



LUIS GUSTAVO FERREIRA VERNUCCI

**UTILIZAÇÃO DO ALGORITMO TLBO PARA OBTENÇÃO DE
GANHOS PARA O CONTROLE DO CONVERSOR CC-CC
BUCK**

LAVRAS – MG

2021

LUIS GUSTAVO FERREIRA VERNUCCI

**UTILIZAÇÃO DO ALGORITMO TLBO PARA OBTENÇÃO DE GANHOS PARA O
CONTROLE DO CONVERSOR CC-CC BUCK**

Trabalho de conclusão de curso apresentado à
Universidade Federal de Lavras, como parte
das exigências do Programa de Graduação em
Engenharia de Controle e Automação.

Prof. Dr. Vinicius Miranda Pacheco

Orientador

LAVRAS – MG

2021

**Ficha catalográfica elaborada pela Coordenadoria de Processos Técnicos
da Biblioteca Universitária da UFLA**

Vernucci, Luis Gustavo Ferreira

Utilização do algoritmo TLBO para obtenção de ganhos para o controle do conversor CC-CC BUCK / Luis Gustavo Ferreira Vernucci. – Lavras : UFLA, 2021.

56 p. : il.

TCC–Universidade Federal de Lavras, 2021.

Orientador: Prof. Dr. Vinicius Miranda Pacheco.

Bibliografia.

1. TCC.

CDD-808.066

LUIS GUSTAVO FERREIRA VERNUCCI

**UTILIZAÇÃO DO ALGORITMO TLBO PARA OBTENÇÃO DE GANHOS PARA O
CONTROLE DO CONVERSOR CC-CC BUCK**

Trabalho de conclusão de curso apresentado à
Universidade Federal de Lavras, como parte
das exigências do Programa de Graduação em
Engenharia de Controle e Automação.

APROVADA em 30 de Novembro de 2021.

Prof. Dr. Vinicius Miranda Pacheco	UFLA
Prof. Dr. Bruno Henrique Groenner Barbosa	UFLA
Profa. Dra. Silvia Costa Ferreira	UFLA

Prof. Dr. Vinicius Miranda Pacheco
Orientador

**LAVRAS – MG
2021**

Espaço reservado a dedicatória.

AGRADECIMENTOS

Enfim o final de uma jornada está no horizonte, o próximo desafio é incerto, mas vem com certezas. Certeza de que trilhei e segui o caminho certo em uma instituição única e que me deu orgulho desde o primeiro dia.

Nada na vida é fácil, mas me formar na Universidade Federal de Lavras me mostrou que a dificuldade e persistência geralmente são recompensadas com crescimento. Crescimento que me fez e me faz melhor a cada dia.

Primeiramente agradeço a Deus por me dar a força espiritual e quiçá sorte por encontrar pessoas tão únicas e incríveis nesta longa jornada.

Agradeço especial ao meu pai, Roberto Vernucci, e minha mãe, Cláudia Ferreira Vernucci, que além de me dar a educação e base necessárias para trilhar este longo caminho, me deram força e paz para poder seguir em frente e buscar a excelência a cada dia da minha vida. Sem eles nada seria possível e nada seria viável, esta conquista e todas as outras que irão vir são igualmente compartilhadas com eles.

De maneira mais que especial agradeço minha esposa Flávia, que sempre me incentivou e me apoiou nesta e em tantas outras jornadas, Ela e nossa filha Beatriz são os pilares que faltavam para que eu me tornasse uma pessoa e um profissional cada vez melhor.

Agradeço aos amigos que fizeram ou fazem parte do Núcleo de Estudos do Laboratório de Mobilidade Terrestre por sempre proporcionar e acreditar nos projetos e trabalhos por mim propostos. Eles formaram a base técnica e teórica necessária para a realização deste e de tantos outros projetos.

Por fim, agradeço a todos os membros da comunidade acadêmica (professores, técnicos administrativos e profissionais auxiliares) por proporcionarem um ambiente único e de excelência para que minhas realizações fossem possíveis.

Espero um dia poder retribuir minimamente a todos.

RESUMO

Este trabalho apresenta a utilização do algoritmo *Teaching-Learning-Based Optimization* (TLBO) na determinação dos ganhos para o controle de fontes chaveadas utilizando realimentação de estados. Fonte chaveada é um dispositivo capaz de converter energia elétrica, adequando tensão e/ou corrente aos requisitos das cargas alimentadas por estes conversores, por meio do chaveamento de um dispositivo semicondutor. O controle destas fontes pode ser feito por realimentação de estados. Este método envolve o uso do vetor de estado para calcular a ação de controle para a dinâmica do sistema especificado. O TLBO é um método baseado em interações sociais entre professor e estudantes que pode ser uma ferramenta útil no auxílio do projeto de um sistema de controle eficaz para fontes chaveadas. A partir da compreensão do algoritmo TLBO e da sua utilização na obtenção de ganhos de controle por realimentação de estados, foram realizadas simulações, no Matlab para obtenção dos ganhos pelo algoritmo TLBO, e no Simulink com um circuito elétrico de um conversor CC-CC e utilizando blocos para avaliar a eficácia do controle.

Palavras-chave: Fontes chaveadas. Controle. Algoritmo TLBO.

ABSTRACT

This work presents the use of the algorithm in determining the gains for source control switched using state feedback. Switched power supply is a device capable of converting electrical energy, adapting voltage and/or current to the requirements of the loads supplied by these converters, through the switching of a semiconductor device. Control of these sources can be done by state feedback. This method involves using the state vector to calculate the control action for the dynamics of the specified system. TLBO is a method based on social interactions between teacher and students that can be a useful tool in helping to design an effective control system for switching sources. From the understanding of the TLBO algorithm and its use in obtaining control gains by state feedback, simulations were performed in Matlab to obtain the gains by the TLBO algorithm, and in Simulink with an electrical circuit of a CC-DC converter and using blocks to evaluate the control efficiency.

Keywords: Switched sources. Control. TLBO algorithm.

LISTA DE FIGURAS

Figura 2.1 – Regulador Linear Básico.	15
Figura 2.2 – Conversor CC-CC chaveado básico.	16
Figura 2.3 – Etapas de operação do conversor BUCK.	17
Figura 2.4 – Diagrama de blocos de um sistema de controle linear.	19
Figura 2.5 – Funções de transferência do conversor.	20
Figura 2.6 – Servossistema sem integrador (tipo 1).	22
Figura 3.1 – Algoritmo TLBO parte 1.	24
Figura 3.2 – Algoritmo TLBO parte 2.	25
Figura 3.3 – Definição das Variáveis.	26
Figura 3.4 – Inicialização do Algoritmo.	27
Figura 3.5 – Cálculo da média das posições e definição do indivíduo Professor.	28
Figura 3.6 – Fase professor.	28
Figura 3.7 – Fase de aprendizado.	30
Figura 3.8 – Armazenamento da melhor solução.	30
Figura 3.9 – Gráfico logarítmico com os resultados finais.	31
Figura 3.10 – Função custo.	32
Figura 3.11 – Conversor Buck no <i>Simulink</i>	33
Figura 3.12 – Resposta da simulação para ganhos iguais a zero. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	34
Figura 3.13 – Resposta da simulação para ganhos iguais a 0,01. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	35
Figura 3.14 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 0,05. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	35
Figura 3.15 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 0,05. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	36
Figura 3.16 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 1. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	36

Figura 3.17 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	37
Figura 3.18 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 10. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	37
Figura 3.19 – Resposta da simulação para ganhos k_1 igual a 0,05, k_2 igual a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	38
Figura 3.20 – Resposta da simulação para ganhos k_1 igual a 0,025, k_2 igual a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	39
Figura 3.21 – Resposta da simulação para ganhos k_1 igual a 0,005, k_2 igual a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	39
Figura 3.22 – Resposta da simulação para ganhos k_1 igual a 0,0075, k_2 igual a 0,01 e k_i igual a 5, A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	40
Figura 3.23 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,05 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	41
Figura 3.24 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,025 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	41
Figura 3.25 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,005 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	42
Figura 3.26 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,001 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	42

Figura 3.27 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,002 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	43
Figura 3.28 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,003 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	43
Figura 3.29 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,004 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	44
Figura 4.1 – Evolução do custo em relação ao número de iterações com o número máximo de iterações igual a 5.	45
Figura 4.2 – Evolução do custo em relação ao número de iterações com o número máximo de iterações igual a 10.	46
Figura 4.3 – Evolução do custo em relação ao número de iterações com o número máximo de iterações igual a 15.	46
Figura 4.4 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0 à 0,01, k_2 variando de 0 à 0,003 e k_i variando de 2 à 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	49
Figura 4.5 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0 à 0,01, k_2 variando de 0 à 0,003 e k_i variando de 4 à 6. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	50
Figura 4.6 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0 à 0,01, k_2 variando de 0 à 0,003 e k_i variando de 4,5 à 5,5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	51
Figura 4.7 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,0025 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	51

Figura 4.8 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,0025 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. Com a população igual a 50 e máximo de iterações igual a 15. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	52
Figura 4.9 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,005 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. Com a população igual a 10 e máximo de iterações igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	52
Figura 4.10 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,0025 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. Com a população igual a 50 e máximo de iterações igual a 15. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.	53

LISTA DE TABELAS

Tabela 3.1 – Descrição dos testes realizados para definição dos limites máximos do ganho k_i	38
Tabela 3.2 – Descrição dos testes realizados para definição dos limites máximos do ganho k_1	40
Tabela 3.3 – Descrição dos testes realizados para definição dos limites máximos do ganho k_2	44
Tabela 4.1 – Apresentação dos dados relativos aos testes de identificação do comportamento do algoritmo com a mudança no número de iterações.	47
Tabela 4.2 – Apresentação dos dados relativos aos testes de identificação do comportamento do algoritmo com a mudança no tamanho da população.	48
Tabela 4.3 – Apresentação dos dados relativos aos testes de identificação do comportamento do algoritmo com as alterações dos limites dos ganhos.	53

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivos	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	14
2	Revisão da Literatura	15
2.1	Conversores CC-CC	15
2.1.1	Conversor Linear Comum	15
2.1.2	Conversor Chaveado Básico	16
2.1.3	Conversor BUCK ou Abaixador	17
2.2	Modelagem de Conversores CC-CC BUCK	18
2.2.1	Método de Variáveis de Estado	18
2.2.2	Representação de fontes chaveadas no espaço de estados	19
2.2.3	Característica dinâmica do conversor Buck no MCC	20
2.3	Controle por realimentação de estados	21
2.4	Algoritmo TLBO	22
3	Metodologia	24
3.1	Algoritmo TLBO	24
3.2	Desenvolvimento do algoritmo TLBO no Matlab	26
3.3	Definição da função custo	29
3.4	Implementação do conversor Buck no <i>Simulink</i>	32
3.5	Definição das faixas de ganhos a serem utilizadas no algoritmo TLBO	34
3.5.1	Definição do limite máximo para o ganho do integrador (k_i)	35
3.5.2	Definição do limite máximo para o ganho k_1	38
3.5.3	Definição do limite máximo para o ganho k_2.	40
3.5.4	Apresentação dos limites máximos dos ganhos obtidos nos itens anteriores.	44
4	Resultados e Discussões	45
4.1	Análise da interferência do número máximo de iterações na resposta do algoritmo.	45
4.2	Análise da interferência do tamanho da população na resposta do algoritmo.	47
4.3	Refinamento dos limites dos ganhos k_i, k_1 e k_2.	48
5	CONCLUSÃO	54

REFERÊNCIAS 55

1 INTRODUÇÃO

Equipamentos eletrônicos necessitam de uma fonte de alimentação para funcionar, cada um com um determinado grau de estabilização. Em equipamentos de pequeno porte é possível que essa alimentação seja realizada por baterias ou pilhas, mas em geral, é necessário que tal dispositivo esteja conectado à rede elétrica local e isso é possível através de conversores (MEHL, 2004).

De acordo com as diretrizes para futuras produções de energia elétrica, que foram estabelecidas no protocolo de Quioto (BRASIL, 1988), empresas, organizações públicas, governos e a sociedade como um todo se mostram cada vez mais preocupados com a produção de energia de forma mais limpa possível. Segundo IRENA (Renewable Energy Agency), as emissões de CO_2 durante os anos de 2017 e 2018 tiveram aumentos significativos, observados principalmente pelo uso de combustíveis fósseis (IRENA, 2019). Por este motivo, cresce a necessidade da redução do uso de combustíveis fósseis e cresce a necessidade da utilização de fontes cada vez mais sustentáveis e renováveis (THIJS, 2012).

Percebe-se também que ao longo do tempo existe um crescente aumento populacional e isso induz a um contínuo crescimento por energia elétrica. A fim de suprir essas necessidades, as concessionárias de energia precisam se adaptar para conseguirem acompanhar o crescimento da demanda de energia. Estudos apontam que, ao longo dos últimos anos e em um futuro próximo, cada vez mais é crescente a parcela da população atendida por autoprodução, que por sua vez pode chegar a 10% de toda energia consumida em 2026 (EPE, 2017).

Com o crescimento do uso de equipamentos eletrônicos e também do uso de redes com Geração Distribuída (GD), a Qualidade de Energia Elétrica (QEE) vem ganhando espaço na preocupação das concessionárias de energia e também de consumidores, uma vez que a QEE está diretamente ligada à vida útil dos equipamentos eletrônicos contidos no sistema (NAGATA, 2018).

A eficiência das fontes chaveadas está diretamente ligada a qualidade do projeto de controle que elas possuem. Atualmente a modelagem de fontes chaveadas é realizada através da obtenção de um modelo matemático para que seja possível projetar um sistema de controle eficiente. Este pode ser obtido através do método de variáveis de estado. Neste método busca-se a função de transferência através da linearização do estágio de potência utilizando valores médios das variáveis de estado. Inicialmente o sistema é descrito no espaço de estados, após

uma pequena perturbação e realizada a separação das componentes CC e CA. O modelo é finalizado com a obtenção da função de transferência (POMILIO, 2005).

Com o objetivo de otimizar as soluções atuais, a obtenção dos parâmetros de controle de fontes chaveadas podem ser realizadas através de soluções que empregam Redes Neurais Artificiais (RNA) que podem ser definidas como técnicas de aprendizado de máquina baseadas na disposição de células neurais de animais. Dentre vários algoritmos disponíveis, é possível destacar o *Teaching-Learning Based Optimization* (TLBO). Este algoritmo é baseado em iterações professor-alunos, simulando uma sala de aula. O TLBO já foi empregado tanto em problemas de regressão como em classificação. Vários estudos comprovam que o TLBO possui resultados interessantes comparados com algoritmos semelhantes (GARCIA CRISTIANO; CATALANO, 2018).

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é realizar a implementação do algoritmo *Teaching-Learning Based Optimization* (TLBO) para determinar os ganhos de controle de fontes chaveadas, k_i , k_1 e k_2 , utilizando realimentação de estados.

1.1.2 Objetivos Específicos

Como objetivos específicos deste trabalho, tem-se:

- Formar uma base de conhecimento em fontes chaveadas CC-CC do tipo BUCK para melhorar a compreensão do seu princípio de funcionamento.
- Compreender de maneira profunda o funcionamento do algoritmo TLBO para ser capaz de implementar um programa para a determinação dos ganhos do sistema de controle.
- Definir valores mínimos e máximos utilizados no algoritmo TLBO.
- Analisar a interferência do número de iterações e do número da população nos resultados finais.
- Realizar simulações de uma fonte chaveada no *Simulink* para executar e analisar os resultados obtidos.

2 REVISÃO DA LITERATURA

Neste capítulo serão apresentadas algumas teorias para formar base de conhecimento para o completo entendimento dos temas abordados no trabalho.

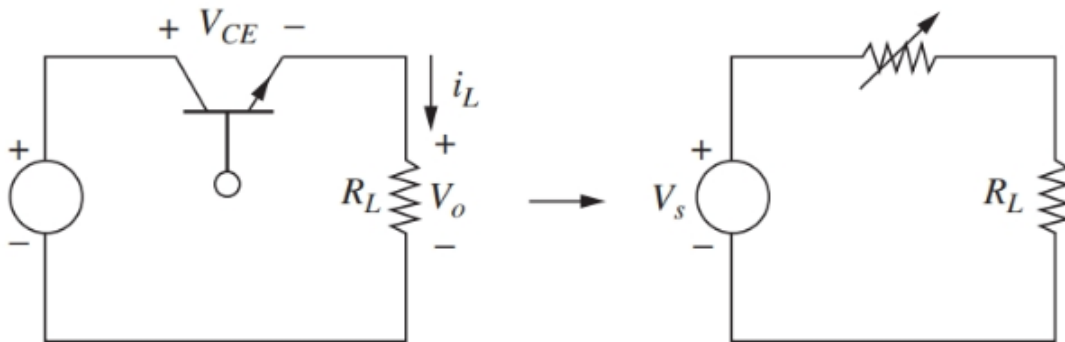
2.1 Conversores CC-CC

Para que seja possível melhor compreender o conversor utilizado neste trabalho, algumas teorias serão apresentadas.

2.1.1 Conversor Linear Comum

Na Figura 2.1 abaixo é possível observar o circuito de um regulador de tensão linear básico. Em um regulador linear comum a potência absorvida pela carga é $V_o I_L$ e a potência absorvida pelo transistor é $V_{ce} I_L$. Neste caso, o transistor está operando na região linear, sendo assim, ele se comporta como uma resistência variável como mostrado no circuito a direita da Figura 2.1. Supondo uma baixa corrente na base, a perda de potência no transistor torna este circuito ineficiente (HART, 2011).

Figura 2.1 – Regulador Linear Básico.



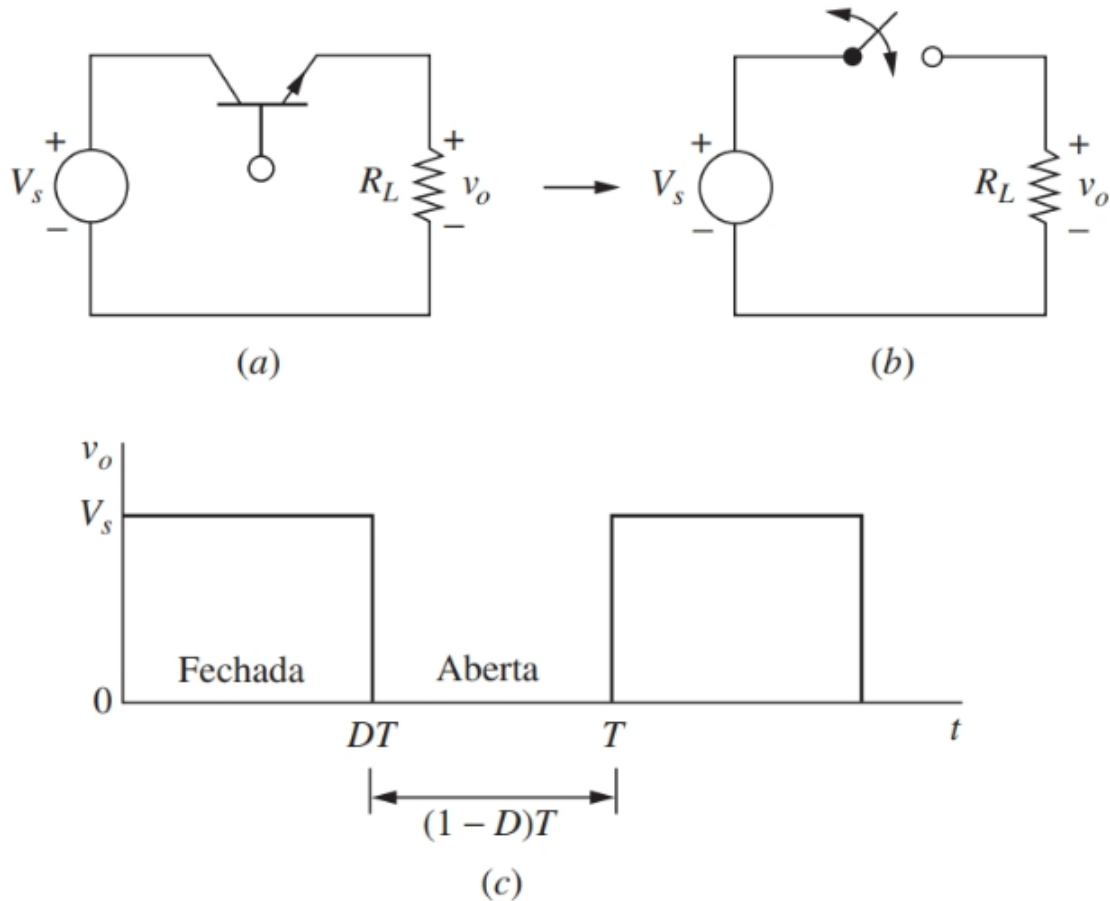
Fonte: Hart (2011)

Como exemplo se a tensão de saída for 25% da tensão de entrada (ou seja, com uma eficiência de 25%), o transistor absorverá os outros 75% da potência que foi fornecida pela fonte.

2.1.2 Conversor Chaveado Básico

Como alternativa eficiente para o regulador linear tem-se o conversor chaveado, sua funcionalidade pode ser descrita como: o transistor opera como uma chave eletrônica, sendo completamente ligado ou completamente desligado, podendo ser observado na Figura 2.2.

Figura 2.2 – Conversor CC-CC chaveado básico.



Fonte: Hart (2011)

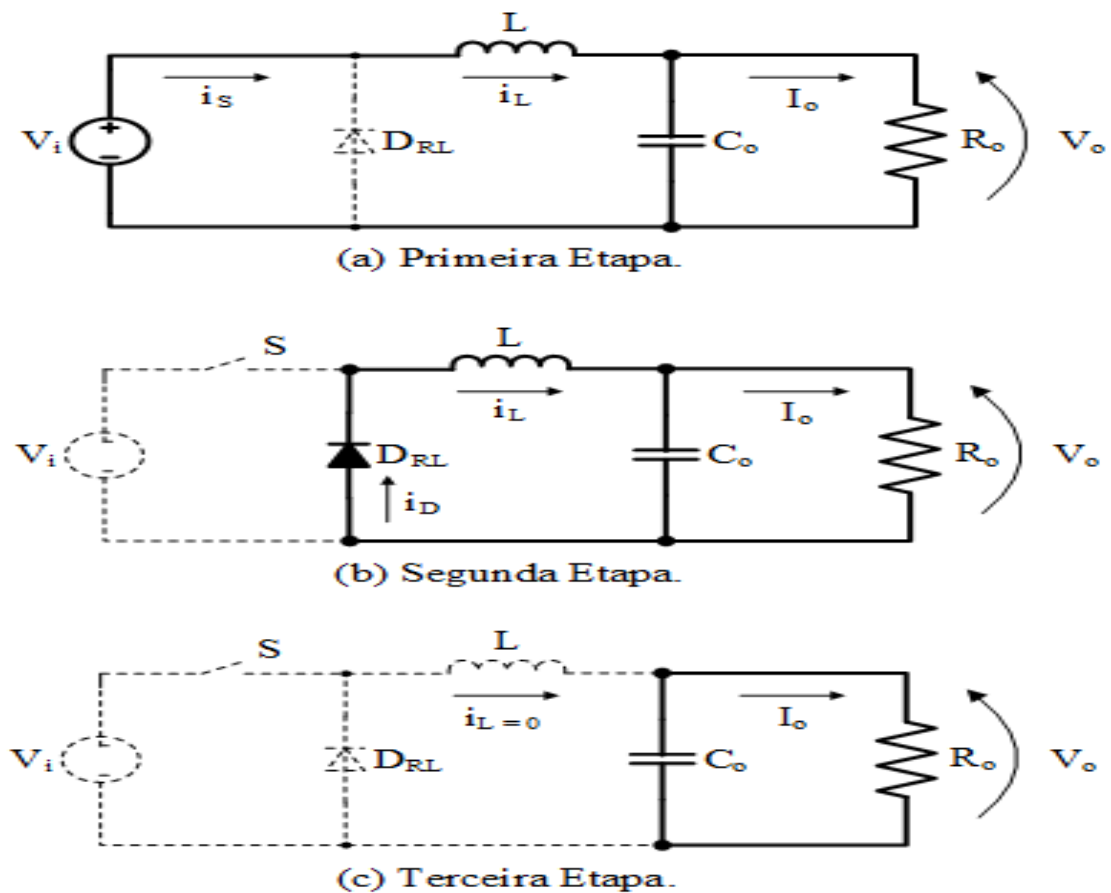
Neste caso, a potência absorvida pela chave ideal é zero, ou seja, quando a chave está aberta não existe corrente e quando a chave está fechada não existe tensão. Portanto, a potência é completamente absorvida pela carga e sua eficiência é de 100%. Porém, em uma chave real ocorrem perdas, uma vez que quando existe condução a tensão não é zero e a chave deve passar pela região de saturação quando fizer a mudança de um estado para outro (HART, 2011).

No item C da Figura 2.2 é possível observar o ciclo de trabalho (*Duty cycle*). Ciclo de trabalho pode ser definido como a proporção de tempo que uma carga ou circuito está ligado quando compara-se com o tempo em que a carga ou circuito está desligado (HART, 2011).

2.1.3 Conversor BUCK ou Abaixador

Este conversor funciona controlando a componente CC com uma tensão pulsada na saída. Para eliminar o chaveamento e criar uma saída puramente CC, coloca-se um filtro passa baixa depois da chave. Imaginando que este filtro seja ideal, a tensão da sua saída é a média da tensão de entrada (HART, 2011). Este conversor pode ser observado nos circuitos abaixo ilustrados na Figura 2.3.

Figura 2.3 – Etapas de operação do conversor BUCK.



Fonte: Moraes (2018)

Na Figura 2.3 é possível observar todas as etapas de funcionamento do conversor BUCK. A primeira etapa representa apenas o momento em que existe condução na chave S . A etapa seguinte representa o momento em que o diodo D_{LR} entra em condução e ela segue até que a corrente no indutor se anule. A última etapa representa o momento em que o diodo é bloqueado e a carga passa receber energia apenas do capacitor C (MORAES, 2018).

A escolha da frequência de chaveamento e da indutância são essenciais para sua utilização em projetos. Quanto maior a frequência de chaveamento, menores ficam os valores

mínimos para o indutor produzir o modo de condução contínua e menores ficam os valores para o capacitor limitar a ondulação na saída (HART, 2011).

Porém, para altas frequências existe um aumento nas perdas de potência nas chaves, o que significa produção de calor. Isso diminui sua eficiência, o que pode acarretar na necessidade de um dissipador de calor maior (HART, 2011).

2.2 Modelagem de Conversores CC-CC BUCK

2.2.1 Método de Variáveis de Estado

Para compreensão do método de variáveis de estado, é importante entender o que é estado. Estado de um sistema dinâmico é o menor conjunto de variáveis (conhecidas como variáveis de estado). O conhecimento dessas variáveis no instante $t = t_0$ e também para os instantes em que $t \geq t_0$ determinam completamente o comportamento do sistema em qualquer instante de tempo (OGATA, 2011).

Neste contexto, variáveis de estado de um sistema dinâmico são definidas como aquelas variáveis que constituem o menor conjunto de variáveis que juntas conseguem determinar o estado de um sistema dinâmico (OGATA, 2011).

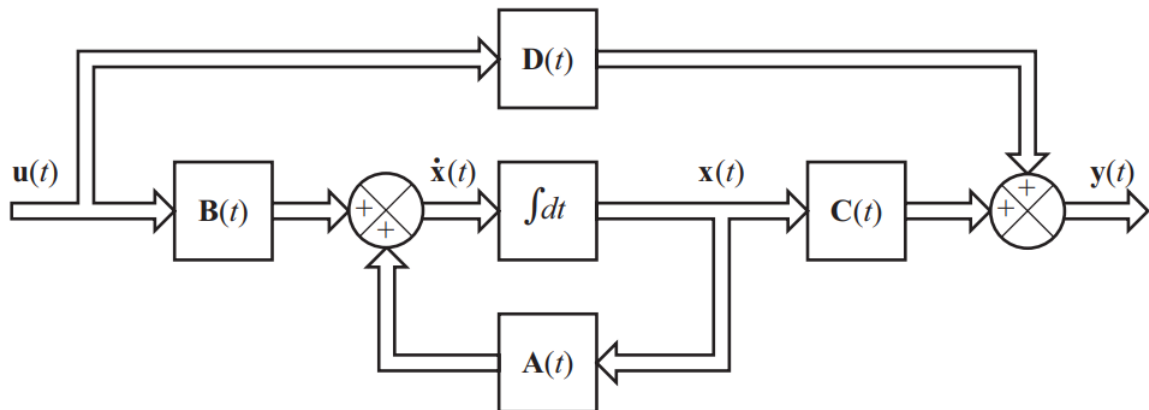
Levando em conta o sistema operando em modo contínuo, tem-se uma configuração possível para cada etapa de operação do conversor eletrônico. No caso de uma fonte chaveada, tem-se duas etapas possíveis, uma quando existe condução na chave e outra quando não existe.

Sendo que espaço de estado pode ser descrito em função de eixos coordenados formados pelos eixos de x_1, x_2, \dots, x_n . Estes eixos são as variáveis de estado e o espaço n-dimensional destes eixos, é denominado espaço de estados (OGATA, 2011).

Em complemento, a análise no espaço de estados envolve três tipos de variáveis, as variáveis de entrada, saída e de estado (OGATA, 2011). Em seguida, na Figura 2.4 é apresentado um exemplo que ilustra o diagrama de blocos de um sistema de controle linear de tempo contínuo, representado no espaço de estados.

A Figura 2.4 representa um sistema com uma entrada $u(t)$ e saída $y(t)$, onde $A(t)$ é a matriz de estado, $B(t)$ é a matriz de entrada, $C(t)$ é a matriz de saída e $D(t)$ é a matriz de transmissão direta (OGATA, 2011).

Figura 2.4 – Diagrama de blocos de um sistema de controle linear.



Fonte: Ogata (2011)

2.2.2 Representação de fontes chaveadas no espaço de estados

Para a elaboração de um modelo linear para o estágio de potência, onde se inclui um filtro de saída, foi desenvolvida uma técnica que tinha por finalidade obter um modelo de variáveis médias no espaço de estado (MIDDLEBROOK; CUK, 1976).

Com o objetivo de obter o cálculo das variáveis médias das variáveis de estado utilizando o ciclo de trabalho, é necessário realizar a descrição do sistema no espaço de estado.

Em cada subintervalo o vetor de estado x é composto pelas tensões nos capacitores e as correntes nos indutores, podendo ser inclusas as respectivas resistências nos elementos citados e algumas interferências nos interruptores (pensando em um sistema real). Por isso é incluído um elemento parasita, uma resistência, na saída do capacitor (POMILIO, 2005).

Para que seja possível realizar a descrição média das variáveis em um período de chaveamento, é necessário realizar um ponderamento das equações correspondentes as duas variações possíveis em relação ao tempo (POMILIO, 2005).

O termo variáveis médias é utilizado para descrever o valor médio de cada variável considerada (que em vias normais são a corrente no indutor e a tensão no capacitor). O valor médio é aquele calculado a cada período de comutação. Compreende-se então que o modelo não seria capaz de representar o *ripple* da tensão ou da corrente, mas traz a representação evolutiva do valor médio destas variáveis (POMILIO, 2005).

Para que o modelo atue adequadamente e reproduza fielmente o comportamento do sistema é necessário que as realimentações sejam devidamente filtradas para que sinais com alta frequência não interfiram no resultado. Como o modelo não foi previsto para casos em que o

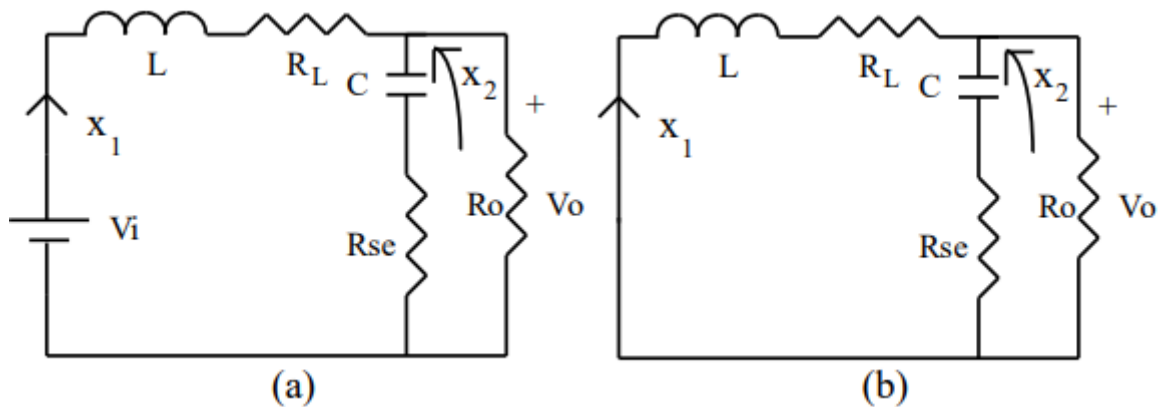
ripple de alta frequência esteja presente, este deve ser suficientemente atenuado (POMILIO, 2005).

A representação no domínio da frequência, ou seja, em termos de funções de transferência, onde cada bloco do sistema pode ser representado por uma função de transferência. As letras v , d e i representam os pequenos sinais causadores, ou resultados, da perturbação.

2.2.3 Característica dinâmica do conversor Buck no MCC

A Figura 2.5 apresenta os circuitos equivalentes do conversor BUCK para cada etapa de operação.

Figura 2.5 – Funções de transferência do conversor.



Fonte: Pomilio (2005)

Na Figura 2.5 é possível observar uma resistência em série com capacitor, representada por R_{se} e uma resistência em série com o indutor representada por R_L . A corrente que passa pelo indutor e pelo capacitor são respectivamente representadas por x_1 e x_2 (POMILIO, 2005)

Levando em consideração a malha externa no circuito exposto na Figura 2.5, tem-se:

$$-V_i + L.\dot{x}_1 + R_L.X_1 + R_0.(x_1 - C.\dot{x}_2) = 0$$

A equação de tensões para a malha de saída pode ser representada por:

$$-x_2 - C.R_{se}.\dot{x}_2 + R_0.(x_1 - C.\dot{x}_2) = 0$$

Assumindo um intervalo normalizado, é possível rescrever a equações anteriores de forma matricial, e seu resultado ficaria:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} -(R_0.(R_{se} + R_L) + R_{se}.R_L)/(L.(R_{se} + R_0)) & -R_0/(L.(R_{se} + R_0)) \\ R_0/(C.(R_{se} + R_0)) & -1/(C.(R_{se} + R_0)) \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} .V_i$$

Para a obtenção das equações de estado para o circuito quando o transistor está desligado, devemos simplesmente observar que $V_i = 0$, sendo assim, temos $A_2 = A_1$ e $B_2 = 0$, logo a tensão de saída nos dois casos é dada por:

$$V_0 = R_0 \cdot (x_1 - C \cdot \dot{x}_2) = (R_0 \cdot R_{se} \cdot x_1 + R_0 \cdot x_2) / (R_{se} + R_0)$$

Logo:

$$C_1 = C_2 = \left[(R_0 \cdot R_{se}) / (R_0 + R_{se}) \quad R_0 / (R_0 + R_{se}) \right]$$

Então, tem-se a matriz e os vetores medianos representados por:

$$A = A_1$$

$$B = B_1 \cdot D$$

$$C = C_1$$

onde $A(t)$ é a matriz de estado, $B(t)$ é a matriz de entrada, $C(t)$ é a matriz de saída.

Realizando simplificações, onde normalmente R_0 é muito maior que R_{se} e, R_L e R_{se} são pequenos, temos:

$$A = A_1 = A_2 = \begin{bmatrix} (R_L \cdot R_{se}) / L & -1/L \\ 1/C & -1/(C \cdot R_0) \end{bmatrix}$$

$$B = B_1 \cdot D = \begin{bmatrix} 1/L \\ 0 \end{bmatrix}$$

$$C = C_1 = C_2 \approx \begin{bmatrix} R_{se} & 1 \end{bmatrix}$$

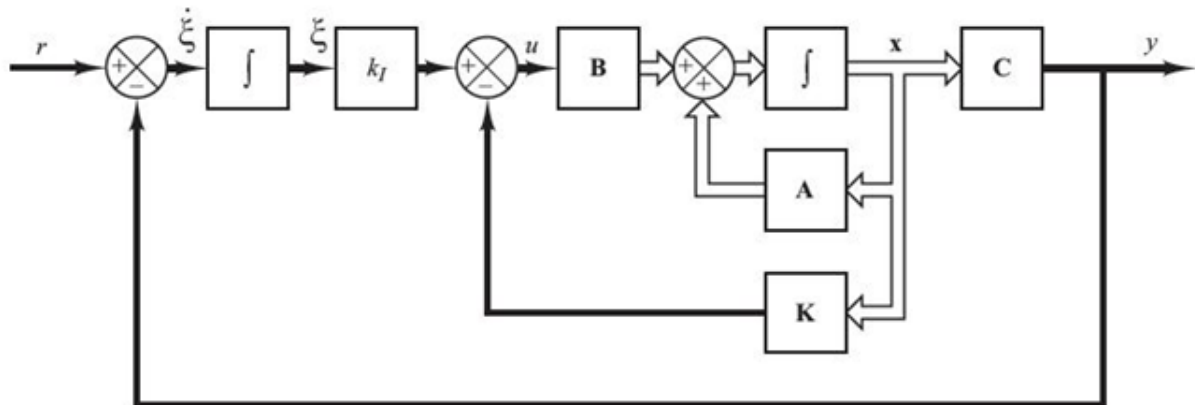
2.3 Controle por realimentação de estados

O controle por realimentação de estados consiste em realimentar todos os estados do sistemas, multiplicando o resultado por uma matriz K em vez de utilizar a saída do sistema como realimentação (MAXWELL, 2018).

Em um sistema de controle, a realimentação ocorre quando utiliza-se um sinal de saída como comparativo para o sinal de referência. Com o conhecimento da saída desejada, é possível gerar um sinal proporcional ao erro entre a resposta esperada e a resposta real (SILVA; PRADO, 2018).

A Figura 2.6 representa o diagrama de blocos de uma planta com o controle por realimentação de estados. Este sistema é conhecido como servossistemas do tipo 1, quando a planta não possui integrador (OGATA, 2011).

Figura 2.6 – Servossistema sem integrador (tipo 1).



Fonte: Ogata (2011)

Na Figura 2.6 é possível observar r como sendo um sinal de entrada, ligado diretamente em um somador. Este sinal passa por um integrador e aplica-se um ganho (k_i), sua saída está ligada a um somador com saída definida como sinal de controle u e como segunda entrada uma matriz de ganho K (OGATA, 2011).

Ainda na Figura 2.6 o sinal de controle passa por uma matriz de controle B e é inserida em outro somador que também tem como entrada a matriz de estado A . Sua resposta é integrada e a resposta é chamada de vetor de estado da planta x , este retorna como entrada para a matriz A e também para a matriz de ganhos K (OGATA, 2011)..

Para finalizar a explicação do servossistema na Figura 2.6, o vetor de estado x é inserido em uma matriz de saída C e sua resposta é chamada de sinal de saída y que também é inserida no somador contendo o sinal de entrada r , representando o erro do sistema (OGATA, 2011).

É possível projetar um servossistema do tipo 1 quando a planta não possui um integrador de forma que os polos de malha fechada se localizem nas posições desejadas. Assim, o sistema projetado será assintoticamente estável, logo a saída y tenderá ao valor de referência predefinido (OGATA, 2011).

2.4 Algoritmo TLBO

Para uma compreensão do algoritmo TLBO (*Teaching-Learning-Base Optimization*) é desejável a observação sobre o comportamento inteligente de algoritmos.

O comportamento inteligente vem das interações entre as unidades de processamento. A resposta desse algoritmo é a média ponderada dos sinais recebidos de cada unidade de pro-

cessamento, sabendo que esses sinais foram multiplicados cada um por seu peso ou relevância (CARVALHO P, 2009).

A otimização de sistemas é a área da pesquisa operacional que utiliza um método científico para apoiar a tomada de decisões, buscando opções melhores para projetar e operar um sistema (SOUZA, 2017).

Dentre os algoritmos atualmente estudados, foi dada ênfase ao método TLBO, que é um algoritmo que está baseado em interações professor – estudante em uma sala de aula. E este pode ser dividido em duas fases, a primeira seria a fase de aprendizado a partir do professor e a segunda seria a fase de aprendizado entre os estudantes.

À cada interação, na fase *professor* seleciona-se um professor, que pode ser considerado como o indivíduo com maior aptidão. Feita a escolha do professor, inicia-se a segunda fase, em que os estudantes aprendem com o professor escolhido, movendo em sua direção por meio do fator de aprendizado (que é calculado de forma aleatória com um valor entre 1 e 2). Na sequência, cada indivíduo passa a ser comparado com outro indivíduo escolhido de maneira aleatória, criando assim um novo indivíduo a fim de se mover na direção do melhor indivíduo.

Diversos trabalhos na literatura sugerem a utilização do algoritmo TLBO para a obtenção de resultados. Como exemplo é possível destacar alguns como: a sua utilização no treinamento de redes neurais artificiais para problemas de classificação (GARCIA CRISTIANO; CATALANO, 2018), sua utilização em problemas de otimização de projetos mecânicos restritos (RAO; SAVSANI; VAKHARIA, 2011) ou ainda em ajustes de parâmetros de controle de um PLL baseado na referência síncrona (SOUZA A. G. ; SILVA, 2018).

3 METODOLOGIA

Este capítulo trará toda a metodologia utilizada no trabalho, procurando explicitar como foram pensados os procedimentos e ensaios e quais pontos foram abordados.

Como é desejado implementar o algoritmo TLBO para a obtenção dos ganhos de controle de fontes chaveadas CC-CC do tipo BUCK, alguns pontos serão abordados para explicar quais os passos necessários.

Primeiramente é indispensável a análise do algoritmo TLBO, buscando a compreensão profunda de cada etapa de seu funcionamento. É desejável também explicitar o desenvolvimento do código no Matlab e também a implementação do circuito no *Simulink*.

Para finalizar, este capítulo buscará definir as faixas máximas para os ganhos k_i , k_1 e k_2 , fornecendo assim a base necessária para a sequência do desenvolvimento do trabalho.

3.1 Algoritmo TLBO

Os códigos utilizados neste trabalho foram baseados no algoritmo TLBO descrito nas Figuras 3.1 e 3.2 .

Figura 3.1 – Algoritmo TLBO parte 1.

Algoritmo 1: Teaching-Learning-based Optimization ((RAO; SAVSANI; VAKHARIA, 2011); (GARCIA; CATALANO; BARBOSA, 2018))

```

(1) Require: numDimensoes  $\leftarrow$  número de dimensões do problema
(2) Require: funcCusto  $\leftarrow$  função que calcula o custo que guiará cada indivíduo durante as iterações
(3) Require: numeroDeIndividuos  $\leftarrow$  quantidade de individuos presentes na população
(4) atribui um valor aleatório para a população pop e calcula o custo para cada indivíduo
(5) MelhorSolucao  $\leftarrow$  null
(6) for  $i \leftarrow 1$  : numeroIteracoes do
(7)   media  $\leftarrow$  mediadasposiesdosindividuos
(8)   selecionateacher() {atribui o melhor usuario}
(9)   {Começo da fase teacher}
(10)  for  $j \leftarrow 1$  : numeroDeIndividuos do
(11)    cria novoIndividuo
(12)     $TF \leftarrow \text{round}(1 + \text{rand}(0, 1))$  {TF = fator de aprendizagem}
(13)    novoIndividuo.Posicao  $\leftarrow$ 
      pop(j).Posicao + rand() * (teacher.Posicao - (TF * media))
(14)    novoIndividuo.Custo = funcCusto(novoIndividuo.Posicao)
(15)    if novoIndividuo.Custo < pop(j).Custo then
(16)      pop(j) = novoIndividuo;
(17)      if pop(j).Custo < melhorSolucao.Custo then
(18)        MelhorSolucao = pop(j);
(19)      end
(20)    end
(21)  end
(22) end

```

Fonte: Do autor (2021)

Figura 3.2 – Algoritmo TLBO parte 2.

```

(23)
(24) {Final da fase teacher}
(25) {Inicio da fase learner}
(26) for j ← 1 : numeroDeIndividuos do
(27)   k ←
      |   valorentraleavariavelnumeroDeIndividuoquesejadiferentedej
      |   if pop(k).Custo < pop(j).Custo then
(28)     |   passo ← pop(j).Posicao - pop(k).Posicao
(29)   end
(30)   else
(31)     |   passo ← pop(k).Posicao - pop(j).Posicao
(32)   end
(33)   cria novoIndividuo
(34)   novoIndividuo.Posicao ← pop(j).Posicao + rand() * passo
(35)   novoIndividuo.Custo = funcCusto(novoIndividuo.Posicao)
(36)   if novoIndividuo.Custo < pop(j).Custo then
(37)     |   pop(j) = novoIndividuo
(38)     |   if pop(j).Custo < melhorSolucao.Custo then
(39)       |   MelhorSolucao = pop(j);
(40)     end
(41)   end
(42) end
(43) end

```

Fonte: Do autor (2021)

De maneira geral, o algoritmo pode ser dividido em sete etapas, sendo elas: a definição das variáveis, inicialização, cálculo da média das posições e definição do *Professor*, fase *Professor*, fase *Aprendizado*, armazenamento da melhor solução e exposição da melhor solução. Cada etapa será detalhada profundamente no item 3.2.

No entanto, o algoritmo funciona da seguinte forma:

1. Define-se todos os limites de ganhos, número de iterações e número da população;
2. Inicialização das variáveis que serão utilizadas no algoritmo;
3. Calcula-se as médias das posições de cada indivíduo e define-se quem será o *Professor* neste momento;
4. Inicia-se a fase professor;
5. Faz-se uma aproximação (aprendizado) de todos os indivíduos para o *Professor*;
6. Armazena-se a melhor solução e caso o número de iterações ainda não seja atingido, retorna-se ao item 3;
7. finaliza-se o algoritmo mostrando a melhor solução encontrada.

3.2 Desenvolvimento do algoritmo TLBO no Matlab

O programa principal foi implementado no Matlab é apresentado nas Figuras 3.3, 3.4, 3.5, 3.6, 3.7, 3.8 e 3.9.

Após a definição da função custo, explicada no Item 3.3, é iniciado o código definindo o número de variáveis desconhecidas $nVar$, a variável $VarSize$ recebe o tamanho da matriz. A seguir é definido o limite inferior, $VarMin$, e superior, $VarMax$, dos ganhos desejados (k_i , k_1 e k_2). São definidos também dois parâmetros para o funcionamento do algoritmo, o primeiro é o limite para o número de iterações, $MaxIt$, e o segundo é o tamanho da população, $nPop$. Este processo é explicado na Figura 3.3.

Figura 3.3 – Definição das Variáveis.

```

1 %% Problem Definition
2
3 % Cost Function
4 CostFunction = @(x, plotfig) cost_func(x, plotfig);
5
6 nVar = 3;           % Number of Unknown Variables
7 VarSize = [1 nVar]; % Unknown Variables Matrix Size
8
9 VarMinki = 0;      % Unknown Variables Lower Bound
10 VarMaxki = 5;      % Unknown Variables Upper Bound
11 VarMink1 = 0;      % Unknown Variables Lower Bound
12 VarMaxk1 = 0.01;   % Unknown Variables Upper Bound
13 VarMink2 = 0;      % Unknown Variables Lower Bound
14 VarMaxk2 = 0.01;   % Unknown Variables Upper Bound
15
16 %% TLBO Parameters
17
18 MaxIt = 1000;      % Maximum Number of Iterations
19
20 nPop = 50;         % Population Size

```

Fonte: Do autor (2021)

Com as variáveis e parâmetros pré-definidos, passa-se agora a inicialização do algoritmo que é apresentada na Figura 3.4. São inicializados os vetores de posição e o custo para os indivíduos que foram chamados de: *empty_individual.Position* e *empty_individual.Cost*, inicia-se também o vetor de população, onde é atribuída uma repetição do vetor *empty_individual* para cada elemento da população.

O valor *infinito* é atribuído para o custo da melhor solução, passando para um *loop* que adiciona a cada membro da população um valor aleatório para a sua posição e um custo para essa posição. Este valor *infinito* é inserido para garantir que qualquer outra solução seja identificada como sendo uma melhor solução.

A cada iteração, caso esse custo seja menor que o custo da melhor solução, atribui-se esse elemento como melhor solução. Após percorrer todos os elementos da população cria-se uma matriz de melhores custos com zeros em todos os valores e com *MaxIt* igual ao número de linhas e apenas uma coluna. Este processo é explicado na Figura 3.4.

Figura 3.4 – Inicialização do Algoritmo.

```

21
22 %% Initialization
23
24 % Empty Structure for Individuals
25 empty_individual.Position = [];
26 empty_individual.Cost = [];
27
28 % Initialize Population Array
29 pop = repmat(empty_individual, nPop, 1);
30
31 % Initialize Best Solution
32 BestSol.Cost = inf;
33
34 % Initialize Population Members
35 for i = 1:nPop
36     pop(i).Position = [random('unif', VarMink1, VarMaxk1)...
37                       random('unif', VarMink1, VarMaxk1)...
38                       random('unif', VarMink2, VarMaxk2)];
39     pop(i).Cost = CostFunction(pop(i).Position, 0);%com zero, calcula custo e nao mostra grafico
40     disp(i);
41
42     if pop(i).Cost < BestSol.Cost
43         BestSol = pop(i);
44     end
45 end
46 init_pos = BestSol.Position;
47 init_cost = CostFunction(BestSol.Position, 1);
48
49 % Initialize Best Cost Record
50 BestCosts = zeros(MaxIt, 1);

```

Fonte: Do autor (2021)

Com todas as variáveis iniciadas, declaradas e com todos os parâmetros pré-definidos, o algoritmo segue para o *loop* principal. Ele será executado *MaxIt* vezes e será dividido em quatro partes para melhor compreensão.

Na primeira parte é calculada a média do valor da posição de todos os indivíduos e esse valor é inserido na variável *Mean*. Com a média calculada, é definido qual indivíduo será o professor. É atribuída à variável que representa o professor (*Teacher*) o elemento 1 e é realizada uma iteração entre todos os elementos. A cada iteração é verificado se o custo do elemento em questão é menor do que o custo do elemento destacado como professor, em caso positivo, é atribuído esse indivíduo como novo professor. Este processo é explicado na Figura 3.5.

Figura 3.5 – Cálculo da média das posições e definição do indivíduo Professor.

```

51
52 for it = 1:MaxIt
53
54     % Calculate Population Mean
55     Mean = 0;
56     for i = 1:nPop
57         Mean = Mean + pop(i).Position;
58     end
59     Mean = Mean/nPop;
60
61     % Select Teacher
62     Teacher = pop(1);
63     for i = 2:nPop
64         if pop(i).Cost < Teacher.Cost
65             Teacher = pop(i);
66         end
67     end

```

Fonte: Do autor (2021)

Com o professor definido, passa-se para a segunda parte do *loop* principal, que representa a fase *professor*.

Nesta fase é criado um *loop* que passará pelo número de indivíduos da população em questão. Inicialmente a cada iteração é definida uma nova solução e é atribuída uma solução individual vazia na variável *newsol*. Para o fator de ensino, é atribuído um valor aleatório entre 1 e 2, passando para a nova solução o conteúdo do professor, fazendo assim a sua validação. Caso alguma das iterações seja uma solução melhor, ela é atribuída como a melhor solução. A fase professor é explicada na Figura 3.6.

Figura 3.6 – Fase professor.

```

68 % Teacher Phase
69 for i = 1:nPop
70     % Create Empty Solution
71     newsol = empty_individual;
72
73     % Teaching Factor
74     TF = randi([1 2]);
75
76     % Teaching (moving towards teacher)
77     newsol.Position = pop(i).Position ...
78         + rand(VarSize).*(Teacher.Position - TF*Mean);
79
80     % Clipping
81     newsol.Position = max(newsol.Position, VarMin);
82     newsol.Position = min(newsol.Position, VarMax);
83
84     % Evaluation
85     newsol.Cost = CostFunction(newsol.Position);
86
87     % Comparison
88     if newsol.Cost < pop(i).Cost
89         pop(i) = newsol;
90         if pop(i).Cost < BestSol.Cost
91             BestSol = pop(i);
92         end
93     end
94 end

```

Fonte: Do autor (2021)

Terminada a fase correspondente ao professor, inicia-se a fase de aprendizado. De modo similar a fase anterior, é criado um *loop* que percorrerá todos os elementos contidos na população. Dentro deste *loop* cria-se uma matriz A com uma linha e o número de colunas igual ao número da população em questão e é atribuído o valor vazio para a célula que representa o elemento analisado em cada iteração.

É coletado um elemento aleatório e é inserido na variável J . É definida uma variável degrau (*Step*) que contém a diferença de posição entre o elemento analisado e o elemento aleatório contido na variável J . Caso o custo do elemento em J seja menor que o custo do elemento em análise, inverte-se o sinal da variável *Step*.

Passa-se para a aproximação do elemento em questão para o professor e é feita uma validação da mesma. Caso o custo da nova solução seja menor que o custo do elemento analisado, o elemento analisado recebe a nova solução, caso ainda esteja com a melhor solução possível, a variável *BestSol* é atualizada para a solução contida neste elemento. A fase aprendizado é explicada na Figura 3.7.

Na última fase do *loop* principal é realizado o registro da melhor solução e é mostrada no console. Também são mostrados os ganhos obtidos, finalizando o *loop* principal do algoritmo. Este processo é explicado no código na Figura 3.8.

Após finalizado o *loop* principal, os resultados finais são expostos em um gráfico logarítmico para a compreensão exata dos resultados, finalizando o algoritmo TLBO. Este processo é explicado na Figura 3.9.

3.3 Definição da função custo

É possível encontrar formulações matemáticas para índices de desempenho necessários para diversas aplicações. Em se tratando de malhas de controle, perturbações são comuns e podem provocar resultados indesejados. Existem alguns critérios utilizados para quantificar o erro em função de uma perturbação (CAMPOS; TEIXEIRA, 2010).

O IAE (*Integrated Absolut Error*) é um desses critérios. Ele tem uma desvantagem significativa, uma vez que ele exige a necessidade de se obter uma grande precisão. Isso ocorre por que esse critério é baseado em uma integral infinita, exigindo longos períodos de simulação para obter valores desejados. Como o tempo é importante para a análise realizada nesse trabalho, quanto mais rápido o sistema estabilizar, melhor será o conversor, este critério pode não ser o melhor dos casos, pois não leva o tempo em consideração.

Figura 3.7 – Fase de aprendizado.

```

95 % Learner Phase
96     for i = 1:nPop
97
98         A = 1:nPop;
99         A(i) = [];
100        j = A(randi(nPop-1));
101
102        Step = pop(i).Position - pop(j).Position;
103        if pop(j).Cost < pop(i).Cost
104            Step = -Step;
105        end
106
107        % Create Empty Solution
108        newsol = empty_individual;
109
110        % Teaching (moving towards teacher)
111        newsol.Position = pop(i).Position + rand(VarSize).*Step;
112
113        % Clipping
114        newsol.Position = max(newsol.Position, VarMin);
115        newsol.Position = min(newsol.Position, VarMax);
116
117        % Evaluation
118        newsol.Cost = CostFunction(newsol.Position);
119
120        % Comparision
121        if newsol.Cost < pop(i).Cost
122            pop(i) = newsol;
123            if pop(i).Cost < BestSol.Cost
124                BestSol = pop(i);
125            end
126        end
127    end

```

Fonte: Do autor (2021)

Figura 3.8 – Armazenamento da melhor solução.

```

128 % Store Record for Current Iteration
129     BestCosts(it) = BestSol.Cost;
130
131 % Show Iteration Information
132     disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCosts(it))]);
133     ki
134     k1
135     k2
136
137 end

```

Fonte: Do autor (2021)

$$IAE = \int_0^{\infty} |e(t)| dt \quad (3.1)$$

Seguindo esta linha, é possível utilizar, em processos não oscilatórios, IE (*Integrated Error*). Esse critério é semelhante ao anterior, com uma boa resposta em sistemas amortecidos. Entretanto, assim como o IAE, não tem o tempo como variável.

Figura 3.9 – Gráfico logarítmico com os resultados finais.

```

138 %% Results
139
140 figure;
141 %plot(BestCosts, 'LineWidth', 2);
142 semilogy(BestCosts, 'LineWidth', 2);
143 xlabel('Iteration');
144 ylabel('Best Cost');
145 grid on;

```

Fonte: Do autor (2021)

$$IE = \int_0^{\infty} e(t) dt \quad (3.2)$$

O erro quadrático, ou ISE (*Integrated Square Error*), é de fácil implementação. Contudo também não possui o tempo como variável.

$$ISE = \int_0^{\infty} [e(t)]^2 dt \quad (3.3)$$

O ITAE (*Integrated of the Time multiplied by Absolute Error*) pode ser uma opção, pois ele é a integral do tempo multiplicado pelo erro absoluto.

$$ITAE = \int_0^{\infty} t \cdot |e(t)| dt \quad (3.4)$$

Entretanto, como é desejado que o erro tenha a maior relevância possível, o ITSE (*Integrated by the Time multiplied by Square Error*) se mostra como a melhor seletividade, uma vez que ele possui o tempo como variável e multiplica o erro quadrático. Serão propostos alguns ensaios com todos os critérios para efeito comparativo.

$$ITSE = \int_0^{\infty} t \cdot e^2(t) dt \quad (3.5)$$

A função custo, observada na Figura 3.10, começa atribuindo os valores $k(1)$, $k(2)$ e $k(3)$ as variáveis k_i , k_1 e k_2 na área de trabalho base do Matlab. Posteriormente é iniciada a

simulação do conversor “Buck” e coleta-se a variável erro, o seu tamanho é armazenado no vetor $[n, \sim]$, o *til* é inserido para ignorar saídas indesejadas e é atribuído o valor zero a variável *cost_value*.

É então realizado um *loop* de n vezes, o custo é calculado de acordo com a função escolhida que é detalhada no item 3.3 deste trabalho. Ao fim, caso exista uma plotagem (gráfico) na variável *plotfig*, a variável *plt* recebe a *figura*(100) para que seja mostrada. Nela está contida a resposta do conversor (tensão de saída) ao longo de toda a simulação.

Figura 3.10 – Função custo.

```

1
2 function cost_value=cost_func(k, plotfig)
3
4 assignin('base', 'ki', k(1))
5 assignin('base', 'k1', k(2))
6 assignin('base', 'k2', k(3))
7
8 sim('Buck.slx')
9
10 erro=(errosaida);
11 [n,~]=size(erro);
12 cost_value=0;
13
14 for i=1:n
15 % cost_value=cost_value+(erro(i))^2; % ISE
16 % cost_value=cost_value+abs(erro(i)); % IAE
17 % cost_value=cost_value+t(i)*abs(erro(i)); % ITAE tempo importante
18 % (resposta mais rapida)
19 cost_value=cost_value+t(i)*((erro(i))^2); % ITSE
20 end
21
22 if plotfig
23 plt = figure(100);
24 plot(Vsaida, 'LineWidth', 1.2)
25 grid on
26 ylim([-50 200]);
27 end
28 end

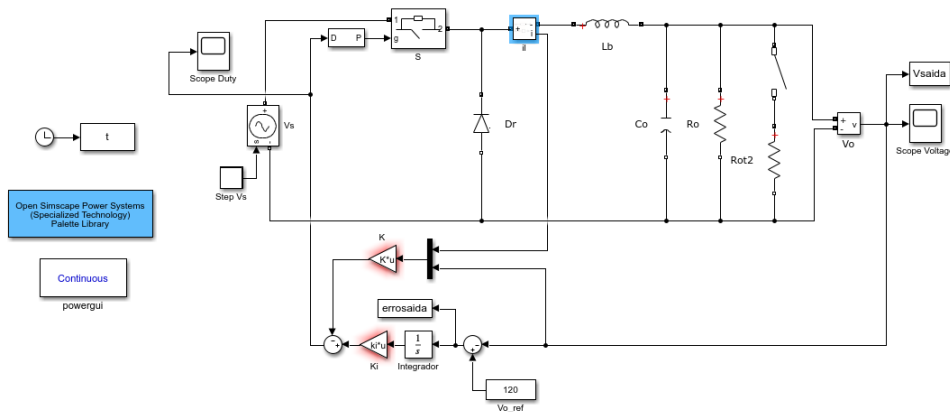
```

Fonte: Do autor (2021)

3.4 Implementação do conversor Buck no *Simulink*

Para que seja possível observar todos os resultados, foi desenvolvido no *Simulink* um conversor do tipo Buck, apresentando o controle de tensão de saída utilizando a técnica de realimentação de estados, conforme mostrado na Figura 3.11.

No conversor é possível alterar os ganhos e observar a saída e o ciclo de trabalho do conversor (*Duty Cycle*). Assim, torna-se possível a análise da resposta do conversor para qualquer ganho inserido. Além disso, para avaliar o desempenho do controle, foram inseridos alguns distúrbios: degrau de subida de tensão, degrau de descida de tensão, ambos com variação de 5% em relação a tensão de inserida no *Step*, que foi de 400V e inserção de carga no sistema, esta carga esta representada por uma resistência de 75 Ω .

Figura 3.11 – Conversor Buck no *Simulink*.

Fonte: Do autor (2021)

O gerador de *Step* (*StepVs*) está ligado em uma fonte de tensão controlada e uma chave ideal (*S*), o gerador opera inicialmente com 400V, após 0,1s ele passa para 420V e no instante 0,3s ele retorna para 400V.

Como a chave (*S*) recebe também o sinal gerado pelo *PWMGenerator*, sua saída varia de acordo com sua entrada. Caso o *PWMGenerator* proporcione uma entrada igual a 0,5, sua saída será 50% da tensão recebida e assim por diante.

É inserido um medidor de corrente em série com a chave ideal (*S*). Sua resposta está ligada em um somador e sua saída diretamente ligada a um indutor com valor de 0,001H. A saída do indutor está ligada a um capacitor, um resistor (carga inicial com valor de 9,6 Ω) e uma chave. Quando a chave, que está em série com o resistor *Rot2* é acionada, este resistor atua adicionando uma nova carga ao sistema com valor de 75 Ω.

Um medidor de tensão foi inserido na saída do conversor, que pode ser analisada graficamente em um osciloscópio (*Scope*) e simultaneamente este valor é registrado na variável *Vsaída*. A tensão de saída retroalimenta o sistema ligada em um somador e um multiplexador, o somador calcula a diferença entre a tensão de referência e a tensão medida (variável “erro-saída”) e a insere em um integrador. O multiplexador combina o sinal de saída com o sinal recebido como resultado do medidor de corrente e o resultado é multiplicado pelo ganho *k1* e *k2*.

A resposta do erro de saída após passar por um integrador multiplicada por *ki* menos o valor obtido pela multiplicação de *k1* e *k2* pelo resultado da multiplexação é enviada ao controlador PWM para que o ciclo recomece.

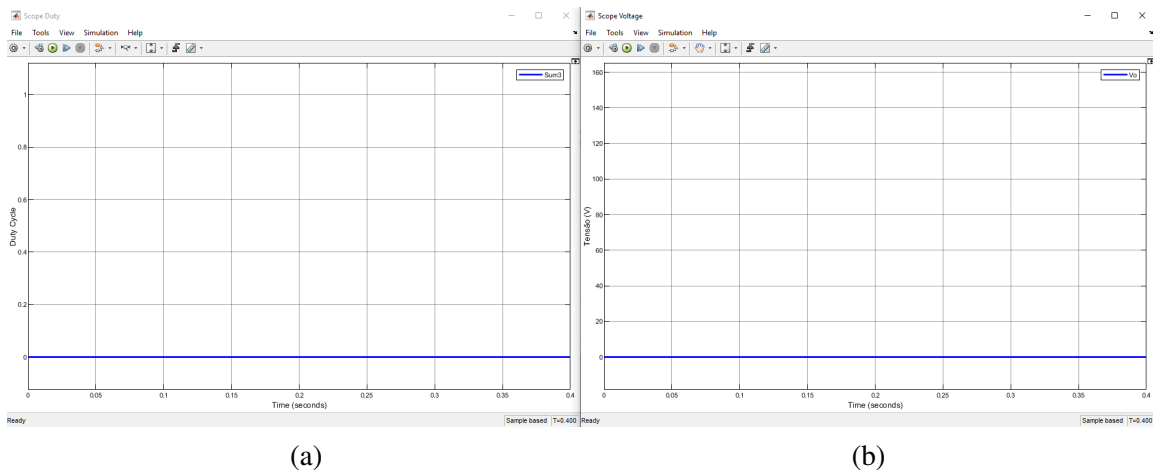
3.5 Definição das faixas de ganhos a serem utilizadas no algoritmo TLBO

Para o funcionamento do algoritmo TLBO, são necessários os parâmetros: limites máximos e mínimos dos ganhos, número de variáveis desejadas, número máximo de iterações e o número da população. Com o intuito de melhor direcionar o algoritmo na obtenção de resultados mais precisos, foram realizados alguns estudos na implementação, que são descrita na item 3.4.

Sabe-se que cada ganho pode influenciar de maneira independente no conversor de acordo com os valores inseridos. Para esta análise foram realizados testes alterando os ganhos e observando a resposta da tensão de saída e saturação do ciclo de trabalho (“Duty Cycle”). É importante a observação da saturação do ciclo de trabalho, pois esta saturação indica que o conversor não tem mais poder de atuação e assim não conseguirá atingir o resultado esperado.

Como primeiro ensaio todos os ganhos foram definidos com valor zero. Com os ganhos iguais a zero, o ciclo de trabalho será zero e a fonte chaveada ficará desligada. A resposta gráfica do ciclo de trabalho e da tensão podem ser observadas na Figura 3.12.

Figura 3.12 – Resposta da simulação para ganhos iguais a zero. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



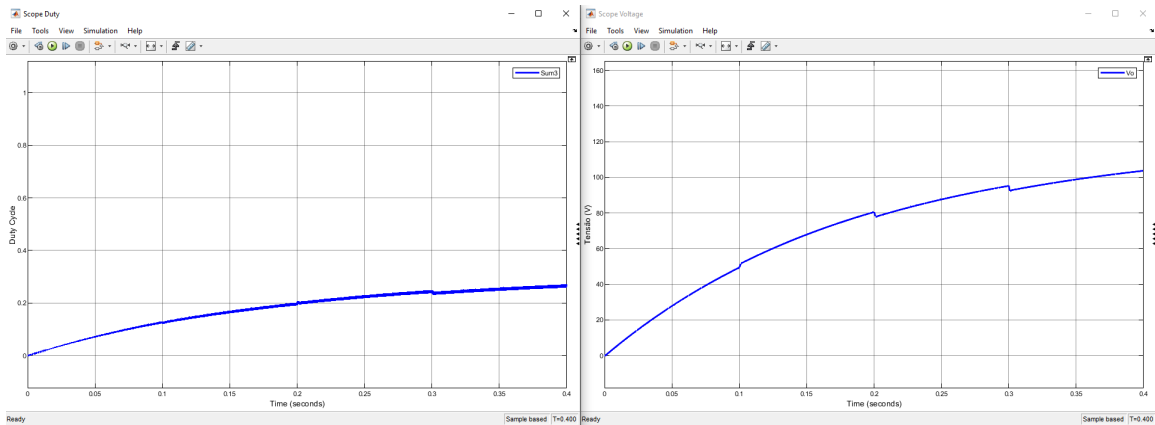
Fonte: Do autor (2021)

Como é desejado encontrar uma faixa de valor para cada ganho, foi adicionado um valor igual a 0,01 para os ganhos, proporcionando assim a influência dos ganhos na resposta. A resposta gráfica deste ensaio do ciclo de trabalho e da tensão podem ser observadas na Figura 3.13.

É possível observar ainda que o tempo de subida está extremamente alto e o valor de referência não foi atingido. Portanto, os ganhos tem relevância no resultado final e não podem

ser iguais a zero e devem ser observados com a devida atenção para que resultados satisfatórios sejam obtidos.

Figura 3.13 – Resposta da simulação para ganhos iguais a 0,01. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

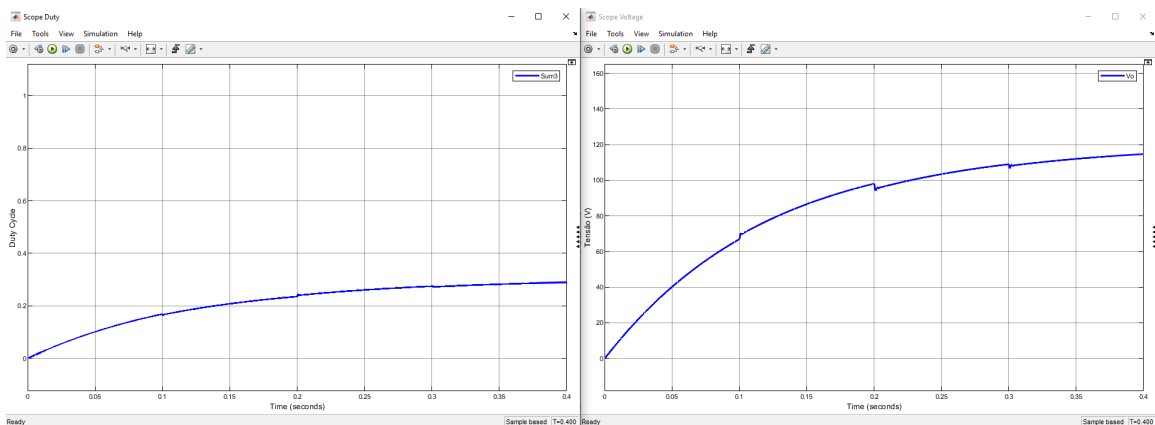
(b)

Fonte: Do autor (2021)

3.5.1 Definição do limite máximo para o ganho do integrador (k_i)

Neste momento, k_1 e k_2 foram mantidos em 0,01 e o ganho referente ao integrador foi aumentado para 0,05, a resposta a este ensaio pode ser observada na Figura 3.14.

Figura 3.14 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 0,05. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



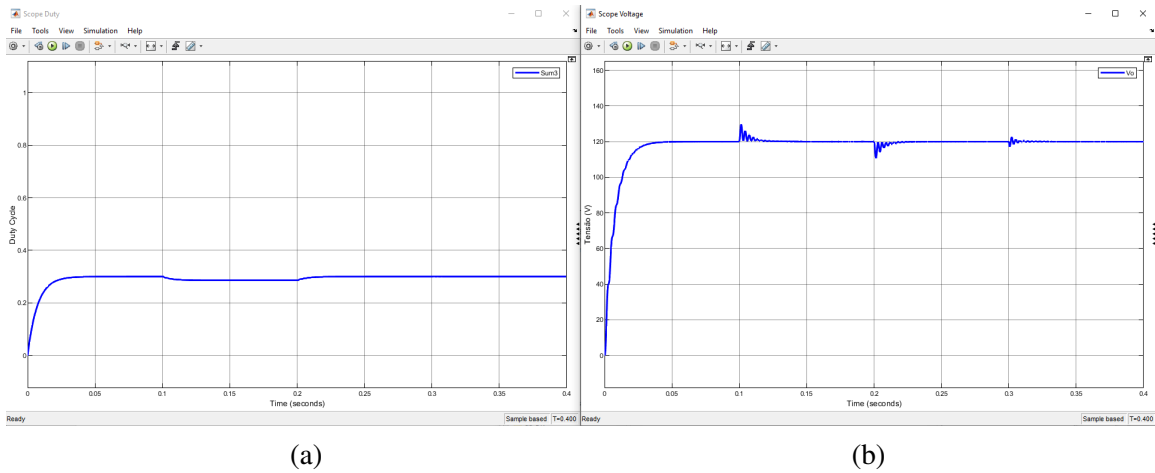
(a)

(b)

Fonte: Do autor (2021)

O ciclo de trabalho teve pouca alteração e o tempo de subida continua alto, porém foi possível observar uma melhora. Os ganhos k_1 e k_2 foram mantidos em 0,01 e o ganho referente ao integrador (k_i) foi novamente aumentado, agora para 0,5, a resposta a este ensaio pode ser observada na Figura 3.15.

Figura 3.15 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 0.05. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.

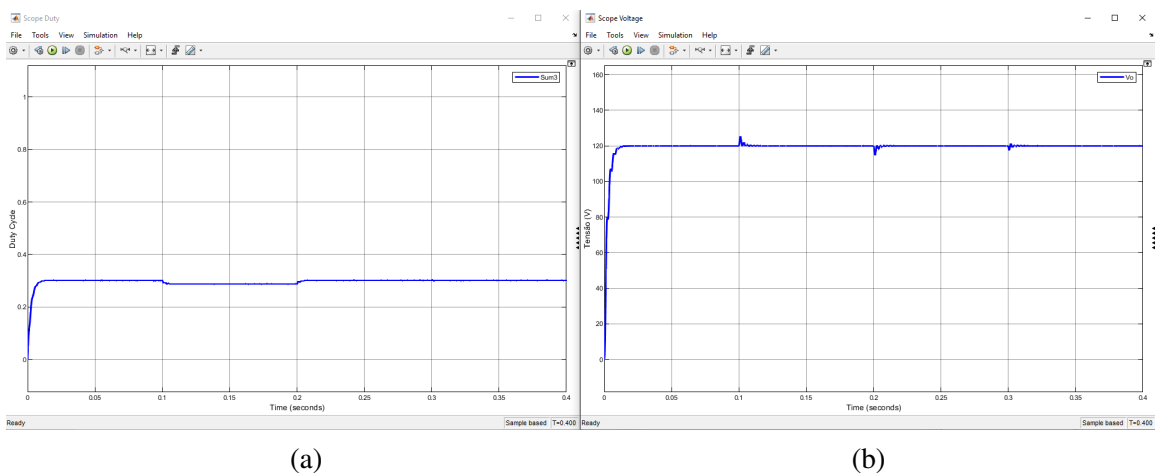


Fonte: Do autor (2021)

O ciclo de trabalho se estabilizou no valor esperado para promover o abaixamento de tensão para o valor desejado e pela primeira vez o conversor atingiu a tensão de referência, com um tempo de subida próximo aos $15ms$ e sobressinal igual a $9,8V$.

Para compreender melhor os resultados, os ganhos k_1 e k_2 foram mantidos novamente em 0,01 e o ganho referente ao integrador (k_i) foi novamente aumentado, agora para 1, a resposta a este ensaio pode ser observada na Figura 3.16.

Figura 3.16 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 1. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.

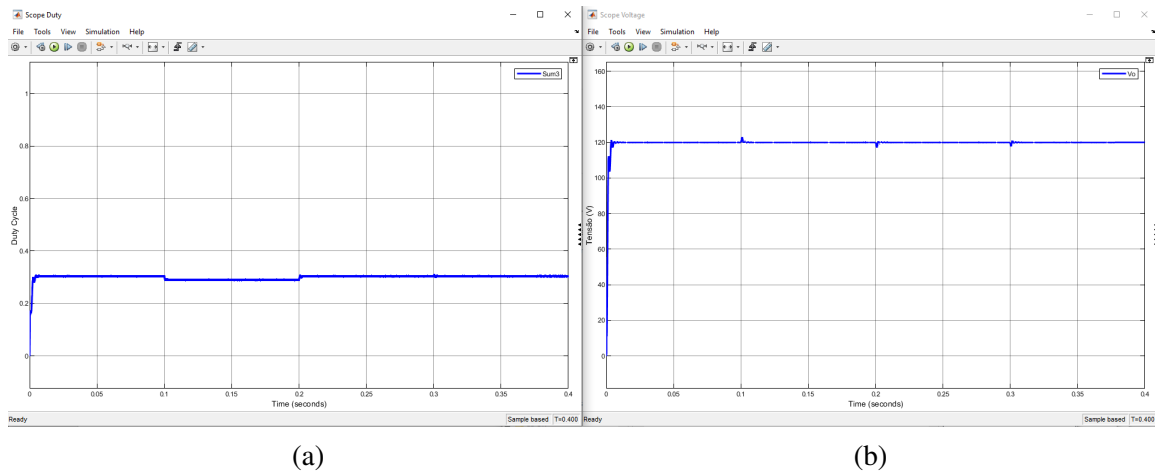


Fonte: Do autor (2021)

Com o aumento de k_i , o tempo de subida caiu para cerca de $5ms$ e o sobressinal caiu para $5,4V$.

Como existe uma indicação de possível melhora na resposta, o ganho k_i foi aumentado para 5 e os ganhos k_1 e k_2 serão mantidos em 0,01. A resposta para este ensaio estão demonstrados na Figura 3.17.

Figura 3.17 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

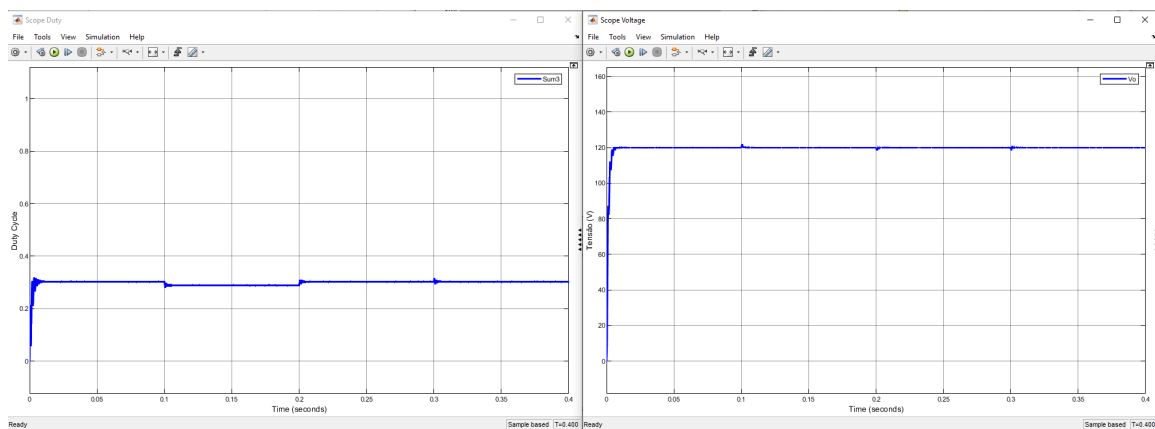
(b)

Fonte: Do autor (2021)

Com o novo aumento de k_i , o tempo de subida caiu para cerca de 1ms e o sobressinal caiu para 2,8V.

Novamente existe uma possibilidade de melhora na resposta, portanto o ganho “ k_i ” foi aumentado para 10 e os ganhos k_1 e k_2 serão mantidos em 0,01. A resposta para este ensaio estão demonstrados na Figura 3.18.

Figura 3.18 – Resposta da simulação para ganhos k_1 e k_2 iguais a 0,01 e k_i igual a 10. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

(b)

Fonte: Do autor (2021)

Com o novo aumento de k_i , o tempo de subida subiu para cerca de 2ms e o sobressinal caiu para 1,8V. Os resultados indicam que variar k_i de 5 para 10 não implica em uma melhora

significativa. Portanto, neste momento k_i máximo foi fixado em 5, uma vez que esse valor proporcionou um tempo de subida menor.

Para melhor entendimento dos testes realizados até o momento para definição do limite máximo de k_i , serão apresentados os resultados de forma numérica na Tabela 3.1.

Tabela 3.1 – Descrição dos testes realizados para definição dos limites máximos do ganho k_i .

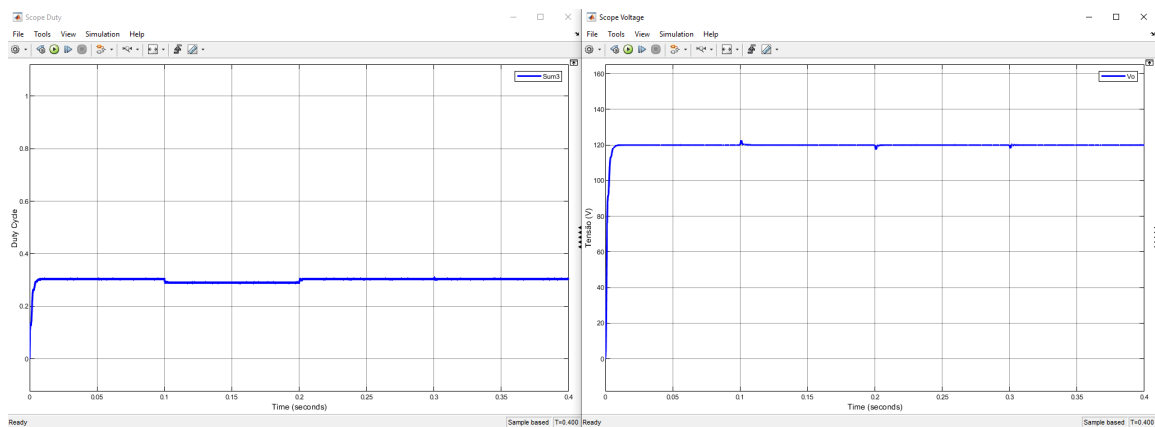
k_1	k_2	k_i	Tempo de Subida (ms)	Duty Cycle Máximo	Sobre Sinal (V)
0,01	0,01	0.05	<i>Não Atingiu</i>	0,2910	0
0,01	0,01	0.5	15,558	0,3009	9,8
0,01	0,01	1	5,233	0,3046	5,4
0,01	0,01	5	1,051	0,3113	2,8
0,01	0,01	10	2,015	0,3182	1,8

Fonte: Do autor (2021)

3.5.2 Definição do limite máximo para o ganho k_1

Com o intuito de identificar a atuação do ganho k_1 na resposta do sistema, foi adotada a mesma metodologia seguida com o ganho k_i . Portanto, inicialmente, o ganho k_i foi mantido com seu ganho máximo encontrado no item anterior, k_2 foi mantido em 0,01 e k_1 será alterado para 0,05. O resultado para este ensaio pode ser observado na Figura 3.19.

Figura 3.19 – Resposta da simulação para ganhos k_1 igual a 0,05, k_2 igual a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



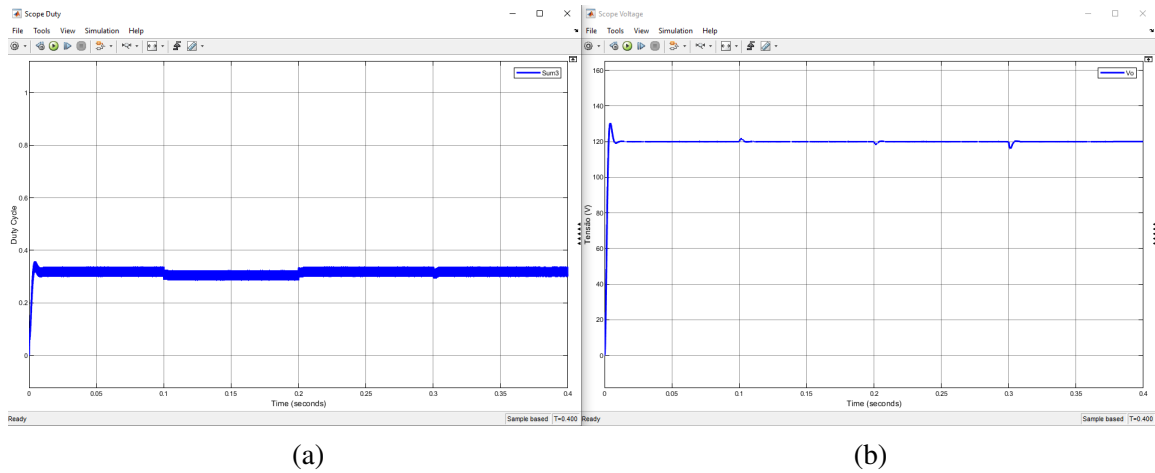
(a)

(b)

Fonte: Do autor (2021)

Comparando com a resposta obtida com $K_i = 5$, $k_1 = k_2 = 0,01$, foi possível observar que o tempo de subida cresceu de 1ms para 2,7ms e o sobressinal se manteve praticamente estável. Sendo assim, foi realizado ensaio com o valor máximo de $k_1 = 0,025$. A resposta para esta situação está na Figura 3.20.

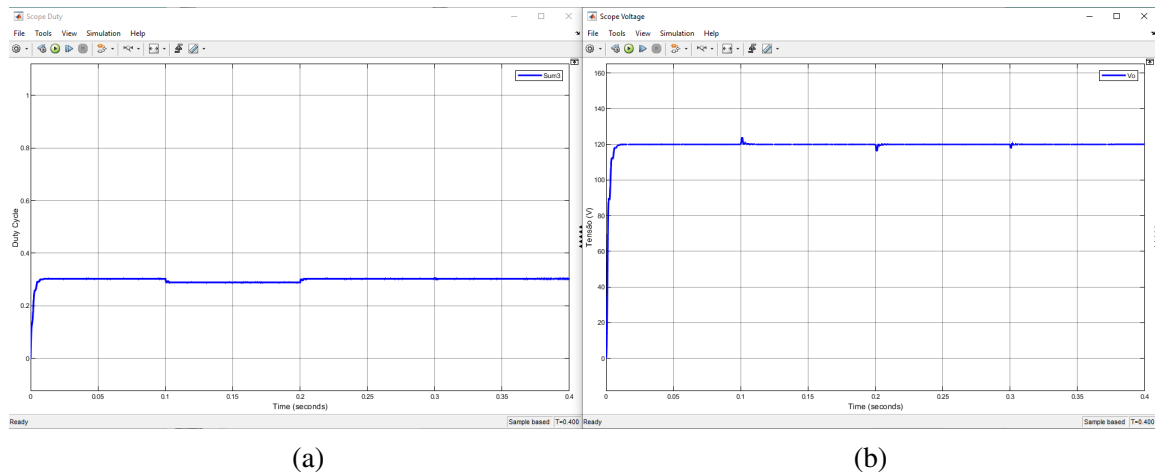
Figura 3.20 – Resposta da simulação para ganhos k_1 igual a 0,025, k_2 igual a 0.01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



Fonte: Do autor (2021)

Comparando novamente com a resposta obtida com $K_i = 5$, $k_1 = k_2 = 0,01$, foi possível observar que o tempo de subida cresceu de $1ms$ para $2ms$. Sendo assim, será feita uma diminuição no valor máximo de k_1 para 0,005. A resposta para estes valores pode ser observada na Figura 3.21.

Figura 3.21 – Resposta da simulação para ganhos k_1 igual a 0,005, k_2 igual a 0,01 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



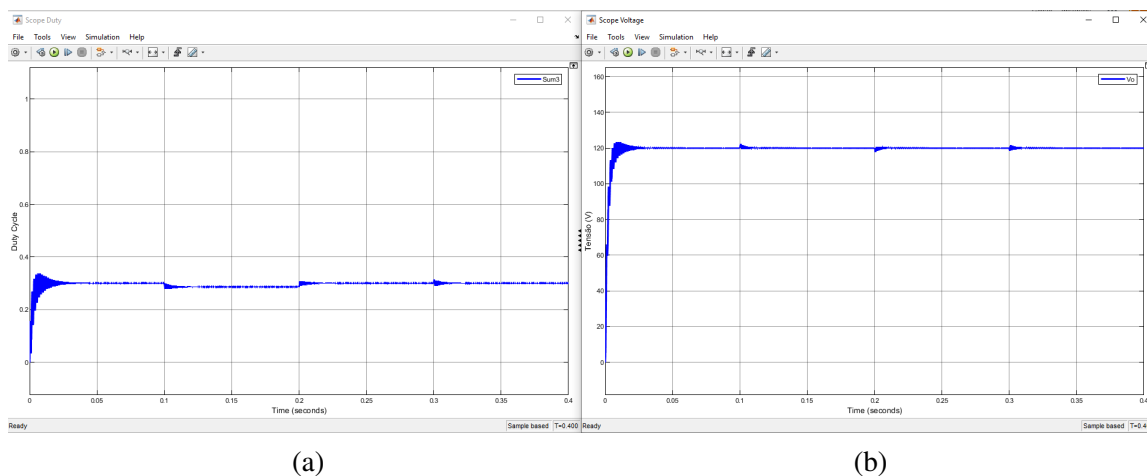
Fonte: Do autor (2021)

Com o valor máximo de $k_1 = 0,005$, o tempo de subida ficou ainda maior que no ensaio anterior e ainda o sobressinal subiu para 3,9V.

Foi proposto agora novo ensaio com um valor máximo de $k_1 = 0,0075$. Sua resposta pode ser observada na Figura 3.22.

Desta forma, com o valor máximo de $k_1 = 0,0075$, o tempo de subida foi de aproximadamente $3,4ms$ e o sobressinal subiu para 3,4V.

Figura 3.22 – Resposta da simulação para ganhos k_1 igual a 0,0075, k_2 igual a 0,01 e k_i igual a 5, A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



Fonte: Do autor (2021)

Para melhor entendimento dos testes realizados até o momento para definição do limite máximo de k_1 , serão apresentados os resultados de forma numérica na Tabela 3.2, incluindo o teste realizado anteriormente com $k_1 = 0,01$.

Tabela 3.2 – Descrição dos testes realizados para definição dos limites máximos do ganho k_1 .

k_1	k_2	k_i	Tempo de Subida (ms)	Duty Cycle Máximo	Sobre Sinal (V)
0,0075	0,01	5	3,396	0,3377	3,4
0,005	0,01	5	3,107	0,3075	3,9
0,01	0,01	5	1,051	0,3113	2,8
0,025	0,01	5	1,983	0,3587	10,03
0,05	0,01	5	2,715	0,3009	2,7

Fonte: Do autor (2021)

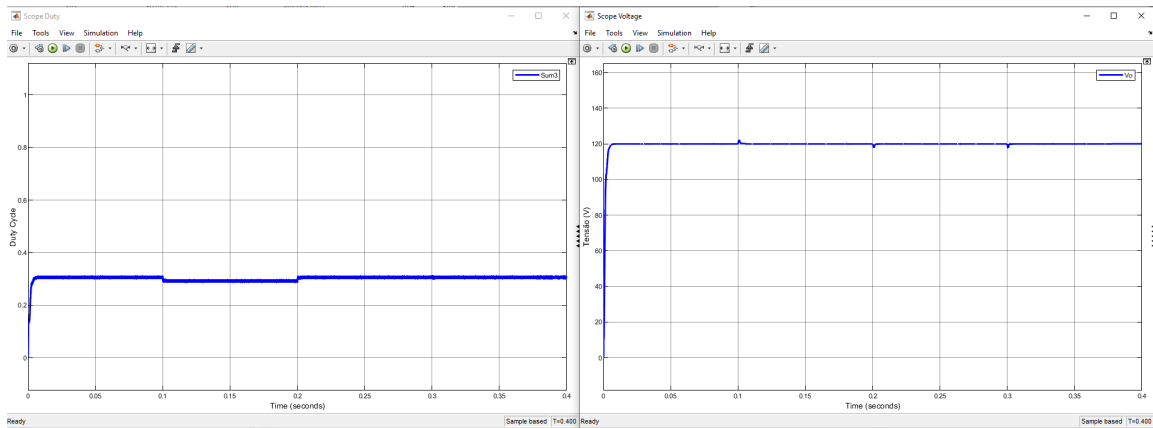
É possível concluir que k_1 deve ser próximo de 0,01, uma vez que qualquer outro valor inserido proporcionou resultados piores. Sempre que o valor de k_1 ficou próximo a 0,01 o conversor teve resultados melhores com relação ao tempo de subida e ao sobressinal. É possível concluir que k_1 deve ser próximo de 0,01, uma vez que qualquer outro valor inserido proporcionou resultados piores. Sempre que o valor de k_1 ficou próximo a 0,01 o conversor teve resultados melhores com relação ao tempo de subida e ao sobressinal.

3.5.3 Definição do limite máximo para o ganho k_2 .

Com o intuito de identificar a atuação do ganho k_2 na resposta do sistema, foi usada a mesma metodologia seguida com os ganhos k_i e k_1 . Logo, inicialmente, foram mantidos os

ganhos encontrados anteriormente ($k_i = 5$ e $k_1 = 0,01$). O ganho k_2 será alterado para $0,05$). O resultado para este ensaio pode ser observado na Figura 3.23.

Figura 3.23 – Resposta da simulação para ganhos k_1 igual a $0,01$, k_2 igual a $0,05$ e k_i igual a 5 . A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



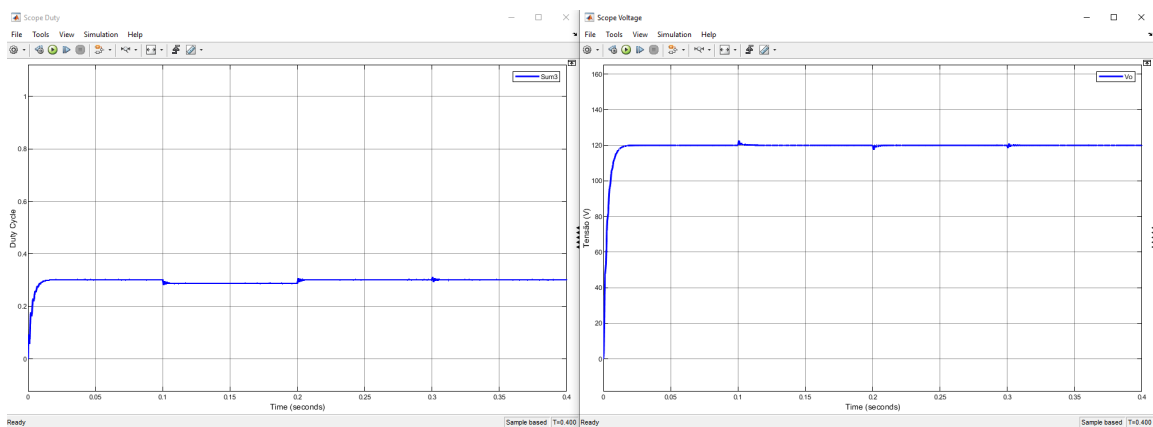
(a)

(b)

Fonte: Do autor (2021)

Comparando com a resposta obtida com $K_i = 5$, $k_1 = k_2 = 0,01$, foi possível observar que o tempo de subida cresceu de $1ms$ para $2,3ms$ e o sobressinal sofreu pouca alteração. Sendo assim, foi feita uma diminuição no valor máximo de k_2 , em relação ao item anterior, para $0,025$. A resposta para estes valores pode ser observada na figura 3.24.

Figura 3.24 – Resposta da simulação para ganhos k_1 igual a $0,01$, k_2 igual a $0,025$ e k_i igual a 5 . A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



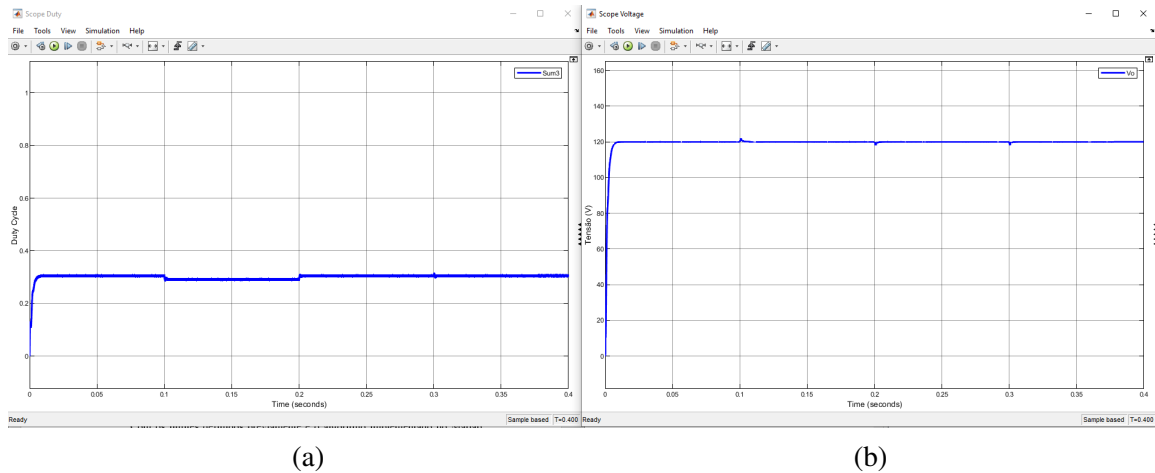
(a)

(b)

Fonte: Do autor (2021)

O tempo de subida ficou acima de $6ms$, indicando que k_2 possa ter limite máximo igual a $0,01$ ou menor. Será proposta uma alteração no limite máximo de k_2 para $0,005$. A resposta para este caso está na Figura 3.25.

Figura 3.25 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,005 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



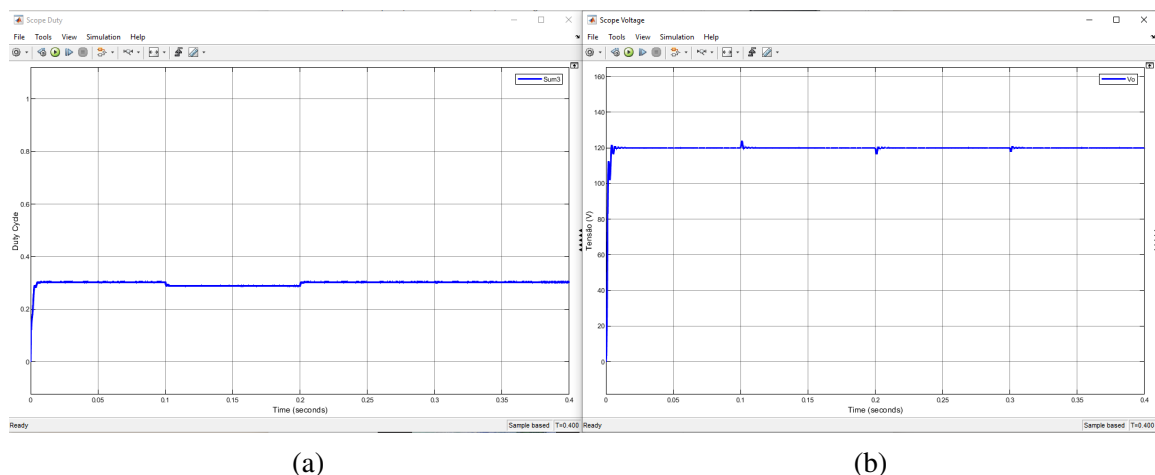
(a)

(b)

Fonte: Do autor (2021)

Observa-se que o tempo de subida ficou mais alto que no resultado obtido com $k_2 = 0,01$. Contudo, o sobressinal ficou abaixo, podendo indicar que limites mais baixos de k_2 possam gerar resultados mais satisfatórios. Para melhor análise, abaixo é apresentado o teste realizado com o limite máximo de k_2 igual a 0,001 e a sua resposta pode ser observada na Figura 3.26.

Figura 3.26 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,001 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



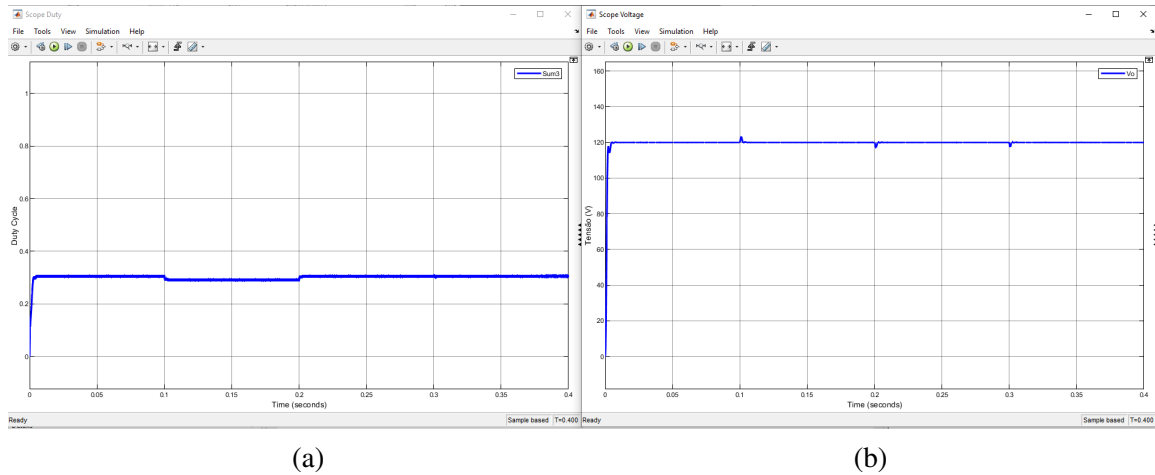
(a)

(b)

Fonte: Do autor (2021)

Neste caso, o tempo de subida ficou muito próximo ao que foi obtido com $k_2 = 0,01$ e o sobressinal ficou um pouco acima do que foi obtido no resultado anterior. Podendo indicar que o limite de k_2 possa estar entre 0,001 e 0,005. Como ferramenta de análise, abaixo serão apresentados os resultados para os testes com k_2 igual a 0,002, 0,003 e 0,004. Na Figura 3.27 abaixo é possível observar os resultados obtidos com $k_2 = 0,002$.

Figura 3.27 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,002 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



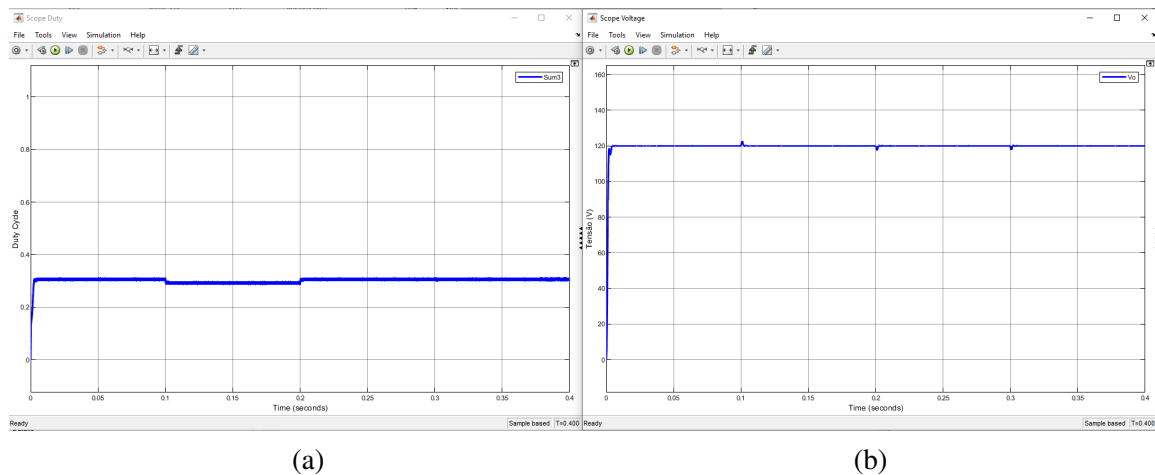
(a)

(b)

Fonte: Do autor (2021)

Com $k_2 = 0,002$, o tempo de subida teve pouca variação em relação ao teste anterior, contudo houve uma redução no sobressinal. Será apresentado na Figura 3.28 abaixo os resultados obtidos com $k_2 = 0,003$.

Figura 3.28 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,003 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

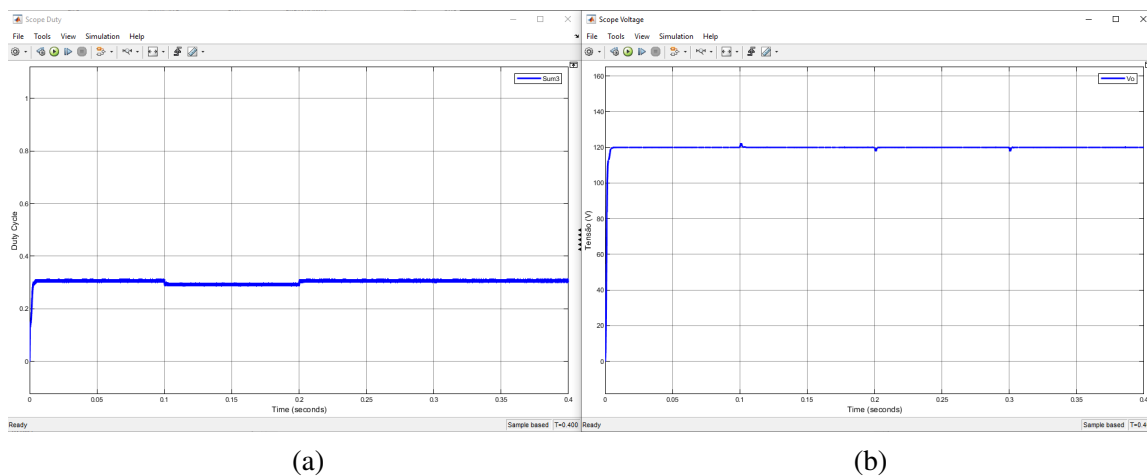
(b)

Fonte: Do autor (2021)

Agora com $k_2 = 0,003$, o tempo de subida teve pouca variação em relação ao teste anterior. É possível identificar uma redução no sobressinal. É apresentado na Figura 3.29 os resultados obtidos com $k_2 = 0,004$.

Já com $k_2 = 0,004$, o tempo de subida ficou um pouco maior que no teste anterior e o sobressinal se manteve estável. Isso indica que, possivelmente, o limite de k_2 como 0,003 proporciona melhores resultados. Para melhor entendimento dos testes realizados até o momento

Figura 3.29 – Resposta da simulação para ganhos k_1 igual a 0,01, k_2 igual a 0,004 e k_i igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



Fonte: Do autor (2021)

para definição do limite máximo de k_2 , são apresentados os resultados de forma numérica na Tabela 3.3 abaixo, incluindo o teste realizado anteriormente com $k_2 = 0,01$.

Tabela 3.3 – Descrição dos testes realizados para definição dos limites máximos do ganho k_2 .

k_1	k_2	k_i	Tempo de Subida (ms)	Duty Cycle Máximo	Sobre Sinal (V)
0,01	0,001	5	1,168	0,3080	3,8
0,01	0,002	5	1,239	0,3112	3,1
0,01	0,003	5	1,220	0,3155	2,23
0,01	0,004	5	1,369	0,3152	2,23
0,01	0,005	5	2,992	0,3155	2,2
0,01	0,01	5	1,051	0,3113	2,8
0,01	0,025	5	6,319	0,3121	2,5
0,01	0,05	5	2,365	0,3150	2,2

Fonte: Do autor (2021)

3.5.4 Apresentação dos limites máximos dos ganhos obtidos nos itens anteriores.

Com os valores obtidos anteriormente, é possível destacar que o melhor resultado foi obtido com os ganhos máximos iguais a:

$$k_1 = 0,01$$

$$k_2 = 0,003$$

$$k_i = 5$$

Como não é possível ciclos de trabalho negativos, define-se que todos os ganhos mínimos sejam zero.

4 RESULTADOS E DISCUSSÕES

Este item irá demonstrar os resultados e discussões refletidos na metodologia. Neste momento, o algoritmo consegue produzir resultados satisfatórios, mas é desejado um aprofundamento maior nestas respostas para que se tenha certeza dos resultados. Por este motivo este item começará realizando uma análise da interferência do número de iterações e do tamanho da população no resultado final.

Posteriormente, com estas análises, serão realizadas tentativas de limitações maiores dos limites dos ganhos, afim de obter o melhor resultado possível com o algoritmo.

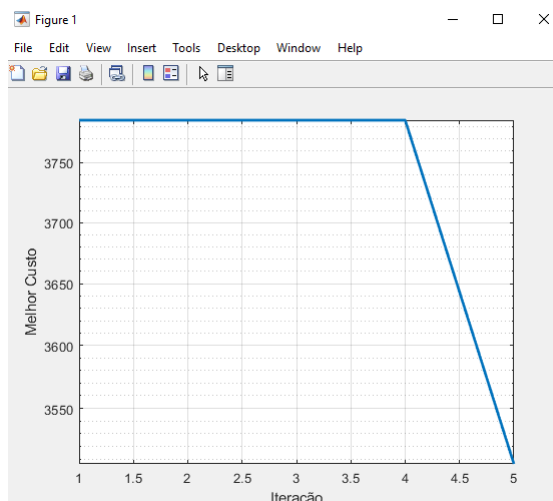
4.1 Análise da interferência do número máximo de iterações na resposta do algoritmo.

Com os limites definidos previamente e o algoritmo implementado no Matlab, foram realizadas algumas simulações para observar a interferência do número máximo de iterações na resposta final do algoritmo. Em todas elas o tamanho da população foi mantido em 5 apenas para agilizar os ensaios. O efeito do tamanho da população será apresentado no item 4.2 deste trabalho.

No primeiro momento o número de máximo iterações foi definido como 1, em seguida foi definido com 5, 10 e 15 iterações.

Para um melhor entendimento da evolução do custo nos testes, apresenta-se o gráfico do custo com relação ao número de iterações. A Figura 4.1 apresenta a evolução para o ensaio com 5 iterações.

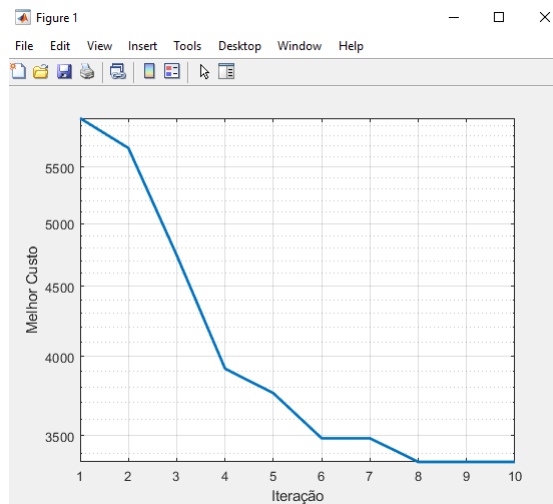
Figura 4.1 – Evolução do custo em relação ao número de iterações com o número máximo de iterações igual a 5.



Fonte: Do autor (2021)

Na Figura 4.1 é possível observar que o custo se manteve estável até a quarta iteração e na quinta iteração houve uma queda. Para melhor analisar a extensão da queda no custo, a Figura 4.2 apresenta a evolução para o ensaio com 10 iterações.

Figura 4.2 – Evolução do custo em relação ao número de iterações com o número máximo de iterações igual a 10.

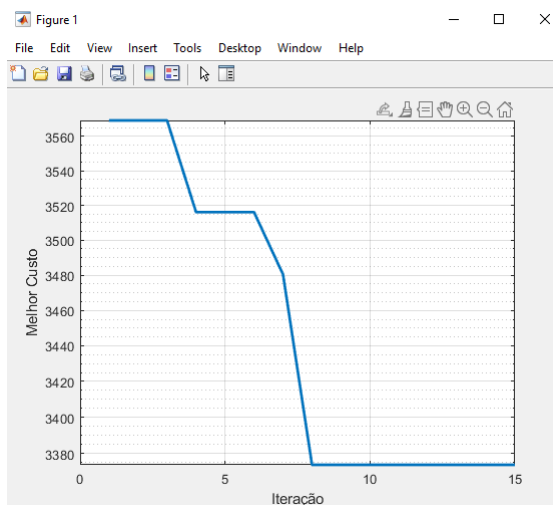


Fonte: Do autor (2021)

Neste ensaio o custo caiu até a oitava iteração, porém houve um comportamento diferente do observado anteriormente. Portanto, entende-se que um ensaio com um número maior de iterações é necessário.

A Figura 4.3 apresenta a evolução para o ensaio com 15 iterações.

Figura 4.3 – Evolução do custo em relação ao número de iterações com o número máximo de iterações igual a 15.



Fonte: Do autor (2021)

É possível observar que o custo de manteve estável após a oitava iteração, idêntico ao observado no ensaio com 10 iterações.

A Tabela 4.1 apresenta os resultados dos ensaios realizados com o número de população igual a 5 e variando o número de iterações.

Tabela 4.1 – Apresentação dos dados relativos aos testes de identificação do comportamento do algoritmo com a mudança no número de iterações.

$k1$	$k2$	ki	Máximo de Iterações	Tempo de Subida (ms)	Duty Cycle Máximo	Sobre Sinal (V)	Melhor Custo
0,0078	0,0031	5	1	1,220	0,3155	2,23	3763,3
0,0045	0,0025	5	5	0,995	0,3113	2,28	3562,3
0,0077	0,0004162	5	10	0,900	0,3370	27,2	3345,2
0,0056	0,0027	5	15	1,225	0,3128	2,26	3206,2

Fonte: Do autor (2021)

É possível indicar que a utilização do algoritmo com um número maior de iterações não causa melhora significativa nos resultados quando são observados o sobressinal e o tempo de subida. Contudo, quando observa-se o custo obtido, verifica-se uma melhora quando o número de iterações é superior à 8.

A princípio, 5 iterações demonstram valor suficiente para a obtenção de um resultado adequado. Isso se torna importante, uma vez que quanto maior o número de iterações, maior o tempo de processamento necessário para a execução do algoritmo.

4.2 Análise da interferência do tamanho da população na resposta do algoritmo.

Para identificar a interferência do tamanho da população na resposta final do algoritmo, foram realizados ensaios utilizando os limites ($k1 = 0,01$, $k2 = 0,03$ e $ki = 5$) definidos previamente e o número de iterações iguais a 5. Abaixo é apresentada a Tabela 4.2 com todos os ensaios realizados e posteriormente estes são explicados.

Mantendo todos os valores máximos dos ganhos e número máximo de iterações, alterou-se o tamanho da população em 10 a cada novo ensaio.

A Tabela 4.2 apresenta os resultados obtidos mantendo a população igual a 5 e variando o número de iterações.

Conforme os resultados apresentados na Tabela 4.2, observa-se que não houve melhora significativa nos resultados quando são observados o sobressinal e o tempo de subida. Entre-

Tabela 4.2 – Apresentação dos dados relativos aos testes de identificação do comportamento do algoritmo com a mudança no tamanho da população.

$k1$	$k2$	ki	Tamanho da População	Tempo de Subida (ms)	Duty Cycle Máximo	Sobre Sinal (V)	Melhor Custo
0,0040	0,0028	4.8497	10	1,121	0,3110	2,9	3575,2
0,0069	0,0020	5	20	1,072	0,3136	2,6	3280,5
0,0055	0,0017	5	30	0,913	0,3137	6,3	3394,7
0,0067	0,0018	5	40	1,015	0,3130	3,4	3389,0
0,0046	0,0024	5	50	0,9262	0,3112	2,9	3147,4

Fonte: Do autor (2021)

tanto, quando observa-se o custo obtido, verifica-se uma melhora quando o tamanho da população fica muito maior (50).

A princípio, o tamanho populacional igual a 10 demonstra ser suficiente para a obtenção de um resultado adequado. Isso se torna importante, uma vez que quanto maior o tamanho da população, maior o tempo de processamento necessário para a execução do algoritmo.

4.3 Refinamento dos limites dos ganhos ki , $k1$ e $k2$.

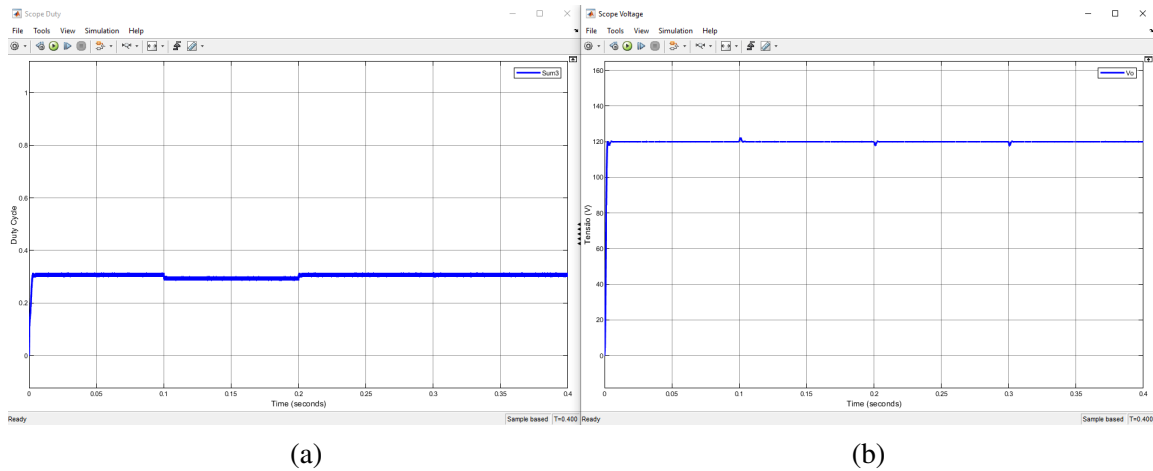
Tomando como base os ganhos ($k1 = 0,01$, $k2 = 0,003$ e $ki = 5$) obtidos na seção 3, o tamanho da população (10) obtido na seção 4.2 e o número máximo de iterações (5) obtido na seção 4.1, foi feito um refinamento maior dos limites de ganhos. Para facilitar o trabalho do algoritmo e proporcionar resultados melhores.

Como o tamanho da população e o número máximo de iterações não representam melhora significativa na qualidade dos resultados, foi fixado o valor de 5 iterações máximas e 10 para o tamanho da população. Estes valores demonstraram agilidade na resolução do algoritmo e proporcionaram resultados semelhantes aos obtidos para outros valores.

Como foi constatado que o ganho ki sempre ficou com valores próximos a 5, foi realizado um ensaio aumentando o valor mínimo de ki de 0 para 2 e a resposta para este caso pode ser observada na Figura 4.4.

Não foi percebida nenhuma variação negativa nos resultados, e os valores de ki se mantiveram acima de 4,9. Buscando direcionar melhor o algoritmo, o limite mínimo de ki foi fixado em 4 e o máximo em 6. O resultado está demonstrado na Figura 4.5.

Figura 4.4 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0 à 0,01, k_2 variando de 0 à 0,003 e k_i variando de 2 à 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

(b)

Fonte: Do autor (2021)

O teste representado na Figura 4.5 proporcionou valores melhores. O tempo de subida se manteve próximo à $1ms$ e o sobressinal foi de $0,7V$. Os ganhos obtidos neste teste foram:

$$k_1 = 0,0049$$

$$k_2 = 0,0023$$

$$k_i = 4,6154$$

Como em todos os ensaios o valor de k_i ficou distante de seus limites, estes foram alterados para 4,5 à 5,5. O resultado está demonstrado na Figura 4.6.

Foi observado que, quando k_i fica igual ou acima de 5, o sistema fica com uma maior oscilação. Por este motivo, o valor máximo de k_i foi definido como 5. Em paralelo, os valores observados de k_1 sempre ficaram entre 0,0025 e 0,0075, por isso estes valores representaram o limite mínimo e máximo de k_1 . Da mesma forma, k_2 nunca atingiu um valor abaixo de 0,001. O resultado pode ser observado na Figura 4.7.

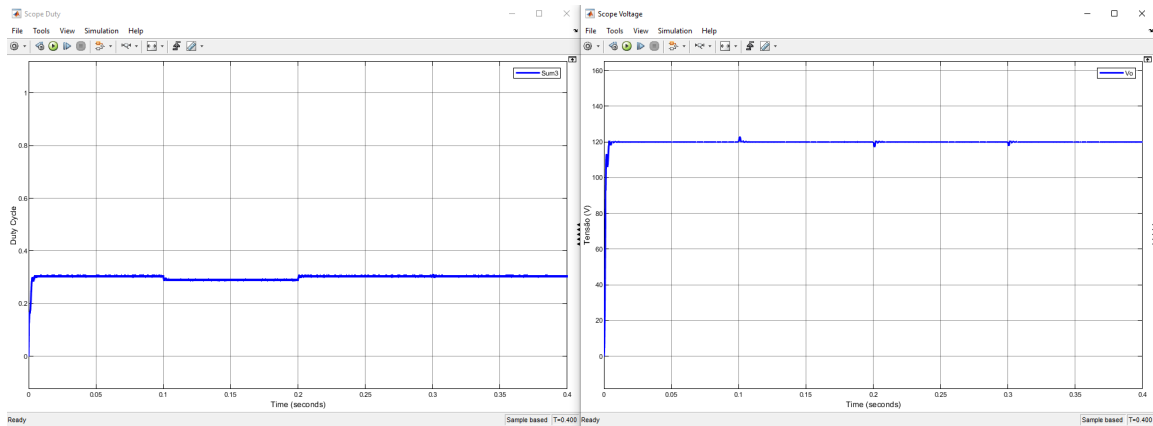
Com estes limites foi possível obter um resultado melhor e com menos oscilações. Os ganhos obtidos neste teste foram:

$$k_1 = 0.0067$$

$$k_2 = 0.0025$$

$$k_i = 5$$

Figura 4.5 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0 à 0,01, k_2 variando de 0 à 0,003 e k_i variando de 4 à 6. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

(b)

Fonte: Do autor (2021)

Apenas para efeito de teste, foi realizado um ensaio sem alteração nos limites dos ganhos, com o tamanho da população igual a 50 e com 15 iterações. O resultado pode ser observado na Figura 4.8.

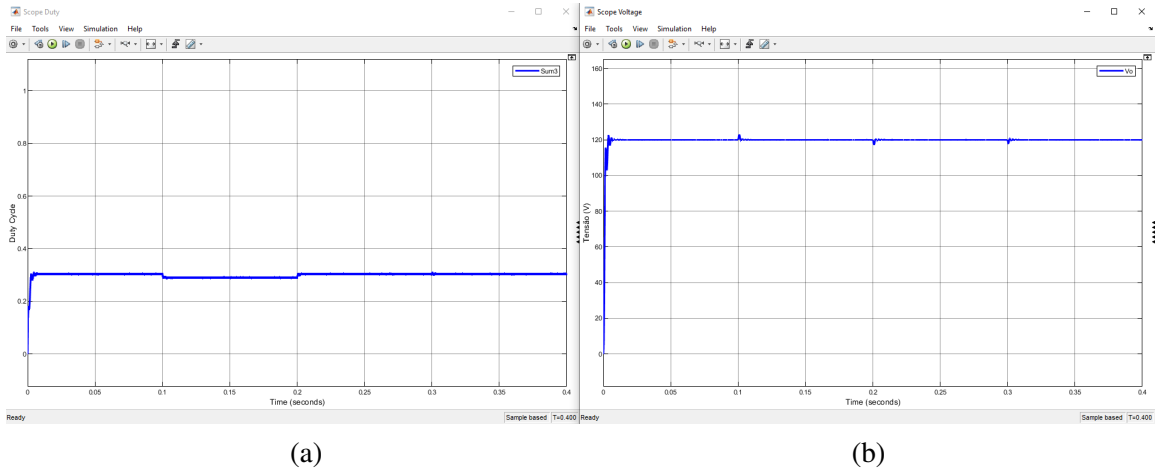
Comparando o ensaio demonstrado na Figura 4.7 com o ensaio da Figura 4.8, constata-se que o ganho k_1 teve a maior maior diferença. Percebe-se que este ganho não pode conter valores abaixo de 0,005, pois isso causou uma oscilação no sistema.

Buscando direcionar melhor o algoritmo, o alterou-se o ganho mínimo de k_1 para 0.005 e todos os outros limites foram mantidos. Para agilidade no teste, definiu-se a população e número de iterações como 10 e 5 respectivamente. . O resultado pode ser observado na Figura 4.9.

Buscando validar os resultados observados anteriormente, foi realizado um ensaio sem alteração nos limites dos ganhos, com o numero da população igual a 50 e com 15 iterações. O resultado pode ser observado na Figura 4.10.

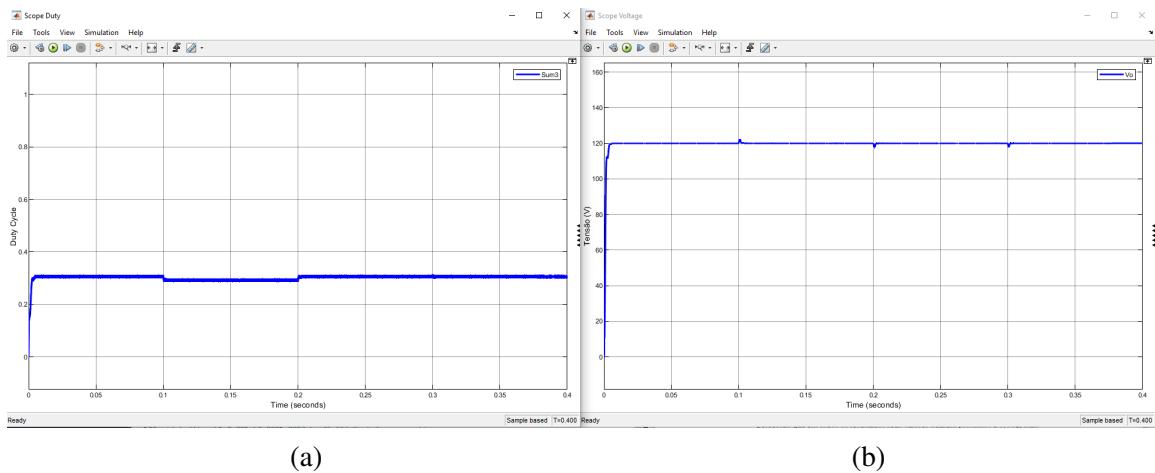
Abaixo é apresentada a Tabela 4.3 com todos os ensaios realizados neste item. Os resultados estão em ordem de execução, portanto o melhor resultado se encontra na última linha desta tabela.

Figura 4.6 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0 à 0,01, k_2 variando de 0 à 0,003 e k_i variando de 4,5 à 5,5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



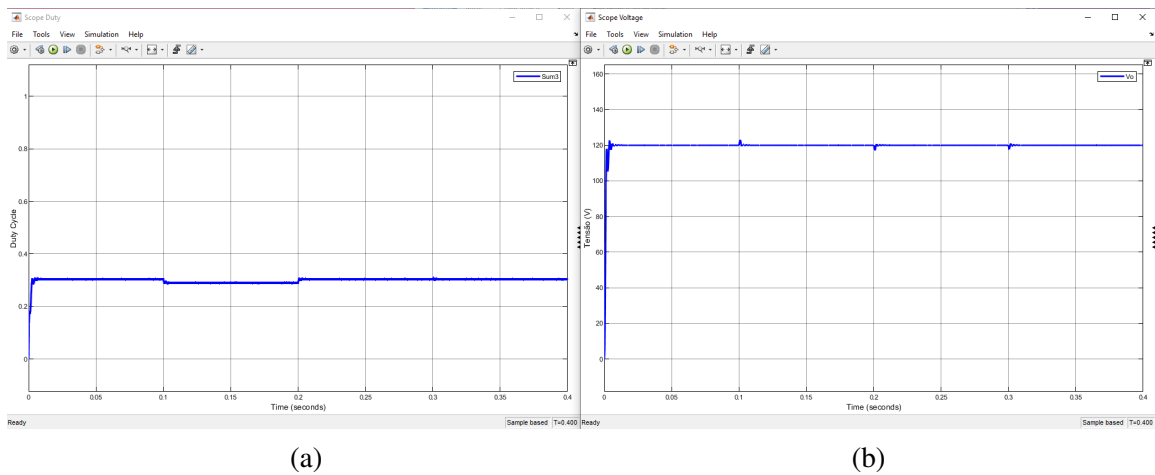
Fonte: Do autor (2021)

Figura 4.7 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,0025 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



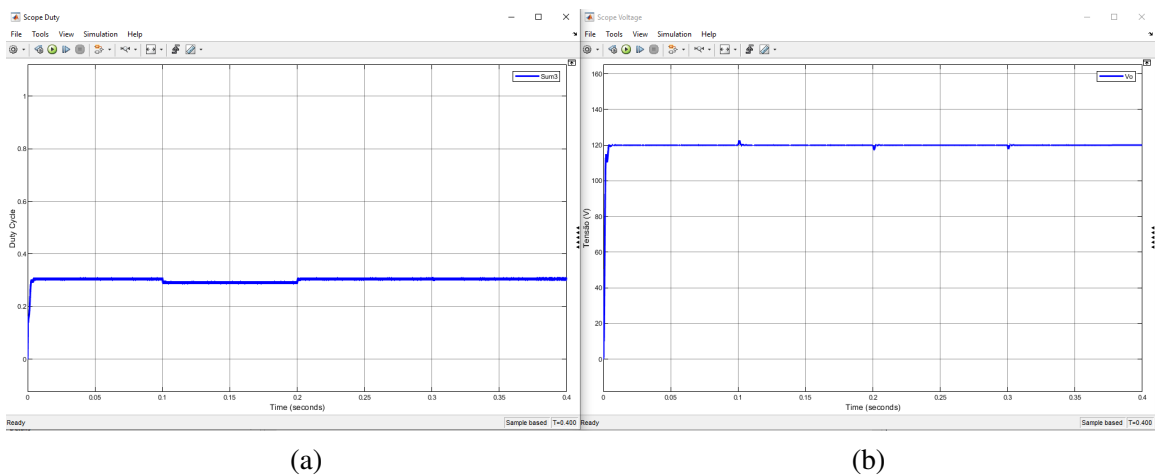
Fonte: Do autor (2021)

Figura 4.8 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,0025 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. Com a população igual a 50 e máximo de iterações igual a 15. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



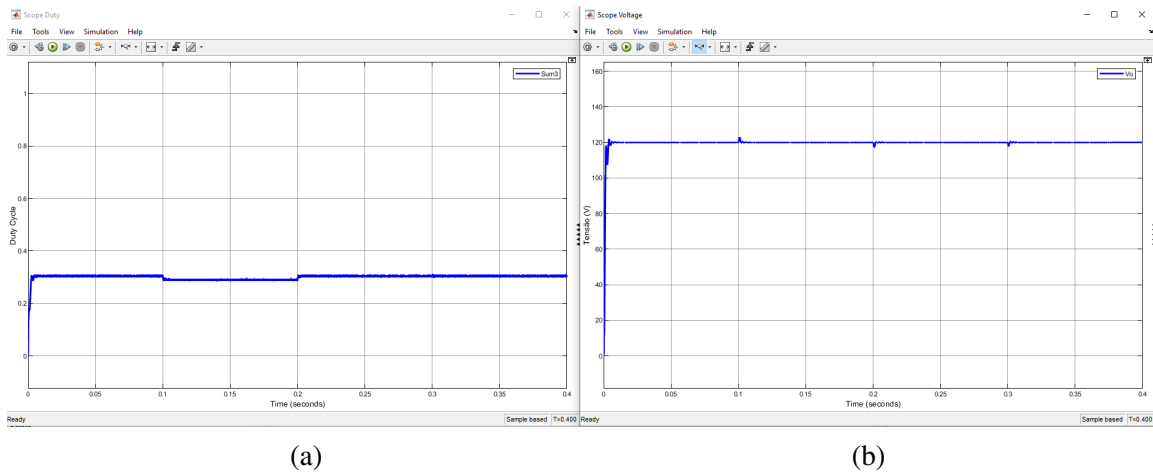
Fonte: Do autor (2021)

Figura 4.9 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,005 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. Com a população igual a 10 e máximo de iterações igual a 5. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



Fonte: Do autor (2021)

Figura 4.10 – Resposta do melhor caso encontrado pelo algoritmo TLBO para ganhos k_1 variando de 0,0025 à 0,0075, k_2 variando de 0,001 à 0,003 e k_i variando de 4,5 à 5. Com a população igual a 50 e máximo de iterações igual a 15. A imagem (a) representa o ciclo de trabalho e a imagem (b) representa a tensão de saída.



(a)

(b)

Fonte: Do autor (2021)

Tabela 4.3 – Apresentação dos dados relativos aos testes de identificação do comportamento do algoritmo com as alterações dos limites dos ganhos.

k_1	k_2	k_i	População	Iterações	Tempo de Subida (ms)	Duty Cycle Máximo	Sobre Sinal (V)	Melhor Custo
0,0085	0,0019	4,9488	10	5	1,204	0,3157	2,4	3584,6
0,0049	0,0023	4,6154	10	5	1,089	0,3004	2,9	3645,8
0,0043	0,0024	4,8951	10	5	0,973	0,3003	2,9	3436,1
0,0067	0,0025	5	10	5	1,212	0,2998	2,5	3407,8
0,0047	0,0023	5	50	15	0,9552	0,3003	2,9	3158,4
0,0059	0,0021	4,6164	10	5	1,1274	0,3013	2,7	3408,2
0,0051	0,0022	4,9934	50	15	0,9754	0,3116	2,8	3196,2

Fonte: Do autor (2021)

5 CONCLUSÃO

Este trabalho pode demonstrar a utilização do algoritmo *Teaching-learning-based optimization* (TLBO) para a determinação dos ganhos de controle de fontes chaveadas utilizando realimentação de estados.

Para o desenvolvimento do trabalho, foram feitos estudos para compreender os princípios de funcionamento dos conversores de maneira geral, porém com foco no conversor BUCK.

De modo geral, o desenvolvimento de conversores passa pela realização de sua modelagem e conseqüentemente pela compreensão da obtenção de um modelo de variáveis médias no espaço de estado.

Ainda no desenvolvimento de conversores BUCK, foram realizados estudos para entender sua característica dinâmica e ainda como se consiste o controle por realimentação de estados.

Foi possível destacar com os resultados e discussões que o algoritmo pode ser uma alternativa para a determinação dos ganhos de controle para fontes chaveadas de conversores CC-CC BUCK. Contudo, o algoritmo necessita que sejam feitas restrições em seus limites máximos e mínimos para que ele forneça resultados satisfatórios.

Durante os testes, houve também a constatação de que o número da população e o número de iterações não reflete, necessariamente, em uma melhora significativa dos resultados.

A maior dificuldade observada na utilização do algoritmo foi a determinação destes limites. Logo, pesquisas futuras, principalmente para o direcionamento do algoritmo através dos limites dos ganhos, podem ser de extrema relevância e contribuição.

Uma vez que o algoritmo TLBO demonstrou grande potencial na determinação de parâmetros adequados de controle de fontes chaveadas, espera-se que este trabalho tenha formado mais base para o desenvolvimento de novos estudos.

REFERÊNCIAS

- BRASIL. Protocolo de quioto: a convenção sobre mudança do clima : O brasil e a convenção – quadro das nações unidas. v. 5, p. 29, 1988.
- CAMPOS, M. C. M. M. d.; TEIXEIRA, H. C. G. Controle Típicos de Equipamentos e Processos Industriais. Blucher, v. 2, 2010.
- CARVALHO P, d. L. F. A. Redes Neurais Artificiais. 2009. Disponível em: <<https://sites.icmc.usp.br/andre/research/neural/#:~:text=Redes%20Neurais%20Artificiais%20s%C3%A3o%20t%C3%A9cnicas,adquirem%20conhecimento%20atrav%C3%A9s%20da%20experi%C3%Aancia>>.
- EPE. Projeção de Demanda de Energia Elétrica para os Próximos 10 anos (2017-2026). **Empresa de Pesquisa Energética**, Janeiro 2017.
- GARCIA CRISTIANO;CATALANO, M. E. B. Teaching-learning-based optimization no treinamento de redes neurais artificiais para problemas de classificação. Setembro 2018.
- HART, D. W. Eletrônica de Potência: Análise e Projetos de Circuitos. AMGH, 2011.
- IRENA. Transformando o Sistema Energerico. **Agência Internacional de Energia Renovável**, 2019. Disponível em: <https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2019/Sep/IRENA_Transforming_2019_PT>.
- MAXWELL. Controle por realimentação de estados. **Simulações em Engenharia Elétrica**, 2018. Disponível em: <<https://www.maxwell.vrac.puc-rio.br/25077/25077.PHP#>>.
- MEHL, E. L. M. Fontes Chaveadas. **Universidade Federal do Paraná**, 2004. Disponível em: <<http://www.eletrica.ufpr.br/mehl/fonteschaveadas.pdf>>.
- MIDDLEBROOK, R. D.; CUK, S. A General Unified Approach to Modelling Switching Converter Power Stages. **IEEE POWER ELECTRONICS SPECIALIST CONFERENCE**, IEEE, Cleveland, OH, 1976.
- MORAES, C. Análise do conversor buck em condução descontínua. **Eletrônica de potência**, 2018. Disponível em: <<https://eletronicadepotencia.com/analise-do-conversor-buck-em-conducao-descontinua/>>.
- NAGATA, E. A. Detecção, segmentação e classificação de afundamentos de tensão em sistemas elétricos de potência. Lavras, 2018.
- OGATA, K. Engenharia de controle moderno. v. 5, 2011.
- POMILIO, J. A. **Fontes Chaveadas**. Rio de Janeiro: FEEC 13/95, 2005.
- RAO, R. V.; SAVSANI, V. J.; VAKHARIA, D. Teaching-Learning-Based Optimization: a Novel Method for Constrained Mechanical Design Optimization Problems. **Computer-Aided Design** **43**, p. 303–315, 2011.
- SILVA, J. M. da; PRADO, M. L. M. Projeto de controladores por aplicação de polos aplicados a um sistema de suspensão ativa. 2018. Disponível em: <<http://www.ecomp.uefs.br/data/tcc/Projeto%20de%20Controladores%20por%20Aloca%C3%A7%C3%A3o%20de%20Polos%20Aplicados%20a%20um%20Sistema%20de%20Suspens%C3%A3o%20Ativa.pdf>>.

SOUZA A. G. ; SILVA, M. J. . F. S. C. . L. W. S. . B. B. H. G. Otimização tlbo para ajuste de parâmetros de controle de um pll baseado na referência síncrona. **XXII Congresso Brasileiro de Automática**, 2018.

SOUZA, M. J. F. Introdução à otimização: Programação linear. 2017.

THIJS, V. d. G. Obsolete or Resurgent? **The International Energy Agency in a changing global landscape**, Energy Policy, p. 233–241, 2012.