

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PEDRO MARTINS WAETGE KIEFER

PROJETO DE DIPLOMAÇÃO

OPTRAFO: SOFTWARE PARA OTIMIZAÇÃO DE TRANSFORMADORES

Porto Alegre

2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

OPTRAFO: SOFTWARE PARA OTIMIZAÇÃO DE TRANSFORMADORES

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Luiz Tiarajú dos Reis Loureiro

Porto Alegre

2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

PEDRO MARTINS WAETGE KIEFER

OPTRAFO: SOFTWARE PARA OTIMIZAÇÃO DE TRANSFORMADORES

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Luiz Tiarajú dos Reis Loureiro, UFRGS

Banca Examinadora:

Prof. Luiz Tiarajú dos Reis Loureiro, UFRGS

Doutor pela UFRGS – Porto Alegre, Brasil

Prof. Roberto Petry Homrich, UFRGS

Doutor pela UNICAMP – Campinas, Brasil

Prof. Aly Ferreira Flores Filho, UFRGS

Doutor pela Universidade de Cardiff – Cardiff, País de Gales

Porto Alegre, Dezembro de 2009.

DEDICATÓRIA

Dedico este trabalho aos meus pais e à minha irmã, que sempre me apoiaram durante toda a graduação.

AGRADECIMENTOS

Aos meus pais, Flávio e Gisela, pelo suporte e carinho.

À minha irmã Luísa que sempre esteve ao meu lado e com quem me divirto há anos.

Aos meus avós que me ensinam tanto.

Aos meus amigos, com quem tive discussões sobre política, economia, artes e tecnologia: Ricardo Wickert, John Kliff Jochens, Rodrigo Kassick, Luciana Muller Brusco, André Spritzer, Sophia Chassot, Christiano Haesbaert, Mirella Peruzzo, Felipe Berger Soares, Wagner Müller Gegler e tantos outros quem foram importantes durante a faculdade.

RESUMO

O projeto de transformadores ótimos já foi estudado, diferentes métodos de otimização surgiram destas pesquisas. A questão fundamental é o que otimizar, o que nos leva a segunda pergunta, como otimizar. Existem diversas respostas. Uma respostas recorrente é otimizar o projeto para mínima massa (e portanto, mínimo custo). Este trabalho apresenta um modelo baseado em programação geométrica (PG). O uso de PG leva a soluções eficientes e confiáveis enquanto garantindo que a solução é ótima globalmente. O modelo utilizado deve englobar como restrições diferentes conjuntos de parâmetros como regulação de tensão, sobreelevação de temperatura, rendimento, etc. Um software para resolver este problema e auxiliar o projetista é proposto. O resultado do uso desta ferramenta é apresentado como uma comparação com um projeto tradicional de transformador.

Palavras-chaves: Otimização Convexa, Programação Geométrica, Transformadores, Projeto de Transformadores, Linux, Python.

ABSTRACT

The design of optimal transformers has been thoroughly studied, different methods arise from this. The fundamental question is what to optimize, which leads to the question how to optimize. Several are the answers. One recurrent answer is to minimize the mass of the transformer (and as a result, the cost). The present work uses a model based on geometrical programming (GP). GP provides efficient and reliable solutions while guaranteeing that the solution is a global optimum. The transformer's model for optimization must encompass different sets of constraints such as voltage regulation, temperature rise, efficiency, etc. Software for solving this problem and to aid the designer is proposed. The result of the use of this software is presented as a comparison with a traditional transformer design.

Keywords: Convex Optimization, Geometric Programming, Transformers, Transformer Projects, Optimization, Linux, Python.

SUMÁRIO

1. INTRODUÇÃO.....	14
2. OTIMIZAÇÕES CONVEXAS.....	18
2.1. CONVEXIDADE	19
2.1.1. Conjuntos	20
2.1.2. Funções	20
2.1.3. Condições de primeira ordem	20
2.1.4. Condições de segunda ordem	21
2.1.5. Exemplos	22
2.2. PROGRAMAÇÃO GEOMÉTRICA	22
2.2.1. Forma Convexa.....	23
2.2.2. Análise de Viabilidade.....	24
2.2.3. Análise de Equilíbrio	25
2.2.4. Análise de Sensibilidade.....	26
2.3. EXTENSÕES A PROGRAMAÇÃO GEOMÉTRICA.....	26
2.3.1. Potências fracionárias de posinômios.....	27
2.3.2. Máximo de posinômios.....	27
2.3.3. Posinômios Generalizados.....	28
3. USO DE PG EM PROJETOS DE TRANSFORMADORES.....	29
3.1. MODELO MATEMÁTICO	30
3.1.1. Função objetivo.....	33
3.1.2. Restrições: Tensão Induzida	33
3.1.3. Restrições: Fator de Preenchimento	34
3.1.4. Restrições: Aumento de Temperatura.....	35
3.1.5. Restrições: Rendimento	35
3.1.6. Restrições: Corrente a Vazio	36
3.1.7. Restrições: Regulação de Tensão	36
3.2. MODELO EM FORMATO DE PG.....	37
3.2.1. Função Objetivo.....	37
3.2.2. Restrições: Tensão Induzida	37
3.2.3. Restrições: Fator de Preenchimento	38
3.2.4. Restrições: Aumento de Temperatura.....	38
3.2.5. Restrições: Rendimento	39
3.2.6. Restrições: Corrente a Vazio	39
3.2.7. Restrições: Regulação de Tensão	39
3.2.8. Extensões: Projeto com Mínimas Perdas.....	40
3.2.9. Extensões: Projeto com Potência de Saída Máxima.....	41
4. DESENVOLVIMENTO DO SOFTWARE.....	42
4.1. BIBLIOTECA DE OTIMIZAÇÕES CONVEXAS: CVXOPT.....	44
4.2. ESTRUTURA DO SOFTWARE.....	46
4.3. FORMATO DE ARQUIVO.....	48
4.4. INTERFACE COM O USUÁRIO	49
5. COMPARAÇÃO.....	51
5.1. TRANSFORMADOR DE 300W	52

6. CONCLUSÃO.....	56
7. REFERÊNCIAS.....	58

LISTA DE ILUSTRAÇÕES

Figura 1 Gráfico de uma função convexa.....	20
Figura 2 $f(x)$ é convexa e diferenciável, então $f(x) + \nabla f(x)^T (y - x) \leq f(y)$ para todos $x, y \in \text{dom } f$	21
Figura 3 Corte lateral e superior de um transformador monofásico com núcleo envolvente..	29
Figura 4 Corte lateral e superior de um transformador monofásico com núcleo envolvente..	30
Figura 5 Diagrama de Classes simplificado, ilustrando as relações entre os diversos objetos.	47
Figura 6 Interface gráfica do software.....	49
Figura 7 Editor de materiais, exibindo os três materiais cadastrados previamente.	50

LISTA DE TABELAS

Tabela 1 - Valores típicos para materiais de núcleo	32
Tabela 2 - Comparação inicial entre o software implementado e o modelo original	52
Tabela 3 - Comparação entre modelo original e otimizado	54

LISTA DE SÍMBOLOS

A_c	Área da coluna central do núcleo
$A_{Cu,p}$	Área de condutores no primário
$A_{Cu,s}$	Área de condutores no secundário
A_m	Área efetiva do núcleo
B	Densidade de fluxo magnético
B_{sat}	Densidade de fluxo magnético de saturação
$\cos \varphi$	Fator de potência
Δ	Razão entre espessura de camada e profundidade de penetração pelicular
ΔT	Sobrelevação máxima de temperatura
δ	Profundidade de penetração pelicular
d	Espessura do condutor
E_p	Tensão nominal no primário
E_s	Tensão nominal no secundário
F_l	Fator de utilização de camada
F_r	Razão entre resistência AC e DC
f	Frequência
η_m	Rendimento mínimo
I_c	Corrente de perdas no núcleo
I_m	Corrente de magnetização
I_p	Corrente nominal no primário
I_φ	Corrente de excitação a vazio
J	Densidade de corrente
k_a	Fator de área de superfície dos enrolamentos
K_c	Coefficiente de perdas específicas do material
k_φ	Relação máxima entre corrente a vazio para carga
k_f	Fator de empilhamento
k_{Cu}	Fator de preenchimento de cobre
L_l	Indutância de dispersão referida ao primário
μ	Permeabilidade
μ_r	Permeabilidade relativa
μ_0	Permeabilidade do espaço livre ($4\pi \cdot 10^{-7}$)
m	Número de camadas por seção de bobina
MLT	Comprimento médio de uma volta

m_c	Massa do núcleo
m_{Cu}	Massa dos enrolamentos
N_p	Número de espiras no primário
N_s	Número de espiras no secundário
P_c	Perdas no núcleo
P_{Cu}	Perdas nos enrolamentos
P_o	Potência de saída
p	Quantidade de segmentos de bobina
ρ_c	Densidade do núcleo
ρ_{Cu}	Densidade do cobre
ρ_w	Resistividade do condutor
S	Potência nominal
T_a	Temperatura ambiente
V_c	Volume do núcleo
V_w	Volume dos enrolamentos
VR_m	Regulação de tensão máxima
V_R	Queda de tensão resistiva
V_X	Queda de tensão reativa
W_a	Área da janela

1. INTRODUÇÃO

Uma das definições à palavra racionalidade é a busca por uma meta ótima, o que não deixa de ser o pensamento por trás de todas as engenharias: maximizar o rendimento de um projeto; otimizar o sistema; aproveitar ao máximo os recursos disponíveis. Estas máximas moveram e revolucionaram o mundo desde as primeiras civilizações, sendo fortemente aceleradas no período pós Revolução Industrial e Revolução Francesa [1]. A figura do engenheiro surge justamente neste período histórico. A maioria dos conceitos com que este trabalho lida foram desenvolvidos nos últimos 200 anos, e portanto são fruto dessas revoluções.

O presente trabalho busca demonstrar um método para a racionalização de projetos de transformadores, ou, em outras palavras, otimizar o desempenho da máquina – transformador – a ser fabricada. Mas, por que necessita-se otimizar o projeto de um transformador? Aqui não há uma resposta única, tão pouco ela é simples. Alguns responderiam que é necessário otimizar o projeto para diminuir seu tamanho, pois existem restrições quanto ao espaço do equipamento, outros mencionariam que o importante é diminuir os custos de fabricação, custos de perdas, etc. Todas estas respostas estão corretas. São elas que instigam o engenheiro a desenvolver novos métodos de otimização.

Os primeiros métodos de projeto são normalmente empíricos, baseados em poucos conhecimentos físicos e a convicção de que o projeto funcionará. Diversos inventos importantes utilizaram este método, e possivelmente novos inventos continuarão a utilizar. Conforme estes projetistas, ou melhor, engenheiros adquirem conhecimentos sobre as diversas características físicas do projeto – com toda a sua matemática, então pode-se alterar os preceitos empíricos por um modelo matemático. Isto possibilita uma análise das equações, o que permite melhorar o desempenho do projeto. Existem diversos métodos de projeto de transformadores, uns fortemente baseados em conceitos empíricos – como os ábacos descritos por Lee e Mullinix [2] – outros utilizando sofisticados métodos computacionais. Os métodos empíricos nor-

malmente geram projetos conservadores [3], que, dependendo da aplicação, não atendem as especificações de forma satisfatória.

Com o advento dos computadores, inicialmente os métodos empíricos foram automatizados como os descritos por Williams *et al.* [4] de 1956, que enfatiza a diminuição de tempo e custo no projeto de transformadores; também no trabalho de Weber e Gallousis [5] de 1959, que descreve os novos conceitos para projeto de transformadores com auxílio de computadores; ou ainda por Shull [6] de 1969, que mostra as vantagens de se utilizar um computador de tempo compartilhado para projetos. Estes métodos funcionavam de forma adequada para a tecnologia da época, mas o custo da sua utilização ainda era altíssimo, o que limitava seu uso à poucas empresas.

Conforme o desempenho computacional aumentou, novas metodologias surgiram, agora não mais automatizando modelos empíricos, mas utilizando modelos matemáticos complexos – baseados nas equações físicas que regem as diferentes partes de um transformador. Dos métodos publicados recentemente vale citar alguns: Odessey desenvolveu um dos primeiros métodos a utilizar modelos matemáticos para diminuir a interação do projetista com a ferramenta [7]; Judd e Kressler propuseram um método para maximizar a potência a partir de um núcleo definido[8]; Andersen descreve um método utilizando uma rotina de Monte Carlo modificada [9], que utiliza números aleatórios para gerar um grande número de projetos e escolher o de menor custo; Geromel e Souza exploraram o uso de redes neurais [10], e Li Hui *et al.* Utilizaram algoritmos genéticos para evoluir modelos e selecionar o mais adequado [11].

O que tornou possível a criação e utilização destes métodos foi o incrível aumento do poder computacional nas últimas décadas. Dos métodos descritos anteriormente, alguns necessitam de poucos recursos computacionais, mas se beneficiam do maior desempenho apresentado pelos computadores atuais.

O método abordado neste trabalho utiliza a técnica de programação geométrica para otimizar o projeto de transformadores. Esta técnica é um tipo de otimização não linear aplicável a problemas com características específicas, e quando estas características são atendidas, então o problema é facilmente resolvido. A principal vantagem deste método é que a solução encontrada é sempre global, pois todo o problema de programação geométrica pode ser convertido em um problema de otimização convexa, como está descrito na seção sobre otimizações convexas.

Quando este método é aplicado a otimização de transformadores, tem-se como vantagens a sua capacidade de otimizar simultaneamente diversos parâmetros do modelo, isto é, as características geométricas, magnéticas e elétricas.

Este trabalho está organizado em capítulos. O primeiro capítulo – Otimizações Convexas – é subdividido em três partes, primeiramente tem-se uma introdução a otimizações convexas, suas principais características e condições para verificar se uma função é convexa. Seguido a isto mostra-se a definição do problema de programação geométrica e também sua ligação com os problemas de otimização convexa. Terminando o capítulo tem-se alguns métodos para flexibilizar a modelagem em programação geométrica.

Com base nas técnicas de modelagem em programação geométrica é apresentado um modelo matemático de um transformador monofásico com núcleo envolvente no segundo capítulo – uso de PG em projetos de transformadores. Este modelo primeiramente é mostrado na sua forma canônica para então ser transformado em um modelo de programação geométrica. Três diferentes problemas de minimização são apresentados: mínima massa, mínimas perdas e potência de saída máxima.

A partir do modelo encontrado é proposto um software para resolver os três problemas. Todas as decisões de implementação do software, como as escolhas de bibliotecas para auxiliar na implementação ou projeto da interface gráfica, são apresentadas no capítulo tercei-

ro – Desenvolvimento do Software. Para testar o software e validar o modelo, foram estudados dois projetos de transformadores. Os resultados obtidos estão descritos no quarto capítulo – Comparação.

Para finalizar este trabalho tem-se as conclusões sobre a utilização do método descrito bem como idéias para trabalhos futuros. No anexo são apresentadas todas as matrizes utilizadas pelo software.

2. OTIMIZAÇÕES CONVEXAS

Antes de definir o que consiste um problema de otimização convexa, é interessante mostrar uma definição genérica dos problemas de otimização. Este tipo de problema consiste em minimizar um objetivo, com algumas restrições impostas a este objetivo. Matematicamente a forma é a seguinte:

$$\begin{aligned} &\text{minimizar } f_0(x) \\ &\text{sujeito a } f_i(x) \leq b_i, \quad i = 1, \dots, m. \end{aligned} \quad (1)$$

Onde $x = (x_1, \dots, x_n)$ é a variável a ser otimizada do problema, f_0 é a função objetivo, f_i são funções de restrição tendo as constantes b_1, \dots, b_m como limites [12]. O vetor x é dito ótimo, quando o valor da função objetivo for o menor entre todos os vetores que satisfazem as restrições.

A classificação dos problemas de otimização é realizada com base nos tipos de função objetivo e das funções de restrição. Pode-se definir duas grandes classes de problemas, os problemas lineares e os não-lineares. O modelo acima seria um problema de programação linear se a função objetivo e suas restrições fossem funções lineares, isto é:

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y) \quad (2)$$

A classe de problemas que será abordada – os problemas de otimização convexa – possuem ambas as funções objetivo e restrições em forma convexa. Portanto, satisfazem a seguinte inequação [12]:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(x) \quad (3)$$

com $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$. Comparando a equação (2) com a equação (3), pode-se concluir que a forma convexa é mais genérica que a forma linear, isto é, uma inequação é menos restritiva que uma relação de igualdade. Portanto, pode-se verificar que os problemas lineares são englobados pelos problemas convexas.

A outra classe de problemas, os de otimização não-linear, são aqueles problemas que possuem sua função objetivo e suas funções de restrição como equações não-lineares e não convexas. Esta classe de problemas não possui nenhum método eficiente de solução.

Um problema de otimização convexa apresenta a seguinte forma:

$$\begin{aligned} & \text{minimizar } f_0(x) \\ & \text{sujeito a } f_i(x) \leq b_i, \quad i = 1, \dots, m. \end{aligned} \tag{4}$$

com as funções $f_0, \dots, f_m : \mathbb{R}^n \rightarrow \mathbb{R}$ convexas, isto é, satisfazendo

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

para quaisquer $x, y \in \mathbb{R}^n$ e todos $\alpha, \beta \in \mathbb{R}$ com $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$.

Não existem soluções analíticas para problemas de otimizações convexas, mas existem diversos métodos de solução [12]. Um dos mais utilizados é o método dos pontos internos, que resolve problemas com centenas de variáveis e milhares de restrições em questão de segundos em um computador normal. Obviamente que, explorando as particularidades de um problema (como a esparcialidade das matrizes), pode-se resolver problemas ainda maiores na mesma ordem de tempo.

O reconhecimento de problemas de otimização convexa, ou de problemas que possam ser transformados em problemas de otimização convexa, não é uma tarefa simples. Porém, depois de encontrado o problema nesta forma, sua solução é fácil. Talvez a principal vantagem dos problemas de otimização convexa é que suas soluções fornecerão sempre o ponto ótimo global.

2.1. CONVEXIDADE

Algumas definições sobre a convexidade de funções e conjuntos são necessárias para o entendimento da transformação que será aplicada aos problemas de programação geométrica, uma vez que estes não são convexas por natureza.

2.1.1. CONJUNTOS

Conjuntos convexos são aqueles que dado um segmento de linha entre quaisquer dois pontos do conjunto, o segmento está contido no conjunto. Matematicamente isto significa que quaisquer $x_1, x_2 \in C$ e qualquer $\theta, 0 \leq \theta \leq 1$ temos:

$$\theta x_1 + (1 - \theta)x_2 \in C .$$

O que significa que um conjunto é convexo quando todos os pontos do conjunto enxergam todos os outros pontos através de uma linha reta, sem obstruções. Para mais detalhes sobre conjuntos convexos ver capítulo 2 de [12].

2.1.2. FUNÇÕES

Uma função é dita convexa se seu domínio for um conjunto convexo e se quaisquer $x, y \in \text{dom } f$ e θ com $0 \leq \theta \leq 1$, então:

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (5)$$

Geometricamente essa inequação representa que um segmento de reta entre $(x, f(x))$ e $(y, f(y))$ está acima do gráfico da função f , como podemos ver na Figura 1.

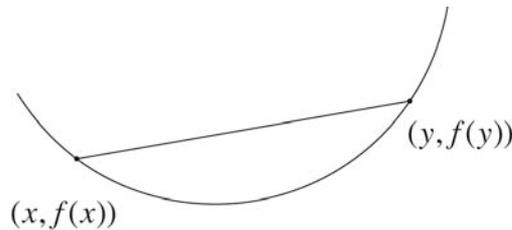


Figura 1 Gráfico de uma função convexa.

Um resultado interessante é que uma função convexa é contínua no interior do seu domínio e pode ter descontinuidades nas suas bordas.

2.1.3. CONDIÇÕES DE PRIMEIRA ORDEM

Para verificar se uma dada função é convexa, pode-se testá-la com duas condições. A condição de primeira ordem para a convexidade de uma função é dada por: seja f derivável –

isto é, seu gradiente ∇f existe em todos os pontos do domínio de f . Então f é convexo se e somente se seu domínio for convexo e

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad (6)$$

para todos os $x, y \in \text{dom } f$.

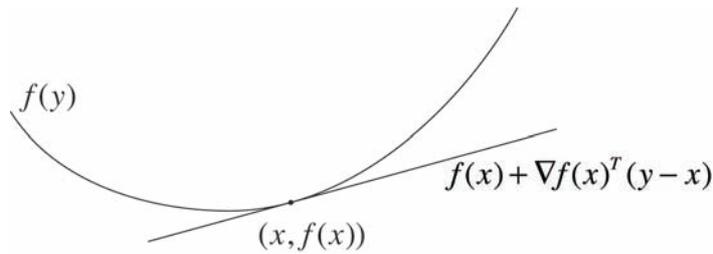


Figura 2 $f(x)$ é convexa e diferenciável, então $f(x) + \nabla f(x)^T (y - x) \leq f(y)$ para todos $x, y \in \text{dom } f$

Na Figura 2 pode-se visualizar este resultado. Pode-se ver que o lado direito da equação (6) é a aproximação de Taylor de primeira ordem da função f entorno de x . E portanto, para uma função convexa esta aproximação é um *subestimador global* da função. Então a partir de informações locais da função – o valor de um ponto e sua derivada – pode-se obter informações globais sobre a função. Como exemplo, se $\nabla f(x) = 0$ então para todos os $y \in \text{dom } f, f(y) \geq f(x)$, portanto x é um mínimo global da função.

2.1.4. CONDIÇÕES DE SEGUNDA ORDEM

Assumindo que a função possua a derivada segunda $\nabla^2 f$ – o hessiano – para todos os pontos do domínio, então ela é convexa se e somente se o domínio for convexo e o hessiano for positivo semidefinido, isto é

$$\nabla^2 f(x) \succeq 0.$$

Para uma função em \mathbb{R} essa condição se reduz para $f''(x) \geq 0$, o que significa que a derivada é não-decrescente. Podemos interpretar essa condição geometricamente como se o gráfico da função em x tenha curvatura positiva.

Outras definições e operações com funções convexas podem ser encontradas no capítulo 3 de [12].

2.1.5. EXEMPLOS

É interessante mostrar alguns exemplos funções convexas agora que já foram definidas algumas de suas principais propriedades.

- Exponencial, e^{ax} é convexa em \mathbb{R} , para qualquer $a \in \mathbb{R}$.
- Potências, x^a é convexo em \mathbb{R}_{++} , quando $a \geq 1$ ou $a \leq 0$ e côncavo para $0 \leq a \leq 1$.
- Potências de valores absolutos, $|x|^p$ para $p \geq 1$ é convexo em \mathbb{R} .
- Função máximo, $f(x) = \max\{x_1, \dots, x_n\}$ é convexo em \mathbb{R}^n .
- Logaritmo – soma – exponencial, a função $f(x) = \log(e^{x_1} + \dots + e^{x_n})$ é convexo em \mathbb{R}^n . Esta função pode ser interpretada como uma aproximação analítica diferencial da função máximo, uma vez que

$$\max\{x_1, \dots, x_n\} \leq f(x) \leq \max\{x_1, \dots, x_n\} + \log n$$

para qualquer x .

A prova da convexidade destas e outras funções encontram-se na seção 3.1.5 de [12].

2.2. PROGRAMAÇÃO GEOMÉTRICA

Problemas de programação geométrica (PG) não são convexas na sua forma usual, mas são facilmente transformados em problemas de otimização convexa, através de trocas de variáveis e uma transformação nas funções objetivo e de restrição.

Dois conceitos são necessários antes de estudar a forma dos problema de programação geométrica, o primeiro é a definição de função monomial:

$$f(x) = cx_1^{a_1} x_2^{a_2} \dots x_n^{a_n} \quad (7)$$

onde $c > 0$ e $a_i \in \mathbb{R}$.

Um exemplo de monômio é $4,5x_1^3x_2^{-0,30}x_3^{2,3}$, que tem x_1, x_2, x_3 como variáveis e coeficiente 4,5.

O segundo conceito é o de soma de monômios, uma função que apresenta a seguinte forma:

$$f(x) = \sum_{k=1}^K c_k x_1^{a_{1k}} x_2^{a_{2k}} \dots x_n^{a_{nk}} \quad (8)$$

onde $c_k > 0$, é chamada de função posinomial, ou posinômio.

O termo é a junção de ‘positivo’ e ‘polinômio’. Todos os monômios são posinômios.

Um exemplo de posinômio é $3(1 + xy)^4 + 0,23z^{-0,12}$.

A forma padrão de um problema de programação geométrica é a seguinte:

$$\begin{aligned} &\text{minimizar } f_0(x) \\ &\text{sujeito a } f_i(x) \leq 1, \quad i = 1, \dots, m, \\ &\quad \quad \quad g_i(x) = 1, \quad i = 1, \dots, p \end{aligned} \quad (9)$$

onde f_0, \dots, f_m são posinômios e g_1, \dots, g_p são monômios.

Em geral este problema não é convexo, mas pode ser transformado num problema convexo. Na próxima seção será apresentada a transformação de PG para a forma convexa.

2.2.1. FORMA CONVEXA

Para transformar um problema de programação geométrica no seu equivalente convexo, inicia-se definindo uma mudança de variáveis como $y_i = \log x_i$, e portanto $x_i = e^{y_i}$. Se a função f é monomial, $f(x) = cx_1^{a_1} x_2^{a_2} \dots x_n^{a_n}$, então

$$\begin{aligned} f(x) &= f(e^{y_1}, \dots, e^{y_n}) \\ &= c(e^{y_1})^{a_1} \dots (e^{y_n})^{a_n} \\ &= e^{a^T y + b} \end{aligned} \quad (10)$$

onde $b = \log c$.

Esta troca de variáveis transforma um monômio em uma exponencial de um mapeamento afim.

Similarmente, para um posinômio, tem-se:

$$f(x) = \sum_{k=1}^K e^{a_k^T y + b_k} \quad (11)$$

onde $a_k = (a_{1k}, \dots, a_{nk})$ e $b_k = \log c_k$. É uma soma de exponenciais de mapeamentos afins.

Com isso pode-se escrever (9) em termos da nova variável y , como:

$$\begin{aligned} & \text{minimizar} \quad \sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}} \\ & \text{sujeito a} \quad \sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}} \leq 1, \quad i = 1, \dots, m, \\ & \quad \quad \quad e^{g_i^T y + h_i} = 1, \quad i = 1, \dots, p \end{aligned} \quad (12)$$

Calculando o logaritmo de (12), obtêm-se:

$$\begin{aligned} & \text{minimizar} \quad \tilde{f}_0(y) = \log \left(\sum_{k=1}^{K_0} e^{a_{0k}^T y + b_{0k}} \right) \\ & \text{sujeito a} \quad \tilde{f}_i(y) = \log \left(\sum_{k=1}^{K_i} e^{a_{ik}^T y + b_{ik}} \right) \leq 0, \quad i = 1, \dots, m, \\ & \quad \quad \quad \tilde{h}_i(y) = g_i^T y + h_i = 0, \quad i = 1, \dots, p \end{aligned} \quad (13)$$

Como as funções \tilde{f}_i são convexas e \tilde{h}_i são funções afins, então este problema é convexo. Esta é a forma convexa de um problema de programação geométrica. Um aspecto importante dessa transformação é que ela não envolve nenhum cálculo, os dados do problema permanecem os mesmos.

2.2.2. ANÁLISE DE VIABILIDADE

Um dos primeiros passos para resolver um PG é determinar se ele é viável, isto é, se as condições impostas

$$f_i(x) \leq 1, \quad i = 1, \dots, m, \quad g_i(x) = 1, \quad i = 1, \dots, p \quad (14)$$

são consistentes. Isto é chamado de problema de viabilidade.

Se um problema não é viável, então com certeza não há uma solução ao PG, uma vez que não existe um ponto que satisfaça todas as condições. Isto significa que as condições, re-

quisitos ou especificações são muito restritivas e não é possível atingir todas simultaneamente. O relaxamento de alguma destas restrições pode tornar o problema viável.

Quando um problema é não-viável, é interessante identificar um ponto \hat{x} que está mais próximo a torná-lo viável. Um método para encontrar este ponto é formar o seguinte PG

$$\begin{aligned} &\text{minimizar } s \\ &\text{sujeito a } f_i(x) \leq 1, \quad i = 1, \dots, m, \\ &\quad g_i(x) = 1, \quad i = 1, \dots, p, \\ &\quad s \geq 1 \end{aligned} \tag{15}$$

Resolvendo este problema, que é sempre viável, assumindo que os monômios são viáveis, determina-se um \bar{x} e \bar{s} . Se $\bar{s} = 1$, então \bar{x} é viável para o PG original. Se $\bar{s} > 1$, então o PG não é viável. O valor de \bar{s} indica a proximidade da viabilidade do problema, por exemplo se $\bar{s} = 1,05$, então o problema é não viável, com afastamento de apenas 5% da viabilidade. Isto é, \bar{x} está a 5% do valor que satisfaça todas as condições impostas.

2.2.3. ANÁLISE DE EQUILÍBRIO

Neste tipo de análise, varia-se os valores das restrições e observa-se os efeitos no valor ótimo do problema. Esta análise é bastante interessante, uma vez que grande parte das restrições de um problema real não são fixas e podem ser alteradas, especialmente se houver motivos para alterar este valor.

Para a análise de equilíbrio, inicia-se com o PG original e cria-se um novo, inserindo perturbações a ele. Troca-se o valor unitário no lado esquerdo das equações de restrições por um parâmetro,

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeito a } f_i(x) \leq u_i, \quad i = 1, \dots, m, \\ &\quad g_i(x) = v_i, \quad i = 1, \dots, p \end{aligned} \tag{16}$$

onde u_i e v_i são constantes positivas.

Quando estas constantes forem iguais a 1, o problema se reduz ao original. Se $u_i > 1$ então a restrição será *reduzida*, isto é, a inequação poderá assumir um intervalo maior de valores. Se $u_i < 1$ então a restrição será *intensificada* em relação ao problema original. Plotando as curvas do valor ótimo dos problemas, pra diversas perturbações, pode-se visualizar a influência de cada restrição no sistema.

2.2.4. ANÁLISE DE SENSIBILIDADE

Esta análise é bastante similar à análise de equilíbrio, mas é realizada com pequenas variações nos parâmetros em torno do valor original, diferente das alterações maiores que podem ser feitas na análise de equilíbrio.

Pode-se definir uma função $p(u, v)$ a partir da equação (16), que representa o valor ótimo do sistema perturbado. Então a função sensibilidade do problema de programação geométrica pode ser definida como

$$S_i = \left. \frac{\partial \log p}{\partial u_i} \right|_{u=1, v=1} \quad (17)$$

A sensibilidade S_i representa uma fração de variação no valor ótimo do sistema por fração de variação no lado direito da inequação i . Por exemplo, se $S_1 = -0,3$ então uma alteração de 1% na primeira inequação significa que o valor ótimo deverá diminuir por cerca de 0,3%. Este tipo de análise ajuda a identificar quais variáveis possuem maior influência sobre o sistema.

2.3. EXTENSÕES A PROGRAMAÇÃO GEOMÉTRICA

Algumas extensões [13] à Programação Geométrica facilitam a escrita de problemas neste formato, sejam elas simples manipulações algébricas ou conceitos não tão óbvios como posinômios generalizados. A seguir serão descritas algumas destas extensões.

2.3.1. POTÊNCIAS FRACIONÁRIAS DE POSINÔMIOS

Posinômios são preservados quando elevados a potências inteiras positivas, mas não podemos dizer o mesmo de

$$f_1(x)^{2,3} + f_2(x)^{3,2} \leq 1 \quad (18)$$

uma vez que as potências são fracionárias. Adicionando variáveis e inequações as restrições do problema pode-se lidar com este tipo de equação.

Para isso, introduz-se duas nova variáveis t_1 e t_2 , assim como as inequações

$$f_1(x) \leq t_1, \quad f_2(x) \leq t_2 \quad (19)$$

que são compatíveis com o PG. Estas variáveis funcionam como condição de contorno superior aos posinômios $f_1(x)$ e $f_2(x)$. Agora pode-se substituir a equação (18), com

$$t_1^{2,3} + t_2^{3,2} \leq 1$$

que é um posinômio válido. Pode-se aplicar esta técnica recursivamente a funções, adicionando as variáveis e restrições necessárias.

2.3.2. MÁXIMO DE POSINÔMIOS

Pode-se aplicar as mesmas idéias que utilizamos para lidar com potências fracionárias a função máximo de posinômios. Sendo f_1, f_2 e f_3 posinômios, então é claro que a restrição

$$\max\{f_1(x), f_2(x)\} + f_3(x) \leq 1 \quad (20)$$

não é um posinômio. Uma vez que o máximo de dois posinômios não é uma função diferenciável, e todo o posinômio é diferenciável em todos seus valores.

Para lidar com a equação (20), pode-se introduzir uma nova variável t , e duas novas inequações

$$t + f_3(x) \leq 1, \quad f_1(x) \leq t, \quad f_2(x) \leq t \quad (21)$$

com isso tem-se restrições equivalentes às originais. Esta técnica também pode ser aplicada recursivamente e pode ser misturada com o método para lidar com potências fracionárias.

2.3.3. POSINÔMIOS GENERALIZADOS

Pode-se definir como posinômio generalizado uma função f de variáveis positivas x_1, \dots, x_n que é formada por posinômios usando as operações de adição, multiplicação, potências positivas (fracionárias) e a operação de máximo. Portanto utiliza-se as duas técnicas vistas anteriormente para trabalhar com posinômios generalizados em programação geométrica. Um exemplo de posinômio generalizado seria

$$h(x) = (x_1 + \max\{x_1, x_3\})(\max\{x_2, x_1^{0.4} + x_2x_3^{-1}\} + (0,1x_1x_2 + x_3^{1.3})^{1.5})$$

Uma vez que x_1, x_2 e x_3 são variáveis e posinômios, então a aplicação destas operações a elas, geram um posinômio generalizado.

Uma propriedade importante dos posinômios generalizados é que eles também satisfazem a propriedade de convexidade. Portanto se f é um posinômio generalizados, então a função

$$F(y) = \log f(e^y)$$

é convexa, então para qualquer y, \tilde{y} e qualquer θ com $0 \leq \theta \leq 1$, tem-se

$$F(\theta y + (1 - \theta)\tilde{y}) \leq \theta F(y) + (1 - \theta)F(\tilde{y}).$$

Em termos do posinômio generalizado original e das variáveis x e \tilde{x} temos a seguinte inequação para θ com $0 \leq \theta \leq 1$,

$$f(x_1^\theta \tilde{x}_1^{1-\theta}, \dots, x_n^\theta \tilde{x}_n^{1-\theta}) \leq f(x_1, \dots, x_n)^\theta f(\tilde{x}_1, \dots, \tilde{x}_n)^{1-\theta}.$$

3. USO DE PG EM PROJETOS DE TRANSFORMADORES

Antes de definir o modelo matemático a ser otimizado, é interessante explorar as variáveis e condições que serão otimizadas. Um transformador é formado por diversos elementos, cada um possui parâmetros e valores que podem ser alterados de acordo com o objetivo do projeto. Um modelo em programação geométrica é capaz de otimizar todas essas variáveis simultaneamente, sejam elas parâmetros da geometria do transformador ou valores magnéticos e elétricos. As restrições impostas à otimização podem ser: nível mínimo de eficiência, temperatura máxima, regulação de tensão, corrente à vazio, entre outros.

Em um projeto de transformador tem-se normalmente como objetivo minimizar a massa do núcleo e do cobre utilizado, sem que isto afete os parâmetros já mencionados, pois isto irá diminuir os custos de fabricação. Quando há um catálogo de núcleos a ser utilizado, pode-se otimizar o uso em duas etapas: primeiro minimiza-se a massa do núcleo e cobre, e com isso determina-se o núcleo do catálogo que apresenta valores mais próximos ao da otimização; numa segunda etapa, altera-se o objetivo da otimização para mínimas perdas do transformador, ou ainda para uma máxima potência de saída. Assim o uso do núcleo escolhido será maximizado.

O modelo utilizado neste trabalho [14] será de um transformador monofásico à seco com núcleo envolvente representado na Figura 3, alimentado por correntes e tensões senoidais, utilizando ar natural como sistema de refrigeração. Para ampliar este modelo para transformadores trifásicos, com refrigeração à óleo, é necessário reescrever algumas equações para adaptar o modelo.

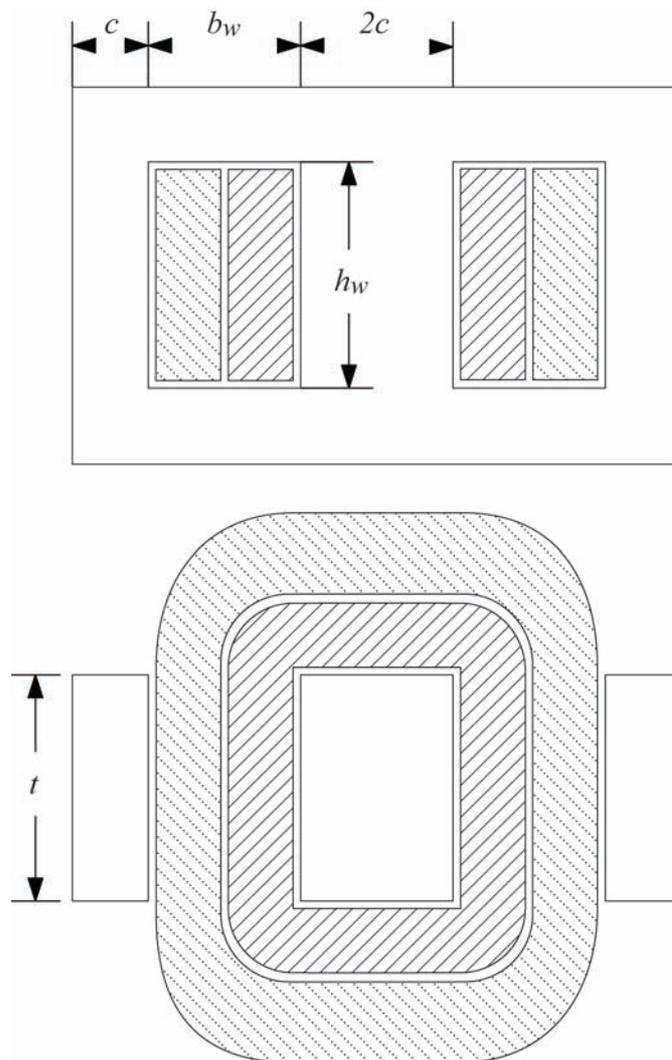


Figura 4 Corte lateral e superior de um transformador monofásico com núcleo envolvente.

3.1. MODELO MATEMÁTICO

Este modelo será a base para o problema de programação geométrica, o qual será criado a partir de manipulação algébricas sobre as equações definidas no modelo. Alguns parâmetros precisam ser definidos antes, especialmente os que dizem respeito ao núcleo do transformador, a partir da Figura 3 tem-se:

A_c Área da coluna central do núcleo, ($A_c = 2ct$);

A_m Área efetiva do núcleo, ($A_m = k_f A_c$), onde $k_f \cong 0,95$ é o fator de empilhamento do núcleo;

- MLT Comprimento médio de uma volta ($MLT = 4c + 2t + \pi b_w$);
- W_a Área da janela ($W_a = b_w h_w$);
- V_c Volume do núcleo ($V_c = A_c(2h_w + 2b_w + 4c)$);
- V_w Volume do enrolamento ($V_w = MLT \cdot W_a$).

Além destes valores, é necessário definir algumas variáveis, que dizem respeito aos valores nominais do transformador:

- E_p Tensão nominal no primário (V);
- E_s Tensão nominal no secundário (V);
- S Potência nominal aparente (VA);
- $\cos \varphi$ Fator de potência;
- f Frequência (Hz);
- T_a Temperatura ambiente (°C);
- ΔT Sobrelevação máxima de temperatura (°C);
- η_m Rendimento mínimo;
- VR_m Regulação de tensão máxima;
- k_φ Relação máxima entre corrente à vazio para carga total.

A temperatura ambiente, bem como a sobrelevação máxima de temperatura indicam indiretamente a classe de isolamento que será usada. A relação entre temperatura e classes de isolamento pode ser encontrada em [15].

Dois outros fatores devem ser considerados em relação aos materiais utilizados, o primeiro diz respeito a curva de magnetização e a curva de perdas do material utilizado no núcleo, uma vez que seus valores dependem da densidade de fluxo e da frequência. A curva de magnetização pode ser aproximada por uma região linear com μ_r como inclinação da reta e uma região com valor fixo correspondente à saturação B_{sat} . Já a curva de perdas, segundo [3] e [16] pode ser aproximada por:

$$p_c = K_c B^\alpha f^\beta \quad (22)$$

onde p_c é a perda em watts por quilograma, B é a densidade de fluxo em tesla e f é a frequência em hertz. Os valores de K_c , α e β são características específicas de cada material, estes valores são obtidos a partir dos dados dos catálogos de fabricantes de aço ou ferrites. Em [16] explica-se como obter estes valores. Na Tabela 1 são apresentados alguns valores para estas características obtidas utilizando o método citado. Com isso, as perdas no núcleo são dadas por:

$$P_c = m_c p_c = \rho_c k_f K_c V_c B^\alpha f^\beta \quad (23)$$

onde ρ_c é a densidade do material do núcleo.

Tabela 1 - Valores típicos para materiais de núcleo						
Material	B_{sat}	μ_r	ρ_c	K_c	α	β
	(T)		(kg/m ³)	(W/kg)		
Aço Silício	2,0	10.000	7.650	$0,5 \cdot 10^{-3}$	1,7	1,9
Ni-Mo	0,8	250	13.000	$5,0 \cdot 10^{-3}$	1,2	2,2
Ferrite (Mn-Zn)	0,4	2.000	4.800	$1,9 \cdot 10^{-3}$	1,24	2,0
Ferrite (Ni-Zn)	0,3	400	4.800	$2,5 \cdot 10^{-3}$	1,6	2,3

O outro fator que deve ser levado em conta no modelo são as propriedades do material condutor, isto é, do cobre dos enrolamentos. É necessário especificar a resistividade ρ_w do condutor na temperatura máxima especificada ($T_a + \Delta T$). O fator de preenchimento de cobre k_{Cu} , isto é a razão entre a área de condutores e a área da janela também deve ser especificada. Valores práticos para esta constante variam entre 0,4 para condutores redondos e 0,6 para condutores retangulares.

Para um modelo genérico, é importante levar em conta os efeitos que ocorrem em altas frequências nos condutores, isto é, o efeito pelicular e o efeito de proximidade. A razão entre a resistência em corrente alternada para corrente contínua (F_r) deve ser especificada, uma vez que ela é diretamente influenciada por estes efeitos.

F_r pode ser escrita em função de m , número de camadas por seção de bobina, e Δ , a razão entre a espessura da camada e a profundidade de penetração pelicular δ . Um valor ótimo pode ser estimado para Δ :

$$\Delta_{\text{opt}} \cong \sqrt[4]{\frac{15}{5m^2 - 1}} \quad (24)$$

com este valor, tem-se $(F_r)_{\text{opt}} \cong 4/3$. Então, para um valor ótimo de F_r , o projetista deve escolher um valor para m e dimensionar a espessura do condutor de acordo:

$$d = \Delta_{\text{opt}} \delta = \Delta_{\text{opt}} \sqrt{\frac{\rho_w}{\pi F_l f \mu_0}} \quad (25)$$

onde $\mu_0 = 4\pi \cdot 10^{-7}$ H/m é a permeabilidade do espaço livre e $F_l \cong 0,90$ é o fator de utilização de camada. Portanto, para um transformador de alta frequência, devem ser especificados os valores de m e d juntamente com o valor ótimo de $F_r = 4/3$, já em baixas frequências deve-se utilizar $F_r = 1,05$ e não é necessário especificar m e d .

3.1.1. FUNÇÃO OBJETIVO

Como é de interesse minimizar a massa de cobre e de núcleo do transformador, o objetivo será dado por

$$m_c + m_{\text{Cu}} = \rho_c k_f V_c + \rho_{\text{Cu}} k_{\text{Cu}} V_w \quad (26)$$

onde ρ_{Cu} é a densidade do cobre em kg/m³.

Esta função (26) pode ser alterada para uma função monetária em relação ao valor do cobre e do núcleo [17].

3.1.2. RESTRIÇÕES: TENSÃO INDUZIDA

A valor rms da tensão induzida, para uma onda senoidal, no enrolamento primário é dada por

$$E_p = \sqrt{2} \pi f N_p B A_m = \sqrt{2} \pi k_f N_p f B A_c \quad (27)$$

onde N_p é o número de espiras do primário e B é o valor de pico da densidade de fluxo magnético.

O número de espiras do secundário é dado por

$$N_s = \frac{E_s N_p}{E_p} \quad (28)$$

3.1.3. RESTRIÇÕES: FATOR DE PREENCHIMENTO

Um critério para o projeto de transformadores ótimos é que as perdas no primário sejam iguais as perdas no secundário. Matematicamente, isto significa

$$A_{Cu,s}/A_{Cu,p} = N_p/N_s \quad (29)$$

onde $A_{Cu,s}$ e $A_{Cu,p}$ são as áreas de condutores do secundário e primário, respectivamente.

Dois aspectos importantes de (29) são: tanto o primário, quanto o secundário do transformador, possuem a mesma densidade de corrente J ; os condutores do primário e secundário partilham a mesma área de janela do núcleo. Pode-se escrever o fator de preenchimento como:

$$k_{Cu} = \frac{N_p A_{Cu,p} + N_s A_{Cu,s}}{W_a} = \frac{2N_p A_{Cu,p}}{W_a} = \frac{2N_p I_p}{W_a J} \quad (30)$$

onde $I_p = S / E_p$, é a corrente nominal do primário.

Outra restrição necessária, é em relação a espessura da camada d e o número de camadas m . Assumindo que a espessura de camada é a mesma para o primário e secundário. A partir de (29) obtêm-se a seguinte relação para o número de espiras por camada:

$$N_p/N_s = N_p/N_s$$

Para garantir que os enrolamentos caibam na janela do núcleo, pode-se adicionar a seguinte restrição

$$d = \frac{A_{Cu,s}}{F_l h_w / N_s} = \frac{(N_p I_p) / (N_s J)}{F_l h_w / N_s} = \frac{N_p I_p N_s}{N_s J F_l h_w} \quad (31)$$

onde F_l é o fator de utilização de camada.

3.1.4. RESTRIÇÕES: AUMENTO DE TEMPERATURA

É necessário relacionar as perdas no núcleo e nos enrolamentos com o aumento de temperatura do transformador. O fator dominante de transferência de calor em um transformador à seco é a convecção. Pode-se escrever a resistência de térmica em °C/W como [3]:

$$R_{\theta} = \frac{1}{1,42A_t} \sqrt[4]{\frac{h_w + 2c}{\Delta T}} \quad (32)$$

onde $A_t = k_a \sqrt{2ct h_w b_w}$ é a área da superfície do transformador.

Em [3] é sugerido o uso de $k_a \cong 40$. Para calcular o aumento de temperatura é necessário saber os valores das perdas; as perdas do núcleo já foram definidas em (23). As perdas nos enrolamentos podem ser calculadas como:

$$\begin{aligned} P_{Cu} &= 2F_r \rho_w \frac{N_p \text{MLT}}{A_{Cu,p}} I_p^2 \\ &= F_r \rho_w (2N_p A_{Cu,p}) \text{MLT} \cdot J^2 \\ &= F_r \rho_w (k_{Cu} W_a) \text{MLT} \cdot J^2 \\ &= F_r \rho_w k_{Cu} V_w \cdot J^2 \end{aligned} \quad (33)$$

Portanto, pode-se escrever a restrição de temperatura como sendo:

$$R_{\theta}(P_c + P_{Cu}) = \frac{1}{1,42A_t} \sqrt[4]{\frac{h_w + 2c}{\Delta T}} (P_c + P_{Cu}) \leq \Delta T \quad (34)$$

3.1.5. RESTRIÇÕES: RENDIMENTO

Um dos requisitos imposto ao problema é que o rendimento seja maior ou igual ao que foi especificado, matematicamente isso significa que:

$$\frac{P_o}{P_o + P_c + P_{Cu}} \geq \eta_m \quad (35)$$

onde $P_o = S \cdot \cos \varphi$ é a potência de saída.

Pode-se reescrever (35) como:

$$\frac{\eta_m}{1-\eta_m}P_c + \frac{\eta_m}{1-\eta_m}P_{Cu} \leq P_o \quad (36)$$

3.1.6. RESTRIÇÕES: CORRENTE A VAZIO

É importante limitar a corrente máxima a vazio do transformador para ser uma fração k_ϕ da corrente nominal. A corrente de excitação a vazio pode ser escrita a partir da corrente de perda no núcleo e da corrente de magnetização:

$$I_\phi = \sqrt{I_c^2 + I_m^2} \quad (37)$$

A corrente de perdas no núcleo é dada por:

$$I_c = P_c/E_p \quad (38)$$

A corrente de magnetização é dada segundo a lei de Ampère:

$$I_m = \frac{B}{\mu} \frac{2h_w + 2b_w + 4c}{\sqrt{2}N_p} \quad (39)$$

onde $\mu = \mu_r \mu_0$ é a permeabilidade do núcleo e B está sujeito a restrição $B \leq B_{sat}$.

Portanto o limite para a corrente a vazio é dado por:

$$I_c^2 + I_m^2 \leq k_\phi^2 I_p^2 \quad (40)$$

3.1.7. RESTRIÇÕES: REGULAÇÃO DE TENSÃO

Um último requisito que deve-se atender é a regulação de tensão do transformador, isto é, a queda de tensão aceitável. A resistência série e a reatância do transformador influenciam diretamente na queda de tensão. A queda de tensão resistiva referida ao primário pode ser escrita como:

$$V_R = 2F_r \rho_w \frac{N_p MLT}{A_{Cu,p}} I_p \cdot \cos \phi = 2F_r \rho_w N_p MLT \cdot J \cdot \cos \phi \quad (41)$$

Já a queda de tensão reativa vista pelo primário pode ser escrita como:

$$V_X = 2\pi f L_l I_p \sin \phi \quad (42)$$

onde $\sin\varphi = \sqrt{1 - (\cos\varphi)^2}$ e L_l é a indutância de dispersão referida ao primário. Pode-se aproximar o valor da indutância de dispersão por:

$$L_l \cong \frac{\mu_0 N_p^2 \cdot \text{MLT} \cdot b_w}{3h_w p^2} \quad (43)$$

onde p é a quantidade de segmentos de bobina. Para enrolamentos com mais de um segmento de bobina, p deve ser um inteiro par, dado pelo dobro de segmentos do secundário:

$$p = \frac{2 N_s}{m N_{ls}} \quad (44)$$

Com isso, pode-se escrever a restrição de regulação de tensão como:

$$\frac{V_R + V_X}{E_p} \leq VR_m \quad (45)$$

3.2. MODELO EM FORMATO DE PG

Utilizando todas as equações vistas anteriormente e as técnicas descritas sobre modelagem em programação geométrica em [12] e [13], o sistema de equações pode ser reescrito de forma a obter-se um problema de programação geométrica. Todas as variáveis do modelo assumem somente valores positivos. Utilizando como variáveis $c, t, h_w, b_w, N_p, N_s, N_{ls}, B, J$ e como limites superiores $\bar{P}_c, \bar{P}_{Cu}, \bar{I}_m, \bar{V}_R, \bar{V}_X$ e \bar{h} pode-se reescrever as equações anteriores.

3.2.1. FUNÇÃO OBJETIVO

A função objetivo (26) pode ser reescrita como:

$$4k_f \rho_c c t h_w + 8k_f \rho_c c^2 t + 4k_f \rho_c c t b_w + 4k_{Cu} \rho_{Cu} c b_w h_w + 2k_{Cu} \rho_{Cu} t b_w h_w + \pi k_{Cu} \rho_{Cu} b_w^2 h_w \quad (46)$$

3.2.2. RESTRIÇÕES: TENSÃO INDUZIDA

Depois de substituir-se $A_c = 2ct$, a restrição (27) pode ser escrita como:

$$2^{\frac{3}{2}} \pi f k_f E_p^{-1} N_p B c t = 1 \quad (47)$$

E a relação de espiras como:

$$E_s E_p^{-1} N_s^{-1} N_p = 1 \quad (48)$$

3.2.3. RESTRIÇÕES: FATOR DE PREENCHIMENTO

Substituindo W_a por $b_w h_w$, então (30) torna-se:

$$2k_{Cu}^{-1} N_p I_p J^{-1} b_w^{-1} h_w^{-1} = 1 \quad (49)$$

Já a restrição na dimensão do condutor pode ser escrita em formato de posinômio como:

$$F_l^{-1} d^{-1} I_p J^{-1} N_p N_s^{-1} N_{ls} h_w^{-1} = 1 \quad (50)$$

3.2.4. RESTRIÇÕES: AUMENTO DE TEMPERATURA

Para reescrever (34) em formato de PG, substitui-se $(h_w + 2c)$ por \bar{h} [13]. Para simplificar a formulação, pode-se substituir as perdas no núcleo e nos enrolamentos pelos seus limites superiores (\bar{P}_c e \bar{P}_{Cu}). Então (34) fica como:

$$\frac{\bar{P}_c \bar{h}^{\frac{1}{4}} c^{-\frac{1}{2}} t^{-\frac{1}{2}} b_w^{-\frac{1}{2}} h_w^{-\frac{1}{2}}}{1,42\sqrt{2}k_a \Delta T^{\frac{5}{4}}} + \frac{\bar{P}_{Cu} \bar{h}^{\frac{1}{4}} c^{-\frac{1}{2}} t^{-\frac{1}{2}} b_w^{-\frac{1}{2}} h_w^{-\frac{1}{2}}}{1,42\sqrt{2}k_a \Delta T^{\frac{5}{4}}} \leq 1 \quad (51)$$

O limite superior para $(h_w + 2c)$ é:

$$h_w \bar{h}^{-1} + 2c \bar{h}^{-1} \leq 1 \quad (52)$$

Substituindo V_c por $2ct(2h_w + 2b_w + 4c)$ em (23) pode-se escrever o limite superior para as perdas no núcleo como:

$$4k_f K_c \rho_c f^\alpha B^\beta c t h_w \bar{P}_c^{-1} + 8k_f K_c \rho_c f^\alpha B^\beta c^2 t \bar{P}_c^{-1} + 4k_f K_c \rho_c f^\alpha B^\beta c t b_w \bar{P}_c^{-1} \leq 1 \quad (53)$$

Já o limite superior para as perdas nos enrolamentos pode ser escrito a partir de (33) fazendo a substituição de $V_w = b_w h_w (4c + 2t + \pi b_w)$ como:

$$4k_{Cu} F_r \rho_w c b_w h_w J^2 \bar{P}_{Cu}^{-1} + 2k_{Cu} F_r \rho_w t b_w h_w J^2 \bar{P}_{Cu}^{-1} + \pi k_{Cu} F_r \rho_w b_w^2 h_w J^2 \bar{P}_{Cu}^{-1} \leq 1 \quad (54)$$

3.2.5. RESTRIÇÕES: RENDIMENTO

A restrição de rendimento (36) pode ser reescrita utilizando os limites superiores de \bar{P}_c e \bar{P}_{Cu} como:

$$\frac{\eta_m}{(1-\eta_m)P_o} \bar{P}_{Cu} + \frac{\eta_m}{(1-\eta_m)P_o} \bar{P}_c \leq 1 \quad (55)$$

3.2.6. RESTRIÇÕES: CORRENTE A VAZIO

Adicionando a variável \bar{I}_m como o limite superior da corrente de magnetização, então a restrição de corrente de magnetização (39) fica:

$$\frac{\sqrt{2}}{\mu} BN_p^{-1} h_w \bar{I}_m^{-1} + \frac{\sqrt{2}}{\mu} BN_p^{-1} b_w \bar{I}_m^{-1} + \frac{2\sqrt{2}}{\mu} BN_p^{-1} c \bar{I}_m^{-1} \leq 1 \quad (56)$$

A restrição para a saturação do núcleo é escrita como:

$$B_{sat}^{-1} B \leq 1 \quad (57)$$

Como \bar{P}_c/E_p é o limite superior para I_c , então a restrição para a corrente a vazio (40) é reescrita como:

$$k_\varphi^{-2} I_p^{-2} \bar{I}_m^2 + E_p^{-2} k_\varphi^{-2} I_p^{-2} \bar{P}_c^2 \leq 1 \quad (58)$$

3.2.7. RESTRIÇÕES: REGULAÇÃO DE TENSÃO

Substitui-se em (41) MLT por $4c + 2t + \pi b_w$, então o limite superior da queda de tensão resistiva \bar{V}_R , fica como:

$$8F_r \rho_w \cdot \cos \varphi \cdot N_p c J \bar{V}_R^{-1} + 4F_r \rho_w \cdot \cos \varphi \cdot N_p t J \bar{V}_R^{-1} + 2\pi F_r \rho_w \cdot \cos \varphi \cdot N_p b_w J \bar{V}_R^{-1} \leq 1 \quad (59)$$

Para o limite da queda de tensão reativa, substitui-se em (42) a expressão de L_l (43) e também o valor de MLT, então tem-se:

$$\begin{aligned} & \frac{8\pi}{3p^2} I_p \cdot \sin \varphi \cdot f \mu_0 N_p^2 c b_w h_w^{-1} \bar{V}_x^{-1} + \frac{4\pi}{3p^2} I_p \cdot \sin \varphi \cdot f \mu_0 N_p^2 t b_w h_w^{-1} \bar{V}_x^{-1} \\ & + \frac{2\pi^2}{3p^2} I_p \cdot \sin \varphi \cdot f \mu_0 N_p^2 b_w^2 h_w^{-1} \bar{V}_x^{-1} \leq 1 \end{aligned} \quad (60)$$

Como a restrição (50) necessita que N_{ls} seja uma das variáveis a serem otimizadas, então m é especificado pelo projetista e p deve ser substituído em (60) pela expressão em (44). Pode-se ainda adicionar as seguintes condições:

$$N_{ls}^{-1} \leq 1, \quad mN_s^{-1}N_{ls} \leq 1 \quad (61)$$

para garantir que N_{ls} seja maior ou igual a um e que N_s seja maior que mN_{ls} .

Usando as variáveis definidas para os limites superiores, então a regulação de tensão (45) fica como:

$$E_p^{-1}VR_m^{-1}\bar{V}_R + E_p^{-1}VR_m^{-1}\bar{V}_X \leq 1 \quad (62)$$

Com isto, temos a representação em PG do problema de mínima massa de um transformador, o objetivo é dado pela função (46) e as restrições são dadas pelas equações de (47) à (62), que são monômios ou posinômios.

Este modelo leva em conta os efeitos de alta frequência nos condutores, e portanto deve ser alterado para otimizar transformadores de baixa frequência. Para tal, altera-se o valor de F_r para 1,05 conforme sugerido em [14]. Como a espessura do condutor é muito menor que a profundidade pelicular a 60Hz, então não é necessário definir os valores de m e d , e portanto N_{ls} não é mais uma das variáveis a ser otimizadas. Neste caso, as restrições (50) e (61) não devem ser incluídas no cálculo.

3.2.8. EXTENSÕES: PROJETO COM MÍNIMAS PERDAS

O modelo formulado pode ser alterado para ter como objetivo minimizar as perdas, no lugar de minimizar a massa do transformador. Neste caso as dimensões do núcleo são definidas pelo projetista, que podem ser obtidas de um banco de dados de núcleos. As variáveis a serem otimizadas, neste caso, se reduzem a $N_p, N_s, N_{ls}, B, J, \bar{P}_c, \bar{P}_{Cu}, \bar{I}_m, \bar{V}_R, \bar{V}_X$ e \bar{h} .

Como o problema pede por mínimas perdas, então a função objetivo é dada por

$$\bar{P}_c + \bar{P}_{Cu} \quad (63)$$

todas as restrições do problema original são utilizadas, com exceção da equação (55), uma vez que minimizar as perdas é um problema equivalente a maximizar o rendimento.

Para um transformador de baixa frequência, utiliza-se a mesma analogia do problema original, então N_{ls} não é mais uma variável a ser otimizada, e as equações (50) e (61) não são incluídas no conjunto de restrições.

3.2.9. EXTENSÕES: PROJETO COM POTÊNCIA DE SAÍDA MÁXIMA

Outra formulação possível é maximizar a potência de saída, para uma estrutura de núcleo dada. Para isso, adiciona-se às variáveis otimizadas a corrente do primário, então o conjunto de variáveis a serem otimizadas é $I_p, N_p, N_s, N_{ls}, B, J, \bar{P}_c, \bar{P}_{Cu}, \bar{I}_m, \bar{V}_R, \bar{V}_X$ e \bar{h} .

Como o objetivo do problema é maximizar I_p , então em forma de PG, minimiza-se a seguinte função:

$$I_p^{-1} \quad (64)$$

Em baixas frequências, a variável N_{ls} e as equações (50) e (61) não são incluídas no conjunto de restrições do PG.

4. DESENVOLVIMENTO DO SOFTWARE

Existem diversas maneiras para se criar um software que resolva o problema de programação geométrica descrito anteriormente. Pode-se utilizar uma ferramenta pronta como o software MOSEK [18], usar o Matlab em conjunto com uma Toolbox para otimizações convexas ou ainda implementar um software que resolva somente este problema.

Um dos requisitos impostos ao software era a necessidade de rodar o programa sem depender de ferramentas pagas, portanto resolveu-se por implementar todo o software que resolvesse a otimização das equações vistas anteriormente. Uma vez que os algoritmos de solução de problemas convexos não são triviais de serem implementados, realizou-se uma pesquisa por uma biblioteca que realizasse a solução numérica deste problema. A melhor biblioteca encontrada foi a CVXOPT [19], implementada pelos autores já citados [12][13].

Uma das vantagens de se utilizar uma biblioteca é que não é necessário “reinventar a roda” cada vez que se necessita uma função específica. A sua utilização acaba diminuindo a quantidade de problemas encontrados no software, uma vez que o programador pode focar-se em escrever somente a lógica para resolver seu problema sem precisar entender em detalhes os algoritmos utilizados pela biblioteca. A utilização de bibliotecas é uma forma de reuso de software, o que diminui o tempo de desenvolvimento de aplicações, pois diminui-se a quantidade de código repetido no software. Então em última instância, reuso de software acaba levando à melhores aplicações e a um *time-to-market* menor.

A biblioteca CVXOPT é implementada utilizando a linguagem Python [20]. Esta linguagem tem sido amplamente utilizada para a resolução de problemas numéricos, havendo diversas bibliotecas disponíveis que auxiliam sua resolução. Também tem sido adotada como uma alternativa livre à programas como Matlab e Mathematica, utilizando a biblioteca Numpy [21] para realizar esta tarefa. Algumas faculdades [22] tem adotado Python como primeira linguagem a ser ensinada em cursos de graduação.

Algumas vantagens que Python apresenta são: é uma linguagem interpretada, portanto o tempo entre escrever código e testá-lo é mínimo; python é gratuito, mantido por uma comunidade extremamente ativa, e multiplataforma; como linguagem de programação apresenta suporte a diferentes paradigmas de programação (orientação a objetos, imperativo e funcional), tipagem dinâmica e gerenciamento automático de memória; possibilita o desenvolvimento extremamente ágil de softwares; e finalmente, é uma linguagem divertida de se utilizar.

Outra área que tem crescido para a linguagem Python é sua utilização em plataformas embarcadas como *smart-phones*, diversos modelos já oferecem suporte a Python com bibliotecas específicas para acessar as principais funções do celular: GPS, lista de contatos, etc.

Para a interface gráfica, optou-se pela utilização do *toolkit* gráfico GTK+ [23] na sua versão para Python, conhecida como PyGTK [24]. GTK+ é uma plataforma para desenvolvimento de interfaces gráficas *open source*, com suporte a diferentes sistemas operacionais e também possui *bindings* (lógica de cola entre diferentes linguagens de programação) da sua API (*Application Programming Interface*) para diversas linguagens de programação.

O projeto do software foi dividido em duas partes, a primeira consiste em como representar as equações no formato requerido pela biblioteca CVXOPT. A segunda parte é constituída da interface gráfica para a interação do projetista e demais requisitos para a boa interação com o software – por exemplo, a estrutura do arquivo para salvar o modelo do transformador otimizado. Nas próximas seções será demonstrado o desenvolvimento destas duas etapas.

4.1. BIBLIOTECA DE OTIMIZAÇÕES CONVEXAS: CVXOPT

A biblioteca CVXOPT oferece ao usuário algoritmos para a resolução de diversas formas de otimizações convexas, entre eles para problemas lineares, quadráticos, cônicos de segunda ordem, não-lineares convexas e de programação geométrica. Cada um desses métodos de resolução recebe diversos parâmetros como matrizes e vetores para resolver o problema. Apesar disto ser a principal função da biblioteca, ela também dispõe de diversos objetos e funções auxiliares, como por exemplo uma interface para o uso da FFTW [25] (biblioteca de transformada de Fourier), BLAS (*Basic Linear Algebra Subprograms*) [26], conjunto de rotinas em Fortran para operações com matrizes e vetores, e LAPACK (*Linear Algebra PACKage*) [27], uma biblioteca para resolução de problemas lineares, fatoração de matrizes e problemas de autovalores.

Como os dados para todas estas funções são matrizes, os programadores de biblioteca criaram um objeto extremamente versátil para representar isto. O objeto *matrix* possibilita a criação de matrizes, que podem ser preenchidas com dados posteriormente, acessando diretamente a linha e coluna necessária. Este objeto também permite extrair partes da matriz de forma bastante simples (operação de *slicing*), assim como suporte a operações aritméticas simples (adição, subtração, multiplicação e divisão). Várias dessas funções foram utilizadas na criação do programa.

A função para resolução de problemas de programação geométrica apresenta a seguinte forma:

```
solvers.gp(K, F, g[, G, h[, A, b]])
```

que representa a solução do problema:

$$\begin{aligned}
&\text{minimizar } f_0(x) = \log \sum_k \exp(F_0 x + g_0) \\
&\text{sujeito a } f_i(x) = \log \sum_k \exp(F_i x + g_i) \leq 0, \quad i = 1, \dots, m \\
&Gx \preceq h \\
&Ax = b
\end{aligned} \tag{65}$$

onde:

K é uma lista de inteiros representando o número de linhas em F_i ;

F é a uma matriz real com os coeficientes das variáveis a serem otimizadas das inequações;

g é uma matriz-coluna com o valor do logaritmo das constantes que multiplicam as variáveis;

A é matriz que representa os coeficientes das igualdades impostas;

b é uma matriz-coluna com o valor do logaritmo das constantes multiplicada por -1.

Para resolver o problema de mínima massa, utilizando as equações (46) à (62), pode-se escrever as seguinte matrizes, aqui escritas de forma reduzida. No Anexo I são apresentadas as matrizes completas para os três problemas de otimização.

$$x = \begin{pmatrix} c \\ t \\ h_w \\ \vdots \\ \bar{h} \end{pmatrix} \tag{66}$$

$$F = \begin{pmatrix} 1 & 1 & 1 & \dots & 0 \\ 2 & 1 & 0 & & 0 \\ 1 & 1 & 0 & & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \quad g = \log \begin{pmatrix} 4k_f \rho_c \\ 8k_f \rho_c \\ 4k_f \rho_c \\ \vdots \\ E_p^{-1} V R_m \end{pmatrix} \tag{67}$$

$$A = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \\ 0 & 0 & -1 & \dots & 0 \end{pmatrix} \quad b = -\log \begin{pmatrix} 2^{\frac{3}{2}} \pi f k_f E_p^{-1} \\ E_s E_p^{-1} \\ 2k_{\text{Cu}}^{-1} I_p \\ F_l^{-1} d^{-1} I_p \end{pmatrix} \quad (68)$$

Com estas matrizes, e as demais para os dois outros problemas, criou-se o objeto Solver para abstrair a geração das matrizes e a resolução dos problemas. Os parâmetros definidos pelo usuário do software são passados a uma das três funções de otimização, que então cria as matrizes necessárias e executa o algoritmo.

4.2. ESTRUTURA DO SOFTWARE

O software foi implementado utilizando o paradigma de programação de orientação a objetos, onde cada objeto é responsável por uma função específica, agrupando-se conceitos e simplificando a manutenção do software. Por exemplo, como visto na seção anterior, o objeto Solver é responsável por toda a interação com a biblioteca CVXOPT, sendo ali definido os diferentes métodos para cada tipo de otimização.

Na Figura 5 pode-se ver um diagrama simplificado das classes envolvidas no software. Este diagrama foi utilizado como guia para a implementação do software, alguns ajustes durante a implementação foram necessários, mas os conceitos do que cada objeto deveria fazer foram preservados. Em alguns casos separou-se um objeto em dois distintos, aplicando as técnicas de refactoring [28], para melhorar a qualidade e manutenibilidade do software.

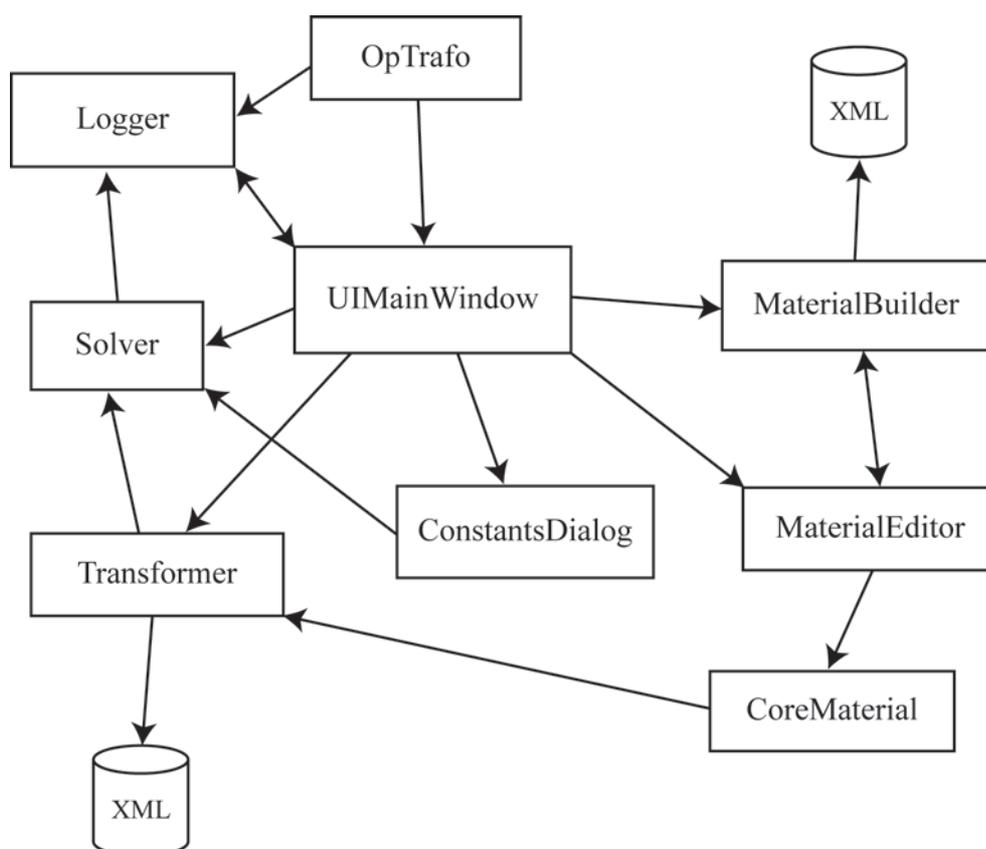


Figura 5 Diagrama de Classes simplificado, ilustrando as relações entre os diversos objetos.

Outra técnica utilizada em projetos de software orientados a objetos é a utilização de design patterns [29], isto é, algumas construções de interação entre objetos foram catalogadas para facilitar e padronizar suas implementações. Aplicou-se a pattern de Factory para o objeto CoreMaterial, criando-se um objeto CoreMaterialBuilder, responsável criar os objetos a partir de um arquivo XML e retornar o objeto correto quando necessário.

4.3. FORMATO DE ARQUIVO

Como um requisito necessário em um software de auxílio a projetos é a possibilidade de salvar os projetos para que eles possam ser abertos em um outro momento, por exemplo, para ajustar os valores após uma consulta com fabricantes de núcleos – caso seja decidido utilizar-se um núcleo comercial. Criou-se uma representação em XML para salvar os dados entrados no software pelo projetista.

XML (Extensible Markup Language) é um conjunto de regras para codificar documentos eletrônicos, de modo que seja possível criar um sistema próprio para cada tipo de documento. Algumas vantagens deste sistema é que os dados são gravados em formato de texto e que a linguagem preza pela simplicidade.

Para representar os dados de projeto de um transformador, foi criada uma estrutura como a que pode ser vista na listagem abaixo.

```
<?xml version='1.0' encoding='UTF-8'?>
<optrafo version="2.0">
  <transformer>
    <primary_voltage>220</primary_voltage>
    <secondary_voltage>12</secondary_voltage>
    <power>3000</power>
    <power_factor>0.9</power_factor>
    <frequency>50</frequency>
    <temperature>30</temperature>
    <temperature_rise>60</temperature_rise>
    <efficiency>0.95</efficiency>
    <voltage_regulation>0.05</voltage_regulation>
    <current_ratio>0.03</current_ratio>
    <core_material>0</core_material>
    <optimization_parameters type="loss">
      <c>4e-2</c>
      <t>8e-2</t>
      <bw>3.5e-2</bw>
      <hw>22e-2</hw>
    </optimization_parameters>
    <optimization_parameters type="power">
      <c>1e-2</c>
      <t>1e-2</t>
      <bw>1e-2</bw>
      <hw>1e-2</hw>
    </optimization_parameters>
  </transformer>
</optrafo>
```

Nesta estrutura estão representados todos os valores necessários para que se tenha um modelo completo do transformador que se deseja otimizar.

4.4. INTERFACE COM O USUÁRIO

Para a interface gráfica com o usuário, buscou-se simplificar a interação do usuário com o software, separando-se os parâmetros comuns às três otimizações numa área fixa e utilizando um modelo de abas para informar os valores específicos de cada otimização. Para exibir os valores calculados e mensagens sobre o estado do software utilizou-se uma área de texto com barra de rolagem.

A partir destes conceitos buscou-se a melhor forma para implementar esta interface. O *toolkit* gráfico GTK+ oferece a ferramenta Glade Interface Designer, onde é possível montar qualquer interface gráfica arrastando os componentes desejados para um modelo de janela. Esta ferramenta permite a prototipação e estudos de interfaces sem que seja necessário escrever o código da interface, o que diminui bastante o tempo de desenvolvimento.

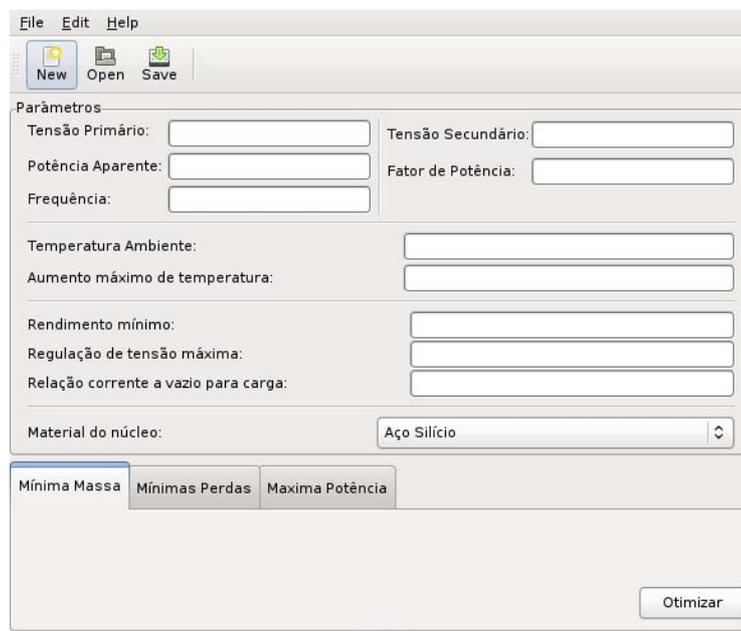
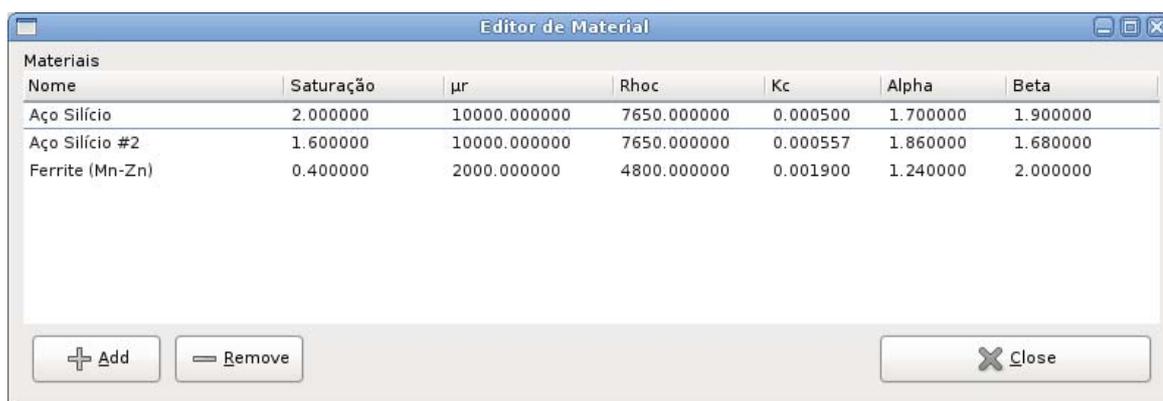


Figura 6 Interface gráfica do software

Na Figura 6 apresenta-se o resultado da interface gráfica criada, com uma barra de menus, uma barra de ícones para as funções mais utilizadas, a área com os parâmetros comuns – basicamente os dados de placa de um transformador, as abas com os diferentes tipos de otimização e por último a área de texto que apresentará os valores calculados.

Na barra de menus vale destacar o conteúdo do item Edit, onde está localizado o acesso ao editor de materiais e a janela de constantes. A janela de constantes permite alterar e visualizar o valor de algumas constantes utilizadas na otimização, são elas: o fator de preenchimento de cobre na janela do núcleo, o fator de utilização de camada, o fator de empilhamento do núcleo e o fator da área de enrolamento. O software utiliza como valores padrões para estas constantes: 0,60; 0,90; 0,95 e 40,00 respectivamente.

O editor de materiais consiste em uma lista de materiais cadastrados no software e as constantes que caracterizam cada um deles. Estas constantes são aquelas descritas na Tabela 1 (página 32). A principal função deste editor é possibilitar o usuário adicionar e alterar as propriedades do material utilizado no núcleo do transformador. Toda a alteração nas propriedades dos materiais é refletida em um arquivo XML que representa a lista exibida. A Figura 7 mostra o editor de materiais.



Nome	Saturação	μ_r	Rhoc	Kc	Alpha	Beta
Aço Silício	2.000000	10000.000000	7650.000000	0.000500	1.700000	1.900000
Aço Silício #2	1.600000	10000.000000	7650.000000	0.000557	1.860000	1.680000
Ferrite (Mn-Zn)	0.400000	2000.000000	4800.000000	0.001900	1.240000	2.000000

Figura 7 Editor de materiais, exibindo os três materiais cadastrados previamente.

5. COMPARAÇÃO

Realizou-se uma comparação inicial em relação aos resultados exibidos no trabalho original [14], este passo foi de extrema importância para mostrar se as matrizes implementadas estavam corretas ou se haviam problemas no software. Alguns erros nos sinais dos termos nas matrizes foram encontrados dessa maneira, e foram facilmente corrigidos.

O modelo usando nesta comparação inicial foi um transformador de potência com as seguintes características:

- Tensão no primário 480V
- Tensão no secundário 240V
- Frequência 60Hz
- Fator de potência 0,8
- Temperatura ambiente 65°C, máxima sobrelevação de temperatura 85°C
- Rendimento mínimo 97%
- Regulação de tensão máxima 5%
- Relação de corrente a vazio para carga 2%
- Largura da coluna 3,1 cm, profundidade da coluna 9,3 cm, largura da janela 6,2 cm e altura da janela 15,5 cm
- Material do núcleo Aço Silício

A partir destes dados, utilizou-se o software para maximizar a potência do transformador projetado. Os resultados obtidos foram bastante satisfatórios, o que possibilitou validar a implementação desta metodologia. Na Tabela 2 estão apresentados na segunda coluna os valores obtidos no software e na terceira coluna os resultados apresentados em [14].

Tabela 2 - Comparação inicial entre o software implementado e o modelo original

Quantidade	Projeto com OpTrafo	Projeto original
Potência	10.094,4 VA	10.082,6 VA
Espiras primário	164,36 (164)	164,36 (164)
Espiras secundário	82,18 (82)	82,18 (82)
Densidade de fluxo magnético	2,00 T	2,00 T
Densidade de corrente	1,199 A/mm ²	1,20 A/mm ²
Perdas no núcleo	45,99 W	45,99 W
Perdas no cobre	113,9 W	113,89 W
Rendimento	98,06 %	98,06%
Regulação de tensão	1,469 %	1,47%
Corrente de magnetização	0,382 A	0,382 A
Corrente de perdas no núcleo	0,096 A	0,096 A
Corrente a vazio	0,394 A	0,394 A
Sobrelevação de temperatura	85 °C	85 °C

Para fazer uma análise mais extensa da qualidade do método, escolheu-se um projeto de transformador que possuísse todos os seus dados de cálculo publicados. Na próxima seção será desenvolvida esta comparação.

5.1. TRANSFORMADOR DE 300W

Este modelo foi extraído do primeiro exemplo do segundo capítulo de [30], onde é calculado um pequeno transformador elevador de tensão. O método de projeto apresentado neste livro consiste em dimensionar a seção magnética do núcleo, e a partir deste valor escolhe-se uma laminação padronizada para o núcleo. Os dados fornecidos no livro para o cálculo são:

- Tensão no primário 120V
- Tensão no secundário 220V

- Frequência 50Hz
- Potência 300VA
- Sobrelevação máxima de temperatura 80°C

Além destes dados, são necessários mais alguns parâmetros para que se possa utilizar o software. Como nem todos estes parâmetros são dedutíveis a partir dos dados do exemplo, arbitrou-se estes parâmetros para valores comumente aceitos.

- Fator de potência 0,98
- Temperatura ambiente 20°C
- Rendimento mínimo 90%
- Regulação máxima de tensão 5%
- Relação de corrente a vazio para carga 2%

Além destes dados iniciais são necessários mais alguns valores para o correto funcionamento do software. O material selecionado para o núcleo foi o aço silício com as características descritas na Tabela 1. Para o fator de preenchimento do cobre foi utilizado o valor $k_{Cu} = 0,6$, o fator de empilhamento $k_f = 0,95$, o fator de área de enrolamento $k_a = 40$, e a relação entre a resistência AC para DC com valor $F_r = 1,05$.

Na Tabela 3 são apresentados os resultados obtidos, na segunda coluna é apresentado os valores do modelo de referência [30], na terceira e quarta coluna estão os dados calculados pelo software para um projeto de mínima massa e de máxima potência respectivamente. O objetivo de apresentar o cálculo para um transformador de máxima potência é meramente um indicativo de que o transformador de referência está sobredimensionado.

Tabela 3 - Comparação entre modelo original e otimizado

	Transformador	OpTrafo	OpTrafo
	Martignoni	Mínima Massa	Potência Máxima
c	2,50 cm	1,38 cm	2,50 cm
t	4,00 cm	6,26 cm	4,00 cm
h_w	15,00 cm	5,85 cm	15,00 cm
b_w	2,50 cm	2,43 cm	2,50 cm
Espiras primário (N_p)	264	289	142
Espiras secundário (N_s)	532	530	261
Densidade de fluxo magnético	1,13 T	1,133 T	2,00 T
Densidade de corrente	2,63 A/mm ²	1,69 A/mm ²	1,29 A/mm ²
Potência	300 VA	300 VA	1232,87 VA
Perdas no núcleo	8,33 W	1,36 W	9,43 W
Perdas no cobre	21,70 W	14,93 W	23,23 W
Rendimento	91 %	94,75%	90%
Regulação de tensão	–	5%	1,90%
Corrente de magnetização	0,108 A	0,048 A	0,356 A
Corrente de perdas no núcleo	0,069 A	0,012 A	0,078 A
Corrente a vazio	0,128 A	0,050 A	0,365 A
Sobrelevação de temperatura	43 °C	33 °C	47 °C

Comparando primeiramente somente a terceira coluna em relação a segunda, pode-se notar a semelhança de alguns valores, como o número de espiras, o fluxo magnético e a largura da janela. Isto indica que o modelo utilizado pelo software está convergindo para valores de projeto aceitáveis. Uma análise mais detalhada dos valores gerados é necessária para verificar se os valores encontrados representam um transformador de mínima massa.

Calculando a área da coluna central do transformador obtém-se para o transformador de referência 20 cm² de área, enquanto o otimizado possui uma área de 17,32 cm², o que indica uma redução de 13,4% na seção da coluna central do transformador. A altura da janela também diminui de 15 cm para 5,85cm. Com base nesses fatores, fica claro que o núcleo é menor e portanto possui menos massa em relação ao transformador original.

O transformador calculado também apresenta menores perdas, o que por sua vez confirma o aumento no rendimento apresentado – que passou de 91% para 94,75%, um aumento de 4%. Também houve uma melhora na sobrelevação de temperatura, que diminuiu de 43°C para 33°C.

Já na quarta coluna são apresentados os valores da otimização para máxima potência utilizando as dimensões de núcleo do projeto inicial. Pode-se notar que há um enorme aumento, cerca de quatro vezes mais, na potência fornecida pelo transformador otimizado. O aumento nas perdas não é tão significativo quando comparado ao modelo original, um pouco mais de 10% para as perdas no núcleo e um pouco menos de 10% para as perdas no cobre. A sobrelevação máxima de temperatura também é pouco alterada.

Estes valores demonstram que o transformador original está com um núcleo superdimensionado para a potência requerida. Um transformador ótimo para a potência especificada teria suas dimensões físicas bastante reduzidas, como pode ser visto na terceira coluna da tabela, ou então pode-se utilizar o núcleo original para um transformador com potência nominal bastante superior, como indicado na quarta coluna.

6. CONCLUSÃO

Como foi apresentado no desenvolvimento desse trabalho, a otimização de transformadores depende de diversos fatores, mas dois devem ser destacados: o modelo utilizado e a técnica de otimização empregada. Um bom modelo leva a melhores resultados, com características mais próximas à máquina real. Já a técnica de otimização representa o quanto se consegue melhorar as variáveis do modelo, isto é o quão próximo de um ótimo global pode-se chegar a partir de um dado modelo. Portanto é ideal utilizar-se um bom conjunto de modelo e método de otimização.

O método de programação geométrica utilizado apresenta ótimos resultados e possuiu ainda o diferencial de encontrar um ótimo global. Estas vantagens fazem com que este método possua uma área de aplicação bastante vasta, podendo ser usado para otimizar diversos tipos de projetos de engenharia, como projetos de sistemas de comunicação, dimensionamento e posicionamento de portas lógicas, projeto de estruturas ou ainda em engenharia química. Boyd *et al.* listam muitas outras aplicações[13], inclusive fora da área de engenharia. Para isso, é preciso escrever o objetivo e as restrições nos moldes da programação geométrica, e aplicar o método para cada um destes casos.

O modelo apresentado é fortemente baseado nas equações físicas que regem o comportamento elétrico, térmico, mecânico, etc. Equações mais precisas para alguns comportamentos podem ser adicionadas ao modelo para que se tenha resultados ainda melhores. Por exemplo a equação que rege a curva de magnetização pode ser aproximada por um modelo não linear.

Apesar do bom resultado apresentado pelo método de otimização utilizado, mais investigações são necessárias para validar complemente o método. A utilização de um método de elementos finitos para simular o funcionamento do projeto otimizado ou até mesmo a cons-

trução de um transformador seguindo os valores fornecidos pelo software ofereceriam novos dados para serem analisados, e a partir destes seria possível validar totalmente o método.

Outra possibilidade que o trabalho apresentou é que o modelo utilizado pelo software poderia ser aumentado para suportar mais tipos de formatos de núcleos, sistemas trifásicos, sistemas de refrigeração diferentes e um conjunto maior de restrições. Para adicionar estas variações ao modelo é interessante que haja melhorias na infraestrutura do software. Por exemplo, é interessante que haja um método capaz de gerar as matrizes a partir do sistema de equações ao invés de ter uma matriz fixa para cada tipo de otimização, já que isso permitiria alterar mais facilmente o modelo e dar mais liberdade para o projetista criar restrições.

Enfim, dentro do âmbito acadêmico deste trabalho, pode-se dizer que os resultados da pesquisa em torno da otimização de transformadores foram altamente satisfatórios. Fica claro que o caminho percorrido até aqui mostrou a utilidade e qualidade do método proposto. Novas pesquisas e investigações podem ser iniciadas a partir deste ponto, e algumas propostas com este intuito já foram mencionadas neste trabalho.

7. REFERÊNCIAS

1. HOBBSAWM, E. **The Age Of Revolution, 1789 - 1848**. Londres: Weidenfeld & Nicolson, 1962. 356 p.
2. LEE, R.; MULLINIX, N. Transformer Design Chart. **Transactions of the IRE Professional Group on Component Parts**, v. 3, n. 1, p. 10-14, Abril 1955.
3. HURLEY, W. G.; WÖLFLE, W. H.; BRESLIN, J. G. Optimized Transformer Design: Inclusive of High-Frequency Effects. **IEEE Transactions on Power Electronics**, v. 13, n. 4, p. 651-659, Julho 1998.
4. WILLIAMS, S. B.; ABETTI, P. A.; MAGNUSSON, E. F. Application of Digital Computers to Transformer Design. **Power Apparatus and Systems, Part III. Transactions of the American Institute of Electrical Engineers**, v. 75, n. 3, p. 728-735, Agosto 1956.
5. WEBER, H. J.; GALLOUSIS, G. Computers Change Transformer Design Philosophy. **Power Apparatus and Systems, Part III. Transactions of the American Institute of Electrical Engineers**, v. 78, n. 8, p. 215-218, Junho 1959.
6. SHULL JR., D. S. Rapid design of electronic power transformers using a computer time-sharing system. **IEEE Transactions on Magnetics**, v. 5, n. 4, p. 934-938, Dezembro 1969.
7. ODESSEY, P. H. Transformer Design by Computer. **IEEE Transactions on Manufacturing Technology**, v. 3, n. 1, p. 1-17, Junho 1974.
8. JUDD, F. F.; KRESSLER, D. R. Design Optimization of Small Low-Frequency Power Transformer. **IEEE Transactions on Magnetics**, v. 13, n. 4, p. 1058-1069, Julho 1977.
9. ANDERSEN, O. W. Optimized design of electric power equipment. **Computer Applications in Power**, v. 4, n. 1, p. 11-15, Janeiro 1991.
10. GEROMEL, L. H.; SOUZA, C. R. The application of intelligent systems in power transformer design. **IEEE CCECE 2002. Canadian Conference on Electrical and Computer Engineering, 2002**, v. 1, p. 285-290, 2002.
11. HUI, L. et al. Application research based on improved genetic algorithm for optimum design of power transformers. **Proceedings of the Fifth International Conference on Electrical Machines and Systems, 2001. ICEMS 2001**, v. 1, p. 242-245, 2001.
12. BOYD, S.; VANDENBERGHE, L. **Convex Optimization**. [S.l.]: Cambridge University Press.
13. BOYD, S. et al. A tutorial on geometric programming. **Optimization and Engineering**, v. 8, n. 1, p. 67-127, 2007.
14. JABR, R. A. Application of Geometric Programming to Transformer Design. **IEEE Transactions on Magnetics**, v. 41, n. 11, p. 4261-4269, Novembro 2005.
15. RIES, W. **Transformadores - Fundamentos para o Projeto e Cálculo**. Porto Alegre: EDIPUCRS, 2007.
16. CHARAP, S.; JUDD, F. A Core Loss Model for Laminated Transformers. **IEEE Transactions on Magnetics**, v. 10, n. 3, p. 678-681, Setembro 1974.
17. RUBAAI, A. Computer Aided Instruction of Power Transformer Design in the Undergraduate Power Engineering Class. **IEEE Transactions on Power Systems**, v. 9,

- n. 3, p. 1174-1181, Agosto 1994.
18. MOSEK. Disponivel em: <<http://www.mosek.com/>>. Acesso em: Agosto 2009.
 19. DAHL, J.; VANDENBERGHE, L. **CVXOPT - Python Software for Convex Optimization**, Setembro 2009. Disponivel em: <<http://abel.ee.ucla.edu/cvxopt/>>.
 20. PYTHON SOFTWARE FOUNDATION. **Python Programming Language**, Setembro 2009. Disponivel em: <<http://www.python.org/>>.
 21. **NumPy**, Outubro 2009. Disponivel em: <<http://www.numpy.org/>>.
 22. PYTHON in Education, 2009. Disponivel em: <<http://www.python.org/community/sigs/current/edu-sig/>>. Acesso em: Outubro 2009.
 23. GTK+ , 2009. Disponivel em: <<http://www.gtk.org/>>. Acesso em: Novembro 2009.
 24. THE GNOME PROJECT AND PYGTK TEAM. **PyGtk: GTK+ for Python**, Setembro 2009. Disponivel em: <<http://www.pygtk.org/>>.
 25. **FFTW**, Outubro 2009. Disponivel em: <<http://www.fftw.org/>>.
 26. **BLAS (Basic Linear Algebra Subprograms)**, Outubro 2009. Disponivel em: <<http://www.netlib.org/blas/>>.
 27. **LAPACK - Linear Algebra PACKage**, Outubro 2009. Disponivel em: <<http://www.netlib.org/lapack/>>.
 28. FOWLER, M. **Refactoring - Improving the Design of Existing Code**. [S.l.]: Addison Wesley, 1999. 431 p.
 29. GAMMA, E. et al. **Design Patterns - Elements of Reusable Object-Oriented Software**. [S.l.]: Addison Wesley, 1994. 395 p.
 30. MARTIGNONI, A. **Transformadores**. Porto Alegre: Editora Globo, 1973. 307 p.

ANEXO

Este anexo apresenta as matrizes completas que foram montadas para executar o algoritmo de otimização.

- Mínima massa

$$x = \begin{pmatrix} c \\ t \\ h_w \\ b_w \\ N_p \\ N_s \\ N_{ls} \\ B \\ J \\ \bar{P}_c \\ \bar{P}_{Cu} \\ \bar{I}_m \\ \bar{V}_r \\ \bar{V}_x \\ \bar{h} \end{pmatrix} \quad b = -\log \begin{pmatrix} 2^{\frac{3}{2}} \pi f k_f E_p^{-1} \\ E_s E_p^{-1} \\ 2k_{Cu}^{-1} I_p \\ F_l^{-1} d^{-1} I_p \end{pmatrix}$$

$$K = (6 \ 2 \ 2 \ 2 \ 3 \ 3 \ 2 \ 3 \ 1 \ 2 \ 3 \ 3 \ 1 \ 1 \ 2)$$

$$g = \log \begin{pmatrix} 4k_f \rho_c \\ 8k_f \rho_c \\ 4k_f \rho_c \\ 4k_{Cu} \rho_{Cu} \\ 2k_{Cu} \rho_{Cu} \\ \pi k_{Cu} \rho_{Cu} \\ 1 \\ \frac{1}{1,42\sqrt{2}k_a \Delta T^{\frac{5}{4}}} \\ 1 \\ \frac{1}{1,42\sqrt{2}k_a \Delta T^{\frac{5}{4}}} \\ 1 \\ 2 \\ 4k_f K_c \rho_c f^\alpha \\ 8k_f K_c \rho_c f^\alpha \\ 4k_f K_c \rho_c f^\alpha \\ 4k_{Cu} F_r \rho_w \\ 2k_{Cu} F_r \rho_w \\ \pi k_{Cu} F_r \rho_w \\ \frac{\eta_m}{(1-\eta_m)P_o} \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \frac{\eta_m}{(1-\eta_m)P_o} \\ \frac{\sqrt{2}}{\mu} \\ \frac{\sqrt{2}}{\mu} \\ \frac{\sqrt{2}}{\mu} \\ \frac{2\sqrt{2}}{\mu} \\ \frac{1}{\mu} \\ B_{sat}^{-1} \\ k_\phi^{-2} I_p^{-2} \\ E_p^{-2} k_\phi^{-2} I_p^{-2} \\ 8F_r \rho_w \cdot \cos \varphi \\ 4F_r \rho_w \cdot \cos \varphi \\ 2\pi F_r \rho_w \cdot \cos \varphi \\ \frac{8\pi}{3p^2} I_p \cdot \text{sen} \varphi \cdot f \mu_0 \\ \frac{4\pi}{3p^2} I_p \cdot \text{sen} \varphi \cdot f \mu_0 \\ \frac{2\pi^2}{3p^2} I_p \cdot \text{sen} \varphi \cdot f \mu_0 \\ 1 \\ m \\ E_p^{-1} \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & -1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Mínimas Perdas

$$x = \begin{pmatrix} N_p \\ N_s \\ N_{ls} \\ B \\ J \\ \bar{P}_c \\ \bar{P}_{Cu} \\ \bar{I}_m \\ \bar{V}_r \\ \bar{V}_x \\ \bar{h} \end{pmatrix} \quad b = -\log \begin{pmatrix} 2^{\frac{3}{2}} \pi f k_f E_p^{-1} c t \\ E_s E_p^{-1} \\ 2 k_{Cu}^{-1} I_p b_w^{-1} h_w^{-1} \\ F_l^{-1} d^{-1} I_p h_w^{-1} \end{pmatrix}$$

$$K = \begin{pmatrix} 2 & 2 & 2 & 3 & 3 & 3 & 1 & 2 & 3 & 3 & 1 & 1 & 2 \end{pmatrix}$$

$$g = \log \begin{pmatrix} 1 \\ 1 \\ \frac{c^{-\frac{1}{2}} t^{-\frac{1}{2}} b_w^{-\frac{1}{2}} h_w^{-\frac{1}{2}}}{1,42 \sqrt{2} k_a \Delta T^{\frac{5}{4}}} \\ \frac{c^{-\frac{1}{2}} t^{-\frac{1}{2}} b_w^{-\frac{1}{2}} h_w^{-\frac{1}{2}}}{1,42 \sqrt{2} k_a \Delta T^{\frac{5}{4}}} \\ h_w \\ 2c \\ 4k_f K_c \rho_c f^\alpha c t h_w \\ 8k_f K_c \rho_c f^\alpha c^2 t \\ 4k_f K_c \rho_c f^\alpha c t b_w \\ 4k_{Cu} F_r \rho_w c b_w h_w \\ 2k_{Cu} F_r \rho_w t b_w h_w \\ \pi k_{Cu} F_r \rho_w b_w^2 h_w \\ \frac{\sqrt{2}}{\mu} h_w \\ \frac{\sqrt{2}}{\mu} b_w \\ \vdots \end{pmatrix} \begin{pmatrix} \vdots \\ \frac{2\sqrt{2}}{\mu} c \\ B_{sat}^{-1} \\ k_\varphi^{-2} I_p^{-2} \\ E_p^{-2} k_\varphi^{-2} I_p^{-2} \\ 8F_r \rho_w \cdot \cos \varphi \cdot c \\ 4F_r \rho_w \cdot \cos \varphi \cdot t \\ 2\pi F_r \rho_w \cdot \cos \varphi \cdot b_w \\ \frac{8\pi}{3p^2} I_p \cdot \text{sen} \varphi \cdot f \mu_0 c b_w h_w^{-1} \\ \frac{4\pi}{3p^2} I_p \cdot \text{sen} \varphi \cdot f \mu_0 t b_w t b_w h_w^{-1} \\ \frac{2\pi^2}{3p^2} I_p \cdot \text{sen} \varphi \cdot f \mu_0 b_w^2 h_w^{-1} \\ 1 \\ m \\ E_p^{-1} \\ E_p^{-1} \end{pmatrix}$$

- Máxima Potência

$$x = \begin{pmatrix} I_p \\ N_p \\ N_s \\ N_{ls} \\ B \\ J \\ \bar{P}_c \\ \bar{P}_{Cu} \\ \bar{I}_m \\ \bar{V}_r \\ \bar{V}_x \\ \bar{h} \end{pmatrix} \quad b = -\log \begin{pmatrix} 2^{\frac{3}{2}} \pi f k_f E_p^{-1} c t \\ E_s E_p^{-1} \\ 2 k_{Cu}^{-1} b_w^{-1} h_w^{-1} \\ F_l^{-1} d^{-1} h_w^{-1} \end{pmatrix}$$

$$K = (1 \ 2 \ 2 \ 3 \ 3 \ 2 \ 3 \ 1 \ 2 \ 3 \ 3 \ 1 \ 1 \ 2)$$

$$g = \log \left(\begin{array}{c} 1 \\ \frac{c^{-\frac{1}{2}} t^{-\frac{1}{2}} b_w^{-\frac{1}{2}} h_w^{-\frac{1}{2}}}{1,42\sqrt{2}k_a\Delta T^{\frac{5}{4}}} \\ \frac{c^{-\frac{1}{2}} t^{-\frac{1}{2}} b_w^{-\frac{1}{2}} h_w^{-\frac{1}{2}}}{1,42\sqrt{2}k_a\Delta T^{\frac{5}{4}}} \\ h_w \\ 2c \\ 4k_f K_c \rho_c f^\alpha c t h_w \\ 8k_f K_c \rho_c f^\alpha c^2 t \\ 4k_f K_c \rho_c f^\alpha c t b_w \\ 4k_{Cu} F_r \rho_w c b_w h_w \\ 2k_{Cu} F_r \rho_w t b_w h_w \\ \pi k_{Cu} F_r \rho_w b_w^2 h_w \\ \frac{\eta_m}{(1-\eta_m)P_o} \\ \frac{\eta_m}{(1-\eta_m)P_o} \\ \frac{\sqrt{2}}{\mu} h_w \\ \frac{\sqrt{2}}{\mu} b_w \end{array} \right) \left(\begin{array}{c} \vdots \\ \frac{2\sqrt{2}}{\mu} c \\ B_{sat}^{-1} \\ k_\varphi^{-2} \\ E_p^{-2} k_\varphi^{-2} \\ 8F_r \rho_w \cdot \cos \varphi \cdot c \\ 4F_r \rho_w \cdot \cos \varphi \cdot t \\ 2\pi F_r \rho_w \cdot \cos \varphi \cdot b_w \\ \frac{8\pi}{3p^2} I_p \cdot \sin \varphi \cdot f \mu_0 c b_w h_w^{-1} \\ \frac{4\pi}{3p^2} I_p \cdot \sin \varphi \cdot f \mu_0 t b_w t b_w h_w^{-1} \\ \frac{2\pi^2}{3p^2} I_p \cdot \sin \varphi \cdot f \mu_0 b_w^2 h_w^{-1} \\ 1 \\ m \\ E_p^{-1} \\ E_p^{-1} \end{array} \right)$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & -1 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

