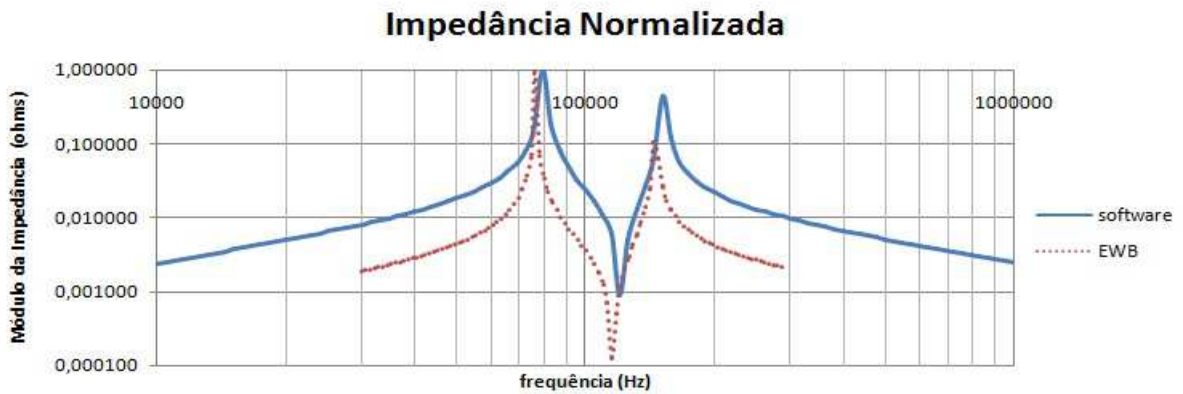


Figura 4. 22 – Resultado gráfico da impedância de entrada (normalizada) plotada no Excel, comparando o resultado do software (linha cheia) e do EWB (ponto).

Para este modelo de transformador os valores gerados pelo *software* da primeira, segunda e terceira frequência de ressonância foram 79,43 kHz, 120,23 kHz e 151,36 kHz, respectivamente. Já para o EWB estas frequências de ressonância foram 76,39 kHz, 115,15 kHz e 144,64 kHz, respectivamente.

Com base nos mesmos valores dos coeficientes utilizados para modelar o transformador da Figura 4.21 criou-se um novo transformador alterando apenas o valor de R_L para 5000 ohms. Comparou-se o resultado gerado pelo *software* para estes mesmos valores de coeficientes, e chegou-se na seguinte resposta, Figura 4.23.

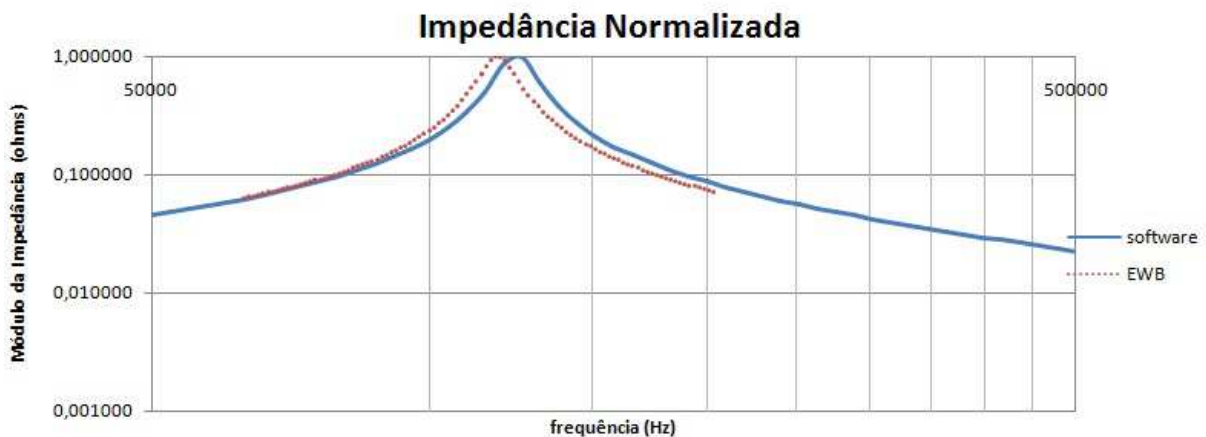


Figura 4. 23 – Resultado gráfico comparando o resultado obtido pelo *software* (linha cheia) e pelo EWB (ponto) para o módulo da impedância de entrada normalizada.

Neste caso o resultado do software para a primeira frequência de ressonância foi 125,89 kHz, já para o EWB este valor foi 117,80 kHz.

Vários exemplos de transformadores foram simulados a fim de se obter o erro percentual relativo sobre a primeira frequência de ressonância, utilizou-se sempre como base os seguintes valores de coeficientes: $r_1 = 0,3$ ohms, $r_2 = 1,5$ ohms, $C_1 = 90$ pF, $C_2 = 70$ pF, $C_{12} = 1$ pF, $L_{11} = 30$ mH, $L_{22} = 40$ mH e $M = 20$ mH. Neste primeiro caso, Tabela 4.1, variou-se apenas o valor da carga de saída (R_L). A primeira coluna da tabela apresenta o valor utilizado para a carga de saída, a segunda apresenta o resultado da f_{01} do *software*, a terceira coluna apresenta o valor da f_{01} do EWB e a quarta coluna o erro percentual relativo.

Tabela 4. 1 – Cálculo do erro percentual relativo, variando apenas o valor de R_L .

Valor de R_L (Ω)	f_{01} do <i>software</i> (kHz)	f_{01} do <i>EWB</i> (kHz)	Epr (%)
50000	79,43	77,83	2,01
100000	79,43	76,59	3,56
500000	79,43	78,52	1,15
1000000	79,43	76,59	3,56
5000000	79,43	76,39	3,82
10000000	79,43	76,59	3,56

Neste caso a média do erro percentual relativo foi de 2,94 %. Valores abaixo de 50 k Ω para a carga de saída são difíceis de serem comparados, pois a primeira frequência de ressonância tende a se aplainar, complicado a identificação do valor de pico.

Para um segundo caso utilizou-se como base os mesmos valores de coeficientes, alterando-se apenas um dos valores dos coeficientes, Tabela 4.2.

Tabela 4. 2 – Cálculo do erro percentual relativo, variando todos os coeficientes

Coeficiente alterado/ valor recebido	f_{01} do <i>software</i> (kHz)	f_{01} do <i>EWB</i> (kHz)	Epr (%)
$C_1 = 900$ pF	31,62	30,12	4,76
$C_2 = 700$ pF	31,62	29,19	7,68
$C_{12} = 100$ pF	75,86	71,98	5,12
$L_1 = 100$ mH	54,95	51,85	5,65
$L_2 = 400$ mH	30,20	29,19	3,33

$R_1 = 30 \Omega$	79,43	76,60	3,57
$R_2 = 150 \Omega$	79,43	76,60	3,57
$R_L = 50000 \text{ k}\Omega$	79,43	77,83	2,01

Para o segundo caso a média do erro percentual relativo foi de 4,46 %.

Os resultados gerados não foram exatamente os esperados, pois durante as comparações realizadas com o EWB notou-se que, este *software* não é capaz de analisar alguns valores de coeficientes utilizados no *software* desenvolvido neste trabalho. Quando se utilizam valores de coeficiente de acoplamento próximos a 1 (valores de transformadores reais), há a tendência que a indutância de entrada e/ou a indutância de saída se tornem negativas através da subtração de M (pelo modelo usado). Valores negativos não são aceitos pelo EWB por se tratar de um *software* que utiliza modelos *spice* de componentes.

Observou-se que em comparação ao EWB há uma grande discrepância. Como o modelo foi verificado em trabalho anterior (SCHNEIDER, 2004), apresentando grande aderência com levantamentos feitos através de vários métodos, infere-se que o EWB não é um programa de grande confiabilidade para o caso do modelo estudado.

Os resultados comparados com o EWB foram insatisfatórios, apresentando erros maiores que 4,46 % apontando uma deficiência do mesmo.

5 CONCLUSÃO

A primeira contribuição do trabalho aqui apresentado é a implementação de um *software* capaz de auxiliar um usuário no desenvolvimento de transformadores, através de dicas e dos modelos de transformadores pré-estabelecidos no *software*. O *software* é mais exato do que simuladores tradicionais, tais como o EWB.

Outra contribuição deste trabalho é a possibilidade de utilização do *software* por pessoas conhecedoras da teoria e da prática de transformadores, mas que necessitam otimizar seu tempo e/ou comparar resultados práticos com resultados teóricos, a fim de validar seus experimentos.

Estas contribuições são muito importantes, pois ainda não se conhece no meio acadêmico *software* que apresente modelos pré-estabelecidos de transformadores e que gere apenas com os valores dos coeficientes resultados numéricos e gráficos do módulo da impedância de entrada e da fase.

Uma função bastante interessante deste *software* é sua capacidade de exportar os resultados gerados em arquivos no formato CSV, podendo ser então importados por outros *softwares* de análise numérica e gráfica, facilitando assim, as possíveis comparações com outros valores, sejam eles valores reais ou também teóricos.

Os resultados comparativos com o EWB não foram tão bons quanto se esperava, mesmo estando com média de 4,46 % de erro percentual relativo em relação ao EWB. Durante a comparação entre o *software* e o EWB, usando o modelo do transformador Delta-Estrela descobriu-se que o mesmo não aceita valores de indutância negativos, impossibilitando comparações com k maiores que 0,5, por exemplo, impossibilitando uma boa comparação com os resultados obtidos pelo *software* desenvolvido neste trabalho. Mesmo quando um conjunto de valores possibilita esta comparação, notou-se muita diferença tanto na posição espectral dos picos e vales quanto nas intensidades dos mesmos. Como o modelo já foi verificado e comparado com sucesso com equipamentos padrão e outras formas de obtenção de dados, a conclusão que resta é a de que o EWB, que é o principal simulador de circuitos eletrônicos, é muito deficiente, pelo menos quando se está querendo simular modelos de transformadores reais.

5.1 DISCUSSÕES

Dentre os modelos estudados e cobertos pelo *software* o modelo Delta-Estrela é considerado o mais completo, contendo em si mesmo o modelo T clássico e o modelo T com capacitor de saída. Para se chegar ao modelo T a partir do Delta-Estrela basta considerar o valor de R_L como um número muito grande, tendendo ao infinito, e que os valores de $C1$, $C2$ e $C12$ são próximos a zero. Já para se chegar ao modelo T com capacitor de saída, deve-se considerar no modelo Delta-Estrela o valor de R_L tendendo ao infinito e os valores de $C1$ e $C2$ muito próximo de zero.

Durante a realização da verificação do equacionamento do modelo Delta-Estrela algumas dificuldades foram encontradas, por se tratar do modelo de transformador com a equação da impedância de entrada mais complexa, mas a principal dificuldade foi durante a verificação do valor de R_L , pois quando este valor é muito baixo a curva da impedância tende a se aplainar, o que atrapalhou e dificultou muito a comparação com estudos de transformadores (SCHNEIDER, 2004).

Outro ponto bastante importante a ser discutido é o fato de em nenhum dos modelos considerarem no equacionamento a permeabilidade magnética, por levar em consideração que os valores a serem utilizados pelo transformador vão até a segunda frequência de ressonância.

Os modelos T com magnetização e perda e o modelo Gama de transformadores apesar de serem estudados neste trabalho não são apropriados para equipamentos biomédicos, por apresentarem perdas significativas, por isso não são abrangidos pelo *software*.

Um problema verificado durante o desenvolvimento do *software* foi a disposição das telas apresentadas ao usuário, pois em determinadas situações elas ficam sobrepostas umas às outras, tornando para o usuário difícil a visualização de mais de uma tela ao mesmo tempo.

5.2 TRABALHOS FUTUROS

Futuramente deseja-se corrigir problemas como a sobreposição das telas, além disso, estuda-se a possibilidade de inclusão de novos modelos de

transformadores, aumentando a usabilidade do *software*, para que isto aconteça serão geradas novas versões do *software*.

Deseja-se considerar também a influência da permeabilidade complexa nos equacionamentos, a fim de abranger novas aplicabilidades.

REFERÊNCIAS

BLACHE, F., KERADEC, J.P. and COGITORE, B., (1994), **Stray Capacitances of two winding Transformers: Equivalent Circuit, Measurements, Calculation and Lowering**, Industry Applications Society Annual Meeting, 1994, Conference Record of 1994, IEEE, 2-6out 1994, p1211-1217, V2.

BOENTE, A., (2003), **Aprendendo a programar em JAVA 2: orientado a objetos**. Rio de Janeiro, RJ, Brasport, 2003, p125.

BRAGA, N. C., (2012), **Curso de Eletrônica – Eletrônica Básica**, São Paulo, Newton C. Braga, p184-185.

BRAGA, N. C., (2012). Disponível em <http://www.newtoncbraga.com.br>. Acessado em agosto de 2012.

DALL'OGGIO, P. **PHP Programando com Orientação a Objetos**. 1 ed. São Paulo: Novatec, 2007.

EDMINISTER, J. A., (1991), **Circuitos Elétricos**, São Paulo, Makron, McGraw-Hill do Brasil, p365-373.

FILCHEV, T., CARASTRO, F., (2010), **High Voltage High Frequency Power Transformer for Power Application**, In: Proceedings of the 14th International Power Electronics and Motion Control Conference, Ohrid, 6-8 Sept.

FOWLER, M., SCOTT, K., **UML essencial: um breve guia para a linguagem padrão de modelagem de objetos**. 2. Ed, Porto Alegre, RS, Bookman, p32.

GUSSOW, M., (1985), **Eletricidade Básica**. 1. Ed, São Paulo, Mc Graw-Hill do Brasil, p20-22.

GNU General Public License. Disponível em <http://www.gnu.org>. Acessado em Julho de 2012.

HARA, M. S., (2003), **Desenvolvimento de uma unidade implantável não ressonante com dois componentes para sistema de biotelemetria passiva**, Dissertação de Mestrado, Universidade Tecnológica Federal do Paraná, 2003.

HAYT, W. H., KEMMERLY, J. E., (1975), **Análise de Circuitos em Engenharia**, São Paulo, McGraw-Hill do Brasil, p413-442.

HAYT, W. H. (1983). Eletromagnetismo, 1ª Edição de 1978, Rio de Janeiro, RJ, Livros Técnicos e Científicos Editora S.A., p403.

LaCOURSE, J. R., VOGT, M. C., MILLER, W. T. & SELIKOWITZ, S. M., (1988), **Spectral Analysis Interpretation of Electrosurgical Generator Nerve and Muscle Stimulation**, IEEE Trans. Biomed. V.35, No 7, July, 1988, p505-509.

LINDHOLM, T., YELLIN, F., (1999). **The Java Virtual Machine Specification**. Sun Microsystems Inc. 1999. Disponível em <http://www.javaseries.com>. Acessado em Agosto de 2012.

LOPES, A., (2008), **Análise de Circuitos e Formas de Onda**, São Paulo, Eccel Eletrônica, p6-11.

MORIMOTO, C. E. (2005). Disponível em <http://www.hardware.com.br/termos/bytecode>. Acessado em Agosto de 2012.

NETBEANS (2000). Disponível em http://netbeans.org/index_pt_PT.html. Acessado em Agosto de 2012.

NBR IEC 60601-2-2:2001, (2001), **Equipamento Eletromédico, Parte 2: Prescrições particulares de Segurança para Equipamento Cirúrgico de Alta Frequência**, ABNT, Associação Brasileira de Normas Técnicas, em dezembro de 2001, válida a partir de 30 de janeiro de 2002, 30p.

NBR IEC 60601-1, (1994), **Equipamento Eletromédico, Parte 1: Prescrições Gerais para segurança** (equivalente à norma IEC 601-1, 1988), ABNT, Associação Brasileira de Normas Técnicas, 149p.

RIBEIRO, L. (2012). **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Disponível em <http://www.devmedia.com.br>. Acessado em Outubro de 2012.

ROCHA NETO, E. (2009). **Material sobre UML**. Disponível em <http://www.dsc.ufcg.edu.br>. Acessado em Outubro de 2012.

SCHNEIDER Jr., B., DIAS Jr., E., ABATTI, P. J., (2010), **How can Electrosurgical Sparks generate Undesirable Effects?**. In: IEEE Latin America Symposium on Circuits and Systems, Foz do Iguaçu, PR. 1st IEEE Latin American Symposium on Circuits and Systems Proceedings. v. 1. p. 112-115.

SCHNEIDER Jr., B., ABATTI, P. J., (2008), **Electrical Characteristics of the Sparks Produced by Electrosurgical Devices**. IEEE Transactions on Biomedical Engineering, v. 55, p. 589-593.

SCHNEIDER Jr., B. **Estudo Teórico-Prático de Parâmetros Técnicos e Fisiológicos Utilizados em Eletrocirurgia, visando a Otimização do Desenvolvimento e Performance de um Bisturi Eletrônico**. 2004. 252f. Tese (Doutorado em Engenharia Biomédica) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná.

SEARS, F. W. & ZEMANSKI, M. W., **Física, Eletricidade, Magnetismo e tópicos de Física Moderna**, Livros Técnicos e Científicos Editora S.A., 1980.

SICA, C. **PHP Orientado a Objetos - Fale a Linguagem da Internet**. 1 ed. Rio de Janeiro: Ciência Moderna, 2006.

SILVA, J. T. **Genetic transcript analyzer -ferramenta computacional para análise de transcrição gênica por rna –seq**. 2012. 64f. Dissertação (Mestrado em Bioinformática) – Programa de Pós-Graduação em Bioinformática, Universidade Federal do Paraná.

SIQUEIRA, R., (2012). **UNICODE – Letras, caracteres e símbolos mais comuns**. Disponível em <http://rsiqueira.postbit.com/>. Acessado em Novembro de 2012.

SMITH, K. C. A. & ALLEY, R. E., (1992), **Electrical Circuits, an Introduction**, Cambridge U. Press, Chap.8, p146-148.

SONTAG (1985), **Materias magnéticos para eletrônica e eletrotécnica**, Empresa R. SONTAG, agosto de 1985 p40.

SOUZA, M. (2011). **Diagrama de Estado (UML)**. Disponível em <http://techblog.desenvolvedores.net/>. Acessado em Outubro de 2012.

VENNERS, B. (1998). **The State of the Java Virtual Machine**. Disponível em <http://www.javaworld.com/javaworld>. Acessado em Agosto de 2012.

WEBSTER, J. G., (1988), **Encyclopedia of Medical Devices and Instrumentation**, 1988, Vol. 2, John Wiley & Sons, p1180-1203.

ANEXO A

CÓDIGO FONTE

```

package screens;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.List;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
/** @author Jhonisson Ricardo de Paula
public class Telapincipal extends javax.swing.JFrame {
    JFrame frame = null;
    public Telapincipal() {
        initComponents();
    }
    private void preencherTabela(List<double[]> lista) {
        DefaultTableModel modelo = (DefaultTableModel) valuesTable.getModel();
        while (modelo.getRowCount() > 0) {
            modelo.removeRow(0);
        }
        for (double[] ds : lista) {
            modelo.addRow(new Object[]{ds[0], ds[1], ds[2]});
        }
    }
    private void calcularCompleteModel (java.awt.event.ActionEvent evt) {
        List<double[]> listaparaGrafico =
transformercalculator.TransformerCalculator.gerarLista(Double.parseDouble(r1Coef.getText()),
Double.parseDouble(r2Coef.getText()), Double.parseDouble(c1Coef.getText()) * Math.pow(10, -12),
        Double.parseDouble(c2Coef.getText()) * Math.pow(10, -12),
Double.parseDouble(c12Coef.getText()) * Math.pow(10, -12), Double.parseDouble(l1Coef.getText())
* Math.pow(10, -3), Double.parseDouble(l2Coef.getText()) * Math.pow(10, -3),
Double.parseDouble(rlCoef.getText()), Double.parseDouble(kCoef.getText()));
        preencherTabela(listaparaGrafico);
        JFreeChart chart = transformercalculator.TransformerCalculator.gerarGrafico("Complete
Model", "Abscissa n where frequency is 1e+n[Hz]", "m where impedance module is 1e+m[ohms]",
false, listaparaGrafico, true);
        JFreeChart chart2 = transformercalculator.TransformerCalculator.gerarGrafico("Complete
Model", "Abscissa n where frequency is 1e+n[Hz]", "Phase[degrees]", false, listaparaGrafico, false);
        exibirFrame("Delta Star", "deltaestrela.jpg");
        preencherGrafico(chart);
        preencherGrafico2(chart2);
    }
}

```

```

private void r2CoefActionPerformed(java.awt.event.ActionEvent evt) {}
private void c2CoefActionPerformed(java.awt.event.ActionEvent evt) {}
private void l1CoefActionPerformed(java.awt.event.ActionEvent evt) {}
private void l2CoefActionPerformed(java.awt.event.ActionEvent evt) {}
private void r1CoefActionPerformed(java.awt.event.ActionEvent evt) {}
private void exportarDados (java.awt.event.ActionEvent evt) {
    JFileChooser fileChooser = new JFileChooser();
    fileChooser.setSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    fileChooser.setApproveButtonText("SAVE");
    fileChooser.setDialogTitle("PLEASE CHOOSE A VALID DIRECTORY AND NAME THE FILE");

    int option = fileChooser.showOpenDialog(null);
    if (option == JFileChooser.APPROVE_OPTION) {
        StringBuilder sb = new StringBuilder("Frequency:Modulus:Phase\n");
        for (int corretabela = 0; corretabela < valuesTable.getRowCount(); corretabela++) {
            sb.append(valuesTable.getValueAt(corretabela,
0).toString()).append(";").append(valuesTable.getValueAt(corretabela,
1).toString()).append(";").append(valuesTable.getValueAt(corretabela, 2).toString()).append("\n");
            try {
                BufferedWriter bf = new BufferedWriter(new
FileWriter(fileChooser.getSelectedFile().getAbsolutePath()));
                bf.write(sb.toString());
                bf.flush();
                bf.close();
            } catch (Exception e) {
                System.out.println(e.getMessage());
            }
        }
    }
}
private void c1CoefActionPerformed(java.awt.event.ActionEvent evt) {}
private void calcularQuadripoleT (java.awt.event.ActionEvent evt) {
    if(null!= frame)
    {
        frame.setVisible(false);
    } // TODO add your handling code here:
    List<double[]> listaparaGrafico =
transformercalculator.TransformerCalculator.gerarLista(Double.parseDouble(r1Coef.getText()),
Double.parseDouble(r2Coef.getText()), (1) * Math.pow(10, -12),
(1) * Math.pow(10, -12), Double.parseDouble(c12Coef.getText()) * Math.pow(10, -12),
Double.parseDouble(l1Coef.getText()) * Math.pow(10, -3), Double.parseDouble(l2Coef.getText()) *
Math.pow(10, -3), Double.parseDouble(r1Coef.getText()), Double.parseDouble(kCoef.getText()));
    preencherTabela(listaparaGrafico);
    JFreeChart chart = transformercalculator.TransformerCalculator.gerarGrafico("Quadripole T",
"Abcissa n where frequency is 1e+n[Hz]", "m where impedance module is 1e+m[ohms]", false,
listaparaGrafico, true);
    JFreeChart chart2 = transformercalculator.TransformerCalculator.gerarGrafico("Quadripole T",
"Abcissa n where frequency is 1e+n[Hz]", "Phase[degrees]", false, listaparaGrafico, false);
    exibirFrame("Quadripole T", "t.jpg");
    preencherGrafico(chart);
    preencherGrafico2(chart2);
}

```

```

private void preencherGrafico(JFreeChart chart)
{
    ChartPanel painelChart = new ChartPanel(
        chart, true);
    painelChart.setSize(chartPanel.getSize());
    chartPanel.removeAll();
    chartPanel.add(painelChart);
    chartPanel.revalidate();
    chartPanel.repaint();
}
private void preencherGrafico2(JFreeChart chart)
{
    ChartPanel painelChart = new ChartPanel(
        chart, true);
    painelChart.setSize(chartPanel1.getSize());
    chartPanel1.removeAll();
    chartPanel1.add(painelChart);
    chartPanel1.revalidate();
    chartPanel1.repaint();
}
private void exibirFrame(String titulo,String caminhoDalmagem)
{
    if(null!=frame)
    {
        frame.setVisible(false);
    }
    frame = new JFrame(titulo);
    Container container = frame.getContentPane();
    ImagemIcon img = new ImagemIcon(caminhoDalmagem);
    JLabel labelteste = new JLabel(img);
    JPanel testePanel = new JPanel();
    testePanel.add(labelteste, BorderLayout.NORTH);
    container.add(testePanel, BorderLayout.CENTER);
    frame.pack();
    frame.setVisible(true);
}
private void calcularQuadripoleTC2 (java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    List<double[]> listaparaGrafico =
transformercalculator.TransformerCalculator.gerarLista(Double.parseDouble(r1Coef.getText()),
Double.parseDouble(r2Coef.getText()), 1 * Math.pow(10, -12),
    Double.parseDouble(c2Coef.getText()) * Math.pow(10, -12),
Double.parseDouble(c12Coef.getText()) * Math.pow(10, -12), Double.parseDouble(l1Coef.getText())
* Math.pow(10, -3), Double.parseDouble(l2Coef.getText()) * Math.pow(10, -3),
Double.parseDouble(rlCoef.getText()), Double.parseDouble(kCoef.getText()));
    preencherTabela(listaparaGrafico);
    JFreeChart chart = transformercalculator.TransformerCalculator.gerarGrafico("Quadripole T plus
C2", "Abscissa n where frequency is 1e+n[Hz]", "m where impedance module is 1e+m[ohms]", false,
listaparaGrafico, true);
    JFreeChart chart2 = transformercalculator.TransformerCalculator.gerarGrafico("Quadripole T
plus C2", "Abscissa n where frequency is 1e+n[Hz]", "Phase[degrees]", false, listaparaGrafico, false);
    exibirFrame("Quadripole T plus C2", "tcapacitorsaida.jpg");
}

```

```

    preencherGrafico(chart);
    preencherGrafico2(chart2);
}
private void exibirVariation(java.awt.event.ActionEvent evt) {
    exibirFrame("Variation according on electrical parameters", "todasvariaco.es.jpg");
}
private void exibirCalculatek(java.awt.event.ActionEvent evt) {
    exibirFrame("Calculate K", "k.jpg");
}
private void exibirDicas(java.awt.event.ActionEvent evt) {
    exibirFrame("Tips", "tips.jpg");
}
private void exibirAbout(java.awt.event.ActionEvent evt) {
    exibirFrame("About", "about.jpg");
}
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Telaprincipal().setVisible(true);
        }
    });
}
private javax.swing.JButton botaoCalcular;
private javax.swing.JTextField c12Coef;
private javax.swing.JTextField c1Coef;
private javax.swing.JTextField c2Coef;
private javax.swing.JPanel chartPanel;
private javax.swing.JPanel chartPanel1;
private javax.swing.JPanel coefPanel;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextField kCoef;
private javax.swing.JTextField l1Coef;
private javax.swing.JTextField l2Coef;
private javax.swing.JTextField r1Coef;
private javax.swing.JTextField r2Coef;
private javax.swing.JTextField rlCoef;

```

```

private javax.swing.JPanel valuesPanel;
private javax.swing.JTable valuesTable;

package transformercalculator;

import java.awt.Color;
import java.util.ArrayList;
import java.util.List;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.axis.NumberAxis;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.chart.plot.XYPlot;
import org.jfree.chart.renderer.category.LineAndShapeRenderer;
import org.jfree.chart.renderer.xy.XYLineAndShapeRenderer;
import org.jfree.data.xy.XYSeries;
import org.jfree.data.xy.XYSeriesCollection;
* @author Jhonisson Ricardo de Paula
*/
public class TransformerCalculator {
    public static JFreeChart gerarGrafico (String titulo, String nomeDoEixoX, String nomeDoEixoY,
boolean exibirLegenda, List <double[]> printarGrafico, boolean seleciona_grafico){
        final XYSeries series1 = new XYSeries(nomeDoEixoX);
        if (seleciona_grafico==true){
            for (double[] ds : printarGrafico) {
                series1.add(Math.log10(ds[0]),Math.log10(ds[1]));
            }
        } else if(seleciona_grafico==false){
            for (double[] ds : printarGrafico){
                series1.add(Math.log10(ds[0]), ds[2]);
            }
        }
        final XYSeriesCollection dataset = new XYSeriesCollection();
        dataset.addSeries(series1);
        final JFreeChart chart = ChartFactory.createXYLineChart(
            titulo, // chart title
            nomeDoEixoX, // x axis label
            nomeDoEixoY, // y axis label
            dataset, // data
            PlotOrientation.VERTICAL,
            exibirLegenda, // include legend
            true, // tooltips
            true // urls
        );
        // NOW DO SOME OPTIONAL CUSTOMISATION OF THE CHART...
        chart.setBackgroundPaint(Color.white);
        // get a reference to the plot for further customisation...
        final XYPlot plot = chart.getXYPlot();
        plot.setBackgroundPaint(Color.lightGray);
        plot.setDomainGridlinePaint(Color.white);
        plot.setRangeGridlinePaint(Color.white);
        final LineAndShapeRenderer renderer = new LineAndShapeRenderer();

```

```

    renderer.setSeriesLinesVisible(0, false);
    renderer.setSeriesShapesVisible(1, false);
    // change the auto tick unit selection to integer units only...
    final NumberAxis rangeAxis = (NumberAxis) plot.getRangeAxis();
    rangeAxis.setStandardTickUnits(NumberAxis.createIntegerTickUnits());
    return chart;
}
public static List<double[]> gerarLista(double resistor1, double resistor2, double capacitor1, double
capacitor2,
double capacitor12, double indutancia1, double indutancia2, double resistorL, double valorK)
{
    double f, M, r1, r2, l11, l22, l11sc, k, C1, C2, C12, Cl, Xc1, Xcl, Xl11, Xl22, Xm, Xc2, Real, Im, Mod,
fase;
    double Xc12, ImZa, ImZb, ImZc, ReZa, ReZb, ReZc, Ra, Rb, Rc, Rl, Ima, Imb, Imc, Ic, j, Zin;
    double alfa_p, alfa_g, beta, gama, delta, teta, variavel;
    List<double[]> listaResultado = new ArrayList<double[]>();
    variavel = 1.08;
    f = 10000;
    if (capacitor1==0)
        capacitor1=1*Math.pow(10, -12);
    if (capacitor2==0)
        capacitor2=1*Math.pow(10, -12);
    if (capacitor12==0)
        capacitor12=1*Math.pow(10, -12);
    do {
        M = valorK*Math.sqrt(indutancia1*indutancia2);
        Xm = 2*Math.PI*f*M;
        Xc2 = 1/(2*Math.PI*f*capacitor2);
        Xc1 = 1/(2*Math.PI*f*capacitor1);
        Xc12 = 1/(2*Math.PI*f*capacitor12);
        Xl22 = 2*Math.PI*f*indutancia2;
        Xl11 = 2*Math.PI*f*indutancia1;
        //equacoes
        ImZc = (Xl22+Xm*(resistor2*resistor1+(Xl11-Xm)*(Xl22-Xm)))/(Math.pow(resistor1,
2)+Math.pow(Xl11-Xm, 2));
        ReZc = resistor2+Xm*((resistor1*(Xm-Xl22)+resistor2*(Xl11-Xm))/(Math.pow(resistor1,
2)+Math.pow(Xl11-Xm, 2));
        ImZb = Xl11+Xl22-2*Xm+((Xl22-Xm)*(Xl11-Xm)-resistor1*resistor2)/Xm;
        ReZb = resistor1+resistor2+((resistor1*(Xl22-Xm)+resistor2*(Xl11-Xm))/Xm);
        ImZa = Xl11+Xm*((resistor1*resistor2+(Xl22-Xm)*(Xl11-Xm))/(Math.pow(resistor2,
2)+Math.pow(Xl22-Xm, 2));
        ReZa = resistor1+Xm*((resistor2*(Xm-Xl11)+resistor1*(Xl22-Xm))/(Math.pow(resistor2,
2)+Math.pow(Xl22-Xm, 2));
        Imc = (ImZc*Math.pow(Xc2, 2)-Xc2*Math.pow(ReZc, 2)-Xc2*Math.pow(ImZc,
2))/(Math.pow(ReZc, 2)+Math.pow(ImZc-Xc2, 2));
        Rc = (ReZc*Math.pow(Xc2, 2))/(Math.pow(ReZc, 2)+Math.pow(ImZc-Xc2, 2));
        Imb = (ImZb*Math.pow(Xc12, 2)-Xc12*Math.pow(ReZb, 2)-Xc12*Math.pow(ImZb,
2))/(Math.pow(ReZb, 2)+Math.pow(ImZb-Xc12, 2));
        Rb = (ReZb*Math.pow(Xc12, 2))/(Math.pow(ReZb, 2)+Math.pow(ImZb-Xc12, 2));
        Ima = (ImZa*Math.pow(Xc1, 2)-Xc1*Math.pow(ReZa, 2)-Xc1*Math.pow(ImZa,
2))/(Math.pow(ReZa, 2)+Math.pow(ImZa-Xc1, 2));
        Ra = (ReZa*Math.pow(Xc1, 2))/(Math.pow(ReZa, 2)+Math.pow(ImZa-Xc1, 2));
    }
}

```



```

delta = (Imc*(Ra+Rb+Rc)+(Rc+resistorL)*(Ima+Imb+Imc)-2*Rc*Imc);
gama = (Ra+Rb+Rc)*(Rc+resistorL)-Imc*(Ima+Imb+Imc)-Math.pow(Rc, 2)+Math.pow(Imc, 2);
beta = (Rc+resistorL)*Ima+Ra*Imc;
alfa_p = (Ra*(Rc+resistorL)-Ima*Imc);
teta = ((beta*gama-alfa_p*delta)/(Math.pow(gama, 2)+Math.pow(delta, 2)));
alfa_g = (1-(alfa_p*gama+beta*delta)/(Math.pow(gama, 2)+Math.pow(delta, 2)));
Real = (Ra*alfa_g+Ima*teta);
Im= (Ima*alfa_g-Ra*teta);
Mod = (Math.sqrt(Math.pow(Real, 2)+Math.pow(Im, 2)));
fase = Math.atan(Im/Real);
fase = (fase/Math.PI*180);
if (Real<=0){
    fase=fase+180;
}
Xc2)/(Math.pow(resistor2, 2)+Math.pow(Xl22-Xc2, 2));
    variavel = variavel+0.02;
    f = Math.pow(10, variavel);
    if(f>20.0)
        listaResultado.add(new double[] {f, Mod, fase});
} while (f < 100000000);
return listaResultado;
}
}

```

ANEXO B

MANUAL DE INSTRUÇÕES DO TRANSFORMER DESIGNER

DESENVOLVIDO POR JHONISSON RICARDO DE PAULA

CURITIBA 2012

SUMÁRIO

1 INTRODUÇÃO	75
2 REQUERIMENTOS MÍNIMOS	75
3 <i>DOWNLOAD DO SOFTWARE</i>	75
4 SELEÇÃO DE MODELOS DE TRANSFORMADORES	76
4.1 Modelo Completo ou Delta-Estrela.....	76
4.2 Modelo T Clássico.....	78
4.3 Modelo T com Capacitor de Saída.....	79
5 VARIAÇÃO DA CURVA DA IMPEDÂNCIA DE ACORDO COM OS COEFICIENTES	80
6 CÁLCULO DA CONSTANTE k.....	81
7 DICAS ÚTEIS PARA O DESENVOLVIMENTO DE TRANSFORMADORES	83
8 EXPORTAR RESULTADOS	86
9 INFORMAÇÕES SOBRE O TRANSFORMER DESIGNER	87

1 INTRODUÇÃO

Este manual tem como objetivo auxiliar o usuário a utilizar o TRANSFORMER DESIGNER, *software* para desenvolvimento de transformadores para equipamentos de eletrocirurgia.

2 REQUERIMENTOS MÍNIMOS

Os requerimentos mínimos para o funcionamento do *software* são:

- processador 1.2 GHz;
- 1024 MB de memória RAM;
- espaço mínimo em disco de 10 MB para manipulação de dados;
- instalação do JRE (Java Runtime Environment) a partir da versão 6.

3 DOWNLOAD DO SOFTWARE

Para fazer uso do TRANSFORMER DESIGNER realize o *download* do mesmo, através do endereço eletrônico http://www.pessoal.utfpr.edu.br/bertoldo/downloads_CEFET.htm clicando sobre o link **Software Livre para Desenvolvimento e Modelagem de Transformadores**.

Após realizar o *download* do arquivo intitulado **TransformerDesigner.rar** faça sua descompactação através da utilização de algum *software* de descompactação compatível com o formato RAR, tomando o cuidado de manter todos os arquivos descompactados dentro do mesmo diretório.

Os arquivos presentes no arquivo **TransformerDesigner.rar** são: “about.jpg”, “deltaestrela.jpg”, “k.jpg”, “t.jpg”, “tcapacitorsaida.jpg”, “tips.jpg”, “todasvariaco.es.jpg”, sendo todos estes arquivos de imagem, além do arquivo “DesignTransformer.jar” este último é um arquivo executável no formato JAR.

4 SELEÇÃO DE MODELOS DE TRANSFORMADORES

Para inicializar o Transformer Designer clique sobre o arquivo DesignTransformer.jar, a tela da Figura 1 será apresentada:

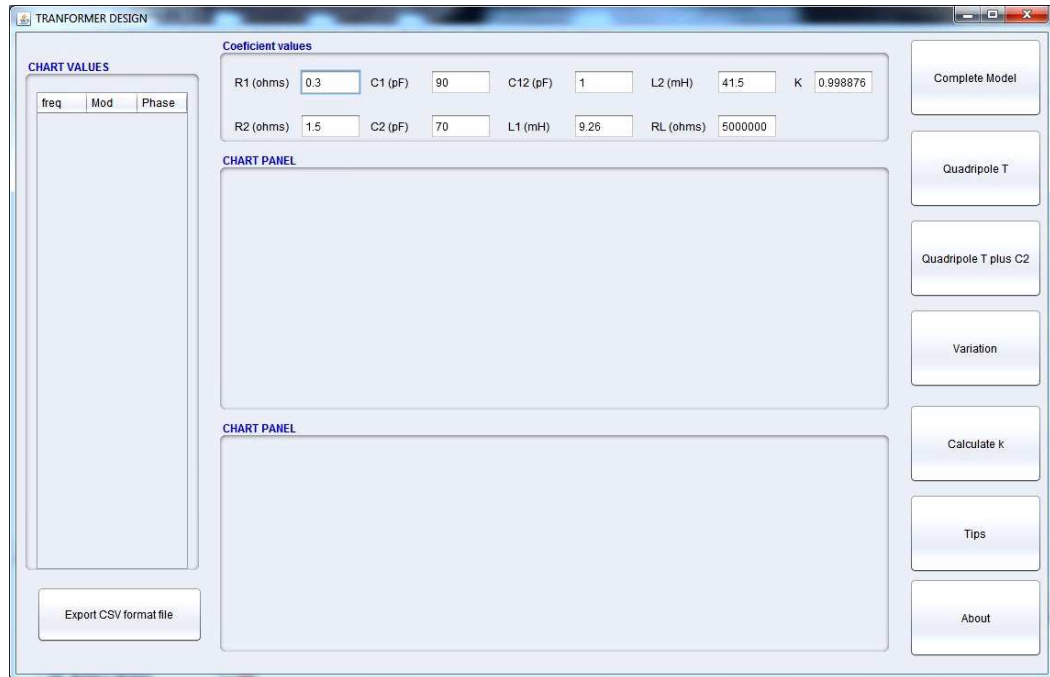


Figura 1 – Tela inicial.

4.1 Modelo Completo ou Delta-Estrela

Após inserir os valores de coeficientes desejados clique sobre o botão “Complete Model” se desejar visualizar os resultados para o modelo de transformador Delta-Estrela, conforme Figura 2.

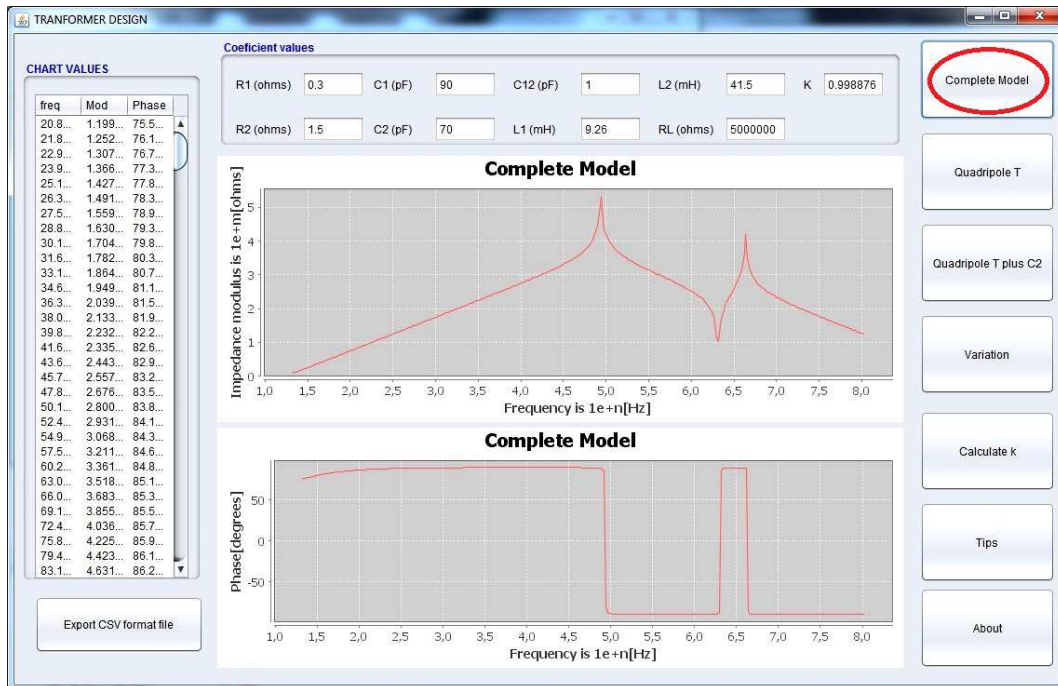


Figura 2 – Seleção do modelo “Delta-Estrela”.

Além de apresentar os valores numéricos e gráficos do módulo e da fase da impedância de entrada, automaticamente uma nova janela é apresentada, nela está representado o modelo esquemático do transformador, além da descrição de cada um dos coeficientes presentes neste modelo de transformador, Figura 3.

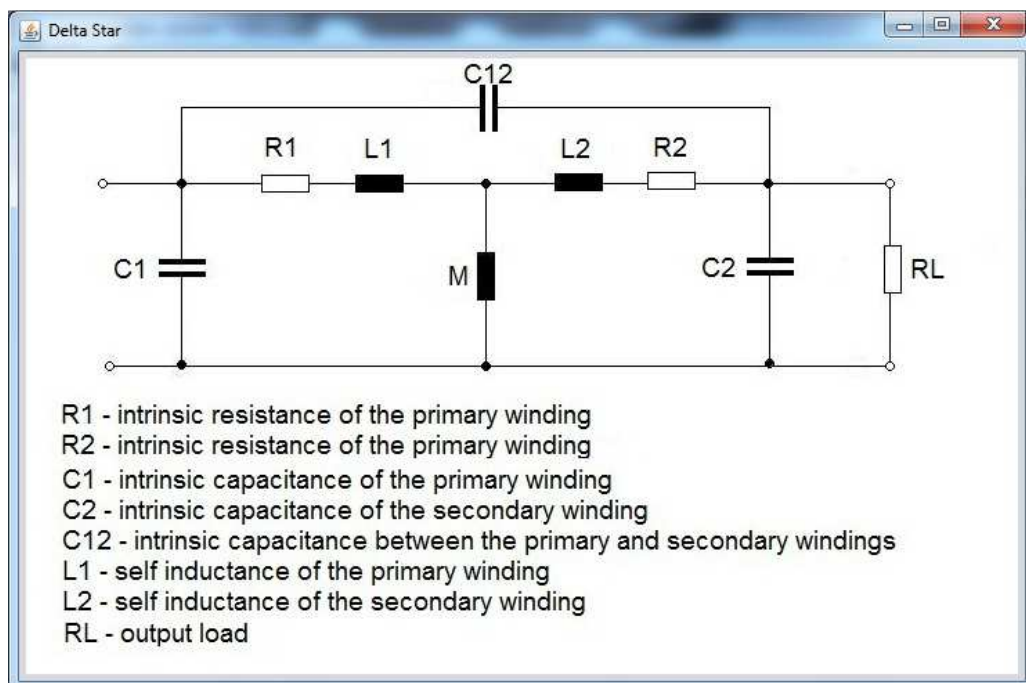


Figura 3 – Janela auxiliar, apresentação do modelo Delta-Estrela.

4.2 Modelo T Clássico

Após inserir os valores de coeficientes desejados clique sobre o botão “Quadripole T” se desejar visualizar os resultados para o modelo de transformador T Clássico, conforme Figura 4.

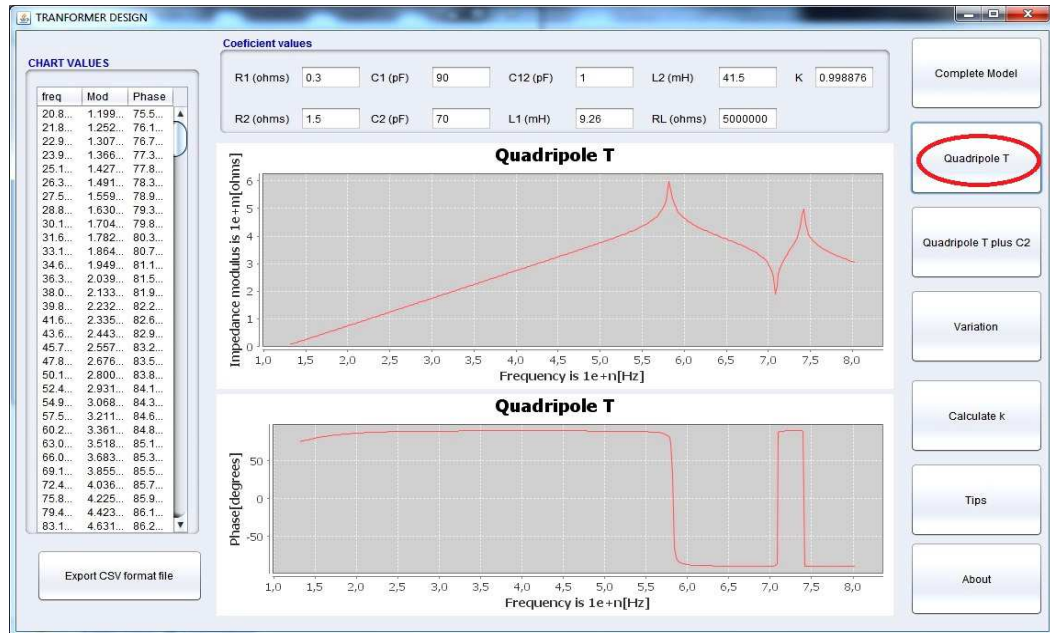


Figura 4 – Seleção do modelo T clássico.

Além de apresentar os valores numéricos e gráficos do módulo e da fase da impedância de entrada, automaticamente uma nova janela é apresentada, nela está representado o modelo esquemático do transformador, além da descrição de cada um dos coeficientes presentes neste modelo de transformador, Figura 5.

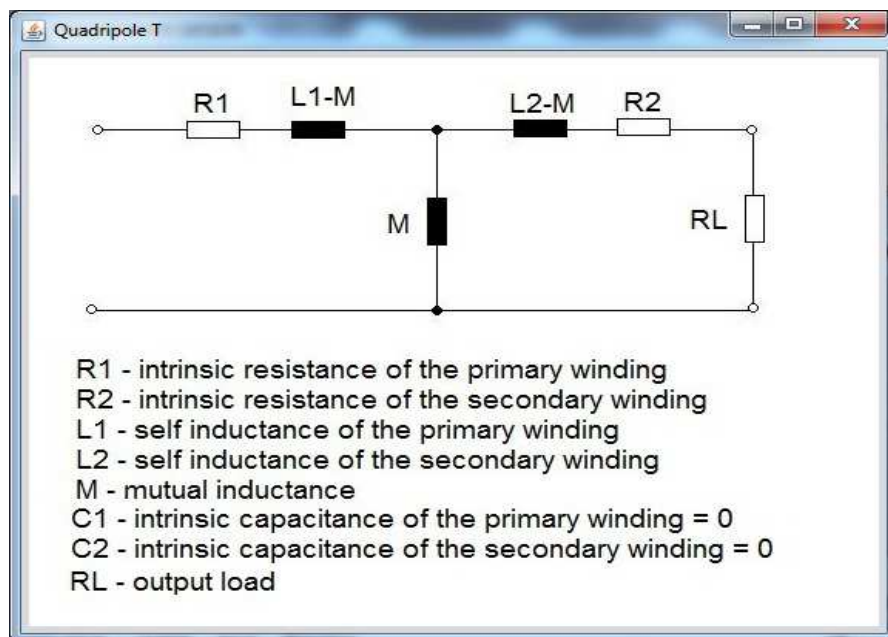


Figura 5 – Janela auxiliar, apresentação do modelo T Clássico.

4.3 Modelo T com Capacitor de Saída

Após inserir os valores de coeficientes desejados clique sobre o botão “Quadripole T plus C2” se desejar visualizar os resultados para o modelo de transformador T com capacitor de saída, conforme Figura 6.

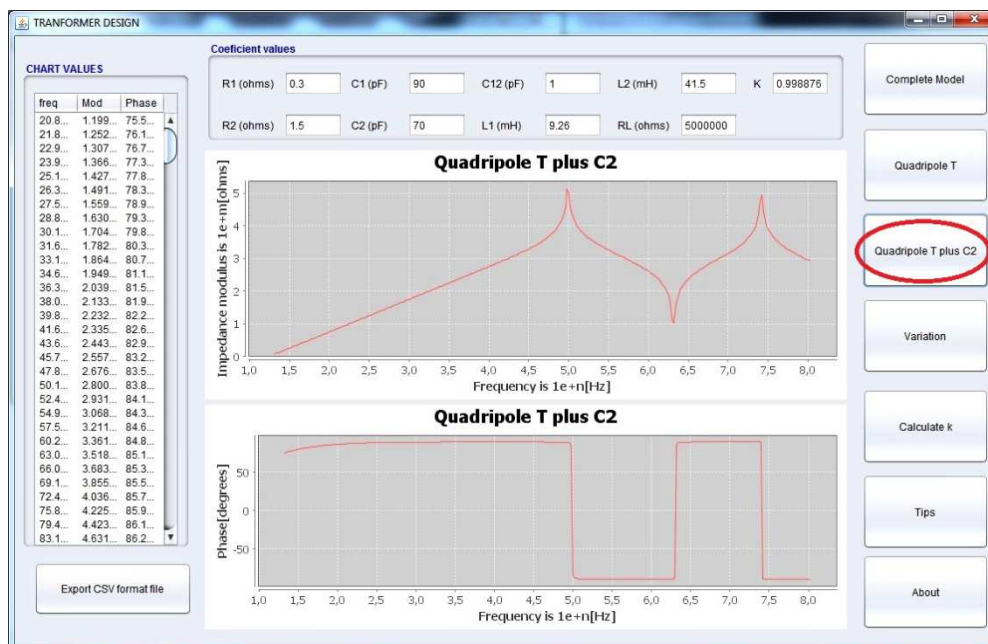


Figura 6 – Seleção do modelo T com capacitor de saída.

Além de apresentar os valores numéricos e gráficos do módulo e da fase da impedância de entrada, automaticamente uma nova janela é apresentada, nela está

representado o modelo esquemático do transformador, além da descrição de cada um dos coeficientes presentes neste modelo de transformador, Figura 7.

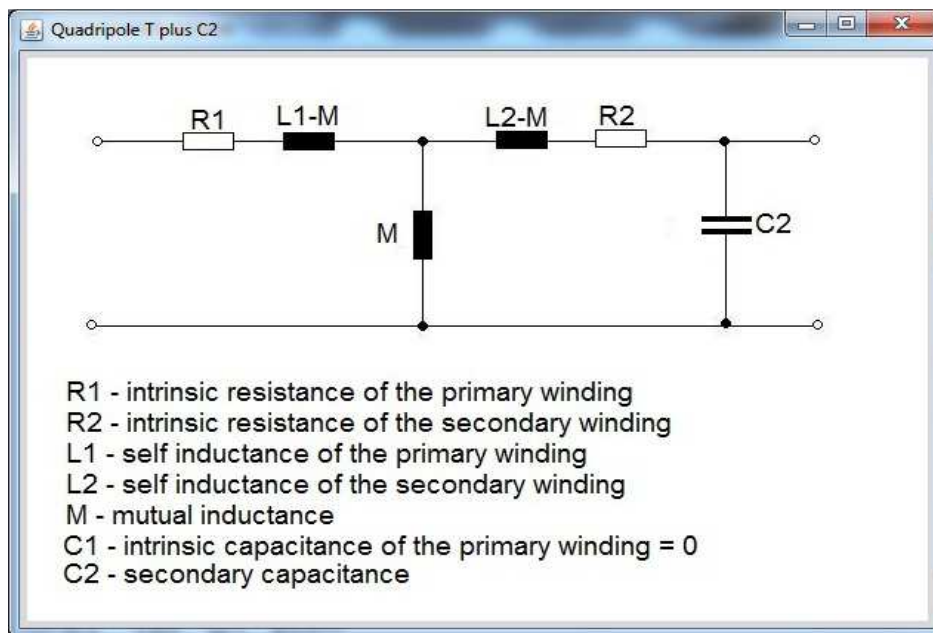


Figura 7 – Janela auxiliar, apresentação do modelo T com capacitor de saída.

5 VARIAÇÃO DA CURVA DA IMPEDÂNCIA DE ACORDO COM OS COEFICIENTES

Para entender melhor a interferência causada por cada um dos coeficientes na curva da impedância de entrada do transformador clique sobre o botão “Variation”, Figura 8.



Figura 8 – Seleção da variação da curva da impedância de entrada de acordo com os coeficientes.

Após clicar sobre o botão a seguinte janela será apresentada:

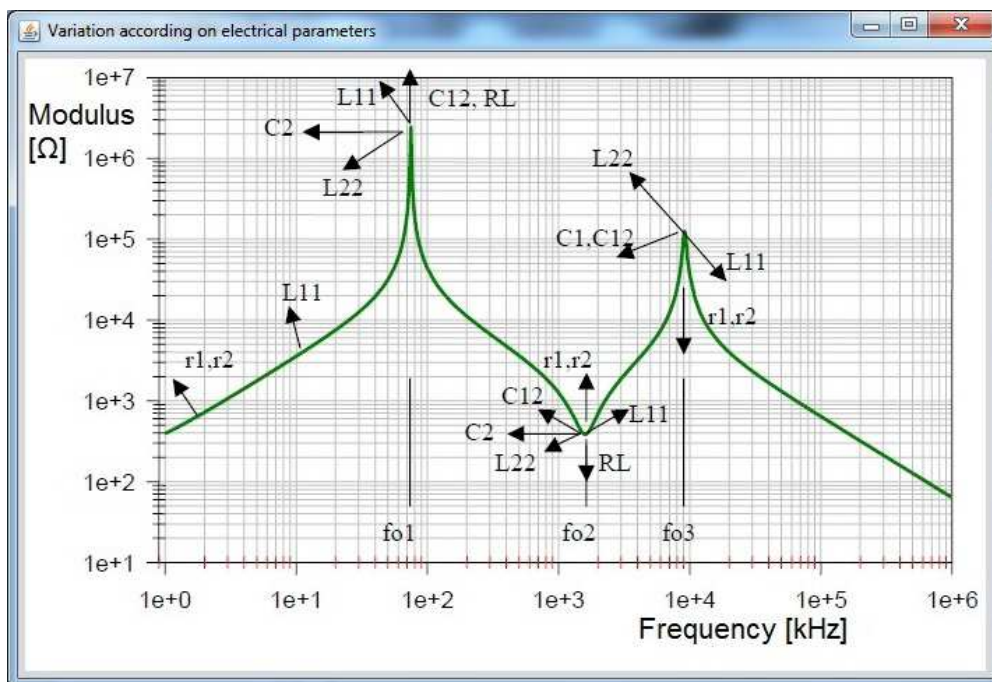


Figura 9 – Variação da curva da impedância de entrada de acordo com os coeficientes.

6 CÁLCULO DA CONSTANTE K

Para visualizar a fórmula utilizada para calcular a constante k clique sobre o botão "Calculate k", Figura 10.

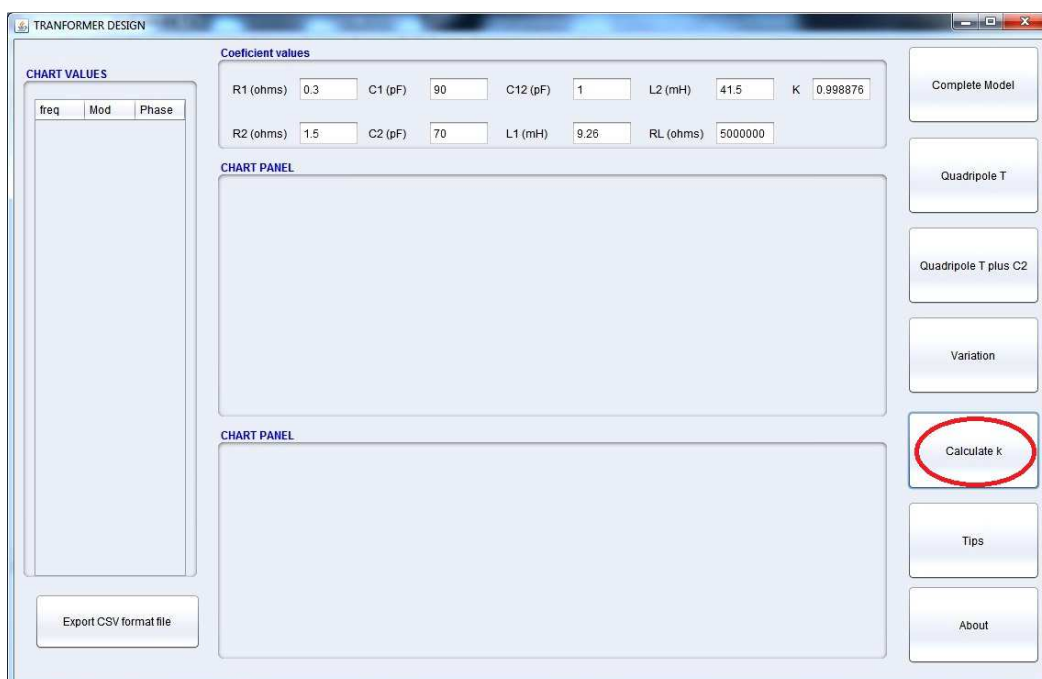
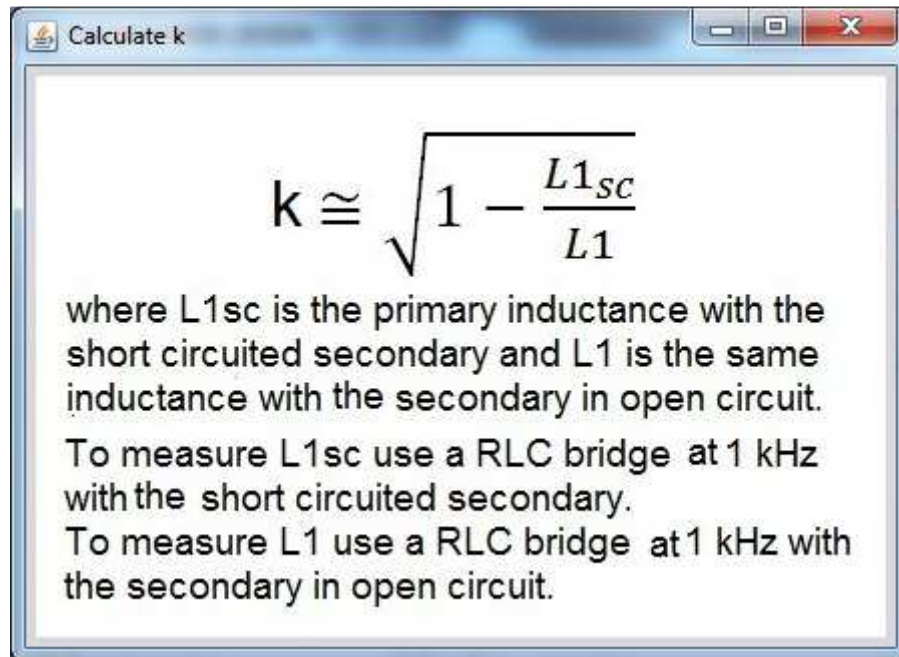


Figura 10 – Seleção do cálculo da constante k.

Após clicar a seguinte janela será apresentada:



The image shows a window titled "Calculate k" with a standard Windows-style title bar (minimize, maximize, close buttons). The window contains the following text:

$$k \cong \sqrt{1 - \frac{L1_{sc}}{L1}}$$

where L1sc is the primary inductance with the short circuited secondary and L1 is the same inductance with the secondary in open circuit.
To measure L1sc use a RLC bridge at 1 kHz with the short circuited secondary.
To measure L1 use a RLC bridge at 1 kHz with the secondary in open circuit.

Figura 11 – Fórmula para calcular a constante k.

7 DICAS ÚTEIS PARA O DESENVOLVIMENTO DE TRANSFORMADORES

Se desejar visualizar algumas dicas essenciais e importantes no desenvolvimento de transformadores, clique sobre o botão “Tips”, conforme Figura 12.



Figura 12 – Seleção de dicas para o desenvolvimento de transformadores.

Após clicar sobre o botão duas janelas serão abertas, as duas serão apresentadas a seguir nas Figuras 13 e 14.

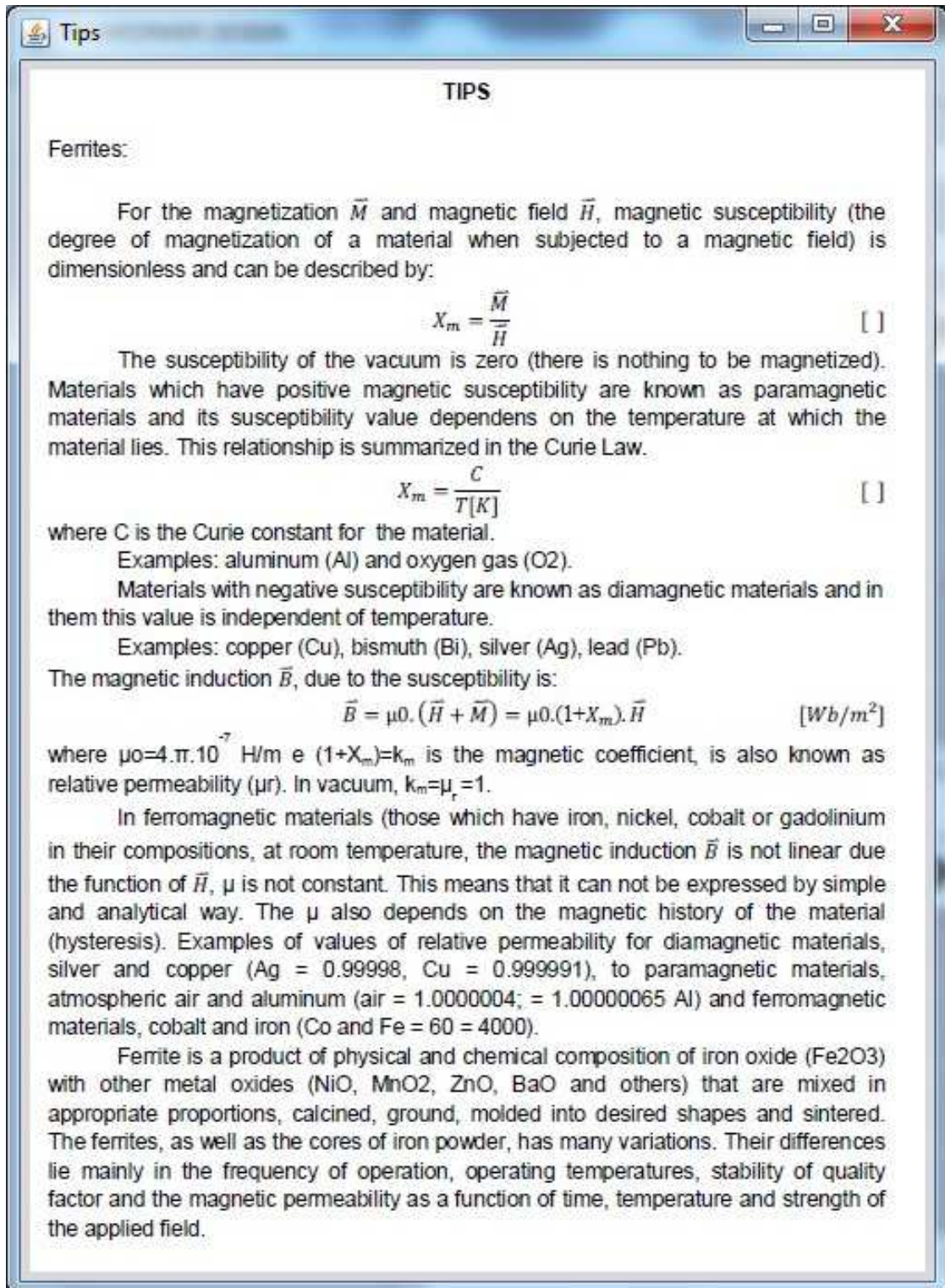


Figura 13 – Dicas úteis para o desenvolvimento de transformadores.

Tips

For a ferrite core, there is one effective permeability, μ_e , considering the gap of:

$$\mu_e = \frac{\mu_r \cdot l_e}{l_e + l_{gap} \cdot (\mu_r - 1)} \quad [H/M]$$

where l_e is the effective length of the magnetic path and l_{gap} the length of the gap. Figure 1 shows the normalized form of the effective permeability as a function of the relative length of the gap in relation to the effective magnetic path.

Effective permeability X gap

Percentage gap on the effective length of the magnetic path	Percentage of relative permeability
0.001	100
0.01	98
0.1	90
1	50
10	10
100	0

Figure 1 - The effective permeability of the ferrite core according to the percentage of the gap magnetic path.

MEASURES:

To measure R1, R2, C1, C2, C12, L1 and L2 you can use a RLC bridge at 1 kHz with the secondary in open circuit.

To measure k verify the button "Calculate k"

WINDINGS:

To move the first resonance frequency to higher frequencies use nylon or cotton wires amongst the turns to minimize the intrinsic capacitance.

To minimize C1 or C2 use nylon or cotton wires amongst coils and a dielectric foil amongst layers.

Any separations between first and second windings minimize C12.

High diameters or multiwires conductors minimize R1 and R2.

Use appropriate enameled wires for the windings. Consider temperature and maximum rupture voltage.

Figura 14 - Dicas úteis para o desenvolvimento de transformadores.

8 EXPORTAR RESULTADOS

Se desejar exportar os resultados apresentados pelo *software* em um arquivo no formato CSV clique sobre o botão “Export CSV format file”, conforme Figura 15.

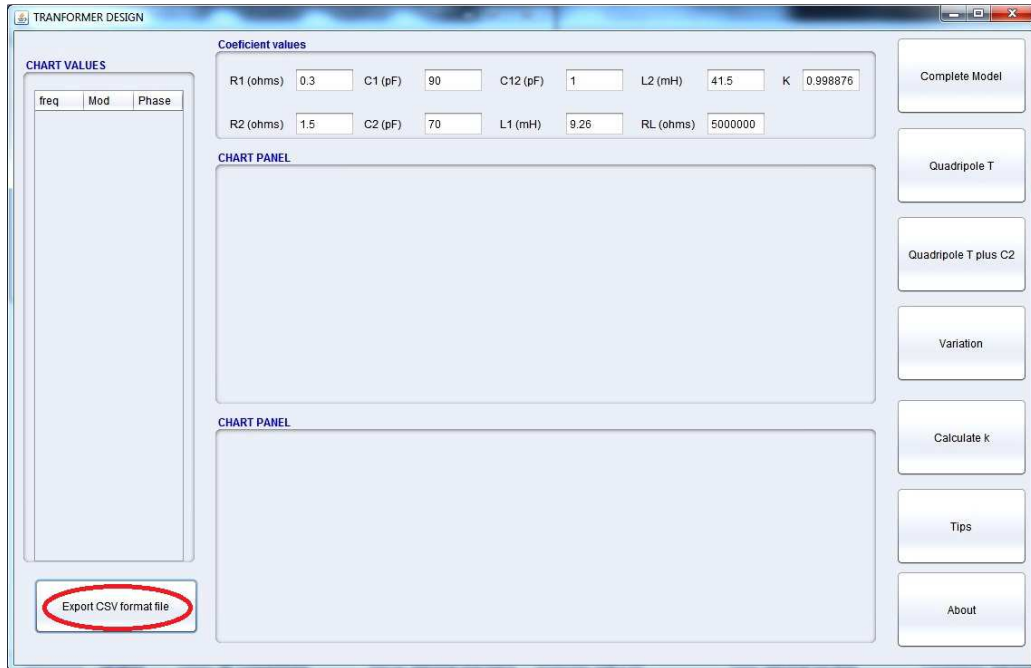


Figura 15 – Seleção de exportação dos resultados obtidos.

Após clicar sobre o botão selecione qual o diretório deseja salvar o arquivo e nomei-o, conforme Figura 16.

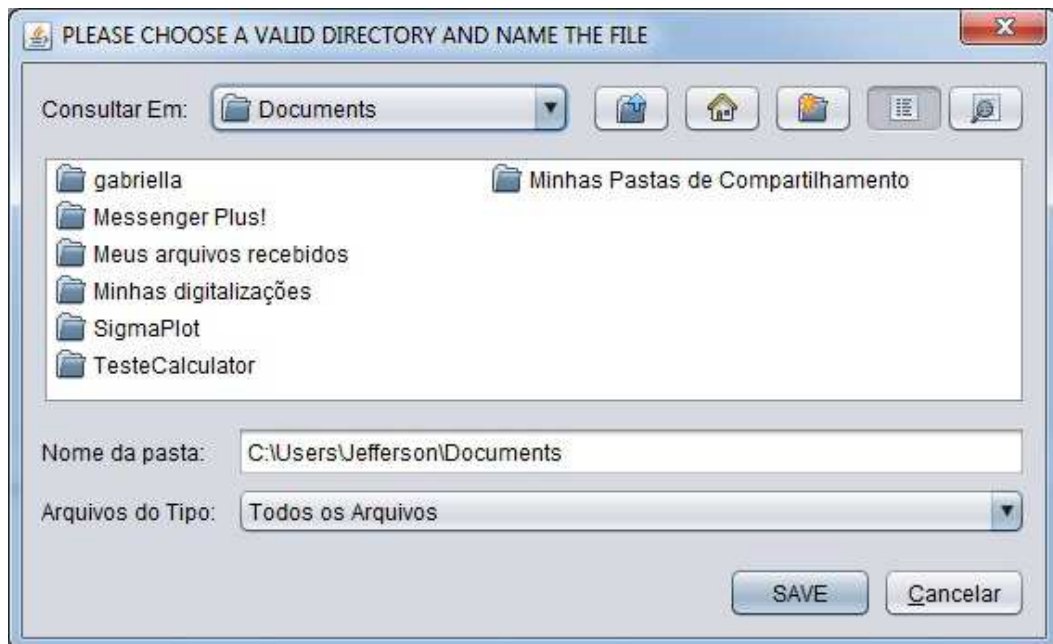


Figura 16 – Tela para selecionar o diretório e nomear o arquivo.

9 INFORMAÇÕES SOBRE O TRANSFORMER DESIGNER

Para visualizar informações sobre os autores, a instituição onde foi desenvolvido o *software* e novas versões clique sobre o botão “About”, Figura 17.

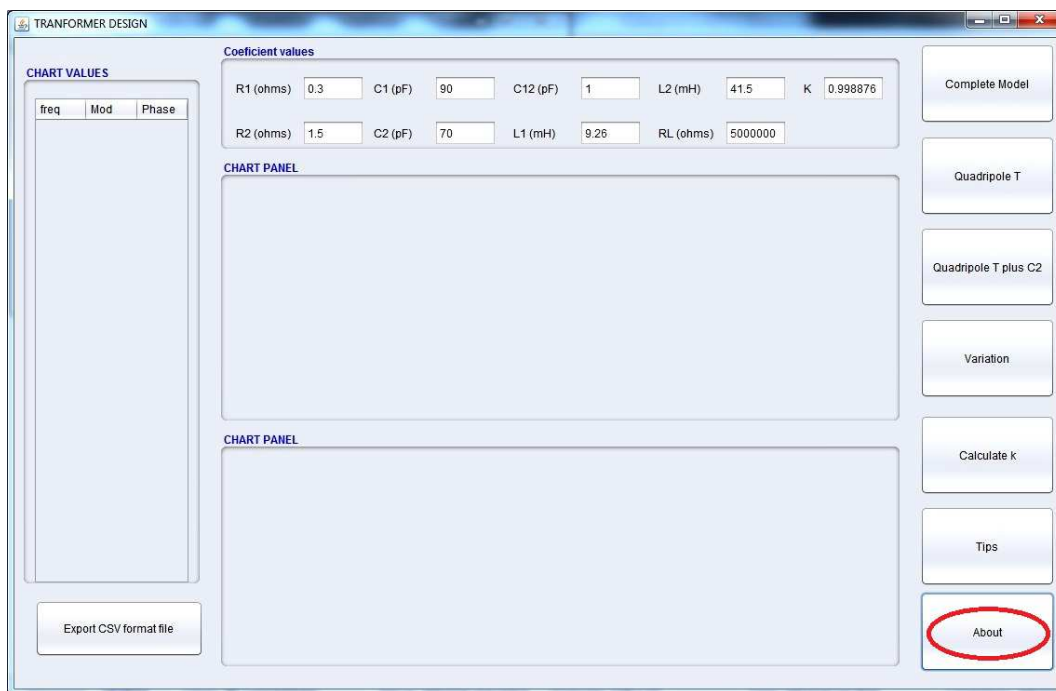


Figura 17 – Seleção do botão “About”.

Após clicar sobre o botão uma tela auxiliar é apresentada, conforme Figura 18.



Figura 18 – Informações sobre o *software*.

ANEXO C

FERRITES

Sejam a magnetização \vec{M} e o campo Magnético \vec{H} , a suscetibilidade magnética, o grau de magnetização de um material quando submetido a um campo magnético, é adimensional e vale (SEARS & ZEMANSKI, 1980):

$$X_m = \frac{\vec{M}}{\vec{H}} \quad []$$

A suscetibilidade do vácuo é zero (no vácuo não há nada a ser magnetizado). Os materiais que têm suscetibilidade magnética positiva são conhecidos por materiais *paramagnéticos* e, nestes, seu valor é dependente da temperatura em que este material se encontra. Esta relação está resumida na Lei de Curie (SEARS & ZEMANSKI, 1980):

$$X_m = \frac{C}{T[K]} \quad []$$

onde C é a constante de Curie para o material.

São exemplos deste tipo de material o alumínio (Al) e o oxigênio gasoso (O₂).

Materiais com suscetibilidade menor que zero são conhecidos por materiais *diamagnéticos* e nestes o valor é independente da temperatura. Como exemplo, tem-se o cobre (Cu), bismuto (Bi), prata (Ag) e chumbo (Pb).

A indução magnética \vec{B} , em função da suscetibilidade é:

$$\vec{B} = \mu_0 \cdot (\vec{H} + \vec{M}) = \mu_0 \cdot (1 + X_m) \cdot \vec{H} \quad [Wb/m^2]$$

onde $\mu_0 = 4 \cdot \pi \cdot 10^{-7}$ H/m e $(1 + X_m) = k_m$ é o coeficiente magnético, também conhecido como permeabilidade relativa (μ_r). No vácuo, $k_m = \mu_r = 1$.

Nos materiais ferromagnéticos (aqueles que têm ferro, níquel, cobalto ou gadolínio em suas composições, na temperatura ambiente (SEARS & ZEMANSKI, 1980), a indução magnética \vec{B} não é linear em função de \vec{H} , isto é, μ não é constante. Isto significa que não se pode expressar de maneira simples e analítica o coeficiente magnético, usando-se normalmente curvas de magnetização dos materiais. O μ depende ainda da história magnética passada do material (histerese). São exemplos de valores de permeabilidade relativa para materiais diamagnéticos a prata e o cobre ($\mu_r = 0,99998$, $\mu_r = 0,999991$) (SEARS & ZEMANSKI, 1980), para materiais paramagnéticos o ar atmosférico e o alumínio ($\mu_r = 1,0000004$;

$\mu_r=1,00000065$) e para materiais ferromagnéticos o cobalto e o ferro ($\mu_r=60$ e $\mu_r=4000$) (HAYT, 1983).

O ferrite é produto da composição físico-química de óxido de ferro (Fe_2O_3) com outros óxidos metálicos (NiO , MnO_2 , ZnO , BaO e outros) que são misturados em proporções adequadas, calcinados, moídos, moldados nas formas desejadas e sinterizados (SONTAG, 1985). Os ferrites, assim como os núcleos de pó de ferro, têm muitas variações. Suas diferenças estão principalmente nas frequências de operação, temperaturas de operação, estabilidade do fator de qualidade e da permeabilidade magnética em função do tempo, da temperatura e da intensidade do campo aplicado.

Para um núcleo de ferrite, tem-se uma permeabilidade efetiva, μ_e , considerando o entreferro, de:

$$\mu_e = \frac{\mu_r \cdot l_e}{l_e + l_{gap} \cdot (\mu_r - 1)} \quad [H/M]$$

onde l_e é o comprimento efetivo do caminho magnético e l_{gap} o comprimento do entreferro. A Figura 1 mostra a forma normalizada da permeabilidade efetiva em função do comprimento relativo do entreferro em relação ao caminho efetivo magnético.

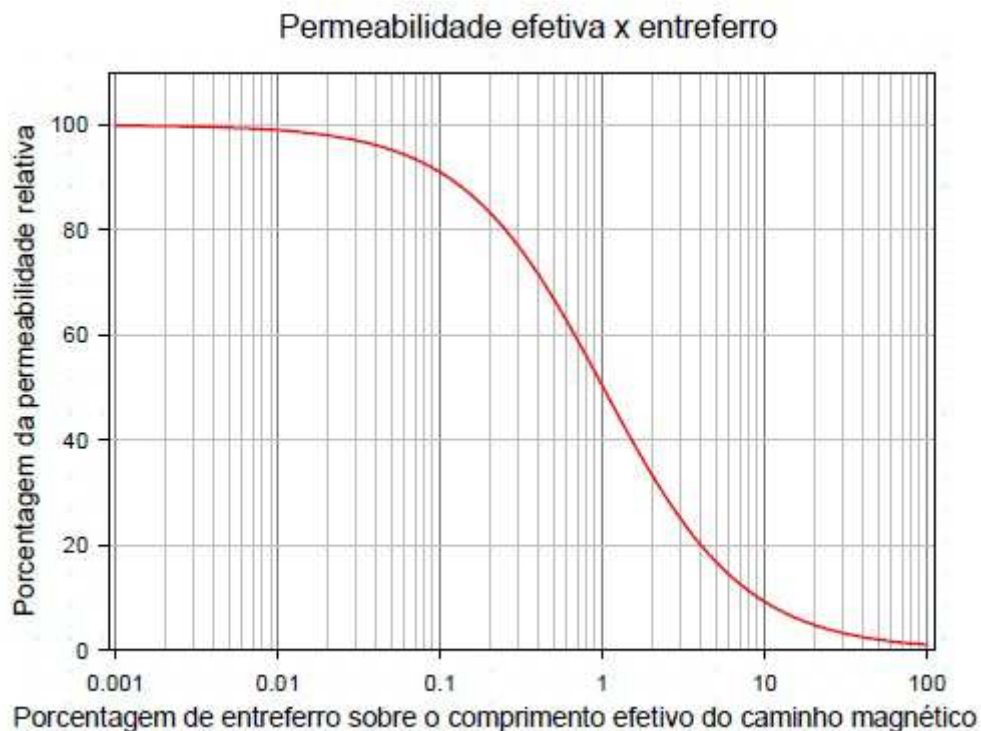


Figura 19 - Permeabilidade efetiva do núcleo de ferrite em função da porcentagem de entreferro do caminho magnético. Adaptado de: Schneider (2004).