

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS E TECNOLOGIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
BACHARELADO EM ENGENHARIA ELÉTRICA**

CARLOS BONIFÁCIO EBERHARDT FRANCO

**DESENVOLVIMENTO DE UM APLICATIVO
PARA PROJETO E ANÁLISE
DE CONVERSORES DE POTÊNCIA**

JOINVILLE – SC

2021

**UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS E TECNOLOGIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
BACHARELADO EM ENGENHARIA ELÉTRICA**

CARLOS BONIFÁCIO EBERHARDT FRANCO

**DESENVOLVIMENTO DE UM APLICATIVO
PARA PROJETO E ANÁLISE
DE CONVERSORES DE POTÊNCIA**

Trabalho de Conclusão de Curso submetido ao curso de Bacharelado em Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina como requisito parcial para obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Dr. Alessandro Luiz Batschauer.

JOINVILLE – SC

2021

RESUMO

A otimização é um processo inerente ao projeto de um conversor de potência, seja uma otimização de perdas, volume ou custo, porque é ela que o torna competitivo. Neste trabalho foi desenvolvido um programa computacional robusto capaz otimizar as perdas do conversor c.c-c.c *Boost Half Bridge*, escolhendo todos os componentes, frequência de chaveamento e valores de indutância. Foram utilizadas técnicas de otimização numéricas e heurísticas, após ser realizar uma análise das fontes de perdas do conversor e implementada em código uma simulação das variáveis do conversor em regime permanente. Por fim, obteve-se um aplicativo com interface gráfica que permite fácil acesso à ferramenta.

Palavras-chave: conversores de potência, métodos numéricos, otimização, algoritmo genético.

ABSTRACT

Optimization is an inherent part in the design of a power converter, whether it is loss, volume or cost, it is what makes the converter competitive. In this paper a robust computer program was developed capable of optimizing the losses of the d.c-d.c Boost Half Bridge converter, choosing all components, switching frequency and inductance values. After conducting an analysis of the converter's loss sources and implementing a simulation of the converter variables in steady state in code, numerical and heuristic optimization techniques were used to optimize the converter. At last, a GUI application was built to allow a user-friendly access to the tool.

Keywords: power converters, numerical methods, optimization, genetic algorithm.

LISTA DE FIGURAS

Figura 1: Fluxograma do Funcionamento do GA.

Figura 2: Fluxograma do Funcionamento do otimizador.

Figura 3: Conversor c.c.-c.c. *Boost Half-Bridge*.

Figura 4: Etapas de Operação do Conversor *Boost Half-Bridge*.

Figura 5: Formas de Onda do Conversor *Boost Half-Bridge*.

Figura 6: Conversor *Boost Half-Bridge* com Carga Resistiva.

Figura 7: Comparação das Curvas de Ganho Ideal e Ganho Real Simplificado.

Figura 8: Diagrama de Classes da Implementação dos Componentes.

Figura 9: Diagrama de Classe da Implementação do Conversor *Boost Half-Bridge*.

Figura 10: Perdas x Frequência, para o Conversor de Knaesel (2018).

Figura 11: Fluxograma do Otimizador Contínuo.

Figura 12: Tela Inicial do Aplicativo.

Figura 13: Janela de Seleção de Componente.

Figura 14: Janelas de Configuração de Indutor e Transformador.

Figura 15: Perdas por Geração do Algoritmo Genético.

LISTA DE ABREVIATURAS

- ALAG - *Augmented Lagrangian* (Lagrangiano Aumentado).
- c.a. - Corrente Alternada.
- c.c. - Corrente Contínua.
- COBYLA - *Constrained Optimization by Linear Approximation* (Otimização Restrita por Aproximação Linear).
- GA - *Genetic Algorithm* (Algoritmo Genético).
- GPCO - *General Power Converter Optimizer* (Otimizador Geral de Conversores de Potência).
- GUI - *Graphics User Interface* (Interface Gráfica de Usuário)
- EMI - *Electromagnetic Interference* (Interferência Eletromagnética).
- HF - *High Frequency* (Alta Frequência).
- PWM - *Pulse Width Modulation* (Modulação por Largura de Pulso).
- SLSQP - *Sequential Least Squares Quadratic Programming* (Programação Quadrática Sequencial de Mínimos Quadrados).
- SUMT - *Sequential Unconstrained Minimization Technique* (Técnica de minimização sequencial irrestrita).
- VSI - *Voltage Source Inverter* (Inversor de Fonte de Tensão).
- ZVS - *Zero Voltage Switching* (Comutação em Tensão Nula).

SUMÁRIO

SUMÁRIO	7
1. INTRODUÇÃO	10
1.1. OBJETIVO GERAL	10
1.2. OBJETIVOS ESPECÍFICOS	10
1.3. CONTRIBUIÇÕES	11
2. OTIMIZAÇÃO DE CONVERSORES	12
2.1. MÉTODOS PENALIZADORES	13
2.1.1. SUMT – Sequential Unconstrained Minimization Technique	14
2.1.2. ALAG – Augmented Lagrangian Method	15
2.2. MÉTODOS APROXIMATIVOS	15
2.2.1. SLSQP – Sequential Least Squares Quadratic Programming	15
2.2.2. COBYLA – Constrained Optimization by Linear Approximation	16
2.3. MÉTODOS HEURÍSTICOS.....	16
2.3.1. Algoritmo genético	17
2.3.1.1. <i>Gerando nova população</i>	18
2.4. PONTOS VIÁVEIS	19
2.5. ESCOLHA DOS ALGORÍTMOS	20
3. CONVERSOR C.C-C.C BOOST HALF-BRIDGE	22
3.1. ANÁLISE SIMPLIFICADA	22
3.1.1. Análise Qualitativa	23
3.1.2. Análise Quantitativa	25
3.1.2.1. <i>Primeiro estágio</i>	25
3.1.2.2. <i>Segundo estágio</i>	27
3.2. INFLUÊNCIA DA INDUTÂNCIA AUXILIAR	28
4. METODOLOGIA DE PROJETO DO CONVERSOR	31
4.1. INDUTOR DE ENTRADA.....	33
4.1.1. Perdas no Indutor de Entrada	34
4.1.2. Restrições do Indutor de Entrada	35
4.1.2.1. <i>Densidade de fluxo máxima no núcleo</i>	35
4.1.2.2. <i>Produto das áreas do núcleo</i>	36

4.1.2.3.	Área de janela.....	36
4.1.2.4.	Efeito skin.....	36
4.1.2.5.	Densidade máxima de corrente.....	36
4.2.	INDUTOR AUXILIAR.....	37
4.2.1.	Perdas no Indutor Auxiliar.....	37
4.2.2.	Restrições do Indutor Auxiliar.....	37
4.2.2.1.	Densidade de fluxo máxima no núcleo.....	37
4.2.2.2.	Produto das áreas do núcleo.....	37
4.3.	TRANSFORMADOR.....	38
4.3.1.	Perdas no Transformador.....	38
4.3.2.	Restrições do transformador.....	38
4.3.2.1.	Densidade de fluxo máxima no núcleo.....	38
4.3.2.2.	Produto das áreas do núcleo.....	39
4.4.	CAPACITORES C1 E C2 – BARRAMENTO PRIMÁRIO.....	39
4.4.1.	Perdas nos Capacitores do Barramento Primário.....	39
4.4.2.	Restrições dos Capacitores do Barramento Primário.....	40
4.5.	CAPACITORES C3 E C4 – BARRAMENTO SECUNDÁRIO.....	40
4.5.1.	Perdas nos Capacitores do Barramento Secundário.....	40
4.5.2.	Restrições dos Capacitores do Barramento Secundário.....	41
4.6.	CHAVES.....	41
4.6.1.	Perdas nas Chaves.....	41
4.6.2.	Restrições das Chaves.....	42
4.7.	DIODOS D3 E D4.....	42
4.7.1.	Perdas nos Diodos.....	42
4.7.2.	Restrições dos Diodos.....	43
5.	IMPLEMENTAÇÃO DO OTIMIZADOR.....	44
5.1.	DEFINIÇÃO DAS VARIÁVEIS.....	44
5.2.	CÓDIGO BASE.....	45
5.3.	SIMULAÇÃO DO CONVERSOR.....	47
5.4.	CÁLCULO DAS PERDAS.....	49
5.5.	DEFINIÇÃO DAS RESTRIÇÕES.....	50
5.5.1.	Restrições de pré-seleção.....	51
5.5.2.	Restrições de conjunto.....	52
5.5.3.	Restrições de limite.....	52
5.5.3.1.	<i>Limites compatíveis.....</i>	<i>53</i>

5.5.4. Restrições contínuas	53
5.5.4.1. <i>Restrição de razão cíclica</i>	54
5.6. OTIMIZADOR CONTÍNUO.....	54
5.7. ALGORITMO GENÉTICO.....	56
5.8. INTERFACE GRÁFICA.....	57
6. RESULTADOS DO OTIMIZADOR	61
6.1. TESTE DO OTIMIZADOR CONTÍNUO.....	61
6.2. TESTES DO GA.....	62
7. CONCLUSÃO	64
REFERÊNCIAS	66
APÊNDICE A – RESTRIÇÕES DE LIMITE	69
LIMITES DE FREQUÊNCIA.....	69
LIMITES DA INDUTÂNCIA AUXILIAR.....	69
LIMITES DA INDUTÂNCIA DE ENTRADA.....	70
APÊNDICE B – EXEMPLO DE ARQUIVO DE COMPONENTE	71

1. INTRODUÇÃO

O aumento da demanda por veículos elétricos, fontes renováveis de energia e equipamentos ecologicamente mais corretos faz com que a eficiência dos conversores de potência que alimentam e suportam essas aplicações seja um fator cada vez mais importante, pois garante mais autonomia aos carros elétricos e melhor utilização da matriz energética.

Sendo assim, é importante que sejam projetados conversores eficientes e que o processo seja rápido e confiável. Todavia, como se trata de um problema de otimização multivariável, com muitas restrições, realizar manualmente o procedimento de cálculo iterativo é extremamente exaustivo.

Assim, seria de extrema valia um ambiente onde fosse possível analisar as variáveis físicas do conversor, mas também as restrições de projeto, e por fim otimizar a eficiência do conversor do conversor com relação aos componentes utilizados e frequência de operação.

1.1. OBJETIVO GERAL

A intenção desse trabalho é compreender e sumarizar o projeto do conversor *c.c.-c.c boost half-bridge*, tal como proposto em Jiang, Cao, et. al (2012) e apresentado em Cardoso (2017) e Knaesel (2018), e criar uma plataforma digital para analisar, projetar e otimizar as perdas do conversor. Também almeja-se apresentar visualmente as relações entre os diferentes aspectos do conversor, como por exemplo a frequência de chaveamento e perdas.

1.2. OBJETIVOS ESPECÍFICOS

- Realizar um estudo bibliográfico de métodos de otimização de conversores de potência.
- Analisar qualitativamente e quantitativamente o conversor proposto e sumarizar as variáveis e passos de projeto.
- Aplicar uma metodologia de otimização de perdas sobre o conversor e comparar com resultados já publicados.

- Implementar a metodologia em um aplicativo com interface gráfica.

1.3. CONTRIBUIÇÕES

Este trabalho se faz relevante pelo aumento da compreensão do conversor Boost Half-Bridge, que foi introduzido em (ZENG; YING; ZHANG, 2002), para aplicações fotovoltaicas e abordado em Cardoso (2017) e Knaesel (2018), ambos pelo Núcleo de Processamento de Energia (NPPE) da Universidade do Estado de Santa Catarina – Centro de Ciências e Tecnologias (UDESC - C.C.T).

O trabalho também contribui com o domínio das metodologias de otimização, que levam à redução das iterações dos testes de hardware em bancada, bem como a um maior entendimento do compromisso entre as diferentes variáveis do conversor (MEJBRI, et. al, 2014).

O aplicativo produzido através deste trabalho servirá de apoio à futuras pesquisas na área da eletrônica de potência, e almeja-se que este seja facilmente generalizável e entendível, a fim de que contribuições futuras possam ser feitas sobre ele.

2. OTIMIZAÇÃO DE CONVERSORES

Otimização é uma palavra-chave nas maiorias das aplicações da engenharia, visto que é essencial o estabelecimento de ciclos de projeto cada vez mais produtivos. (MEBRI, AMMOUS, et. al, 2014). Portanto, é interessante aplicar metodologias de projeto otimizadas computacionalmente, pois estas reduzem o tempo e custo de projeto.

Assim, em Balachandran & Lee (1981) é apresentada a noção de que projetar um conversor com alguma característica ótima e que satisfaça características restritivas, pode ser traduzido matematicamente como:

$$\begin{aligned} \text{Minimizar: } & f(x, \lambda) \\ & x \\ \text{Sujeito à: } & B(x, \lambda) \geq 0 \\ & C(x, \lambda) = 0 \end{aligned} \tag{2.1}$$

Onde x é o vetor de tamanho N que representa as características ajustáveis do conversor, como por exemplo: valores de componentes passivos, frequência de chaveamento, componentes discretos escolhidos. Já λ contém os requisitos de performance do conversor ótimo, os quais incluem: tensões de entrada e saída, ondulação máxima na saída e restrições físicas dos componentes, como correntes e tensões máximas.

A função $f(x)$ é o critério de otimização e pode ser qualquer parâmetro desejado, sendo no caso deste trabalho, as perdas do conversor. Já $B(x, \lambda)$ e $C(x, \lambda)$ são vetores com P e Q funções, respectivamente, que representam as restrições que o ponto ótimo não pode violar. É interessante notar que em uma na otimização de perdas a ser realizada, o custo e peso podem ser considerados restrições, tal que o ponto ótimo de rendimento nunca resulte em uma solução com um custo mais elevado do que o viável, ou com um peso que inviabilize o projeto. No caso de um conversor, essas restrições incluem: limites de ondulação de corrente e tensão e limites físicos dos componentes (tensão e corrente máxima).

Tentativas prévias de otimização de conversores de potência coincidiram com o aumento da computação compartilhada nos meados de 1970. Tais esforços, como em Yu, et. al (1979) consideravam que o vetor x consiste apenas de variáveis

contínuas e fizeram uso de métodos clássicos para otimização restrita, como a técnica de otimização sequencial irrestrita (SUMT) e o método do lagrangiano aumentado (ALAG). (MEJBRI, AMMOUS, et. Al; 2014).

Ainda que os primeiros estudos em otimização de conversores tenham deixado um impacto significativo no entendimento das dificuldades e relações de compromissos envolvidos, a maior parte falha em capturar a natureza discreta de algumas variáveis envolvidas (MONGE, et. al, 2004). Isso é relevante porque com variáveis discretas é possível que variáveis do vetor x sejam componentes comerciais de um banco de dados, assim após o processo de otimização, o projeto é realizável imediatamente.

Mais recentemente no âmbito da eletrônica de potência, algoritmos genéticos (GA) têm recebido atenção, pela capacidade inerente de resolver problemas discretos, como em Duro, Ramdsen e Muttik (1999), no qual um GA é utilizado para projetar um filtro ativo e Zhang, et. al (2000), no qual o transiente de um conversor *Buck* é otimizado (MONGE, et. al, 2004).

Neste trabalho as variáveis discretas, como componentes comerciais, são tratadas separadamente das variáveis contínuas. Isso é feito para que o usuário possa realizar diferentes otimizações:

- Otimizar a frequência para um dado conjunto de componentes.
- Otimizar os componentes e a frequência.

2.1. MÉTODOS PENALIZADORES

Tanto o método SUMT como ALAG, muito estudados a partir de 1970, se baseiam em transformar o problema de minimização restrita, em um ou mais problemas de minimização irrestrita (RARDIN, 1998). De forma geral, isso é feito penalizando a função objetivo $f(x, \lambda)$ quando uma das restrições é violada. A melhor parte desses algoritmos de transformação é possibilidade de fazer uso de algoritmos já bem estabelecidos para otimização irrestrita (BALACHANDRAN & LEE, 1981).

Um dos reverses dessa classe de método, é que apesar de não depender explicitamente das derivadas da função objetivo e funções restritivas, o método utilizado para resolver o subproblema geralmente depende.

2.1.1. SUMT – Sequential Unconstrained Minimization Technique

Essa técnica foi desenvolvida por Fiacco e McCormick (1968) e consiste em substituir o problema de otimização restrita por uma sequência de otimizações irrestritas da forma, conforme as equações (2.2) e (2.3)

$$\underset{x}{\text{Minimizar}} \quad P(x, \lambda, r_k) \quad (2.2)$$

Onde:

$$P(x, \lambda, r_k) = f(x, \lambda) - r_k \sum_{i=1}^P \ln(b_i(x, \lambda)) + \frac{1}{r_k} \sum_{j=1}^Q c_j(x, \lambda)^2 \quad (2.3)$$

A definição da função P é tal que ela aumenta à medida que as restrições de igualdade se distanciam de zero, e as restrições de desigualdade se aproximam de zero. A presença do logaritmo no segundo termo da função faz com que seja necessário que o ponto inicial do algoritmo não viole as restrições $b_i \in B$, visto que a violação de uma dessas restrições faz com que a função não seja calculável.

Já $r_k \in \Re$ é uma sequência monotonicamente decrescente que tende a zero. A ideia é que inicialmente r_k faz com que as restrições de inequação tenham o maior peso, fazendo com que o ponto solução permaneça na região viável, e conforme r_k tende a zero as restrições de igualdade passam a ter mais peso, pois são multiplicadas pelo inverso de r_k (RARDIN, 1998; CHINNECK, 2018; BALACHANDRAN & LEE, 1981).

O subproblema geralmente é resolvido utilizando o método de Newton, ou algum dos métodos quasi-newtonianos, que aproximam as derivadas e a matriz hessiana do problema.

2.1.2. ALAG – Augmented Lagrangian Method

O método foi proposto ao mesmo tempo por alguns autores, incluindo Hestenes (1969) e é uma tentativa de concertar alguns problemas existentes em outros métodos penalizadores, como o mau condicionamento da matriz hessiana e a necessidade de iniciar em um ponto que não viola as inequações (BALACHANDRAN & LEE, 1981; HESTENES, 1969).

2.2. MÉTODOS APROXIMATIVOS

Essa classe de método consiste em aproximar a função objetivo e as funções de restrição por outras mais simples, resolver o problema na forma aproximada, e iterar esse processo.

2.2.1. SLSQP – Sequential Least Squares Quadratic Programming

Neste método a função objetivo é aproximada por um polinômio de segundo grau multivariável e as restrições são aproximadas por equações lineares. Uma das maneiras de realizar essa aproximação é utilizando as primeiras e segundas derivadas de $f(x, \lambda)$ e as funções restritivas $B(x, \lambda)$ e $C(x, \lambda)$, tal como apresentado na Equação (2.4).

$$\begin{aligned} \text{Minimizar:} \quad & f(x_k, \lambda) + \nabla f(x_k, \lambda)^T d + \frac{1}{2} d^T \Delta_{x_k} f(x_k, \lambda) d \\ & d \\ \text{Sujeito à:} \quad & B(x_k, \lambda) + \nabla B(x_k, \lambda)^T d \geq 0 \\ & C(x_k, \lambda) + \nabla C(x_k, \lambda)^T d = 0 \end{aligned} \tag{2.4}$$

Onde o operador ∇ representa o gradiente da função e Δ_{x_k} é o laplaciano da função com relação ao vetor de variáveis x_k . Como $B(x, \lambda)$ e $C(x, \lambda)$ são vetores de funções, seus respectivos gradientes também são, e as expressões que definem as restrições do subproblema (2.4) são conjuntos com P e Q inequações, respectivamente.

Em cada iteração k o algoritmo resolve o problema aproximado utilizando o método dos mínimos quadrados. Por fim o vetor d encontrado é somado à x_k para obter x_{k+1} .

Quando as derivadas não estiverem disponíveis ou forem difíceis de calcular, existem várias maneiras de aproximar as funções, a mais comum sendo a amostragem de pontos próximo de x_k .

2.2.2. COBYLA – Constrained Optimization by Linear Approximation

Este método é semelhante ao método SLSQP, porém a função objetivo é aproximada por uma função linear, ao invés de quadrática, e em geral o método faz uso de aproximações baseadas na amostragem de pontos próximos à x_k , ao invés de utilizar derivadas.

Para realizar a amostragem, o algoritmo calcula a cada iteração o que é chamada de região de confiança, que é uma esfera multidimensional, dentro da qual o algoritmo está livre para amostrar os pontos sem violar as restrições (POWELL, 1998).

Apesar do algoritmo só suportar restrições de desigualdade, isso é compensando com robustez e estabilidade superiores a maioria dos métodos de otimização restrita.

2.3. MÉTODOS HEURÍSTICOS

Um método é heurístico quando este não garante que uma solução será encontrada, mas geralmente obtém bons resultados. Ainda assim esses métodos são interessantes pela velocidade com que geram soluções suficientemente boas (CHINNECK, 2018).

Muitos desses algoritmos são inspirados em fenômenos naturais, como o algoritmo genético (GA), otimização por exame de partículas, otimização por cozimento simulado, otimização por colônia de formigas. São utilizadas regras probabilísticas que incentivam a busca por pontos ótimos, mas que dão ampla oportunidade de explorar o espaço das variáveis, antes de convergir numa única

solução. Isso faz com que os métodos heurísticos sejam muito úteis quando a função possui diversos mínimos locais.

Outra vantagem dos algoritmos heurísticos é a facilidade com a qual lidam com variáveis discretizadas, enquanto nenhum dos métodos apresentados até agora tem essa flexibilidade.

2.3.1. Algoritmo genético

O nome desse algoritmo vem do modo com o qual este tenta mimetizar vagamente a seleção natural. Em um GA, cada variável é um gene e o vetor x é chamado de genoma. Assim, cada candidato à ponto ótimo é um organismo na população, e a otimização se baseia na reprodução dos organismos com a maior aptidão, que é calculada através da função $f(x, \lambda)$.

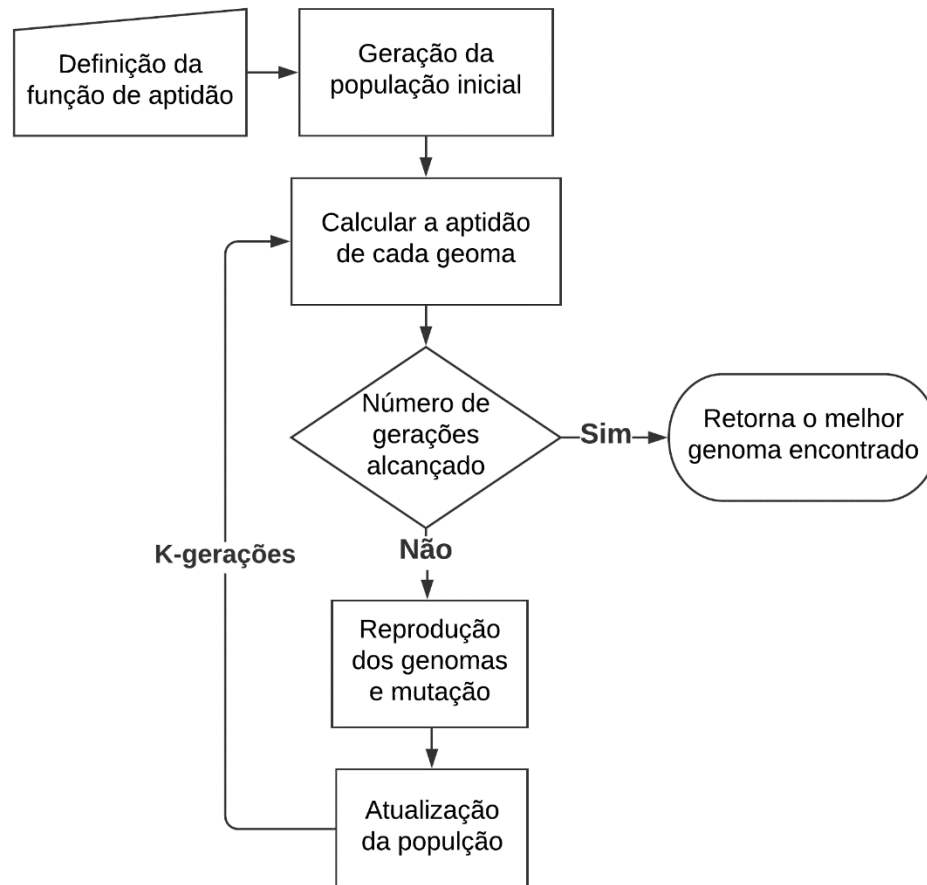
As vantagens desse algoritmo são que este consegue otimizar funções descontínuas e não diferenciáveis, e não requer nenhum conhecimento prévio do problema, além da função $f(x, \lambda)$ (MEBRI, AMMOUS, et. al, 2014). Todavia, ele requer algum nível de customização para que seja capaz de otimizar com pouca sensibilidade aos parâmetros do problema.

Como apresentado na Figura 1, o algoritmo inicia com a definição da função de aptidão. No caso de um problema de minimização, é comum utilizar a função de aptidão da Equação (2.5), onde C é uma constante de escala.

$$\alpha(x, \lambda) = \frac{C}{f(x, \lambda)} \quad (2.5)$$

Depois é feita criação da primeira população que pode ser aleatória ou pré-definida. Calcula-se então a aptidão de cada genoma, com a qual são selecionados os organismos que irão se reproduzir e passar adiante seus genes (CHINNECK, 2018). Após a reprodução, são aplicadas mutações aos genomas criados e então o processo é reiterado até que o número de iterações seja alcançado. Quando isso ocorre é retornado o melhor genoma encontrado.

Figura 1: Fluxograma do Funcionamento do GA.



Fonte: Koza (2012), adaptado.

2.3.1.1. Gerando nova população

Como explica Haupt e Haupt (2004), são realizados *crossovers* entre os genomas atuais para gerar os da próxima população. O *crossover* consiste em gerar um novo genoma selecionando aleatoriamente os genes dos dois genomas parentes.

Para selecionar da população os genomas que irão se reproduzir, existem diversos métodos, os principais sendo:

- **Seleção por torneio:**

Na seleção por torneio são selecionados da população dois ou mais genomas aleatórios. Então verifica-se dentro destes qual tem a melhor aptidão, e esse genoma é então selecionado para ser parente de um novo genoma.

- **Seleção por roleta:**

Na seleção por roleta, cada genoma k tem uma probabilidade de ser escolhido proporcional à sua aptidão dividida pela soma de aptidão da população inteira, como é apresentado na Equação (2.5), onde a função a_i representa a aptidão de cada genoma.

$$p_k = \frac{a_k}{\sum_i a_i} \quad (2.5)$$

2.4. PONTOS VIÁVEIS

Um ponto x_v é chamado de viável quando não viola nenhuma das restrições definidas pelo problema (2.1). Já o espaço desses pontos é chamado de região viável. Com isso, muitos algoritmos de otimização requerem ou se tornam mais eficientes se iniciados em um ponto inicial viável (ELWAKEIL e ARORA; 1995). Em especial, no caso de um conversor de potência, um ponto viável garante o funcionamento do conversor, e um ponto fora da região viável pode até mesmo ser não simulável.

O processo para se obter um ponto viável é um subproblema complexo, com diversas técnicas já desenvolvidas para resolvê-lo. Em Elwakeil e Arora (1995) são apresentados os métodos de penalização e os métodos primais, todavia, nenhum dos métodos consegue determinar previamente se existe uma região viável e os autores destacam o fato de que pode ser necessário reiterar o processo de pontos iniciais diferentes, caso o resultado não seja válido.

Neste trabalho, optou-se por utilizar o método de penalização apresentado em Freund (2004) porque o subproblema desse método, apresentado na Equação (2.6), é resolvido utilizando um método de otimização, e assim é possível utilizar a mesma biblioteca na implementação algoritmo em código.

$$\text{Minimizar: } v_T(x, \lambda) \quad (2.6)$$

x

Onde $v_T(x, \lambda)$ é a função de violação total, dada pelo somatório da violação de cada restrição. Para as restrições de igualdade e desigualdade, as funções de violação são dadas pelas equações (2.7) e (2.8), respectivamente.

$$v_j(x, \lambda) = |c_j(x, \lambda)|^k \quad (2.7)$$

$$v_i(x, \lambda) = \max(0, -b_i(x, \lambda))^k \quad (2.8)$$

As definições das equações de violação fazem com que a violação seja nula caso a restrição não seja violada e k é uma constante maior ou igual à um, mas geralmente utiliza-se o valor $k = 2$, sendo assim possível remover o operador de módulo da violação de igualdade.

2.5. ESCOLHA DOS ALGORÍTMOS

A escolha dos algoritmos a utilizados tem como critério principal o tipo das variáveis envolvidas, sendo estas discretas ou contínuas. Para as variáveis contínuas, como frequência de chaveamento e valores de indutância, foram escolhidos os métodos SLSQP e COBYLA, por serem mais robustos e não necessitarem da derivada da função objetivo. Para as variáveis discretas, o algoritmo genético com seleção por roleta é escolhido, por já apresentar resultados positivos na otimização de conversores de potência (DURO, RAMDSEN e MUTTIK, 1999; CARVALHO, 2017).

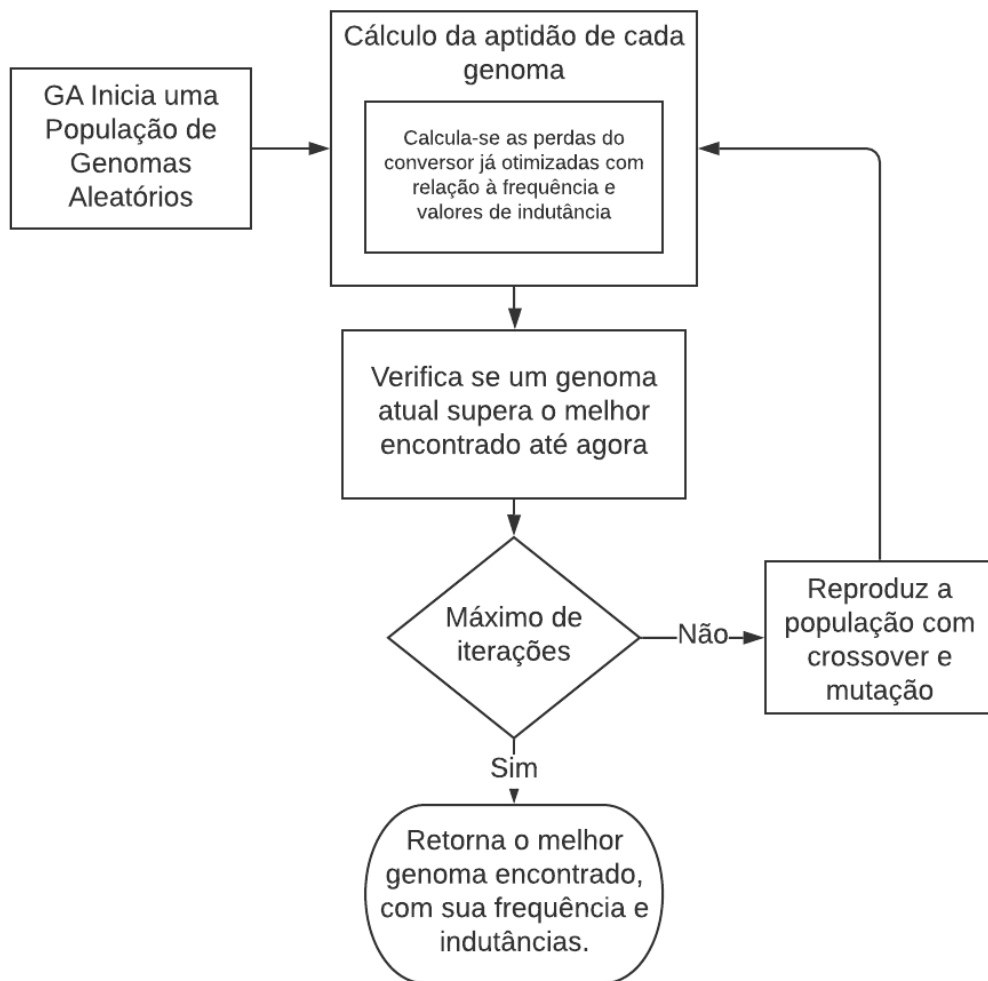
As variáveis discretas mais comuns em conversores de potências estão listadas abaixo:

- Componentes discretos, como chaves, diodos e capacitores.
- Núcleos a serem utilizados em transformadores e indutores.
- Fios a serem utilizados em transformadores e indutores.
- Número de espiras em transformadores e indutores.
- Número de fios em paralelo em transformadores e indutores.

A distinção das variáveis é utilizada porque o algoritmo genético tem certa dificuldade de resolver problemas de otimização com muitas restrições, e os métodos contínuos apresentados são eficientes e robustos quando o número de variáveis envolvidas é pequeno. Assim, a separação garante um número reduzido de restrições a serem lidadas pelo GA, e apenas $n + 1$ variáveis contínuas para o algoritmo contínuo, onde n é o número de indutores no conversor. Todas as demais variáveis são consideradas discretas, pois são restritas pelos componentes comerciais disponíveis.

Com isso tem-se uma otimização concatenada, na qual o algoritmo escolhido para as variáveis contínuas otimiza a frequência de chaveamento e valores das indutâncias, para um dado conjunto de componentes, e o algoritmo genético otimiza o conjunto de componentes. Para fins de compreensão, essa otimização concatenada é apresentada no fluxograma da Figura 2.

Figura 2: Fluxograma do Funcionamento do Otimizador.



Fonte: Próprio Autor.

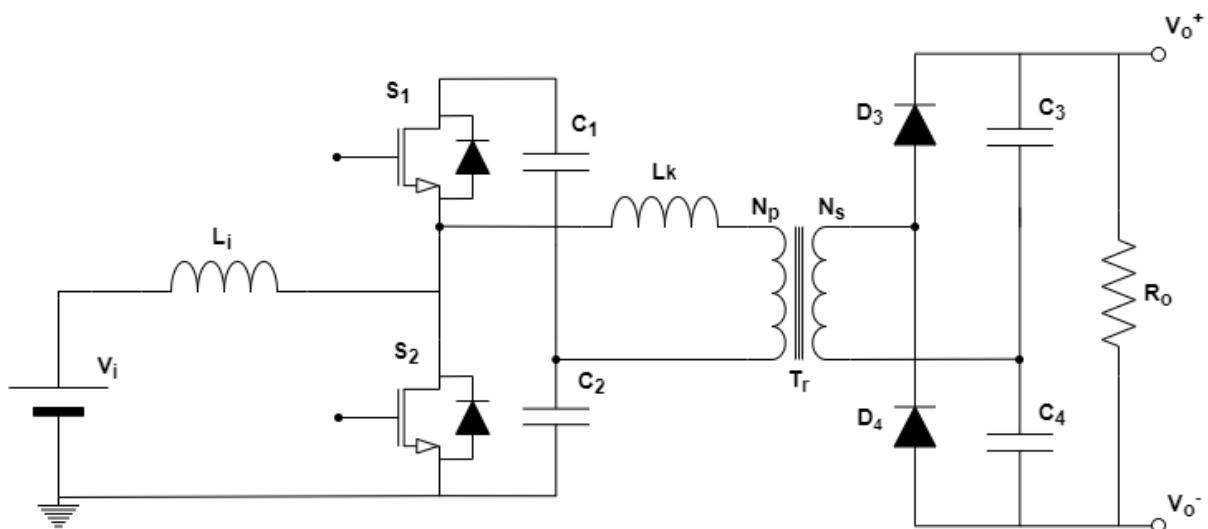
Sendo assim, para realizar a otimização do conversor *Boost Half-Bridge*, é necessário o equacionamento completo das perdas e restrições, em função das variáveis discretas contínuas. Também é necessário criar um banco de dados de componentes, para que o GA tenha um bom espaço de procura para a otimização.

3. CONVERSOR C.C.-C.C BOOST HALF-BRIDGE

A topologia *Boost Half-Bridge*, apresentada na Figura 3, opera com o chaveamento complementar das chaves, o que realiza uma conversão c.c.-c.a. da tensão entrada, depois a tensão alternada alimenta o transformador, que em geral é elevador. Por último, a tensão no secundário do transformador é retificada pelos diodos D_3 e D_4 , e filtrada pelos capacitores C_3 e C_4 .

A presença do indutor de entrada faz com que o primeiro estágio tenha característica elevadora e com entrada em corrente. Já a indutância auxiliar L_k serve para que haja comutação suave na entrada da condução de S_1 e S_2 .

Figura 3: Conversor c.c.-c.c. *Boost Half-Bridge*.



Fonte: Próprio autor.

3.1. ANÁLISE SIMPLIFICADA

Ainda que no equacionamento das perdas sejam consideradas algumas não idealidades, bem como a presença da indutância auxiliar, nessa seção, para que o funcionamento do conversor possa ser compreendido de maneira sucinta são feitas as seguintes considerações:

- Todos os componentes são ideais.
- As tensões sobre os capacitores são constantes.
- A corrente sobre o indutor é constante.

- Não há indutância auxiliar.

3.1.1. Análise Qualitativa

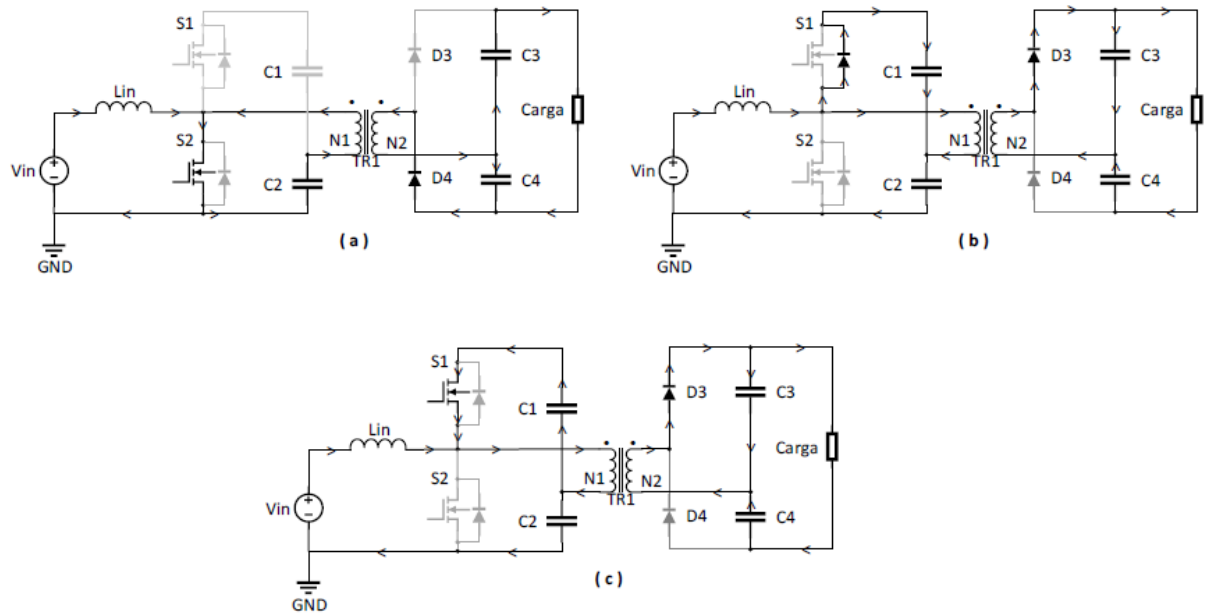
O conversor *Boost Half Bridge* trabalha com duas etapas de operação, que dependem diretamente do acionamento de S_1 e S_2 . Todavia, independentemente da etapa de operação, determina-se por inspeção do circuito que a tensão de saída sobre R_o é a soma das tensões sobre C_3 e C_4 .

A 1ª Etapa de operação (Figura 4a) ($t = t_o \rightarrow t = t_1 = DT$) inicia com o acionamento da chave S_2 , com isso, a tensão de entrada é aplicada sobre a indutância L_{in} , tal que a sua corrente cresce linearmente. Sobre o transformador é aplicada uma tensão de $-V_{C2}$, a qual é refletida para o secundário e comanda o diodo D_4 a conduzir. Desta forma o capacitor C_4 é carregado com o inverso da tensão do secundário.

Assim, a 2ª Etapa (Figura 4b) ($t = t_1 \rightarrow t = t_3 = T$) inicia quando S_2 é desativada e S_1 é comandada a conduzir, com isso é aplicada uma tensão $V_{in} - (V_{C1} + V_{C2})$ sobre L_{in} , isso resulta em um valor negativo e a corrente sobre a indutância decresce. Sobre o primário do transformador é aplicada uma tensão positiva igual à V_{C1} , que é refletida no secundário, comandando o diodo D_3 a conduzir e carregando C_3 com a tensão do secundário.

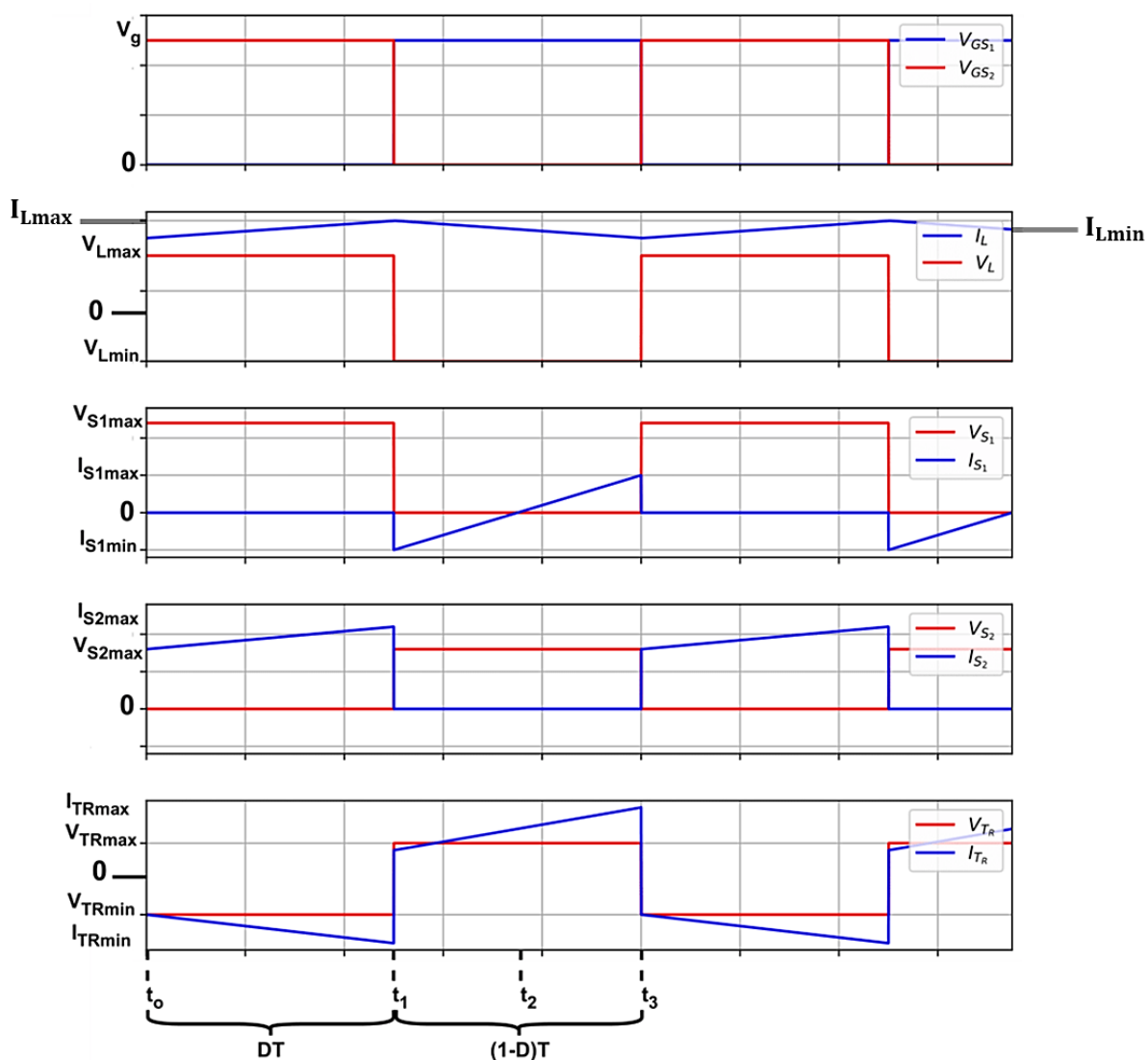
A segunda etapa pode ainda ser subdividida (Figura 4c). No primeiro momento, que inicia em t_1 a energia armazenada no indutor de entrada faz com que o diodo D_1 , que está em antiparalelo com S_1 , conduza, e assim parte dessa energia carrega o capacitor C_1 . Essa sub-etapa termina em t_2 , quando capacitor C_1 carrega, e a partir disso o capacitor passa a alimentar o primário do transformador juntamente com o indutor.

Figura 4: Etapas de Operação do Conversor *Boost Half Bridge*.



Fonte: CARDOSO (2017).

Figura 5: Formas de Onda do Conversor *Boost Half Bridge*.



Fonte: Próprio Autor.

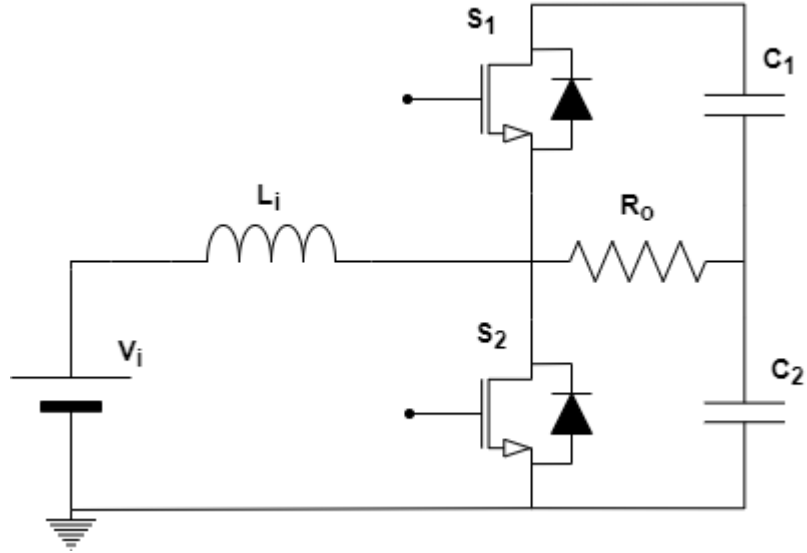
3.1.2. Análise Quantitativa

3.1.2.1. Primeiro estágio

Para facilitar a análise do primeiro estágio do conversor, inicialmente é considerado que no lugar do transformador há uma carga resistiva, como representado na Figura 6. Isso facilita a compreensão das correntes nos capacitores

C_1 e C_2 , bem como o cálculo da tensão sobre a resistência R_o , que representa o primário do transformador.

Figura 6: Conversor *Boost Half-Bridge* com Carga Resistiva.



Fonte: Próprio Autor.

Com base na Figura 6, e levando em consideração que em regime permanente a tensão média sobre o indutor e a corrente sobre o capacitor são nulas, é possível obter as equações (3.1) e (3.2), tal como expostas em Jiang, Cao, et. al (2012), referenciando a razão cíclica em S_2 .

$$V_{C1} = V_{in} \left(\frac{D}{1-D} \right) \quad (3.1)$$

$$V_{C2} = V_{in} \quad (3.2)$$

Na primeira etapa, o acionamento de S_2 faz com que a tensão aplicada sobre R_o seja $-V_{C2}$ e na segunda etapa, o acionamento de S_1 faz com C_1 esteja em paralelo com R_o . Assim a tensão média sobre a carga R_o é dada por (3.3).

$$\langle V_{Ro} \rangle = -V_{C2} \cdot DT + V_{C1} \cdot (1-D)T \quad (3.3)$$

Substituindo (3.1) e (3.2) em (3.3):

$$\langle V_{Ro} \rangle = 0 \quad (3.4)$$

Como a tensão média sobre a carga R_o é zero, é de fato possível conectar um transformador no lugar dela.

3.1.2.2. Segundo estágio

Baseando-se na seção 3.1 e nas equações (3.1) e (3.2), determina-se que a tensão aplicada no primário e secundário do transformador é dada por (3.5) e (3.6), para a primeira etapa de operação, e por (3.7) e (3.8) para a segunda etapa.

$$V_{pri}^{1a} = -V_{in} \quad (3.5)$$

$$V_{sec}^{1a} = -V_{in} \frac{N_s}{N_p} \quad (3.6)$$

$$V_{pri}^{2a} = V_{in} \left(\frac{D}{1-D} \right) \quad (3.7)$$

$$V_{sec}^{2a} = V_{in} \frac{N_s}{N_p} \left(\frac{D}{1-D} \right) \quad (3.8)$$

Portanto, conforme a análise qualitativa, as tensões sobre os capacitores C_3 e C_4 são:

$$V_{C3} = V_{in} \frac{N_s}{N_p} \left(\frac{D}{1-D} \right) \quad (3.9)$$

$$V_{C4} = V_{in} \frac{N_s}{N_p} \quad (3.10)$$

E finalmente a tensão de saída é dada por:

$$V_o = V_{C3} + V_{C4} \quad (3.11)$$

Substituindo (3.9) e (3.10) em (3.11), e dividindo a equação por V_{in} , é obtida a equação do ganho estático do conversor:

$$V_o = V_{in} \frac{N_s}{N_p} \left(\frac{1}{1-D} \right) \quad (3.12)$$

Apesar da razão cíclica teoricamente poder variar entre 0 e 1 na Equação (3.12), para diminuir os valores de corrente de pico, limita-se a razão cíclica entre 0,3 e 0,7.

3.2. INFLUÊNCIA DA INDUTÂNCIA AUXILIAR

Conforme apresentado em Knaesel (2018), para que ocorra comutação sob tensão nula (ZVS) dos interruptores S_1 e S_2 , a energia acumulada na indutância auxiliar L_k deve ser suficiente para garantir que os capacitores intrínsecos dos interruptores sejam carregados e descarregados. Para que isso ocorra, Knaesel (2018) determina que a Equação (3.13) seja válida, onde C_{S1} e C_{S2} são as capacitâncias intrínsecas de S_1 e S_2 respectivamente, e $i_{ZVS(S1)}$ é a corrente de S_1 antes de ocorrer a entrada de comutação de S_2 , dada pela Equação (3.14).

$$L_k > \frac{C_{S1} + C_{S2}}{(i_{ZVS(S1)})^2} \left(\frac{V_i}{1-D} \right)^2 \quad (3.13)$$

$$i_{ZVS(S1)} = \frac{V_o^2}{\eta V_i R_o} + \frac{V_i D}{2L_i f_s} + \frac{2nV_o}{DR_o} \quad (3.14)$$

Apesar da comutação suave reduzir as perdas de comutação, e assim permitir que as chaves trabalhem com frequências mais altas, a presença da indutância auxiliar faz com que a corrente do segundo estágio do conversor tenha a derivada limitada, o que leva o conversor a ter um comportamento mais complexo do que o inicialmente considerado, tendo agora oito etapas, que levam em conta a interação de L_k com as capacitâncias do conversor.

O equacionamento preciso dessas oito etapas e das equações de corrente é apresentado em Knaesel (2018), e leva a conclusão de que o ganho estático final do conversor depende de um sistema não linear de equações (3.15).

$$\begin{cases} \frac{2L_k f_s V_i n^3 (V_{C3} + V_{C4})^3}{R_o} + V_{C3}^2 (V_{C4} - nV_i) (V_{C4} (1 - D) + nV_i D) = 0 \\ \frac{2L_k f_s V_i n^3 (V_{C3} + V_{C4})^3}{R_o} + V_{C4}^2 (V_{C3} + nV_i) (V_{C3} (1 - D) - nV_i D) = 0 \\ V_{C3} + V_{C4} - V_o = 0 \end{cases} \quad (3.15)$$

A resolução desse sistema depende de um polinômio de oitavo grau, o que não possui solução fechada. Sendo assim, é necessário utilizar um método iterativo para calcular o valor da razão cíclica para se obter a tensão de saída desejada.

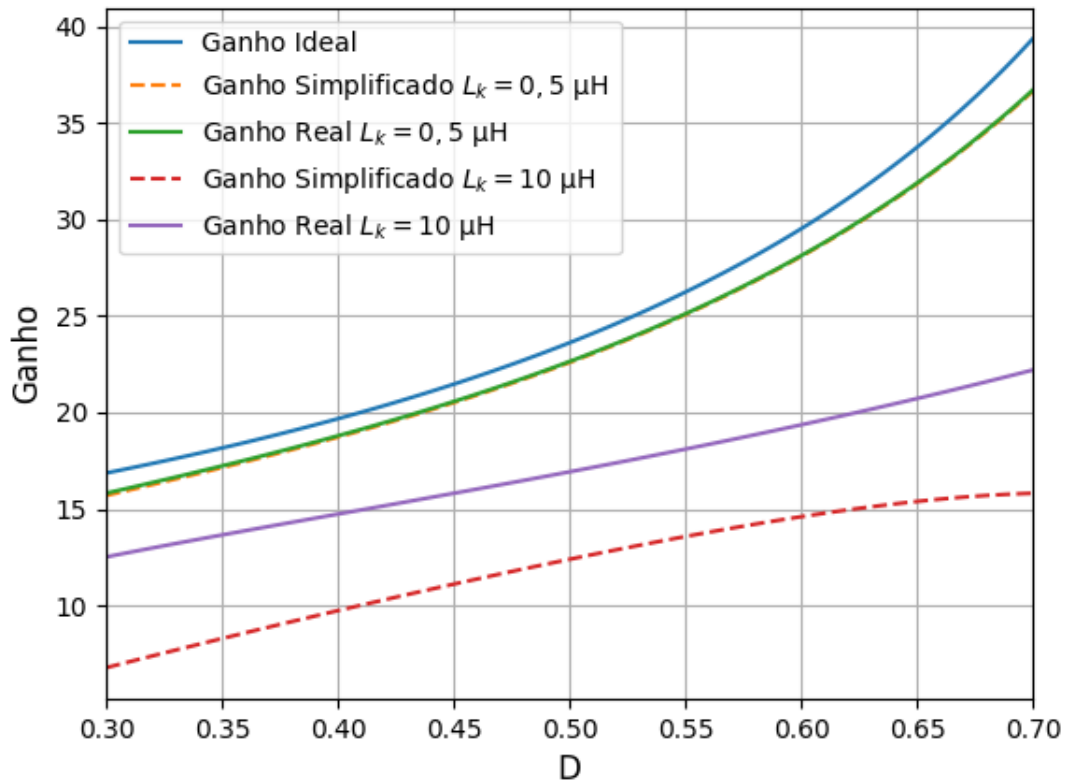
Ainda assim, para se obter uma relação mais clara entre a razão cíclica e a tensão de saída, mas que leve em conta a influência de L_k , é apresentada em Knaesel (2018), uma metodologia simplificada, onde os tempos das menores etapas de operação são considerados nulos. Com isso, é finalmente possível se ter uma equação do ganho, mesmo que simplificada (3.18).

$$\frac{V_o}{V_i} = \frac{nD^2(1-D)}{(2D^2 - 2D + 1) \frac{2L_k f_s n^2}{R_o} + D^2(1-D)^2} \quad (3.18)$$

Analisando a Equação (3.18) nota-se que um aumento do termo $2L_k f_s n^2 / R_o$ faz com que ocorra uma diminuição da tensão de saída. Em especial, a presença da indutância auxiliar e da frequência impõe um limite ao produto das duas variáveis, fora dos quais é impossível obter a tensão de saída requerida com um valor de razão cíclica factível.

A Figura 7 demonstra o efeito de L_k sobre o ganho com a comparação entre o resultado ideal, o ganho real simplificado e o ganho real para diferentes valores de L_k e $V_i = 17,4 \text{ V}$, $R_o = 1231 \text{ } \Omega$, $n = 11,8$ e $f_s = 50 \text{ kHz}$.

Figura 7: Comparação das Curvas de Ganho Ideal e Ganho Real Simplificado.



Fonte: Próprio autor.

Percebe-se que para pequenos valores de L_k , os três valores de ganho tendem à mesma curva, mas conforme o valor de L_k aumenta as curvas se distanciam bastante e passam a ter uma forma diferente. Ainda assim, o ganho ideal e o simplificado são importantes, pois são utilizados para determinar o ponto inicial da razão cíclica para resolução do sistema não linear em (3.15), acelerando a convergência do processo.

4. METODOLOGIA DE PROJETO DO CONVERSOR

Tomando como base o trabalho desenvolvido por Knaesel (2018) e Nilton (2017), determina-se que o projeto de um conversor c.c. *Boost Half-Bridge* inicia com a especificação dos parâmetros da Tabela 1 e os parâmetros de segurança da Tabela 2.

Em termos da formulação matemática da otimização, as variáveis da Tabela 1 compõe o vetor Z , de características restritivas, ou desejáveis, e com esta é possível dimensionar essencialmente todos os componentes do conversor. Todavia, para que seja realizado o projeto ótimo, através de um algoritmo numérico, é necessário que se tenha uma equação que represente a variável a ser otimizada, a perdas do conversor, em função dos componentes utilizados e frequência escolhida, para que seja feita a otimização restrita dessas perdas.

Já os parâmetros da Tabela 2 são multiplicadores aplicados a algumas equações de restrição, com o intuito de aumentar a robustez e segurança do conversor. Os parâmetros de tensão e corrente multiplicam as correntes e tensões máximas calculadas, e fazem com que seja necessário um componente que suporte um pouco mais que o máximo imposto pelo conversor em operação nominal.

Tabela 1: Parâmetros de projeto do conversor *Boost Half-Bridge*.

Parâmetro	Símbolo	Unidade	
Tensão de entrada	V_{in}	V	
Potência de saída	P_{out}	W	
Tensão de saída	V_{out}	V	
Frequência de comutação	f_s	Hz	
Máxima ondulação da corrente de entrada	$\Delta I_{inMáx}$	%	
Máxima ondulação na tensão de saída	ΔV_{out}	%	
Máxima densidade máxima dos condutores	$J_{máx}$		
Densidade máxima de fluxo magnético	Transformador	$B_{máxT}$	T
	Indutor de Entrada	$B_{máxLin}$	T
	Indutor Auxiliar	$B_{máxLr}$	T

Fonte: Knaesel (2018) & Nilton (2017) adaptado.

Tabela 2: Parâmetros de segurança adicionais do conversor *Boost Half-Bridge*.

Parâmetro		Símbolo
Tensão dos Capacitores		f_{VC}
Tensão dos Diodos		f_{VD}
Tensão dos Chaves		f_{VS}
Corrente dos Capacitores		f_{IC}
Corrente dos Diodos		f_{VD}
Corrente dos Chaves		f_{IS}
Fator de Utilização	Transformador	f_{uTr}
	Indutor de Entrada	f_{uLi}
	Indutor Auxiliar	f_{uLk}

Fonte: Knaesel (2018) & Nilton (2017) adaptado.

É interessante notar que em Knaesel (2018), Nilton (2017) e Jiang, Cao, et. al (2012), um dos parâmetros de entrada para iniciar o projeto do conversor é o rendimento, todavia neste trabalho ele não fará parte desses parâmetros, porque ele será calculado a partir da potência de saída e das perdas, através da Equação (4.1).

$$\eta = \frac{P_{out}}{P_{out} + P_{Total}} \quad (4.1)$$

Ainda, como será visto, a corrente de entrada e muitas das equações de perdas dependem do rendimento, sendo assim, é necessário supor um rendimento, calcular as perdas totais do conversor, recalcular o rendimento, e iterar esse procedimento até que o resultado das perdas convirja.

As equações utilizadas para obter as perdas vêm da mesma análise sem simplificações realizada por Knaesel (2018), na qual existem na realidade oito etapas de operação para o conversor e é considerada a indutância auxiliar. Todas as equações de corrente necessárias para obter as perdas se encontram nos Apêndices em Knaesel (2018), e pelo fato de serem extensas, não serão apresentadas no corpo deste trabalho.

4.1. INDUTOR DE ENTRADA

O indutor de entrada é constituído apenas por um núcleo, N_{condLi} condutores enrolados com N_{Li} espiras. Isso é importante, pois essas quatro variáveis, são otimizadas pelo otimizador das variáveis discretas, enquanto o valor da indutância em si é otimizado pelo otimizador contínuo.

O fator mais importante no projeto do indutor de entrada é que a indutância seja grande o suficiente para que a ondulação percentual de corrente seja menor ou igual ao valor máximo $\Delta I_{in\text{máx}}\%$. A amplitude da variação de corrente ΔI_{in} , é definida como:

$$\Delta I_{in} = I_{máx} - I_{min} \quad (4.2)$$

E então tem-se:

$$\frac{\Delta I_{in}}{I_{in(MED)}} \leq \Delta I_{in\text{máx}}\% \quad (4.3)$$

Para que essa inequação seja apresentada em termos da indutância de entrada, pode-se utilizar a primeira etapa de operação, na qual a tensão sobre o indutor L_{in} é a própria tensão de entrada, assim é possível utilizar a Equação (4.2) para obter a expressão (4.6), como segue:

$$v_{Lin} = L_{in} \frac{di_{Lin}}{dt} \quad (4.4)$$

$$\frac{V_{in}}{L_{in}} dt = di_{Lin} \quad (4.5)$$

$$I_{Lin} = I_{min} + \frac{V_{in}}{L_{in}} t \quad (4.6)$$

Como essa etapa termina em $t = DT_s = \frac{D}{f_s}$, é possível realizar as substituições de t , e calcular o valor da corrente máxima, e com isso, a variação de corrente:

$$I_{máx} = I_{min} + \frac{V_{in}D}{L_{in}f_s} \quad (4.7)$$

$$\Delta I_{in} = \frac{V_{in}D}{L_{in}f_s} \quad (4.8)$$

Sabendo que L_{in} está em série com a fonte de tensão da entrada, e considerando que o conversor possui um rendimento η , é possível utilizar as relações de potência para obter a equação de corrente média (4.9) (KNAESEL; 2018).

$$I_{Lin(MED)} = I_{in(MED)} = \frac{P_{out}}{\eta V_{in}} \quad (4.9)$$

Substituindo as equações (4.8) e (4.9) em (4.3) finalmente se obtém:

$$\frac{\eta V_{in}^2 D}{P_{out} L_{in} f_s} \leq \Delta I_{in_{m\acute{a}x}\%} \quad (4.10)$$

Outras variáveis importantes para o projeto do indutor de entrada são a corrente de pico e a corrente eficaz que passa pelo mesmo. Considerando que a corrente no indutor consiste em uma corrente média somada à uma onda triangular de amplitude $\frac{\Delta I_{in}}{2}$, é possível expressar a corrente eficaz sobre a indutância pela equação (4.11)

$$I_{in(RMS)} = \sqrt{I_{in(MED)}^2 + \frac{\Delta I_{in}^2}{12}} \quad (4.11)$$

4.1.1. Perdas no Indutor de Entrada

As perdas do indutor de entrada podem ser separadas em duas, as perdas no núcleo, e as perdas no cobre.

$$P_{Lin} = P_{Lin_Fio} + P_{Lin_Núcleo} \quad (4.12)$$

Para calcular as perdas no cobre será utilizada a metodologia apresentada em Knaesel (2018), na qual são considerados os efeitos c.a da condução nos enrolamentos.

$$P_{Fio} = R_{CC} \cdot \sum_{n=0}^M \frac{k(n f_s) I_n^2}{2} \quad (4.13)$$

Conforme (4.13), nesta metodologia as perdas se tornam uma soma das perdas provenientes de cada harmônica, onde ρ_{Fio} é a resistência por metro do fio, l_t o comprimento médio de cada espira. $k(n)$ é a razão entre resistência c.a da enésima harmônica e a resistência c.c (R_{CC}), que depende da frequência de chaveamento, da ordem da harmônica e do fator de empacotamento do enrolamento, e I_n é a amplitude da corrente dessa harmônica. O equacionamento detalhado é apresentado em Knaesel (2018).

As harmônicas são calculadas através da série de Fourier da forma de onda da corrente do indutor.

Já as perdas no núcleo são calculadas através da expressão genérica (4.14) (OLIVEIRA, 2001; adaptado).

$$P_{Núcleo} = k_c \cdot G(f_s) \cdot \Delta B^\beta \cdot V_e \quad (4.14)$$

Onde k_c e β são constantes que dependem do núcleo, V_e é o volume do núcleo, e $G(f_s)$ é uma função monotônica crescente dependente da frequência.

4.1.2. Restrições do Indutor de Entrada

A primeira restrição do indutor de entrada, e a mais intuitiva, é a restrição ao valor de indutância, já representada pela Equação (4.10). Já as demais restrições advêm dos componentes utilizados em sua construção.

4.1.2.1. Densidade de fluxo máxima no núcleo

O valor da densidade de fluxo magnético no núcleo do indutor dado pelo parâmetro B_{Li} , não pode exceder o valor máximo imposto na Tabela 1. Dessa forma, considerando a corrente na indutância de entrada, tem-se a expressão (4.15).

$$B_{Li} = \frac{L_i \left(I_{in(MED)} + \frac{\Delta I_{in}}{2} \right)}{A_{eLi} N_{Li}} \leq B_{máxLi} \quad (4.15)$$

4.1.2.2. Produto das áreas do núcleo

Considerando um núcleo EE, o produto $A_{eLi}A_{wLi}$ é limitado pela expressão (4.16) (Knaesel, 2018).

$$A_{eLi}A_{wLi} \geq \frac{L_i \left(I_{in(MED)} + \frac{\Delta I_{in}}{2} \right) I_{in(RMS)}}{B_{máxLi} J_{máx} k_{uLi}} \quad (4.16)$$

4.1.2.3. Área de janela

Essa é a restrição física que a área de janela do núcleo impõe no enrolamento do indutor, através da expressão (4.17). Onde S_{cLi} é a área da seção transversal condutor fio utilizado no indutor.

$$A_{wLi} \geq \frac{N_{Li} N_{condLi} S_{fioLi}}{f_{uLi}} \quad (4.17)$$

4.1.2.4. Efeito skin

De acordo com Erickson e Maksimovic (2001, KNAESESEL; 2018). Deve-se selecionar fios condutores com diâmetro menor que o dobro da profundidade de penetração, para evitar a influência do efeito *skin* nos enrolamentos. Isso resulta na expressão (4.18).

$$D_{CUfioLi} \leq 2 \sqrt{\frac{\rho_{fioLi}}{f_s \pi \mu_r \mu_{fioLi}}} \quad (4.18)$$

4.1.2.5. Densidade máxima de corrente.

Para proteger o fio condutor de sobreaquecimento, a densidade de corrente que passa por ele é limitada, utilizando o parâmetro $J_{máx}$ da Tabela 1. Essa restrição não considera

$$\frac{I_{in(RMS)}}{S_{CUfioLi}} \leq J_{máx} \quad (4.19)$$

4.2. INDUTOR AUXILIAR

4.2.1. Perdas no Indutor Auxiliar

O equacionamento das perdas do indutor auxiliar é muito parecido com o do indutor de entrada, modificando-se apenas as equações de corrente e os parâmetros de cálculo da resistência c.a. Assim, utilizam-se as equações de corrente completas, apresentadas em Knaesel (2018) para se obter as perdas no cobre e no núcleo, utilizando as equações (4.13) e (4.14) respectivamente.

4.2.2. Restrições do Indutor Auxiliar

Conforme já mencionado, a presença do indutor auxiliar serve para garantir a comutação suave dos semicondutores. E para que isso seja garantido, já foi apresentada a expressão (3.13), que é a primeira restrição do indutor auxiliar.

As restrições de área de janela, efeito *skin* e densidade máxima de corrente do indutor auxiliar utilizam as mesmas equações apresentadas nas restrições do indutor de entrada, porém com os respectivos parâmetros do indutor auxiliar e não são apresentadas nessa seção.

4.2.2.1. Densidade de fluxo máxima no núcleo

Assim como para a indutância de entrada, o valor da densidade de fluxo máxima no núcleo do indutor auxiliar é limitado pelo seu respectivo valor apresentado na Tabela 1. Todavia, a forma de onda de corrente do indutor de entrada faz com que seja necessário utilizar a Equação (4.20) para o campo máximo, onde $V_{Lk(RMS)}$ é a tensão eficaz e k_{fLk} é um fator de forma igual à 0,4 (KNAESEL, 2018).

$$B_{Lk} = \frac{V_{Lk(RMS)}}{k_{fLk} f_s A_{eLk} N_{Lk}} \leq B_{máxLk} \quad (4.20)$$

4.2.2.2. Produto das áreas do núcleo

Novamente a equação utilizada é similar à equação utilizada pelo indutor de entrada, mas utilizando a tensão eficaz e o fator de forma.

$$A_{eLk}A_{wLk} \geq \frac{V_{Lk(RMS)}I_{Lk(RMS)}}{k_{fLk}f_s B_{máxLk}J_{máx}k_{uLk}} \quad (4.21)$$

4.3. TRANSFORMADOR

4.3.1. Perdas no Transformador

Assim como para o indutor de entrada, as perdas do transformador advêm do cobre dos enrolamentos e do núcleo magnético.

Expressões para essas perdas podem ser calculadas através das equações (4.12) e (4.13), todavia, a expressão (4.12) deve ser aplicada tanto no primário como no secundário, tal como apresentado na Equação (4.22).

$$P_{FioTr} = R_{CC1^{o}Tr} \sum_{n=0}^M \frac{k_{1^{o}Tr}(nf_s)I_{n1^{o}Tr}^2}{2} + R_{CC2^{o}Tr} \sum_{n=0}^M \frac{k_{2^{o}Tr}(nf_s)I_{n2^{o}Tr}^2}{2} \quad (4.22)$$

A dependência de (4.36) com $B_{máxT}$ aparece porque o fluxo magnético excursionsa de 0 à $B_{máxT}$ em todo período.

4.3.2. Restrições do transformador

O transformador possui enrolamento primário e secundário, e sendo assim, devem ser aplicadas as restrições de efeito *skin* e de densidade máxima de corrente nos dois enrolamentos, portanto essas restrições não serão apresentadas explicitamente.

4.3.2.1. Densidade de fluxo máxima no núcleo

Utiliza-se a Equação 4.20 aplicada para o primário do transformador. A equação só é utilizada para o primário porque substituindo os valores do primário pelo

secundário tem o mesmo resultado visto que $\frac{N_{2^{\circ}Tr}}{N_{1^{\circ}Tr}} V_{1^{\circ}Tr} = V_{2^{\circ}Tr}$. Para o caso do transformador, o fator de forma é 0,4.

$$B_{Tr} = \frac{V_{1^{\circ}Tr}}{k_{fTr} f_s A_{eTr} N_{1^{\circ}Tr}} \leq B_{máxTr} \quad (4.23)$$

4.3.2.2. Produto das áreas do núcleo

Para essa restrição do transformador, é utilizada a Equação (4.24).

$$A_{eTr} A_{wTr} \geq \frac{V_{1^{\circ}Tr} I_{1^{\circ}Tr} + V_{2^{\circ}Tr} I_{2^{\circ}Tr}}{k_{uTr} k_{fTr} f_s B_{máxTr} J_{máx}} \quad (4.24)$$

4.4. CAPACITORES C_1 E C_2 – BARRAMENTO PRIMÁRIO

O dimensionamento dos capacitores C_1 e C_2 , é definido de maneira a limitar a ondulação da tensão de saída, assim para cada capacitor é definido uma variável ΔV_C que é o percentual da variação de tensão nesse capacitor. Em Knaesel (2018), são realizadas as deduções necessárias para obter as equações desses capacitores, que são válidas para um rendimento de 100%, todavia neste trabalho estas são modificadas para levar em conta o rendimento real. É interessante notar que assim como no caso da indutância de entrada, os valores das capacitâncias são inversamente proporcionais à frequência de chaveamento.

$$C_1 \geq \frac{P_{out}(1-D)^2}{4\eta f_s D \Delta V_{C1} V_{in}^2} \quad (4.25)$$

$$C_2 \geq \frac{P_{out}(1-D)}{\eta f_s \Delta V_{C2} V_{in}^2} \quad (4.26)$$

4.4.1. Perdas nos Capacitores do Barramento Primário

Todo capacitor tem uma resistência série intrínseca, chamada RSE, que faz com ele dissipe potência, baseado na corrente eficaz que passa por este. Assim:

$$P_{C1} = RSE_{C1} I_{C1RMS}^2 \quad (4.27)$$

$$P_{C2} = RSE_{C2} I_{C2RMS}^2 \quad (4.28)$$

4.4.2. Restrições dos Capacitores do Barramento Primário

As principais restrições dos capacitores são sobre os valores de capacitância, expressas nas equações (4.25) e (4.26). Porém para que possa ser utilizado, um capacitor deve suportar a tensão máxima e a corrente eficaz que o circuito lhe impõe. Assim para os capacitores, são definidas duas restrições adicionais, apresentadas nas equações (4.29) à (4.32), onde $I_{C1máx}$, $I_{C2máx}$, $V_{C1máx}$ e $V_{C2máx}$ são os valores máximos que os capacitores suportam, e f_{VC} e f_{IC} são fatores multiplicativos apresentados na Tabela 2 (KNAESEL, 2018).

$$f_{IC} I_{C1RMS} \leq I_{C1máx} \quad (4.29)$$

$$f_{IC} I_{C2RMS} \leq I_{C2máx} \quad (4.30)$$

$$f_{VC} V_{in} \left(\frac{D}{1-D} \right) \leq V_{C1máx} \quad (4.31)$$

$$f_{VC} V_{in} \leq V_{C2máx} \quad (4.32)$$

4.5. CAPACITORES C_3 E C_4 – BARRAMENTO SECUNDÁRIO

Assim como para o barramento primário, o dimensionamento dos capacitores C_3 e C_4 deve ser feito de tal maneira a garantir uma ondulação mínima da tensão de saída. Novamente, serão utilizadas as equações obtidas em Knaesel (2018), porém nesse caso não é necessário fazer nenhuma compensação do rendimento, porque as equações não dependem da corrente de entrada.

$$C_3 \geq \frac{P_{out} D}{V_o^2 f_s \Delta V_{C3}} \quad (4.33)$$

$$C_4 \geq \frac{P_{out} (1-D)}{V_o^2 f_s \Delta V_{C4}} \quad (4.34)$$

4.5.1. Perdas nos Capacitores do Barramento Secundário

Assim como para os capacitores do primário, as perdas são obtidas por:

$$P_{C1} = RSE_{C1}I_{C1RMS}^2 \quad (4.35)$$

$$P_{C2} = RSE_{C2}I_{C2RMS}^2 \quad (4.36)$$

4.5.2. Restrições dos Capacitores do Barramento Secundário

Assim como para o primário, são aplicadas as restrições de tensão e corrente aos capacitores do secundário.

$$f_{IC}I_{C3RMS} \leq I_{C3m\acute{a}x} \quad (4.37)$$

$$f_{IC}I_{C4RMS} \leq I_{C4m\acute{a}x} \quad (4.38)$$

$$f_{VC}V_{C3} \leq V_{C3m\acute{a}x} \quad (4.39)$$

$$f_{VC}V_{C4} \leq V_{C4m\acute{a}x} \quad (4.40)$$

As tensões V_{C3} e V_{C4} não são expressas explicitamente porque elas dependem da resolução do sistema não linear apresentado na seção 3.2.

4.6. CHAVES

4.6.1. Perdas nas Chaves

Cada chave possui duas fontes de perdas, condução e comutação. No caso de um Mosfet, as chaves em condução podem ser consideradas como resistências, de tal forma que são dissipadas potências dadas pelas equações (4.41) e (4.42). Nessa expressão, considera-se R_{DS} fixo com relação à temperatura.

$$P_{S1_Cond} = R_{DS1}I_{S1RMS}^2 \quad (4.41)$$

$$P_{S2_Cond} = R_{DS2}I_{S2RMS}^2 \quad (4.42)$$

Para o cálculo das perdas de comutação é feita a consideração que o conversor está operando com comutação suave, com isso há apenas perdas no desligamento das chaves (KNAESEL, 2018).

$$P_{S1_Comut} = \frac{V_{pkS1} I_{pkS1} f_s t_{offS1}}{2} \quad (4.43)$$

$$P_{S2_Comut} = \frac{V_{pkS2} I_{pkS2} f_s t_{offS2}}{2} \quad (4.44)$$

Onde:

V_{pkS1} - Tensão máxima na chave S_1 .

I_{pkS1} - Corrente máxima na chave S_1 .

T_{onS1} - Tempo de entrada em comutação para S_1 .

T_{offS1} - Tempo de bloqueio da comutação para S_1 .

V_{pkS2} - Tensão máxima na chave S_2 .

I_{pkS2} - Corrente máxima na chave S_2 .

T_{onS2} - Tempo de entrada em comutação para S_2 .

T_{offS2} - Tempo de bloqueio da comutação para S_2 .

4.6.2. Restrições das Chaves

As restrições das chaves são relativas aos valores máximos de tensão bloqueio e corrente eficaz que o componente pode suportar. Isso resulta nas expressões:

$$f_{IS} I_{S1máx} \leq I_{S2máx} \quad (4.45)$$

$$f_{IS} I_{S2máx} \leq I_{S1máx} \quad (4.46)$$

$$f_{VS} \frac{V_{in}}{1-D} \leq V_{S1máx} \quad (4.47)$$

$$f_{VS} \frac{V_{in}}{1-D} \leq V_{S2máx} \quad (4.48)$$

4.7. DIODOS D_3 E D_4

4.7.1. Perdas nos Diodos

Novamente, considerando que há comutação suave, os diodos sempre comutam com sob corrente nula (KNAESEL, 2018), assim não há perdas de comutação, apenas de condução, de tal maneira que as perdas se resumem à:

$$P_{D3} = V_{D3}I_{D3} + R_{TD3}I_{D3RMS}^2 \quad (4.49)$$

$$P_{D4} = V_{D4}I_{D4} + R_{TD4}I_{D4RMS}^2 \quad (4.50)$$

Onde R_{TD3} e R_{TD4} são as resistências de condução de D_3 e D_4 , respectivamente.

4.7.2. Restrições dos Diodos

Assim como as chaves, os diodos são limitados pela tensão máxima de bloqueio e corrente de pico.

$$f_{ID} \frac{2V_o}{R_o(1-D)} \leq I_{D3máx} \quad (4.51)$$

$$f_{ID} \frac{2V_o}{R_oD} \leq I_{D4máx} \quad (4.52)$$

$$f_{VD}V_o \leq V_{D3máx} \quad (4.53)$$

$$f_{VD}V_o \leq V_{D4máx} \quad (4.54)$$

5. IMPLEMENTAÇÃO DO OTIMIZADOR

5.1. DEFINIÇÃO DAS VARIÁVEIS

Conforme apresentado na seção 3.3, as variáveis do conversor podem ser contínuas ou discretas, e a maneira pela qual tais variáveis serão otimizadas depende do grupo ao qual pertencem. Assim, é importante saber exatamente quais as variáveis de otimização do conversor, e separá-las.

As variáveis discretas incluem, além dos semicondutores e capacitores, os componentes e valores necessários para a construção do transformador e dos indutores, como: núcleo, condutor, número de espiras e condutores em paralelo. O único aspecto do projeto dos elementos magnéticos que é considerado como contínuo é o valor das indutâncias. Considerando isso, a Tabela 3 lista todas as variáveis discretas, totalizando 23.

Tabela 3: Variáveis otimizáveis discretas do conversor *Boost Half-Bridge*.

Variáveis Discretas		
Núcleo do Transformador	Núcleo de L_i	Núcleo de L_k
Condutor do Primário	Condutor de L_i	Condutor de L_i
Condutor do Secundário		
N_p	N_{Li}	N_{Lk}
N_s		
N_{cond_p}	$N_{cond_{Li}}$	$N_{cond_{Lk}}$
N_{cond_p}		
S_1	C_1	
S_2	C_2	
D_3	C_3	
D_4	C_4	

Fonte: O Próprio Autor.

As variáveis contínuas são menos numerosas, e são a frequência de chaveamento, o valor da indutância de entrada e o valor da indutância auxiliar.

5.2. CÓDIGO BASE

Para realizar a implementação do algoritmo apresentado na Seção 3 e criar o aplicativo final, foi escolhida a linguagem de programação Python, por ser uma linguagem orientada a objeto versátil e com uma quantidade expressiva de bibliotecas, tanto para auxiliar a otimização, como a interação com o usuário.

Entretanto, antes de transformar em código as equações de perda do conversor, foi organizado um diagrama de classes (Figura 8), em que cada componente, e suas variáveis são representados. Essa representação, orientada a objeto, facilita na compreensão e escrita do código.

No diagrama da Figura 8 existe uma classe base chamada **Component** que possui apenas a variável **name** que armazena o nome do componente. Dessa forma, como indicado pela flecha em negrito, os componentes do conversor herdam dessa classe a variável name.

Existe então uma classe para cada componente comercial que é utilizado no conversor: capacitores, diodos, chaves, fios, núcleos. Cada classe possui as variáveis necessárias para simular o conversor, calcular as perdas e restrições.

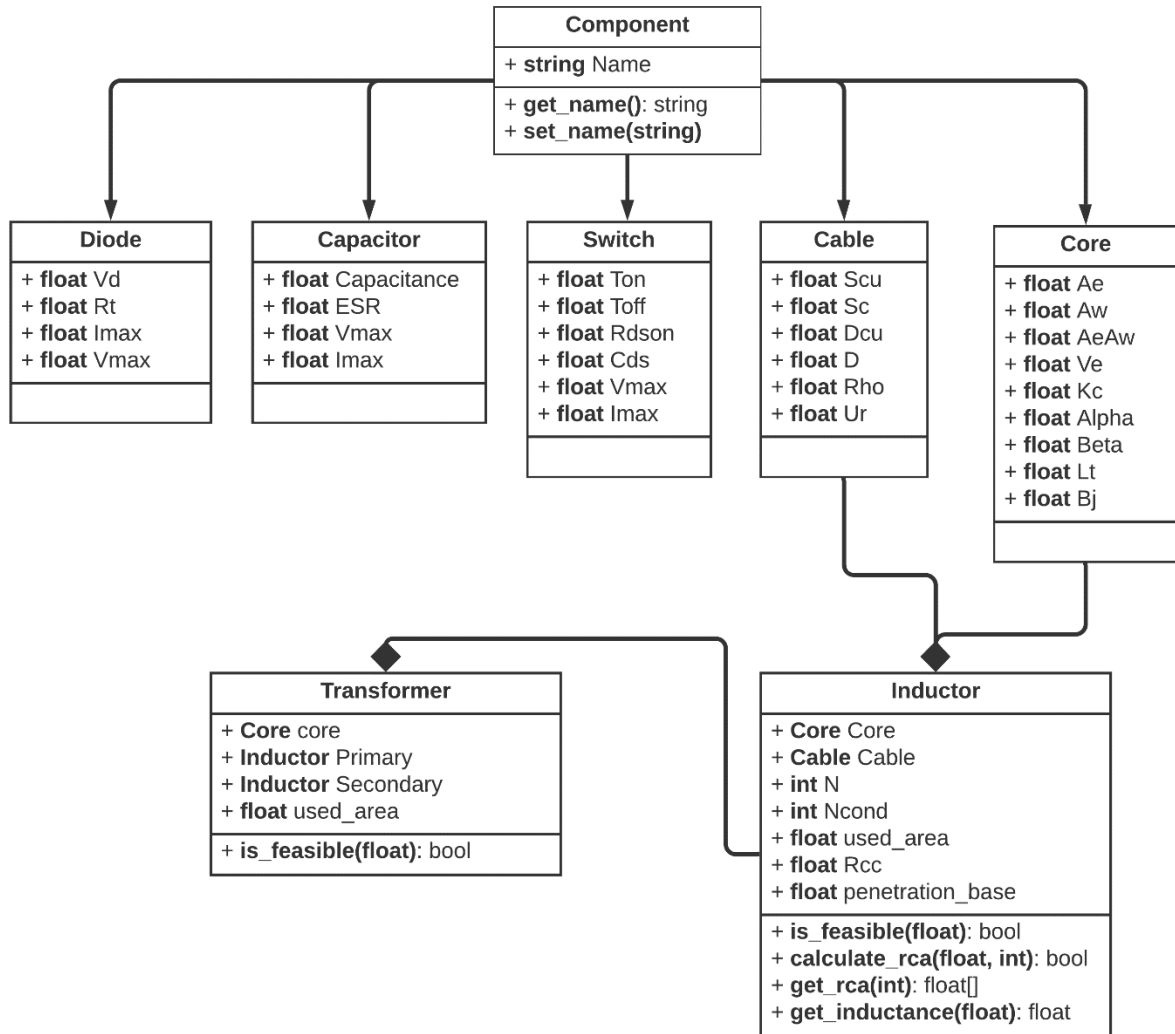
A classe **Inductor** é uma composição de um núcleo, um fio, número de espiras e número de condutores. Além disso essa classe também tem as variáveis **R_{CC}**, **used_area** e **penetration_base**, que são a resistência c.c do indutor, a área utilizada pelos condutores e a base da profundidade de penetração, dada pela Equação (5.1).

$$Penetration_base = \sqrt{\frac{p_{fio}}{\pi\mu_{rfio}}} \quad (5.1)$$

As funções **calculate_rca()** e **get_rca()** da classe do indutor são funções auxiliares que encapsulam o cálculo da resistência c.a do indutor. Já a função **is_feasible()** determina se a ocupação da área de janela está abaixo do permitido. E

por fim a função **get_inductance()** recebe um valor de *gap* e diz qual a indutância resultante do componente.

Figura 8: Diagrama de Classes da Implementação dos Componentes.

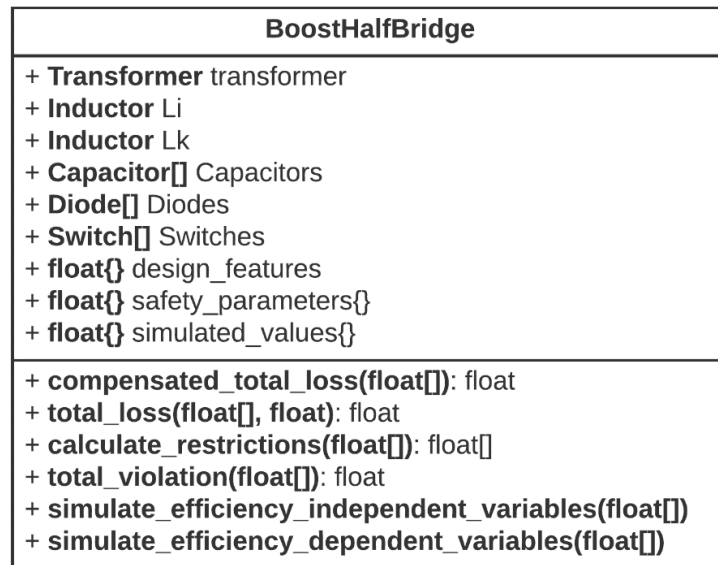


Fonte: Próprio autor.

A última classe do diagrama da Figura 8 é o **Transformer** e nota-se que o primário e secundário do transformador são indutores à parte, isso faz com que seja fácil de se referir ao cabo, número de espiras e condutores de cada lado do transformador. Além dos dois indutores a classe também possui um núcleo, e uma variável que representa a área utilizada pelos condutores. Assim como para os indutores, existe a função **is_feasible()** que verifica a restrição da área de janela.

Com os componentes finalizados, foi então criada uma classe chamada **BoostHalfBridge**, apresentada no diagrama da Figura 9. Uma instância dessa classe tem todos os componentes do conversor, seus parâmetros de projeto, parâmetros de segurança e funções necessárias para simular as variáveis do conversor, calcular suas perdas, restrições e a função de violação total.

Figura 9: Diagrama de Classe da Implementação do Conversor *Boost Half Bridge*.



Fonte: Próprio Autor.

Além disso, também foi feita uma biblioteca chamada de **FileHandler**, que lê um arquivo de texto tal como exemplificado no Apêndice A, e o transforma em uma instância de componente dentro do programa. Com isso torna-se fácil ler e modificar um banco de dados de componentes.

5.3. SIMULAÇÃO DO CONVERSOR

Para simular as correntes e tensões do conversor, foram utilizadas as equações obtidas em Knaesel (2018), com a diferença que todos as integrais e séries de Fourier necessárias foram resolvidas analiticamente, para tornar o código mais eficiente do que utilizando integração numérica.

A simulação do conversor foi separada em duas partes, a que depende da eficiência e a que independe. Com isso a parte independente não precisa ser recalculada toda vez que as perdas forem recalculadas. As duas funções que simulam

o circuito são **simulate_efficiency_dependent_variables()** e **simulate_efficiency_independent_variables()**, apresentadas no diagrama da Figura 9.

Com o código do conversor pronto, foi realizado um teste de confiabilidade do simulador. O teste consistiu em construir em código o mesmo conversor *Boost Half Bridge* utilizado em Knaesel (2018) e simular o mesmo ponto de operação. O resultado comparativo das principais grandezas é apresentado na Tabela 4.

Tabela 4 – Comparação das grandezas do conversor neste trabalho X Knaesel (2018).

Componente	Grandeza	Knaesel	Calculado	Erro
-----	D	0,550	0,530	3%
Transformador	$I_{rms_primário}$ [A]	8,474	9,397	-11%
Indutância de Entrada	I_{med} [A]	7,462	7,996	-7%
	ΔI [A]	0,747	0,722	3%
S_1	I_{rms} [A]	3,789	4,006	-6%
S_2	I_{rms} [A]	10,715	9,775	9%
D_3	I_{med} [A]	0,325	0,325	0%
	I_{rms} [A]	0,557	0,547	2%
D_4	I_{med} [A]	0,325	0,325	0%
	I_{rms} [A]	0,508	0,516	-2%
C_1	I_{rms} [A]	3,789	4,006	-6%
C_2	I_{rms} [A]	7,759	8,239	-6%
C_3	I_{rms} [A]	0,452	0,440	3%
	V_{max} [V]	218,297	211,508	3%
C_4	I_{rms} [A]	0,380	0,401	-5%
	V_{max} [V]	181,496	188,492	-4%

Fonte: O autor.

Percebe-se que a maior parte das grandezas tem um erro percentual bem baixo. A maior parte do erro advém do fato de que em Knaesel (2018), o valor da razão cíclica é definido como 0.55, e não calculado pelo sistema, e que a eficiência é considerada fixa. Em especial a razão cíclica afeta quase todas as variáveis, visto que esta define os intervalos das integrais de corrente eficaz e, como apresentado na

seção 4.1, a razão cíclica altera o valor da tensão sobre o transformador em cada etapa de operação.

Neste trabalho a razão cíclica deve sempre ser calculada por conta de sua dependência com a frequência de chaveamento e indutância auxiliar, que fazem parte do otimizador e, portanto, não são conhecidas previamente.

5.4. CÁLCULO DAS PERDAS

Para validar os cálculos de perdas implementados no código, foram novamente feitas comparações com as perdas calculadas em Knaesel (2018). Esses resultados comparativos estão apresentados na Tabela 5.

Tabela 5 – Comparação das perdas do conversor neste trabalho X Knaesel (2018).

Componente	Fonte das Perdas	Knaesel	Calculado	Erro
Transformador	Núcleo	2,911 W	3,012 W	-3 %
	Fio Primário	0,718 W	1,041 W	-45 %
	Fio Secundário	0,82 W	0,713 W	13 %
Indutância de Entrada	Núcleo	0,036 W	0,033 W	9 %
	Cobre	2,103 W	1,908 W	9 %
Indutância Auxiliar	Núcleo	0,438 W	0,410 W	6 %
	Cobre	0,235 W	0,168 W	28 %
S_1		0,308 W	0,291 W	5 %
S_2		0,957 W	0,866 W	10 %
D_3		0,487 W	0,487 W	0 %
D_4		0,487 W	0,487 W	0 %
C_1		0,148 W	0,166 W	-12 %
C_2		0,181 W	0,204 W	-13 %
C_3		0,112 W	0,106 W	5 %
C_4		0,038 W	0,040 W	-6 %

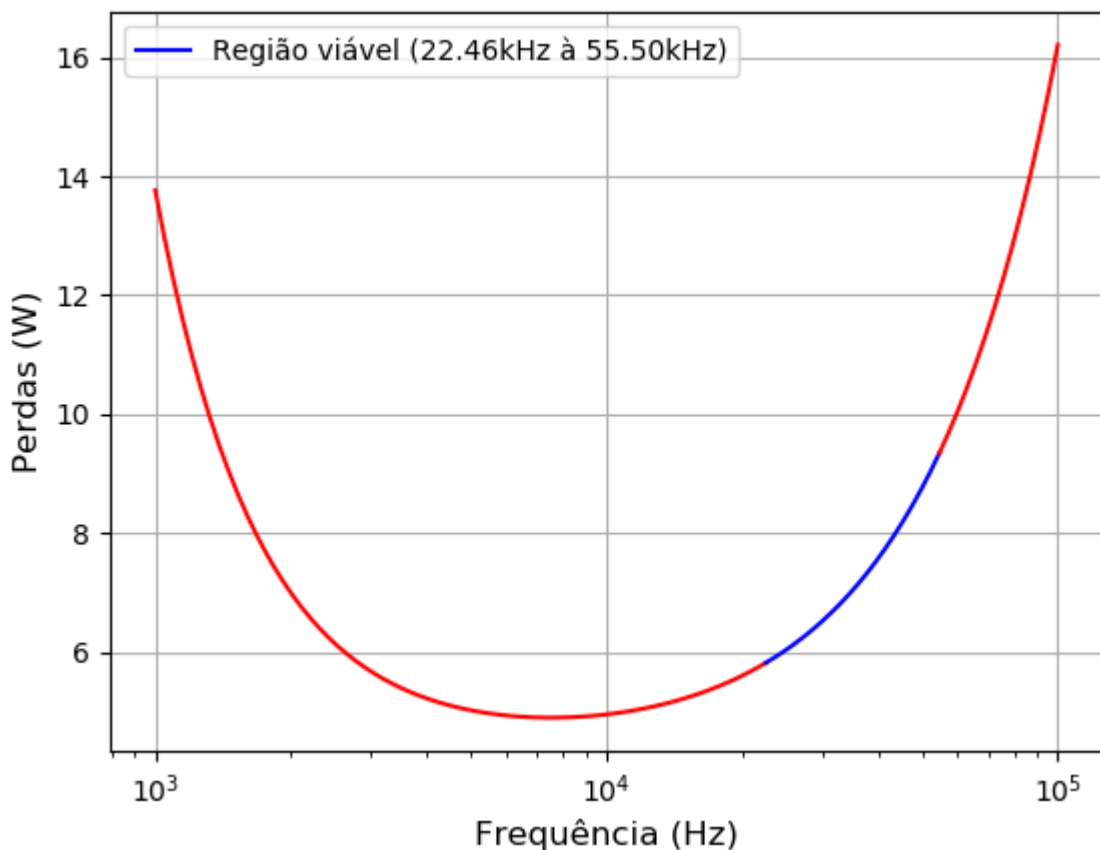
Fonte: O autor.

Novamente, o erro foi no geral bem baixo, exceto em dois casos, no fio do primário do transformador e no fio da indutância auxiliar. Se tratando do cálculo de

perdas mais complexo implementado, é difícil localizar exatamente a fonte da discrepância, além dos fatores já mencionados da eficiência e razão cíclica.

Com o conversor modelado, aproveitou-se para produzir um gráfico das perdas desse conversor com relação à frequência de comutação, mantendo os valores de L_i e L_k fixos. Neste gráfico, apresentado na Figura 10, foi destacada a região em que as restrições do conversor são satisfeitas, e nota-se que a frequência de 50 kHz utilizada em Knaesel (2018) está dentro do intervalo. Pela figura também se percebe que a curva das perdas pela frequência tem uma forma convexa, o que é algo muito positivo para o processo de otimização.

Figura 10: Perdas x Frequência, para o Conversor de Knaesel (2018).



Fonte: O próprio autor.

5.5. DEFINIÇÃO DAS RESTRIÇÕES

As restrições são a parte mais importante do problema de otimização, visto que nada adianta um conversor teoricamente ótimo, sendo que ele não funcionaria na

prática. De fato, a importância dos cálculos das restrições é tão grande, que foi isso que motivou a separação de variáveis em contínuas e discretas, da mesma forma que agora, são separadas as restrições, em quatro categorias.

A categoria da restrição diferencia a maneira com a qual é utilizada no otimizador, e apenas as restrições contínuas fazem parte dos vetores de restrições $B(x, \lambda)$ e $C(x, \lambda)$.

5.5.1. Restrições de pré-seleção

Essas restrições dependem apenas de um componente e dos parâmetros de projeto. Assim essas restrições são usadas para fazer uma pré-seleção dos componentes que podem ser utilizados pelo GA, e depois não precisam mais ser calculadas. Isso é importante para reduzir o espaço de procura do algoritmo genético.

Uma inspeção das restrições apresentadas na Seção 4, leva conclusão de que as únicas restrições que podem ser utilizadas para realizar a pré-seleção são as restrições de tensão máxima dos capacitores, diodos e chaves, e as restrições de corrente dos diodos.

Porém, nem todas as restrições mencionadas podem ser utilizadas da maneira na qual são apresentadas originalmente, porque algumas dependem da razão cíclica, que não é conhecida durante a otimização. Para as restrições em que isso ocorre, foram substituídos os valores limitantes da razão cíclica, 0,3 e 0,7 e verificou-se o caso menos restritivo, e esse caso foi utilizado com a restrição de pré-seleção.

Por exemplo, substituindo $D = 0,3$ e $D = 0,7$ na Equação (4.48), que é a restrição de tensão máxima de S_1 , obtém-se (5.2) e (5.3), respectivamente.

$$1,43f_{VS}V_{in} \leq V_{S2m\acute{a}x} \quad (5.2)$$

$$3,33f_{VS}V_{in} \leq V_{S2m\acute{a}x} \quad (5.3)$$

Fica evidente que a expressão (5.2) é a menos restritiva, portanto é utilizada para fazer a pré-seleção das chaves que o GA pode escolher para atribuir a S_1 na criação de seus genomas, visto que qualquer chave que viola a restrição (5.2), com certeza viola (4.48) para qualquer valor de razão cíclica.

Ainda assim, é necessário que as restrições de pré-seleção que envolvem a razão cíclica, estejam também presentes no vetor de restrições do otimizador numérico.

5.5.2. Restrições de conjunto

São restrições que estabelecem uma relação entre duas ou mais variáveis discretas do conversor sem serem influenciadas pelas variáveis contínuas. Essas restrições são utilizadas para criar a primeira geração de genomas do GA e ajustá-los antes da otimização das variáveis contínuas. Isso impede que o algoritmo calcule as perdas de conversores em que o conjunto das variáveis discretas não é compatível, ainda que o conversor seja simulável.

Assim, na primeira geração o GA escolhe aleatoriamente as variáveis discretas, mas estas só são aprovadas se não violarem as restrições, caso contrário, é repetida a escolha até que a restrição seja satisfeita.

Também pode ser o caso que o *crossover* e a mutação, tornam incompatíveis as variáveis discretas. Dessa forma antes de calcular a aptidão, as variáveis do genoma são modificadas iterativamente até que nenhuma restrição de conjunto seja violada.

No caso do conversor *Boost Half Bridge* as únicas restrições deste tipo são as restrições de área de janela dos elementos magnéticos, já apresentadas, e as restrições de limites compatíveis.

A restrição de limite compatível determina que o limite superior estimado de uma variável deve ser maior que limite inferior. Caso isso não seja verdadeiro é impossível otimizar as variáveis contínuas, e um novo conjunto de variáveis discretas deve ser gerado.

5.5.3. Restrições de limite

Essas restrições dependem apenas de uma variável contínua, dos parâmetros de projeto e de um número qualquer de variáveis discretas. A função dessas

restrições, é limitar os valores mínimos e máximos das variáveis contínuas, e reduzir o espaço de busca.

Infelizmente, não existe nenhuma restrição apresentada em sua forma original, que dependa apenas de uma variável contínua, porque geralmente a razão cíclica, ou o rendimento fazem parte da equação. Dessa forma, nas equações em que apenas uma variável contínua aparece explicitamente, juntamente da razão cíclica ou rendimento, substituem-se os valores limitantes para a razão cíclica (0,3 e 0,7) e para o rendimento (0,5 e 1) e escolhe-se dentre inequações resultantes, a menos restritiva, pois qualquer valor da variável que viola esta, violará todas as demais possibilidades.

Aplicando essa manipulação e análise em inequações diferentes é possível gerar mais de um limite superior e inferior para a mesma variável. Neste caso, deve-se escolher os valores que resultam no menor intervalo para a variável, de tal forma que todos os demais limites sejam satisfeitos. Todo o desenvolvimento das restrições de limite é apresentado no Apêndice A.

Na implementação em código, foi adicionado um fator de encolhimento dos limites de 10%, com a intenção de evitar ainda mais que o otimizador leve as variáveis para uma região de violação das restrições.

5.5.3.1. Limites compatíveis

Conforme já mencionado em 5.5.2 existe a possibilidade de que o limite inferior de uma variável seja maior que o limite superior. Isso claramente significa que não há valor factível para a variável e assim sabe-se antes de iniciar a otimização que há um problema no conjunto de variáveis discretas.

5.5.4. Restrições contínuas

Qualquer restrição que envolve mais que uma variável contínua, ainda que implicitamente, é considerada uma restrição contínua. Todas as restrições contínuas devem estar no formato apresentado na expressão (2.1), e são passadas diretamente ao algoritmo de otimização contínua.

5.5.4.1. Restrição de razão cíclica

A restrição de razão cíclica é a única restrição contínua que não advém de um requerimento de componente. Ao invés disso, essa restrição serve para que o otimizador contínuo nunca saia da região em que existe uma razão cíclica entre 0,3 e 0,7 na qual é possível obter a tensão de saída desejada.

Para isso, inicia-se com a Equação (3.18) e substituem-se os valores de 0,3 e 0,7 para a razão cíclica, isso resulta em um intervalo de tensões de saída possíveis, e a restrição imposta é que a tensão de saída desejada esteja dentro desse intervalo.

$$\frac{0,0063V_i n}{\frac{1,16L_k f_s n^2}{R_o} + 0,0441} \leq V_o \leq \frac{0,147V_i n}{\frac{1,162L_k f_s n^2}{R_o} + 0,0441} \quad (5.4)$$

Por fim são feitas algumas manipulações sobre a inequação (5.4) para se obter as expressões (5.5) e (5.6).

$$\frac{V_o}{V_i n} \left(\frac{1,16L_k f_s n^2}{R_o} + 0,0441 \right) - 0,0063 \geq 0 \quad (5.5)$$

$$0,147 - \frac{V_o}{V_i n} \left(\frac{1,16L_k f_s n^2}{R_o} + 0,0441 \right) \geq 0 \quad (5.6)$$

Essa restrição é muito importante, pois impede o otimizador contínuo de ir para regiões em que o circuito não é simulável, o que aumenta a robustez do otimizador.

5.6. OTIMIZADOR CONTÍNUO

Com todas as equações de perdas e restrições das variáveis contínuas definidas e implementadas em código, foi possível criar e testar o otimizador dessas variáveis.

A função **scipy.minimize()**, da biblioteca Scipy, que é gratuita e *Open Source*, foi a única função do otimizador que não foi desenvolvida pelo autor. Essa função recebe um ponto inicial e realiza o procedimento de otimização, sujeita às restrições

e limites das variáveis, e com a possibilidade de utilizar diferentes métodos, incluindo os dois métodos escolhidos, o SLSQP e COBYLA. É importante mencionar que a função possui uma *flag* de sucesso, que só é verdadeira se o ponto ótimo resultante não viola nenhuma restrição.

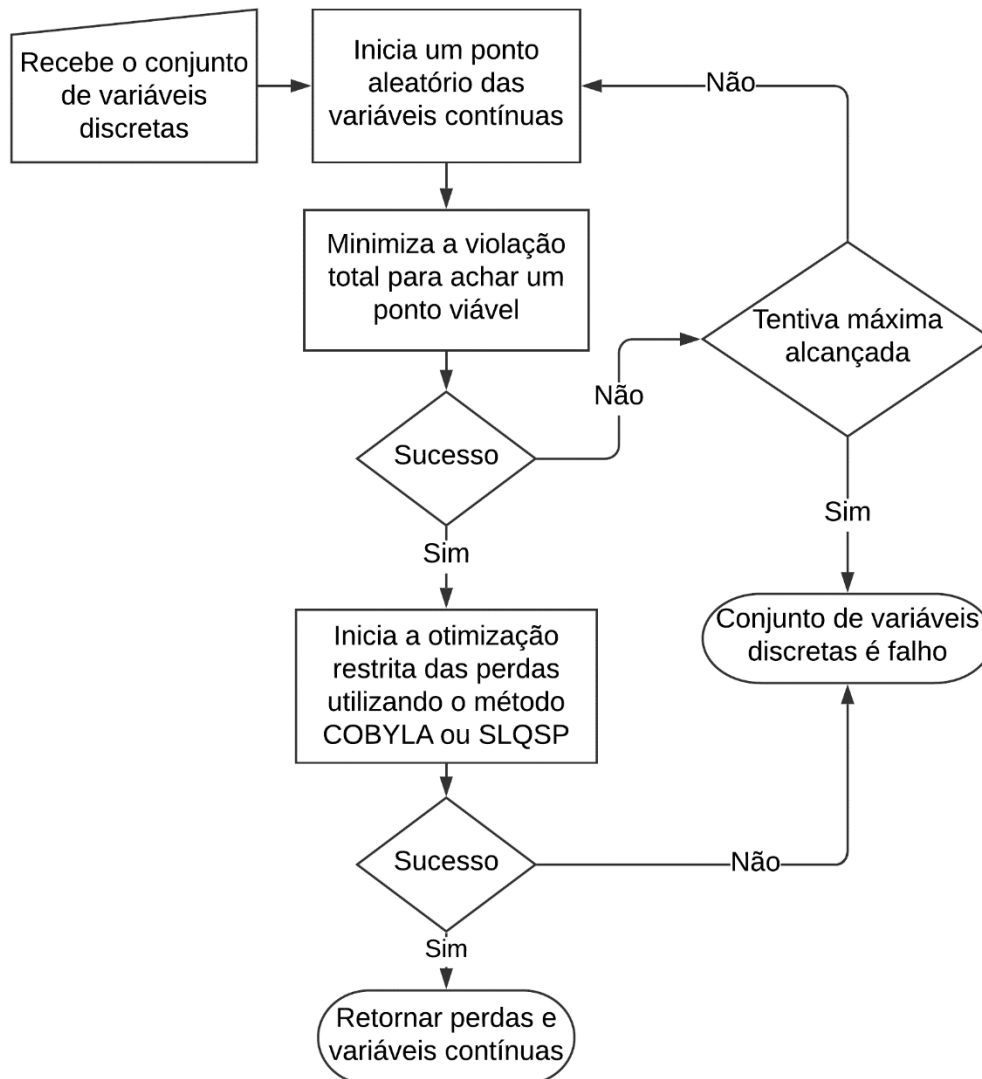
Para facilitar o processo de otimização e a obtenção de um ponto viável, foi criada a função **determine_bounds()** que utiliza as restrições de limite para obter máximos e mínimos para f_s , L_i e L_k .

Já o ponto inicial do otimizador contínuo é obtido pela função **find_feasible_point()** que, dado um conjunto de variáveis discretas, encontra um ponto viável para as variáveis contínuas, através da minimização da função de violação total. Caso essa função não obtenha sucesso, o processo é reiterado até um máximo de 100 tentativas, após as quais sinaliza-se ao GA que o conjunto de variáveis discretas é falho e deve ser penalizado.

Por último, todo o processo foi encapsulado na função **optimize_converter()** que recebe as variáveis discretas, calcula os limites das variáveis contínuas, encontra um ponto factível e otimiza as perdas do conversor. Essa função tem a opção de realizar o procedimento **n** vezes, dando ao algoritmo mais oportunidade de achar o mínimo global.

A Figura 11, apresenta um fluxograma do funcionamento de uma iteração da função **optimize_converter()**. O resultado é um otimizador que ou retorna um ponto ótimo viável para as variáveis contínuas, ou então retorna à informação de que existe um problema no conjunto de variáveis discretas. Não há possibilidade de o ponto final retornado ser inviável.

Figura 11: Fluxograma do Otimizador Contínuo.



Fonte: Próprio Autor.

5.7. ALGORITMO GENÉTICO

Para o algoritmo genético foi criada uma classe denominada **GeneticAlgorithm**, cuja função é implementar o algoritmo genético tal como apresentado na Seção 2.3.1. Dessa forma a classe possui funções para criar a população, testá-la e gerar uma nova população fazendo o *crossover* dos genomas da população anterior e aplicando mutações.

Também foi implementado um método de ajuste de genoma, que utiliza as restrições de conjunto para ajustar o genoma antes de passar ele para a etapa de otimização de variáveis contínuas.

O Algoritmo Genético foi criado para ser o mais genérico o possível. Sendo assim, ele trabalha com: tamanho de genoma, função de criação de genoma, e função de aptidão customizáveis.

Com essa implementação customizável, abstraiu-se o GA dentro da função **optimize_components()**, que recebe os componentes selecionados pelo usuário, ou a biblioteca inteira.

A função de aptidão utilizada é a Equação (5.7), que aumenta a aptidão conforme as perdas do conversor são reduzidas.

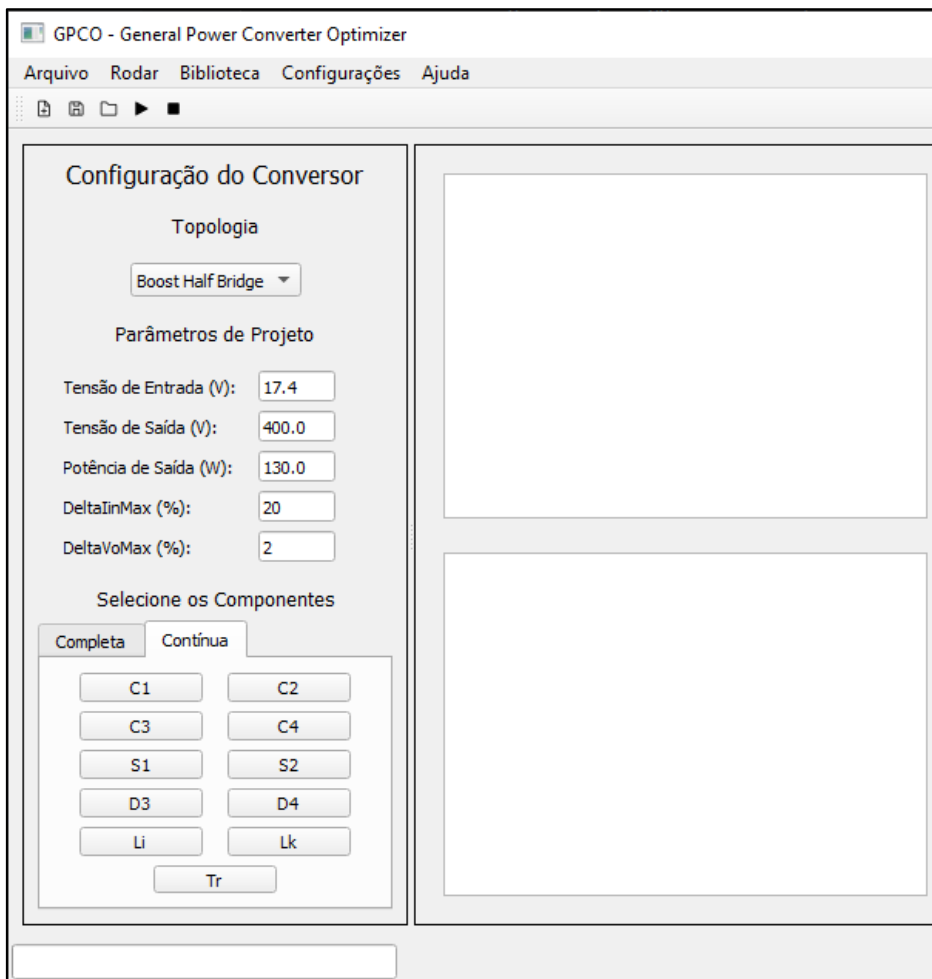
$$a(x, \lambda) = \frac{P_o}{Perdas} \quad (5.7)$$

5.8. INTERFACE GRÁFICA

Após a conclusão do código da representação orientada a objeto, e do equacionamento em código de todas as perdas, foi dado início à criação da interface gráfica, utilizando a biblioteca PyQt5, que é uma implementação em Python da biblioteca Qt.

A Figura 12 mostra como é a tela inicial do aplicativo, nela é possível ver que existem algumas entradas de texto, em que são requeridas informações essenciais para a análise do conversor, muitas das quais presentes na Tabela 1. Ainda na janela inicial existem duas abas, nomeadas “Completa” e “Contínua” que são responsáveis pela escolha dos componentes que farão parte da otimização.

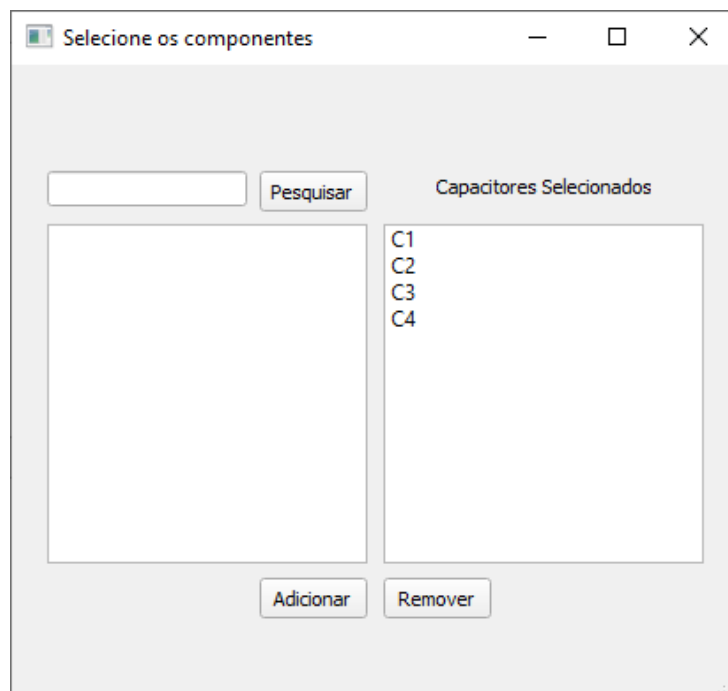
Figura 12: Tela Inicial do Aplicativo.



Fonte: Próprio Autor.

No caso da otimização apenas contínua, além de escolher cada componente, deve-se também selecionar o número de espiras e condutores dos elementos magnéticos. Já para a otimização completa, escolhe-se mais de um componente, pois o GA fará a otimização de componentes. Todas as seleções de componente são feitas através de janelas semelhante à da Figura 13.

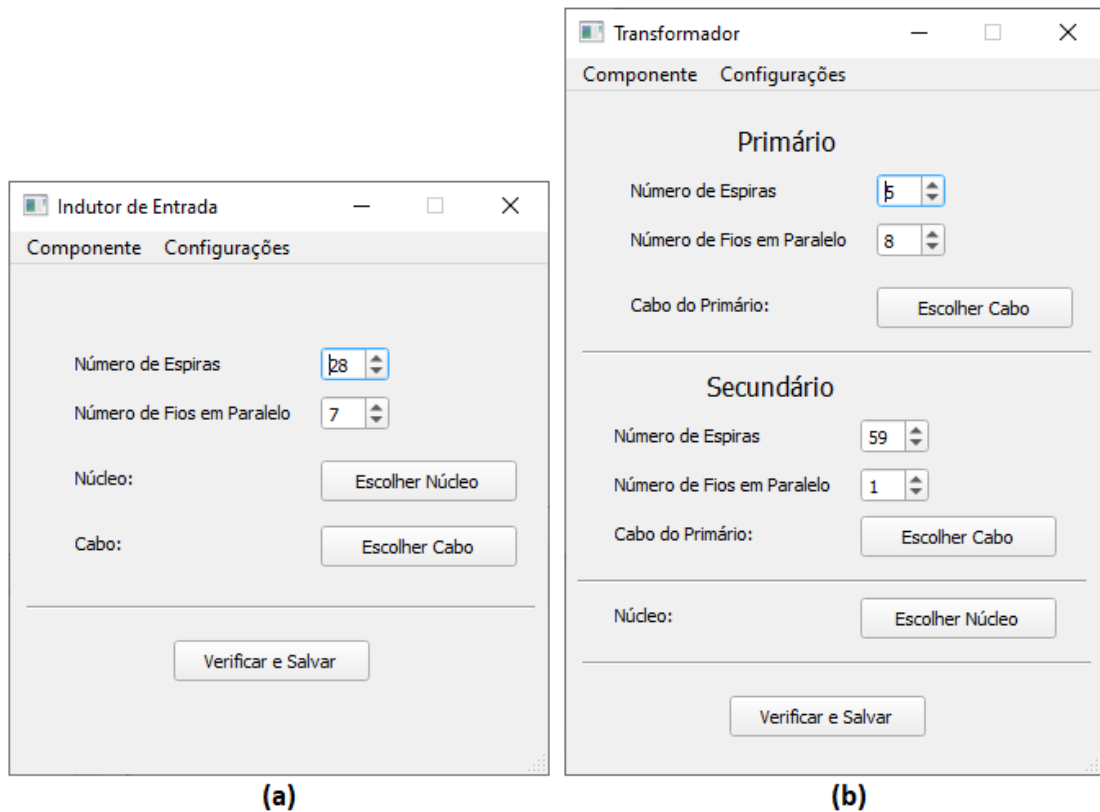
Figura 13: Janela de Seleção de Componente.



Fonte: Próprio autor.

Já as configurações de indutores e transformadores são feitas pelas janelas apresentadas nas figuras 14a e 14b, respectivamente. Nessas janelas, o aplicativo avisa o usuário quando é criado um componente que viola a restrição de área de janela do núcleo escolhido.

Figura 14: Janelas de Configuração de Indutor e Transformador.



Fonte: Próprio autor.

O usuário pode também selecionar a topologia do conversor. No âmbito deste trabalho só há uma topologia disponível, todavia, essa opção está presente para incentivar a introdução de outras topologias no aplicativo.

Depois de configurar todos os parâmetros desejáveis do conversor, escolher o tipo de otimização e componentes, o usuário pode apertar o botão para iniciar a otimização, que quando finalizada apresenta as variáveis otimizadas. Há uma área de gráficos na qual o usuário pode fazer varreduras de algumas variáveis para analisar o comportamento do conversor final.

6. RESULTADOS DO OTIMIZADOR

6.1. TESTE DO OTIMIZADOR CONTÍNUO

Para verificar o funcionamento do otimizador contínuo, foi novamente utilizado o conversor teste, construído na seção 5.3. Com esse conversor foi feita uma varredura manual dos valores das variáveis contínuas para se encontrar o ponto ótimo. Para reduzir o espaço de procura, foi utilizada a função **determine_bounds()** para encontrar os limites máximos e mínimos das variáveis contínuas. Os limites calculados são apresentados na Tabela 6.

Tabela 6: Limites das variáveis para varredura.

Variável	Limite Inferior	Limite Superior
f_s	8,014 kHz	76,032 kHz
L_i	9,458 μ H	179,889 μ H
L_k	39,070 nH	3,716 μ H

Fonte: O autor.

Também foi utilizado o otimizador contínuo para obter o ponto ótimo, usando duas tentativas. Os resultados comparativos entre o otimizador e a varredura manual são apresentados na Tabela 7.

Tabela 7: Comparação de ponto ótimo encontrado.

Variável	Força Bruta	Otimizado	Erro
f_s	29,652 kHz	30,078 kHz	-1,43 %
L_i	183,707 μ H	179,889 μ H	2,08 %
L_k	40,039 nH	36,813 nH	8,06 %
Tempo de Cálculo	8m20s	3s	-----

Fonte: O autor.

Nota-se que o erro das variáveis ótimas é bem pequeno e, mais importante que isso, o tempo que o otimizador leva para obter esse ponto é muito menor que o tempo

que uma varredura manual. Assim o otimizador mostra-se muito eficiente para obter soluções.

6.2. TESTES DO GA

Como o algoritmo genético é um método heurístico, seu resultado é uma solução com alta probabilidade de ser ótima, mas sem garantir que aquela solução é o ponto ótimo global da função.

Ainda assim, é necessário verificar a capacidade do algoritmo de gerar tais soluções sub ótimas, e a velocidade com que ele o faz, e comparando com resultados de uma varredura das variáveis discretas, ter alguma medida da eficácia do método.

Para isso foram criados em código componentes ideais com as mesmas características restritivas que os componentes utilizados em Knaesel (2018), mas com perdas nulas. Com isso, espera-se que o algoritmo genético, se funcional, acabe escolhendo os componentes ideais ao invés dos componentes reais. No conversor, das 23 variáveis discretas, 15 são componentes, então mede-se a exatidão do algoritmo como $n_i/15$ onde n_i é o número de componentes ideias que fazem parte do melhor genoma encontrado.

Esse teste foi realizado quatro vezes, variando-se o tamanho da população e utilizando 100 gerações. O resultado da precisão obtida é apresentado na Tabela 8.

Tabela 8: Resultados do Algoritmo Genético

Tamanho da População	Exatidão	Tempo de Cálculo
10	80,0 %	6 min
25	86,6 %	19 min
50	100 %	34 min
100	100 %	1hr13 min

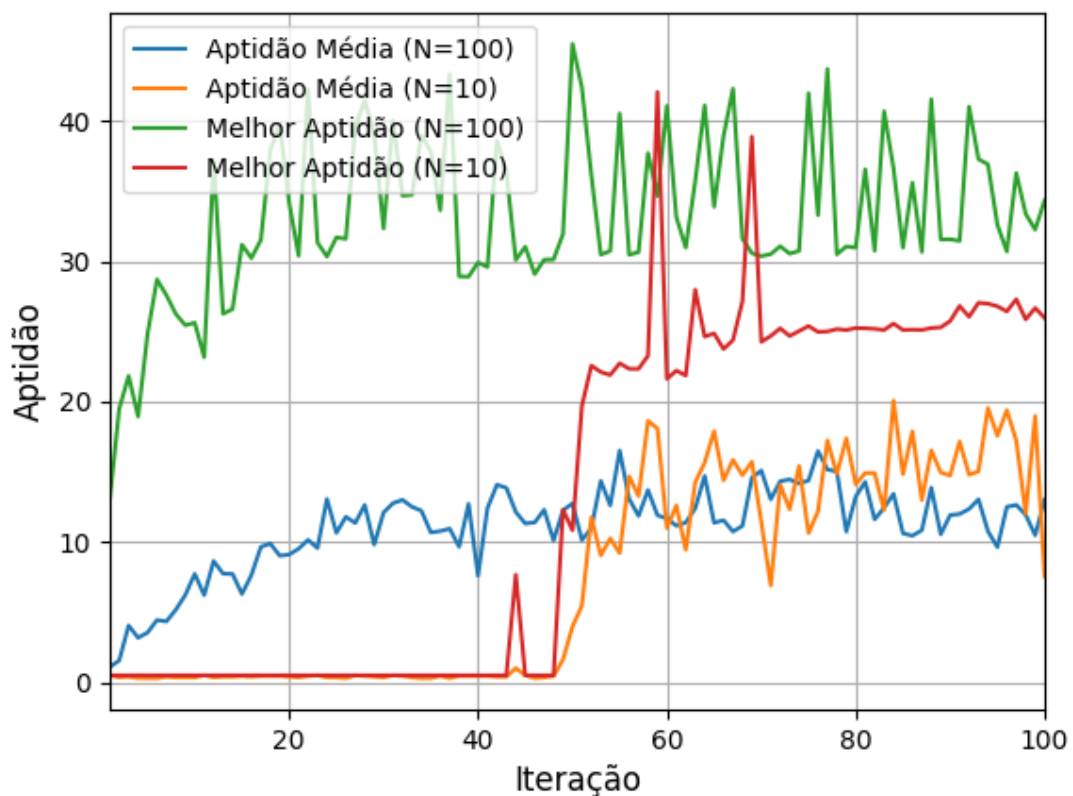
Fonte: O autor.

Com isso, percebe-se que o algoritmo genético é capaz de determinar entre dois componentes semelhantes, qual resulta em menos perdas. Também se nota que

com uma população menor o algoritmo leva menos tempo para realizar a otimização, apesar de obter menor exatidão. Assim é necessário fazer um compromisso entre tamanho de população e número de gerações para obter um resultado rápido e confiável.

Também foram feitos dois testes do algoritmo genético sem os componentes ideais e apenas o banco de dados dos componentes utilizados em Knaesel (2018), utilizando uma população com 10 genomas e outra com 100. Através do resultado desse experimento, apresentado na Figura 15, é possível perceber que a aptidão aumenta com as gerações, e que a população com um número maior de genomas chega a um resultado ótimo mais rapidamente, obtendo um pico de aptidão de 45,6, equivalente à 2,85 W.

Figura 15: Perdas por Geração do Algoritmo Genético.



Fonte: O autor.

Percebe-se também, que assim que o GA encontra uma solução viável, a população inteira começa a convergir.

7. CONCLUSÃO

O projeto de conversores de potência pode ser um processo de cálculo exaustivo e sujeito a erros, devido à grande quantidade de restrições que o circuito e componentes impõe. Com o estudo das metodologias de otimização aplicadas em conversores de potência foi escolhido utilizar uma combinação de um algoritmo genético com um algoritmo numérico.

Foi preciso entender o funcionamento do conversor *Boost Half Bridge*, em especial suas fontes de perdas e as restrições que o conversor e os componentes devem satisfazer. Com isso, foi desenvolvido um simulador do conversor, capaz de calcular todas as variáveis em regime permanente, e com isso calcular as perdas do conversor.

Após criar o simulador foi possível implementar a metodologia de otimização escolhida. Finalmente, desenvolveu-se uma rotina robusta capaz de otimizar todas as variáveis do conversor *Boost Half Bridge*. Os testes realizados sobre a rotina demonstram sua eficácia na otimização, além de levar menos tempo que uma varredura de parâmetros.

O resultado foi embutido numa interface gráfica, nomeada GPCO – *General Power Converter Optimizer*, a qual abstrai os processos internos do usuário e torna o algoritmo mais fácil de usar. Com isso, foi alcançado o objetivo proposto de criar um aplicativo capaz onde é possível analisar, projetar e otimizar o conversor. Além disso, foram adquiridos muitos conhecimentos na área de otimização numérica, análise de conversores de potência e cálculo de perdas.

Cabem ainda algumas sugestões para futuros estudos e implementações no aplicativo.

- Considerar efeitos térmicos na simulação. Isso permitiria utilizar modelos mais realistas dos componentes, além de introduzir os dissipadores à lista de componentes, produzindo um resultado mais completo.
- Realizar testes de mais algoritmos para as variáveis contínuas, aumentando a flexibilidade do aplicativo.
- Comparar o GA com outros algoritmos heurísticos, em especial o anelamento simulado.

- Introduzir especiação ao GA. Isso faz com que genomas muito diferentes uns dos outros, mas com bons valores de aptidão, sejam segregados em diferentes populações que são reproduzidas separadamente. Isso evita que bons genomas, que apenas necessitam de um ajuste fino, sejam extintos precocemente.
- Realizar um estudo para escolha dos parâmetros do GA, fazendo com que o aplicativo possa escolher de maneira automática o tamanho da população e a quantidade de gerações.
- Generalizar as perdas e restrições que advém diretamente dos componentes. Dessa forma quando um conversor novo for implementado, a classe que o representa apenas passa suas variáveis simuladas aos componentes e estes calculam suas respectivas perdas e violações.
- Implementar o algoritmo em uma linguagem de programação compilada.

REFERÊNCIAS

- AUDE, H. T. R; Notes on Quartic Curves. **The American Mathematical Monthly**. v. 56, n. 3, p. 165-170. 1949.
- BALACHANDRAN, Swaminathan; LEE, Fred C.Y. **Algorithms for Power Converter Design Optimization**. **IEEE Transactions on Aerospace and Electronic Systems**. v. 17, n. 3. 1981.
- BARBI, Ivo; MARTINS, Denizar Cruz. **Introdução ao Estudos de Conversores c.c.-c.a.** 1ªEd. Florianópolis: Edição do Autor, 2005. v. 500. 489p.
- BARBI, Ivo. **Projeto de Fontes Chaveadas**. 3ªEd. Florianópolis: Edição do Autor, 2014.
- CARVALHO, Leticia Pivetta. **Sistema de Projetos Otimizados de Conversores Estáticos (Poce) Usando Algoritmos Genéticos**. Tese de Doutorado, UFSM. Santa Maria, RS; 2017.
- CARDOSO, Nilton Pedro. **Inversor Monofásico com Estágio c.c.-c.c. Boost Half Bridge Alimentado a partir de Bateria Veicular**. Trabalho de Conclusão de Curso, UDESC. Joinville, SC; 2017.
- CHINNECK, John W. **Practical Optimization: A Gentle Introduction**. Ottawa, Canada. Edição do Autor. Última edição em 2018.
- DEMONTI, R; MARTINS, D. C.. Photovoltaic Energy Processing for Utility Connected System. **Anais do VI Congresso Brasileiro de Eletrônica de Potência (COBEP 2001)**. Florianópolis, Brasil, pp. 735-739. 2001.
- DURO, B; RAMDSEN, V. S; MUTTIK P. Minimization of active filter rating in high power hybrid filter systems. **Anais da IEEE International Conference on Power Electronics Drive Systems**. Hong Kong, Hong Kong, pp. 1043–1048. 1999.
- ELWAKEIL, Ossama A.; ARORA, Jasbir S.;. Methods for Finding Feasible Points in Constrained Optimization. **AIAA Journal**. v. 33 n. 9. 1995.
- FIACCO, A.V; MCCORMICK, G.P. **Nonlinear Programming: Sequential Unconstrained Minimization Techniques**. New York. Editora Wiley, 1968.
- FREUND, Robert. M. **Penalty and Barrier Methods for Constrained Optimization**. Massachusetts Institute of Technology. Edição do Autor, 2004.
- HAUPT, R. L.; HAUPT, S. E. **Practical Genetic Algorithms**. Edição: 2nd. Hoboken, N.J: Wiley-Blackwell, 2004. ISBN 978-0-471-45565-3.
- HESTENES, Magnus R.. Survey Paper: Multiplier and Gradient Methods. **Journal of Optimization Theory and Applications**. v. 4. n. 5. 1969.
- JEENA, John. Implementation of a Novel Transformerless Inverter Topology for PV Application. **International Journal of Latest Trends in Engineering and Technology**. v. 8, n. 2, p. 301-306.

JIANG, Shuai; CAO, Dong; et. al. Grid-Connected Boost-Half-Bridge Photovoltaic Micro Inverter System Using Repetitive Current Control and Maximum Power Point Tracking. **Twenty-Seventh Annual IEEE Applied Power Electronics Conference and Exposition (APEC)**. 2012.

KNAESEL, Carolina. **Conversão c.c.-c.c. Isolado de Alto Ganho para Integração em Módulos Fotovoltaicos**. Dissertação de Mestrado, UDESC. Joinville, SC; 2018.

MEJBRI, Hanen; AMMOUS, Kaiçar; et al. Bi-objective sizing optimization of power converter using genetic algorithms: Application to photovoltaic systems. **COMPEL - The international journal for computation and mathematics in electrical and electronic engineering**. v. 33, n. 1/2. 2014.

MONGE, Sergio B; et. al. Power Converter Design Optimization: a GA-based design approach to optimization of power electronics. **IEEE Industry Applications Magazine**. 2004.

MOTA, Paulo Vitor de Sousa. **Desenvolvimento de um Inversor para Aplicações Fotovoltaicas com MPPT Integrado**. Tese de Mestrado, Universidade do Minho. Braga; 2013.

PyQt – Python binding for Trolltech’s Qt applications Framework. Disponível em: <<https://riverbankcomputing.com/pyqt/index.php>>

POWELL, Michael James David. **Direct Search Algorithms for Optimizations Calculations**. Universidade de Cambridge, Departamento de Matemática Aplicada e Física Teórica. Cambridge; 1998.

POWELL, Michael James David. **A view of algorithms for Optimization without derivatives**. Universidade de Cambridge, Departamento de Matemática Aplicada e Física Teórica. Cambridge; 2007.

OLIVEIRA, Sérgio Vidal Garcia de. **Otimização de Projeto de Fontes de Alimentação para Centrais de Telecomunicações**. Dissertação de Mestrado, UFSC. Florianópolis, SC; 2001

RARDIN, Ronald. L. **Optimization in Operations Research**. 1ª Edição. Prentice Hall; New Jersey. 1998.

RIDLEY, Raymond. B; ZHOU, Chen; LEE, Fred. C. Application of Nonlinear Design Optimization Tool for Power Converter Components. **IEEE Transactions on Power Electronics**. v. 5, n. 1. 1990

RAHMAN, S; LEE, Fred. C. Nonlinear Program Based Optimization of Boost and Buck-Boost Converter Designs. **32nd International Spring Seminar on Electronics Technology**. v. 32, n. 3, p. 257-281.

VIRTANEN Pauli, et al. e Contribuidores do SciPy 1.0. **SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python**. Nature Methods, v. 17, n. 3, p. 261-272. 2020.

TEIXEIRA, Estêvão Coelho; BRAGA, Henrique Antônio Carvalho; et. al. **Uma visão topológica sobre sistemas fotovoltaicos monofásicos conectados à rede de energia elétrica**. Juiz de Fora. 2003

TIGGEMANN, Henrique. **Análise e desenvolvimento de um inversor monofásico de baixa potência aplicado a sistemas de transporte**. Trabalho de Conclusão de Curso, UNIVANTES. Lajeado, RS; 2008.

XUE, Yaosuo; CHANG, Liuchen; KÆR, Søren Bækthøj; et al. Topologies of Single-Phase Inverters for Small Distributed Power Generators: An Overview. **IEEE Transactions on Power Electronics**. v. 19, n. 5. 2004.

YORK, Ben; YU, Wensong; et al. An Integrated Boost Resonant Converter for Photovoltaic Applications. **IEEE Transactions on Power Electronics**. v. 32, n. 3. 2013.

YU, Yuan; LEE, Fred C.Y; TRINER, James E. Power Converter Design Optimization. **IEEE Transactions on Aerospace and Electronic Systems**. v. 15, n. 3. 1979.

ZHANG, J; et. al. Decoupled optimization technique for design of switching regulators using genetic algorithms. **Anais da IEEE International Conference Circuits and Systems**. Geneva, Switzerland, v. 3. pp. 495–498. 2000.

APÊNDICE A – RESTRIÇÕES DE LIMITE

LIMITES DE FREQUÊNCIA

O limite superior para a frequência de comutação pode ser obtido através da restrição de efeito *skin*, dada pela Equação (A.1). É necessário aplicar essa restrição em todos os fios do conversor e verificar o limite mais restritivo.

$$f_s \leq \left(\frac{2}{D_{cu\text{fio}}} \right)^2 \frac{\rho_{fio}}{\pi \mu_r} \quad (\text{A.1})$$

Já o limite inferior é encontrado isolando a frequência nas expressões (4.25), (4.26), (4.33) e (4.34) que são as restrições dos valores de capacitância. Como essas inequações dependem da razão cíclica e/ou do rendimento, foram substituídos valores para estas variáveis em cada uma, e escolheu-se o caso menos restritivo. Esse procedimento resultou nas inequações abaixo.

$$f_s \geq 1,2857 \cdot 10^{-1} \frac{P_{out}}{4C_1 D \Delta V_{C1} V_{in}^2} \quad (\text{A.2})$$

$$f_s \geq 0,3 \frac{P_{out}}{C_2 \Delta V_{C2} V_{in}^2} \quad (\text{A.3})$$

$$f_s \geq 0,3 \frac{P_{out}}{C_3 V_o^2 \Delta V_{C3}} \quad (\text{A.4})$$

$$f_s \geq 0,3 \frac{P_{out}}{C_4 V_o^2 \Delta V_{C4}} \quad (\text{A.5})$$

Dentro dessas quatro restrições, deve-se escolher a mais restritiva, o que garante a validade das demais.

LIMITES DA INDUTÂNCIA AUXILIAR

A indutância auxiliar não tem nenhuma restrição apresentada na metodologia de projeto que pode ser utilizada como limite. Para isso, é introduzida a restrição de largura de *gap*. O valor da indutância auxiliar possui uma relação direta com a largura de *gap* dado pela equação (A.6). Assim dando um limite para a largura de *gap* do

núcleo, é possível limitar o valor de indutância que o otimizador pode utilizar para fazer a busca do ponto ótimo.

$$L_k = \frac{N_{Lk}^2 \mu_0 A_{eLk}}{l_{gLk}} \quad (\text{A.6})$$

Esse limite não precisa ser exato, mas espera-se pelo menos que ele dê uma noção da dimensão do valor, então utilizou-se $l_{gLk} \in [10^{-4}m, 5 \cdot 10^{-2}m]$. Com isso, tem-se os limites da indutância auxiliar.

$$20N_{Lk}^2 \mu_0 A_{eLk} \leq L_k \leq 10^4 N_{Lk}^2 \mu_0 A_{eLk} \quad (\text{A.7})$$

LIMITES DA INDUTÂNCIA DE ENTRADA

Os primeiros limites da indutância de entrada são obtidos da mesma maneira que os da indutância auxiliar, através da largura de *gap*, então utiliza-se a Equação (A.7) porém com os parâmetros de L_i .

No caso da indutância de entrada, é possível utilizar também a restrição de campo máximo, apresentada na expressão (4.15). Isolando a indutância de entrada nessa tem-se:

$$L_i \leq \frac{B_{\text{máx}Li} A_{eLi} N_{Li}}{\left(I_{in(MED)} + \frac{\Delta I_{in}}{2} \right)} \quad (\text{A.8})$$

Percebe-se que a expressão (A.8) se torna menos restritiva à medida que a ondulação da corrente de entrada diminui, substitui-se $\Delta I_{in} = 0$ A. Também é importante levar em consideração que a corrente média depende do rendimento, mas fica claro que um rendimento maior leva a uma expressão menos restritiva, então utiliza-se $\eta = 1$.

$$L_i \leq \frac{V_i B_{\text{máx}Li} A_{eLi} N_{Li}}{P_o} \quad (\text{A.9})$$

Por fim, o algoritmo deve escolher entre o limite de *gap* e o de campo máximo, o mais restritivo.

APÊNDICE B – EXEMPLO DE ARQUIVO DE COMPONENTE

Todos os componentes são salvos num arquivo de texto, com o formato exemplificado pelo arquivo do núcleo NEE_30_15.

Name: NEE_30_15	# nome do componente
AeAw: 1.037e-8	# produto das áreas
Ae: 1.22e-4	# área do entreferro
Aw: 0.85e-4	# área da janela
Ve: 8.17e-6	# volume
Alpha: 7.9292e-3	# expoente da frequência (perdas volumétricas)
Beta: 1.4017	# expoente do campo (perdas volumétricas)
Kc: 2.3294	# parâmetro multiplicativo (perdas volumétricas)
Lt: 6.7e-2	# comprimento médio da espira
Bj: 6.7e-2	# largura da janela