

**UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS  
UNIDADE ACADÊMICA DE GRADUAÇÃO  
CURSO DE ENGENHARIA ELÉTRICA**

**EVERTON DORO BITTENCOURT**

**SINTONIA PID EM CONVERSORES CC-CC UTILIZANDO ALGORITMOS  
GENÉTICOS (GA) E ENXAME DE PARTICULAS (PSO)**

**São Leopoldo**

**2022**

EVERTON DORO BITTENCOURT

**SINTONIA PID EM CONVERSORES CC-CC UTILIZANDO ALGORITMOS  
GENÉTICOS (GA) E ENXAME DE PARTICULAS (PSO)**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Bacharel em Engenharia, habilitação Engenharia Elétrica, ênfase em Controle e Automação Industrial, pelo Curso de Engenharia Elétrica da Universidade do Vale do Rio dos Sinos - UNISINOS

Orientador: Prof. Dr. João Olegário De Oliveira De Souza

São Leopoldo

2022

A minha família e esposa Gabriela Ludvig Bittencourt.

## **AGRADECIMENTOS**

Em primeiro lugar agradeço muito à Deus por ter me dado saúde e força para enfrentar as dificuldades durante o decorrer do curso.

Em especial, agradeço a minha esposa Gabriela Ludvig Bittencourt, por toda ajuda e compreensão, que esteve sempre comigo desde o início nos muitos momentos em que precisei passar noites sem dormir, finais de semanas sem poder sair, ver os parentes e amigos, ela nunca me deixou desanimar ou desistir, até mesmo quando as dificuldades se apresentavam.

A minha família querida, em especial a minha amada mãe Rosa Maria Doro Bittencourt, uma mulher que me ensinou a correr atrás dos meus objetivos e sempre se manteve preocupada com minha saúde, torcendo pelas minhas vitórias. Aos meus sogros Rejane Ludvig e Inácio Lecio Ludvig, que foram de extrema importância, proporcionando apoio, força e sustentação para seguir firme na caminhada até o final. Suportaram a carga imposta por esses longos anos de estudo e que agora vibram com a minha vitória.

Ao professor Dr. João Olegário de Oliveira de Souza, orientador deste trabalho, por seus conhecimentos, sugestões de pesquisa e por toda dedicação e atenção na revisão deste trabalho, ao qual me foi de grande aprendizado.

Aos demais professores da UNISINOS, por todos valiosos ensinamentos passados ao longo do curso, que me permitiram aplicar tais ensinamentos no desenvolvimento deste trabalho e assim me tornar um profissional melhor. Também um carinho especial aos professores que compuseram a banca avaliadora, que dispuseram de seu tempo para participarem do desfecho deste trabalho.

Enfim, a todos que de alguma forma contribuíram para realização deste trabalho e para a conclusão do curso, fechando comigo mais um ciclo da minha vida.

**“A persistência é o menor caminho do êxito”.**

(Charles Chaplin)

## RESUMO

Este trabalho apresenta a metodologia empregada para a sintonia de controladores PI e PID utilizando meta-heurísticas. É feita uma comparação entre duas meta-heurísticas, *Genetic Algorithm* – GA e *Particle Swarm Optimization* – PSO, com o objetivo de minimizar os parâmetros (*overshoot*, tempo de acomodação e tempo de subida) na resposta dos sistemas no domínio do tempo. Duas plantas foram propostas para aplicação das meta-heurísticas, sendo os conversores CC-CC um do tipo *Buck* e o outro do tipo *Boost* na configuração de carga eletrônica. Foram realizadas simulações a fim de comparar e encontrar os parâmetros para solução de cada problema proposto. As meta-heurísticas também foram submetidas na comparação de sua eficiência com um trabalho já realizado de otimização PID pelos autores Varol e Bingul (2004), onde foi plenamente satisfatório o desempenho de ambas. Todas as meta-heurísticas se mostraram satisfatórias na busca dos parâmetros de sintonia, e para atingir esse objetivo, utilizou-se dos métodos baseados nos critérios de desempenho IAE, ITAE, ISE e ITSE empregados na função objetivo de cada planta. Para o conversor do tipo *Buck*, a sintonia utilizando algoritmo PSO e o índice de desempenho ITSE, foi a que encontrou o melhor conjunto de parâmetros PID. Coincidentemente também para o conversor do tipo *Boost*, a sintonia utilizando algoritmo PSO e o critério de desempenho ITSE, foi a que encontrou o melhor conjunto de parâmetros para o controlador PI.

**Palavras-chave:** PID; Conversores CC-CC; Meta-heurística; *Genetic Algorithm*; *Particle Swarm Optimization*.

## LISTA DE FIGURAS

Figura 1 - Relação entre entrada e saída de um conversor CC-CC.....	19
Figura 2 - Diferentes tipos de Topologia de Conversores CC-CC.....	19
Figura 3 – Conversor CC básico .....	20
Figura 4 - Forma de onda da tensão de saída $V_o$ .....	21
Figura 5 – Obtenção do PWM com sinal de referência contínuo .....	22
Figura 6 – Topologia do Conversor <i>Buck</i> ideal.....	22
Figura 7 - Circuito do conversor <i>Buck</i> representado o transistor como chave .....	23
Figura 8 - Modos de operação do Conversor <i>Buck</i> – M1 chave aberta, M2 fechada	23
Figura 9 - Formas de onda da tensão e corrente no conversor <i>Buck</i> .....	24
Figura 10 - Topologia típica do conversor <i>Boost</i> .....	25
Figura 11 - Circuito do conversor <i>Boost</i> representado o transistor como chave .....	26
Figura 12 - Modos de operação do Conversor <i>Boost</i> .....	26
Figura 13 - Formas de onda do conversor <i>Boost</i> .....	27
Figura 14 - Topologia de uma carga eletrônica baseada em um conversor <i>Boost</i> ....	28
Figura 15 – Circuito da carga eletrônica utilizada como processo neste trabalho.....	29
Figura 16 - Diagrama de blocos do controle PID.....	31
Figura 17 - Diagrama de blocos de um sistema de controle com controlador PI .....	33
Figura 18 - Curva da resposta transitória em degrau unitário .....	34
Figura 19 - Fluxograma do Algoritmo Genético (GA) .....	38
Figura 20 - Seleção dos Pais para Reprodução no GA.....	39
Figura 21 - DNA sendo misturado no cruzamento do GA .....	40
Figura 22 – Efeito da Mutação de um gene num cromossomo do GA .....	40
Figura 23 – Movimento de Partículas dentro de um espaço de busca .....	41
Figura 24 - Modificação do ponto de busca no PSO .....	42
Figura 25 – Bloco <i>SimCoupler</i> do PSIM.....	45
Figura 26 – Exemplo conexão de entrada e saída do módulo <i>SimCoupler</i> no PSIM	45
Figura 27 – Exemplo de ligação do módulo <i>SimCoupler</i> no Simulink® .....	45
Figura 28 - Circuito de potência e controle do conversor CC-CC <i>Buck</i> .....	46
Figura 29 – Circuito de teste dos parâmetros $K_p$ , $K_i$ e $K_d$ do conversor <i>Buck</i> .....	49
Figura 30 - Parâmetros do Controlador no PSIM .....	49
Figura 31 - Configuração do bloco <i>Gating Block</i> no PSIM .....	50
Figura 32 - Circuito de Potência do conversor <i>Boost</i> como Carga Eletrônica.....	51

Figura 33 - Diagrama de blocos do controlador PI para carga eletrônica .....	51
Figura 34 - Circuito de teste para Carga Eletrônica .....	52
Figura 35 - Funções do algoritmo GA .....	53
Figura 36 - Determinação dos parâmetros do PID através das meta-heurísticas .....	56
Figura 37 - Sistema de Controle PID Planta Varol e Bingul .....	57
Figura 38 - Parâmetros do sistema de controle PID planta Varol e Bingul.....	58
Figura 39 - Sistema de controle com o índice IAE .....	58
Figura 40 - Sistema de controle com o índice ISE .....	59
Figura 41 - Função de Custo IAE .....	59
Figura 42 – Parâmetros de simulação para o controle do conversor <i>Buck</i> .....	61
Figura 43 – Função objetivo e Índice de desempenho IAE para o conversor <i>Buck</i> ..	62
Figura 44 - Função objetivo e Índice de desempenho ITAE para o conversor <i>Buck</i> ..	62
Figura 45 - Função objetivo e Índice de desempenho ISE para o conversor <i>Buck</i> ...	63
Figura 46 - Função objetivo e Índice de desempenho ITSE para o conversor <i>Buck</i> ..	63
Figura 47 – Função objetivo e Índice de desempenho IAE para o conversor <i>Boost</i> ..	64
Figura 48 - Função objetivo e Índice de desempenho ITAE para o conversor <i>Boost</i> ..	65
Figura 49 - Função objetivo e Índice de desempenho ISE para o conversor <i>Boost</i> ..	65
Figura 50 - Função objetivo e Índice de desempenho ITSE para o conversor <i>Boost</i> ..	65
Figura 51 - Curvas de desempenho das funções de custo de Varol e Bingul .....	67
Figura 52 - Respostas ao degrau com diferentes funções de custo.....	68
Figura 53 – Resposta ao degrau com novo teste de GA e PSO x Varol e Bingul .....	69
Figura 54 - Resposta do GA no conversor <i>Buck</i> comparando com Souza (2021) ....	72
Figura 55 - Resposta do PSO no conversor <i>Buck</i> comparando com Souza (2021) ..	72
Figura 56 - Teste em rampa de 35 a 50 V com parâmetros do PSO-ITSE. ....	73
Figura 57 – Circuito de teste em rampa para os dados de Souza (2021) .....	74
Figura 58 - Teste em rampa de 35 a 50 V com parâmetros de Souza (2021) .....	74
Figura 59 - Teste de variação de resistência com parâmetros de Souza (2021). ....	75
Figura 60 - Teste de variação de resistência com parâmetros do PSO-ITSE .....	76
Figura 61 - Respostas do GA na carga eletrônica, com tensão de entrada em 25 V ..	78
Figura 62 – Respostas individuais do GA para cada índice de desempenho .....	78
Figura 63 - Respostas PSO na carga eletrônica, com tensão de entrada em 25 V ..	79
Figura 64 - Circuito carga eletrônica com controle PI em bloco .....	79
Figura 65 - Corrente drenada pela carga, na sintonia de Souza (2021).....	80
Figura 66 - Respostas do GA na carga eletrônica, com tensão de entrada em 5 V ..	80

Figura 67 – Comparativo de <i>overshoot</i> no teste de tensão de entrada de 5 V .....	81
Figura 68 - Respostas do PSO na carga eletrônica, com tensão de entrada em 5 V	81
Figura 69 - Resposta PSO-ITSE com <i>overshoot</i> de 6%.....	82
Figura 70 – Teste de rampa de corrente na carga eletrônica com GA e PSO .....	82
Figura 71 – Teste em rampa na carga eletrônica com parâmetros do GA-IAE .....	83
Figura 72 - Teste em rampa na carga eletrônica com parâmetros do PSO-ITSE .....	83
Figura 73 - Circuito de controle PI com rampa de corrente de 10 a 25 A.....	84
Figura 74 – Resposta a rampa de corrente com tensão de entrada em 15 V .....	84

## LISTA DE QUADROS

Quadro 1 – Relação entre Biologia e Problema de Otimização .....	38
Quadro 2 – Parâmetros do GA testados no trabalho de Varol e Bingul .....	60
Quadro 3 - Parâmetros do PSO testados no trabalho de Varol e Bingul.....	60
Quadro 4 - Parâmetros do GA e PSO simulados na planta do Conversor <i>Buck</i> .....	61
Quadro 5 - Parâmetros do GA e PSO simulados na planta do Conversor <i>Boost</i> .....	64
Quadro 6 – Mudanças de parâmetros para nova simulação com GA e PSO .....	68

## LISTA DE TABELAS

Tabela 1 – Parâmetros do Conversor Buck.....	47
Tabela 2 – Parâmetros comparativos do trabalho de Varol e Bingul.....	66
Tabela 3 – Comparativo GA e PSO com diferentes funções de custo .....	67
Tabela 4 – Resultado da simulação com população de 20 e 50 iterações.....	69
Tabela 5 – Média de 5 simulações com 20 indivíduos e 50 iterações.....	70
Tabela 6 – Resultados da otimização no conversor CC-CC <i>Buck</i> .....	71
Tabela 7 – Média das 5 simulações na otimização do conversor CC-CC <i>Buck</i> .....	73
Tabela 8 – Resultados da otimização no conversor CC-CC <i>Boost</i> .....	77

## LISTA DE SIGLAS

ACO	<i>Ant Colony Optimization</i> (Otimização por Colônia de Formigas)
ABC	<i>Artificial Bee Colony</i> (Otimização por Colônia artificial de Abelhas)
CA	Corrente Alternada
CC	Corrente Contínua
GA	<i>Genetic Algorithm</i> (Algoritmo Genético)
IAE	<i>Integral of the Absolute Error</i> (Integral do Erro Absoluto)
IFT	<i>Iterative Feedback Tuning</i>
IGBT	<i>Insulated Gate Bipolar Transistor</i> (Transistor Bipolar de Porta Isolada).
IMC	<i>Internal Model Control</i>
ISE	<i>Integral of the Squared Error</i> (Integral do Erro Quadrático)
ITAE	<i>Integral of Time multiplied by Absolute Error</i> (Integral do Erro Absoluto Ponderado pelo Tempo)
ITSE	<i>Integral of Time multiplied by Squared Error</i> (Integral do Erro Quadrático Ponderado pelo Tempo)
MATLAB®	<i>Matrix Laboratory</i> , produzido pela <i>MathWorks</i> ®
MOSFET	<i>Metal-Oxide-Semiconductor Field-Effect Transistor</i> (Transistor de Efeito de Campo de Óxido Metálico Semicondutor)
PID	Proporcional Integral Derivativo
PSIM	<i>Power Simulation</i> (Software de Simulação de Circuito Eletrônico)
PSO	<i>Particle Swarm Optimization</i> (Otimização por Enxame de Partículas)
PWM	<i>Pulse Width Modulation</i> (Modulação por Largura de Pulsos)
RBEMCE	<i>Reference Based Error With Minimum Control Effort</i> (Erro Baseado em Referência com Controle Mínimo de Esforço)
Simulink®	<i>Simulation and Model-Based Design</i> integrado no software <i>MATLAB</i> ®, produzido pela <i>MathWorks</i> ®
ZN	<i>Ziegler-Nichols</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
<b>1.1 Tema</b> .....	<b>15</b>
<b>1.2 Delimitação do Tema</b> .....	<b>15</b>
<b>1.3 Problema</b> .....	<b>16</b>
<b>1.4 Objetivos</b> .....	<b>16</b>
1.4.1 Objetivo Geral .....	16
1.4.2 Objetivos Específicos .....	16
<b>1.5 Justificativa</b> .....	<b>17</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>18</b>
<b>2.1 Conversores CC-CC</b> .....	<b>18</b>
2.1.1 Topologias de conversores CC-CC: .....	19
2.1.2 Modulação por Largura de Pulso – PWM .....	20
2.1.3 Conversor <i>Buck</i> .....	22
2.1.4 Conversor <i>Boost</i> .....	25
2.1.5 Carga Eletrônica CC .....	27
<b>2.2 Controladores PID</b> .....	<b>29</b>
2.2.1 Controlador PI .....	32
2.2.2 Características das respostas transitórias de um sistema de controle .....	33
2.2.3 Índices de Desempenho .....	34
<b>2.3 Ferramentas de Otimização</b> .....	<b>36</b>
2.3.1 Algoritmo Genético GA .....	37
2.3.2 Algoritmo Enxame de Partículas PSO .....	41
<b>3 METODOLOGIA</b> .....	<b>44</b>
<b>3.1 Ambiente de Simulação</b> .....	<b>44</b>
3.1.1 Simulação compartilhada .....	44
<b>3.2 Planta do Conversor CC-CC tipo <i>Buck</i></b> .....	<b>46</b>
<b>3.3 Planta do conversor CC-CC tipo <i>Boost</i> como carga eletrônica</b> .....	<b>50</b>
<b>3.4 <i>Scripts</i> para as Meta-heurísticas</b> .....	<b>52</b>
3.4.1 <i>Script</i> do Algoritmo Genético GA .....	53
3.4.2 <i>Script</i> do Algoritmo Enxame de Partículas PSO .....	54
<b>3.5 Parâmetros das Meta-Heurísticas</b> .....	<b>55</b>
3.5.1 Teste das Meta-Heurísticas em uma planta com valores conhecidos .....	57

3.5.2 Configurações do GA e PSO na planta do Conversor CC-CC <i>Buck</i> .....	61
3.5.3 Configurações do GA e PSO na planta do Conversor CC-CC <i>Boost</i> .....	63
<b>4 ANÁLISE DOS RESULTADOS E DISCUSSÃO.....</b>	<b>66</b>
<b>4.1 Resultado comparativo do GA e PSO na Planta de Varol e Bingul.....</b>	<b>66</b>
4.1.1 Novos arranjos de parâmetros para uma melhor performance .....	68
<b>4.2 Análise de desempenho do conversor CC-CC <i>Buck</i> .....</b>	<b>70</b>
4.2.1 Teste de variação de tensão de entrada numa faixa de 35 a 50 V no <i>Buck</i> .....	73
4.2.2 Teste no <i>Buck</i> com variação de resistência na carga de 3, 4 e 5 ohms.....	75
<b>4.3 Análise de desempenho do conversor CC-CC <i>Boost</i> – Carga Eletrônica....</b>	<b>76</b>
4.3.1 Teste com tensão de entrada fixa em 25 V e referência de corrente em 20 A.	77
4.3.2 Teste com tensão de entrada em 5 V e referência de corrente em 20 A .....	80
4.3.3 Teste em rampa de corrente de 10 a 25 A com entrada fixa em 25 V .....	82
<b>5 CONCLUSÃO .....</b>	<b>85</b>
<b>REFERÊNCIAS.....</b>	<b>86</b>

## 1 INTRODUÇÃO

Conversores de corrente contínua para corrente contínua (CC-CC) são muito utilizados em várias aplicações industriais, principalmente em fontes de alimentação que necessitam alimentar cargas com tensões contínuas variáveis, sejam elas de valores maiores ou menores que as tensões de entrada. Através de uma fonte de tensão CC constante, essa conversão pode ser obtida com um circuito que utiliza de elementos armazenadores de energia como o indutor e o capacitor, e de um dispositivo de estado sólido que opere no modo de chaveamento em alta frequência.

Para realização deste tipo de conversão utiliza-se de uma técnica de chaveamento de dispositivos semicondutores de potência, chamada de modulação por largura de pulso (PWM), esta modulação permitirá que o valor médio da tensão de saída varie de acordo com o tempo em que o dispositivo de chaveamento permaneça ligado com o sinal de entrada. (ARRABAÇA; GIMENEZ, 2014)

Para controlar a razão cíclica de chaveamento, uma das técnicas mais utilizadas é o uso de controladores PID (proporcional, integral, derivativo, ou uma combinação destes). Nos processos industriais, uma série de estruturas de controladores são desenvolvidas para estabilizar processos estáveis, instáveis e não lineares, onde o ajuste destes torna-se um procedimento preliminar essencial (GOLDBERG, 1989). O PID é o mais utilizado em laços de controle, pois é capaz de eliminar o erro em regime permanente além de estimar linearmente amostras futuras através da ação derivativa. Em um sistema de controle automático o controlador é o cérebro, por fazer o sistema atingir um comportamento desejado a partir de suas configurações (OGATA, 2011).

É grande o desafio da engenharia de encontrar a melhor sintonia dos parâmetros para o controlador por métodos tradicionais, como por Ziegler Nichols ou na tentativa e erro. Atualmente, algoritmos de otimização são aplicados para encontrar os parâmetros do controlador e melhorar sua performance (FRANKLIN; POWELL; EMANI-NAEINI, 2013).

Os procedimentos de otimização baseados em algoritmos heurísticos surgiram como uma ferramenta poderosa para encontrar soluções para uma variedade de problemas de engenharia de controle. Eles são amplamente usados no controle de processos devido à sua simplicidade estrutural e velocidade de resposta (TALBI, 2009).

Devido à sua flexibilidade, os algoritmos de otimização podem se adaptar facilmente aos procedimentos clássicos de projeto de controladores existentes. Eles podem ser usados como uma ferramenta vital para projetar controladores estruturados clássicos e modificados para uma classe de modelos de processos instáveis, independentemente da ordem do modelo. Os métodos heurísticos mais recentes, como *Genetic Algorithm* - GA (Algoritmo Genético), *Particle Swarm Optimization* – PSO (Otimização por Enxame de Partículas), *Ant Colony Optimization* – ACO (Otimização por Colônia de Formigas), e *Artificial Bee Colony* – ABC (Otimização por Colônia artificial de Abelhas), dentre outros, são amplamente abordados pelos pesquisadores para ajustar controladores para uma classe de modelos de processo (SOUZA; VALIATI; GÓMEZ, 2012).

Pautado no exposto, o presente trabalho busca realizar um comparativo entre algoritmos genéticos (GA) e enxame de partículas (PSO) para a sintonia de controladores PID em conversores CC-CC.

## 1.1 Tema

Sintonizar os parâmetros de controladores PID, para o correto funcionamento de conversores CC-CC, utilizando de meta-heurísticas como os algoritmos genéticos (GA) e enxame de partículas (PSO).

## 1.2 Delimitação do Tema

O tema da presente pesquisa é delimitado na comparação de algoritmos genéticos (GA) e enxame de partículas (PSO) para a sintonia de controladores PID em conversores CC-CC. A escolha do GA, foi por ele ser o mais conhecido dentre os algoritmos de controle evolutivo, sendo muito utilizado no campo da Inteligência Artificial (IA). Já o PSO, foi escolhido pelo seu grande sucesso na comunidade científica na resolução de problemas difíceis de otimização. Os conversores a serem estudados serão: o conversor rebaixador de tensão *step-down* (*Buck*) com potência de 100 W e uma carga eletrônica CC baseada em conversor elevador de tensão *step-up* (*Boost*), que terá uma variação de corrente na entrada de 10 a 25 A.

### 1.3 Problema

O controle PID proporciona uma variação contínua da saída dentro de um mecanismo de realimentação de loop de controle, para controlar com precisão o processo, removendo a oscilação e aumentando a eficiência, dessa forma a sintonia define um bom controle de um processo contínuo. Diante desta importância, buscase a redução dos parâmetros da resposta entregue à um sistema no domínio do tempo. Em relação ao tempo de subida ( $tr$ ), tempo de acomodação ( $ts$ ) e do *overshoot* ( $\%UP$ ), quem apresenta melhor capacidade de sintonia de controladores PID em conversores CC-CC, os Algoritmos Genéticos (GA) ou Enxame de Partículas (PSO)?

### 1.4 Objetivos

#### 1.4.1 Objetivo Geral

Comparar duas meta-heurísticas GA e PSO, na obtenção dos parâmetros para sintonia de controladores PID, utilizados em malhas de controle de conversores CC-CC do tipo *Buck* e *Boost*, onde em um determinado uso com estes conversores, se deseja atingir valores de referência minimizando o erro em regime permanente. Para isso, a otimização buscará reduzir os parâmetros da resposta destes sistemas, como o tempo de subida, o *overshoot* e o tempo de acomodação.

#### 1.4.2 Objetivos Específicos

- a) aprofundar o conhecimento em conversores do tipo CC-CC, controladores PID e meta-heurísticas como os algoritmos GA e PSO;
- b) implementar nos algoritmos GA e PSO uma função objetivo para busca dos parâmetros que satisfaça a sintonia dos controladores PID em conversores do tipo CC-CC;
- c) testar os parâmetros encontrados pelos algoritmos GA e PSO nos conversores CC-CC;
- d) comparar os resultados indicando qual teve a melhor performance.

## 1.5 Justificativa

Nos dias atuais, o crescente aumento da demanda por produtos tecnológicos vem forçando a indústria a se reinventar, transformando seus processos de produção manuais em processos automatizados. Para isso, é necessário o uso de diversos equipamentos, máquinas e sistemas que possam de alguma forma serem programados para obterem um resultado muito superior ao processo manual. Neste contexto, a automação industrial entra para atender a demanda e trazer agilidade nos processos, aumentando a produtividade. Dentro da automação, a utilização de controladores de processos contínuos em malha fechada tem aumentado gradativamente, e nesse aspecto saber realizar a sintonia de um controlador pode trazer a estabilidade ao sistema atingindo um controle mais preciso.

Na literatura, diversos trabalhos têm sido feitos na área de controladores para busca de parâmetros por meio de otimização, e ainda são um desafio, principalmente quando não se consegue modelar por meio de uma função de transferência a planta ou processo a ser controlada. Métodos heurísticos empíricos tentam encontrar valores por meio de ensaios práticos, fazendo com que em muitos casos o controlador tenha um desempenho ruim, gerando custos para a indústria. Com a evolução dos computadores, meta-heurísticas por meio de algoritmos de otimização, passam a ser utilizadas na busca de uma melhor sintonia para controladores PID, do que os métodos heurísticos.

Diante disso, e pela importância que a sua discussão tem para a área de conhecimento explorada, é que se justifica a escolha e a motivação do tema para realização desse trabalho, que buscará comparar duas meta-heurísticas GA e PSO, na obtenção dos parâmetros para sintonia de controladores PID, utilizados em malhas de controle de conversores CC-CC do tipo *Buck* e *Boost*.

## 2 FUNDAMENTAÇÃO TEÓRICA

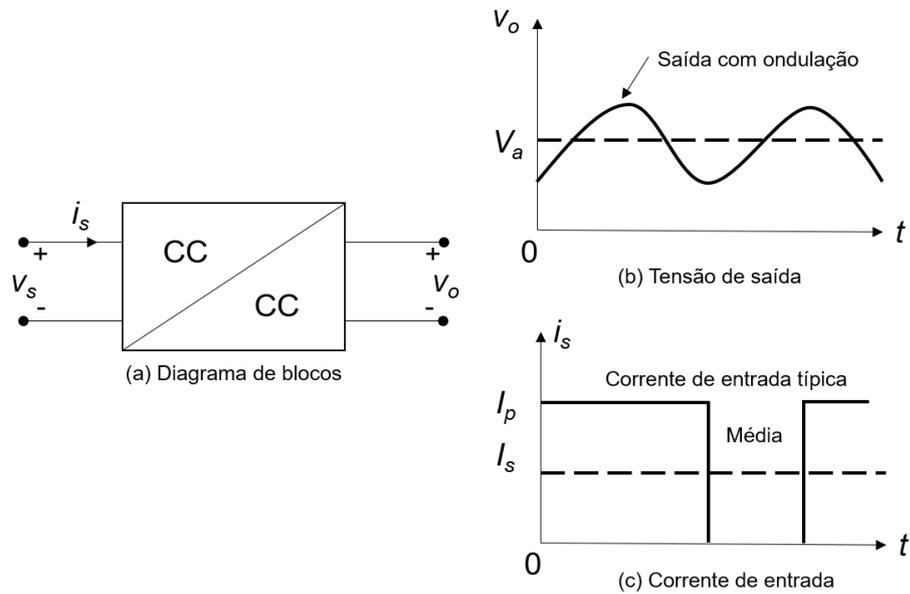
### 2.1 Conversores CC-CC

“Em muitas aplicações industriais, é necessário converter uma fonte de tensão CC fixa em uma variável. Um conversor CC-CC converte diretamente CC em CC, e é chamado apenas de ‘conversor CC’ ” (RASHID, 2014 p. 180). Segundo Mattos *et al.* (2021, p. 1), “conversores CC-CC são largamente utilizados para regular a tensão de alimentação de diversos sistemas, como computadores, equipamentos de comunicação, sistemas aeroespaciais etc.” Rashid (2014), traz outro exemplo de utilização como em veículos elétricos, onde os conversores atuam no controle de tração de motores com aceleração suave, alta eficiência e uma resposta dinâmica rápida, como também na frenagem regenerativa, trazendo economia energética nos momentos de parada do veículo.

Conversores CC-CC são sistemas formados por semicondutores de potência operando como interruptores, e por elementos passivos, normalmente indutores e capacitores que tem por função controlar o fluxo de potência de uma fonte de entrada para uma fonte de saída (LIPING GUO; HUNG; NELMS, 2009). “Um conversor CC pode ser considerado o equivalente CC de um transformador CA com uma relação de espiras continuamente variável. Assim como o transformador, ele pode ser usado para baixar ou elevar uma fonte de tensão CC” (RASHID, 2014, p. 180). Mesmo que se tenha uma variação na carga e na tensão de entrada, os conversores CC-CC apresentam uma importante característica, que é a capacidade de manter a tensão de saída regulada (MATTOS *et al.*, 2021).

A Figura 1 detalha a relação de entrada e saída de um conversor CC-CC trazendo em a) um diagrama de blocos que mostra que este tipo de conversor pode tanto produzir uma saída de tensão fixa ou variável, a partir de uma tensão CC de entrada ser fixa ou variável. Em b) temos a tensão de saída sendo mostrada com ondulação, pois na prática tanto a tensão como a corrente da fonte de entrada mostradas em c), podem vir carregadas de harmônicas ou ondulações (RASHID, 2014).

Figura 1 - Relação entre entrada e saída de um conversor CC-CC

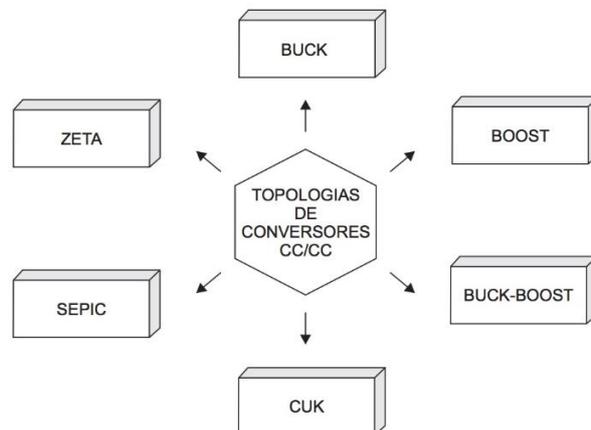


Fonte: Rashid (2014, p. 181).

### 2.1.1 Topologias de conversores CC-CC:

Na construção de conversores, a ideia básica é utilizar o menor número de componentes possível para sintetização dos circuitos, visando apresentarem dimensões de concepção muito menores que uma fonte linear, por exemplo, e serem utilizados em projetos com espaço físico reduzido. Temos seis conversores que podem ser considerados como básicos, ou seja, a maioria dos conversores usados normalmente pode ser derivada desses circuitos. Esses conversores são *Buck*, *Boost*, *Buck-Boost*, *Cuk*, *Sepic* e *Zeta*, que podem ser vistos na Figura 2 (MELLO, 2011).

Figura 2 - Diferentes tipos de Topologia de Conversores CC-CC



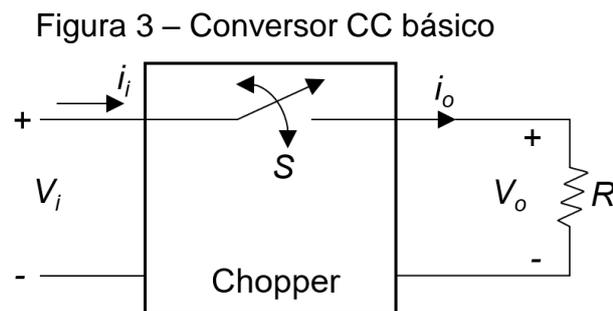
Fonte: Arrabaça e Gimenez (2014, p. 18).

Neste trabalho serão utilizados em nosso estudo os conversores CC-CC com as topologias do tipo *Buck* e *Boost*, as quais serão detalhadas nos tópicos seguintes.

### 2.1.2 Modulação por Largura de Pulso – PWM

É uma técnica empregada em várias áreas da eletrônica para o controle da energia e potência entregues à carga nos sistemas de corrente contínua e muito difundida para o controle de chaveamento dos conversores CC-CC (ZAMBON; VARGAS; FLORIAN, 2020).

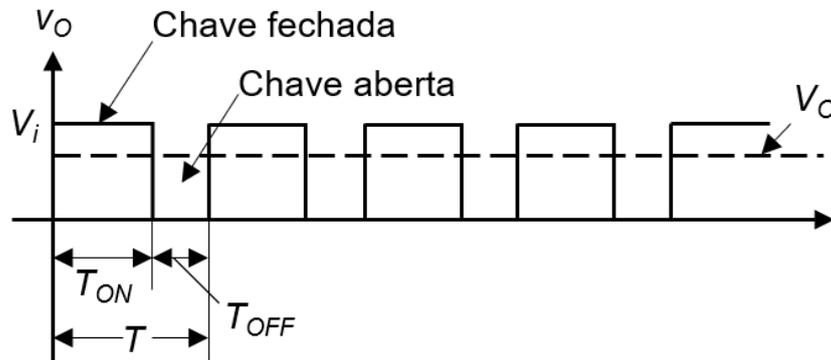
Em PWM, opera-se com frequência constante, variando-se o tempo em que a chave permanece ligada. Considerando a chave  $S$  do circuito da Figura 3 ideal, podemos descrever de forma sucinta o princípio de funcionamento dessa técnica de chaveamento.



Fonte: Ahmed (2000, p. 309).

A chave  $S$  está ligada em série com a fonte de tensão de entrada  $V_i$  e a carga  $R$ , sendo que a tensão média na carga será dada pela relação do tempo em que esta chave permanecer fechada (AHMED, 2000).

Com base na Figura 4, podemos observar a forma de onda da tensão de saída, a qual devido à chave ter permanecido mais tempo fechada ( $T_{ON}$ ) do que aberta ( $T_{OFF}$ ), por um período ( $T$ ) de tempo, a tensão média  $V_o$  se aproximou mais da tensão de entrada  $V_i$ .

Figura 4 - Forma de onda da tensão de saída  $V_o$ 

Fonte: Ahmed (2000, p. 310).

A relação da tensão média de saída pela tensão de entrada pode ser obtida com base na equação (1).

$$V_o = \frac{T_{ON}}{T_{ON} + T_{OFF}} V_i \xrightarrow{\text{como } T = T_{ON} + T_{OFF}} \xrightarrow{\text{então}} V_o = \frac{T_{ON}}{T} V_i \quad (1)$$

A razão cíclica de operação também conhecida como ciclo de trabalho (*Duty Cycle*,  $D$ ), é uma grandeza contínua limitada entre 0 e 1 e geralmente expressa em percentual, e pode ser definida pela relação entre a largura de pulso de chave ativa ( $T_{ON}$ ) e o período da forma de onda, conforme a equação (2) (LUCAS, 2015).

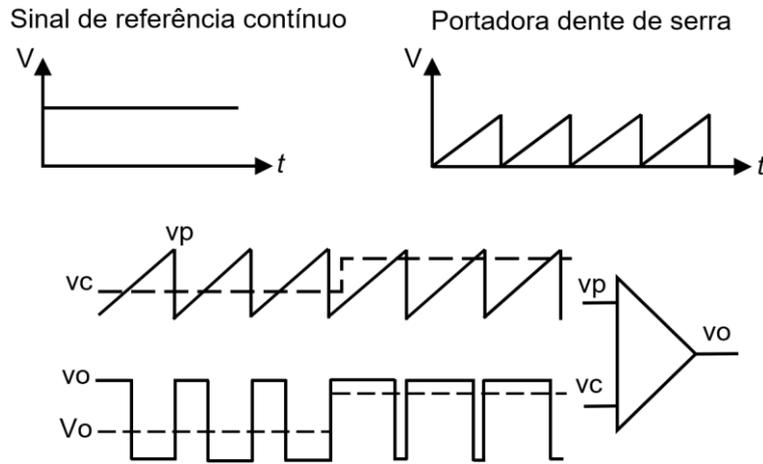
$$D = \frac{T_{ON}}{T} \quad (2)$$

Outra forma de representar a relação da tensão média de saída pela tensão de entrada, é variando o percentual de chave ativa com  $D$ , conforme a equação (3).

$$V_o = D V_i \quad (3)$$

É possível obter um sinal PWM utilizando um comparador de tensão e dois sinais, sendo um contínuo ligado na entrada positiva, e outro sinal triangular (dente de serra) ligado na entrada negativa. Quando  $v_c > v_p$  saturação positiva, e quando o contrário, saturação negativa. Alterando a tensão contínua, pode-se alterar o *Duty Cycle*, pois os intervalos se modificarão (ANGADI; SAIKUMAR; SATYANARAYANA KUMARI, 2013). A Figura 5 mostra um exemplo de sinal de controle ( $v_c$ ) contínuo, sendo comparado com uma portadora dente de serra, onde através da realimentação da tensão de saída do circuito, é possível variar o *Duty Cycle*.

Figura 5 – Obtenção do PWM com sinal de referência contínuo



Fonte: Adaptado de Rashid (2014, p. 185).

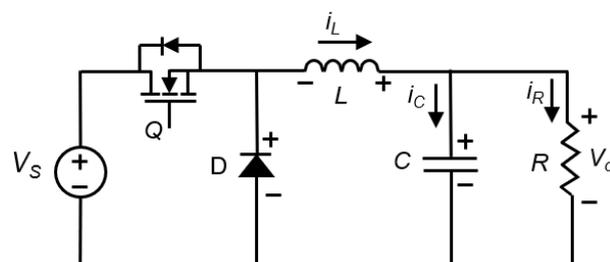
### 2.1.3 Conversor *Buck*

“O conversor *Buck* é característico por sua saída apresentar a tensão regulada sempre menor, ou igual no limite teórico, à tensão de entrada. Assim, por conta de tal característica, também é conhecido como conversor *Step-Down*” (ZAMBON; VARGAS; FLORIAN, 2020).

Segundo Mello (2011, p. 24), “[...] A tensão de saída possui mesma polaridade da tensão de entrada. O ruído gerado para a saída é baixo devido à configuração do circuito L1 C1, que forma um filtro passa-baixa”. O autor também complementa do alto ruído gerado na alimentação de entrada, devido ao chaveamento do transistor.

A Figura 6, apresenta idealmente a topologia desse conversor.

Figura 6 – Topologia do Conversor *Buck* ideal

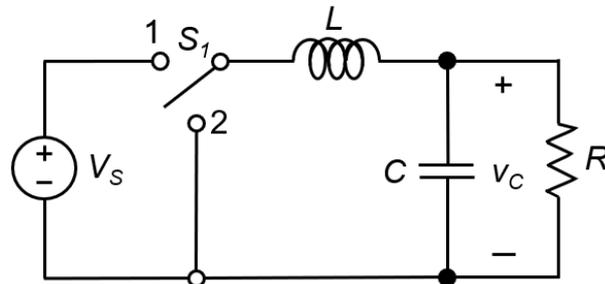


Fonte: Adaptado de Lucas (2015, p. 20).

Tanto o transistor  $Q$  quanto o diodo  $D$  funcionam como duas chaves bidirecionais, e podem ser representadas pelo circuito da Figura 7, onde o transistor

$Q$  atua como uma chave controlada pelo ciclo de trabalho enviado pelo PWM, e o diodo  $D$  como uma chave não controlada (RASHID, 2014).

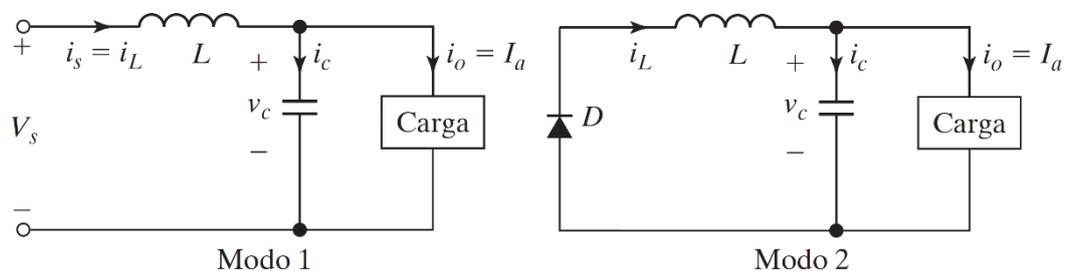
Figura 7 - Circuito do conversor *Buck* representado o transistor como chave



Fonte: Rashid (2014, p. 201).

Seu funcionamento é baseado pelo chaveamento do transistor, representado por uma chave como  $S_1$  na Figura 7 e sua operação pode ser dividida em duas etapas ou modo1 e modo2, conforme estão representados na Figura 8. O modo 1 começa com o acionamento do transistor, fechando a chave no instante de tempo nulo, passando a conduzir. Nesse momento o diodo estará bloqueado, inversamente polarizado, não influenciando no circuito. Devido a tensão de entrada ser maior que a tensão de saída, a corrente de entrada  $i_s$ , começa a circular através do indutor dando origem a corrente  $i_L$  conhecida como corrente de magnetização, e seu valor vai aumentando linearmente com o tempo enquanto a chave permanecer fechada, por um período que é chamado de  $t_{on}$ . Da mesma forma, essa corrente é distribuída para o carregamento do capacitor e para o resistor de carga (RASHID, 2014).

Figura 8 - Modos de operação do Conversor *Buck* – M1 chave aberta, M2 fechada

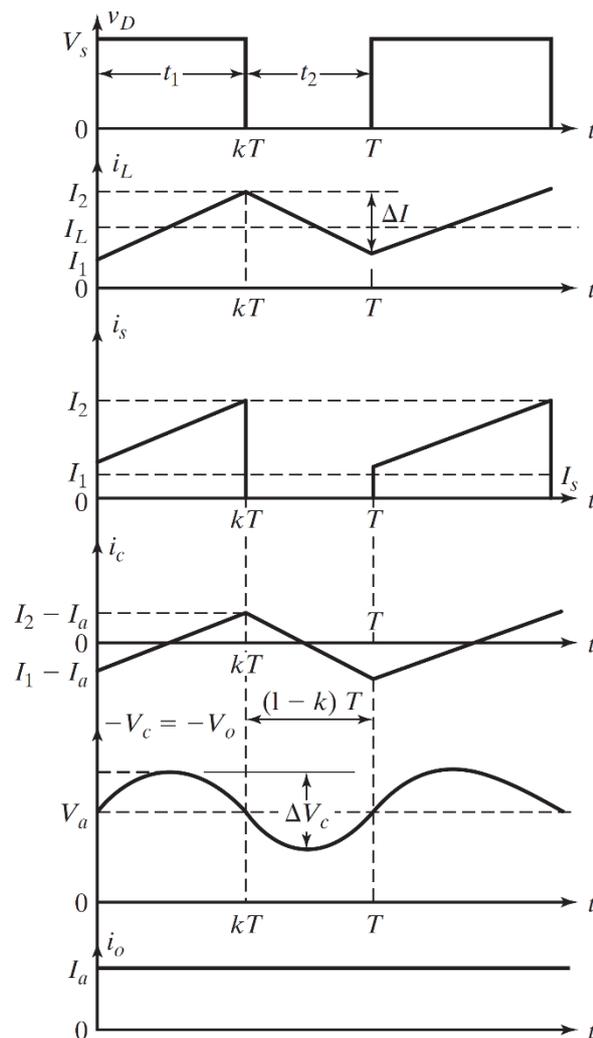


Fonte: Rashid (2014, p. 201).

O modo 2 inicia-se com a abertura da chave  $S_1$ , caracterizando o período em que ela estiver nessa fase por  $t_{off}$ . Neste instante, o diodo de roda livre entra em ação permitindo a continuidade da corrente pelo indutor. Essa corrente que atingiu o seu

valor máximo no modo 1, neste modo 2 passará a decrescer também de forma linear, onde damos o nome de corrente de desmagnetização. Se a corrente não chegar a zero, dizemos que o conversor atuou em modo de condução contínua (MCC), e caso seu valor atinja o zero, dizemos que o conversor atuou em modo de condução descontínua (MCD). A corrente armazenada no indutor é composta de uma componente média e de uma componente alternada, onde a componente alternada é entregue ao capacitor e a componente média à carga. Quando o indutor apresentar corrente menor que a carga, o capacitor começará a se descarregar para manter constante essa corrente. Assim, o ciclo dos dois modos estará completo, e tudo volta a acontecer no instante em que a chave  $S_1$  é novamente acionada. Essa sequência de etapas está ilustrada na Figura 9 (BARBI, 2015; MELLO, 2011; RASHID, 2014).

Figura 9 - Formas de onda da tensão e corrente no conversor *Buck*



Fonte: Rashid (2014, p. 201).

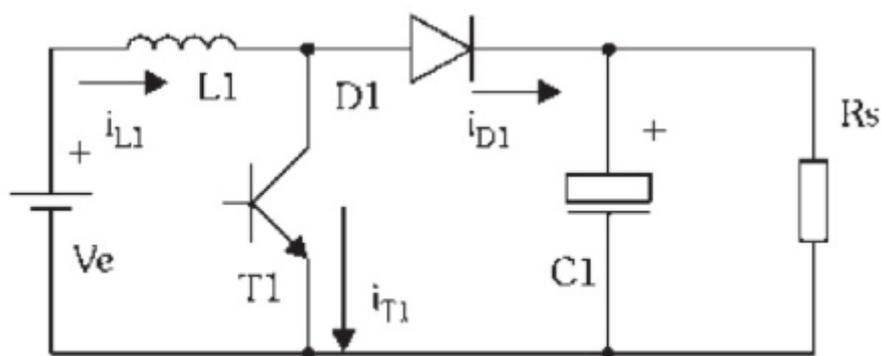
### 2.1.4 Conversor *Boost*

O conversor CC-CC *Boost (step-up)*, assim como o *Buck* é bastante utilizado na indústria e mais recentemente, ele vem ganhando espaço no setor de energia solar como elevador e regulador de tensão dos painéis solares ou fotovoltaicos, assim como pode mediante uma modificação de sua topologia trabalhar como carga eletrônica, que será objeto de estudo deste trabalho. Basicamente ele fornece na saída do circuito uma tensão CC maior do que aquela que é aplicada na sua entrada, por isso o chamam de elevador de tensão. Por ele ter na sua construção o indutor colocado em série com a entrada de alimentação, conforme é mostrado na Figura 10, assim ele é caracterizando como uma fonte de corrente (ARRABAÇA; GIMENEZ, 2014).

Segundo Mello (2011, p. 43), “[...] A tensão de saída possui mesma polaridade que a tensão de entrada. O ruído gerado para a saída é alto, devido aos pulsos de corrente fornecidos ao capacitor. O ruído gerado para a entrada é baixo, sem pulsos devido ao indutor”.

A Figura 10, mostra de forma geral a topologia típica do conversor *Boost* ideal.

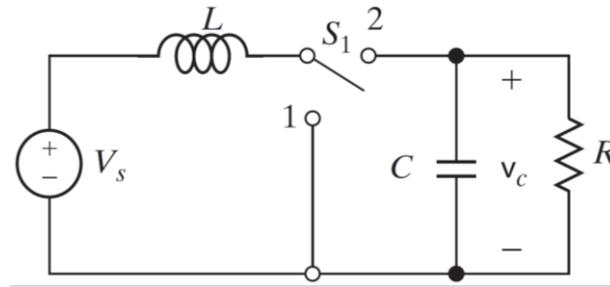
Figura 10 - Topologia típica do conversor *Boost*



Fonte: Mello (2011, p. 44).

O circuito da Figura 10 é muitas vezes representado por uma chave de dois terminais, conforme pode ser visto na Figura 11.

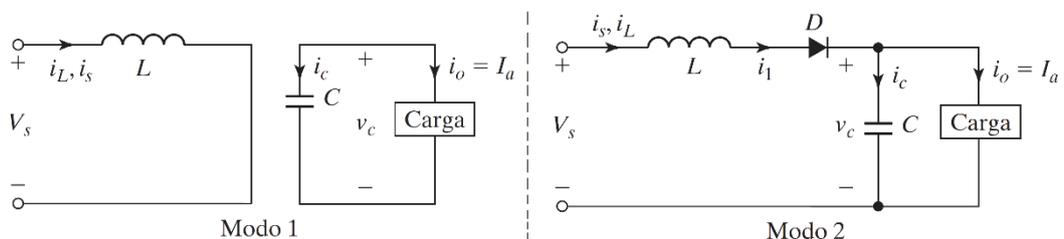
Figura 11 - Circuito do conversor *Boost* representando o transistor como chave



Fonte: Rashid (2014, p. 204).

Assim como no *Buck*, a operação do circuito em modo de condução contínua pode ser dividida em dois modos, conforme estão representados na Figura 12.

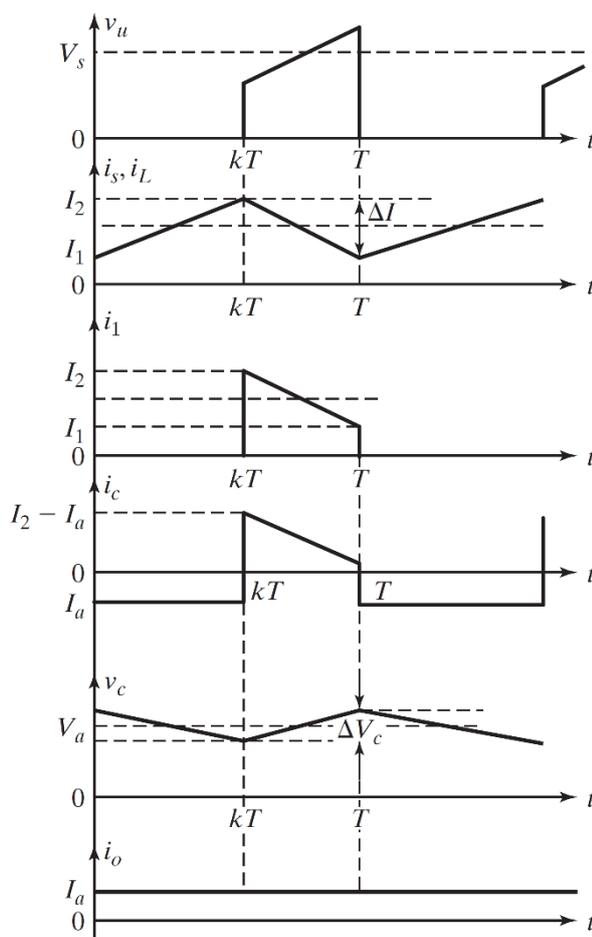
Figura 12 - Modos de operação do Conversor *Boost*



Fonte: Rashid (2014, p. 204).

No modo 1 o transistor estará em condução e passará a ser percorrido pela mesma corrente do indutor  $i_L$ , o indutor será conectado diretamente a alimentação, recebendo a tensão  $V_s$  instantaneamente, porém a corrente de magnetização vai aumentando de maneira gradual. O diodo  $D$  isola a carga na saída, pois encontra-se reversamente polarizado, apresentando-se como um circuito aberto. A razão cíclica para o acionamento do transistor deve ser limitada para evitar a saturação do indutor por corrente, impedindo-o de se tornar um curto-circuito (MELLO, 2011).

Já no modo 2, quando o transistor passar para o estado de não condução como uma chave aberta, o diodo conduzirá a corrente do indutor permitindo que a energia armazenada no indutor seja transferida à carga e ao capacitor. A corrente que o capacitor recebe do indutor deve ser suficiente para repor sua carga, perdida durante o momento em que ficou alimentando a carga, para manter a tensão de saída sem grandes ondulações. A tensão de saída é superior à tensão de entrada, pelo fato de que a tensão induzida no indutor troca de polaridade, somando-se assim com a fonte de alimentação (MELLO, 2011). Na Figura 13 é possível verificar as formas de ondas de um conversor *Boost* (ARRABAÇA; GIMENEZ, 2014).

Figura 13 - Formas de onda do conversor *Boost*

Fonte: Rashid (2014, p. 204).

### 2.1.5 Carga Eletrônica CC

Atualmente os fabricantes de fontes de alimentação de energia precisam recorrer a equipamentos que tenham a capacidade de testar e avaliar as características de resposta em regime permanente e seus transientes, onde o uso da carga eletrônica CC é uma alternativa muito viável para emular cargas de resistências de corrente contínua ou corrente alternada, pois contemplam uma alta gama de aplicações como executar testes funcionais de baterias, fontes de alimentação ou células solares fotovoltaicas, onde estes testes dinâmicos e a simulação do seu funcionamento é imprescindível para aferir o produto antes dele ir ao mercado (BK PRECISION, 2017).

Uma das características do funcionamento de uma carga eletrônica, consiste em drenar de uma fonte de alimentação uma corrente constante, simulando uma carga resistiva, onde um sinal de referência determina a quantidade de corrente que

será drenada. A fonte em teste deve ser capaz de seguir o sinal de referência, como rampas de corrente, para verificar o que ocorreria com uma carga real conectada em suas extremidades, apresentando esse comportamento (DE FARIA *et al.*, 2010).

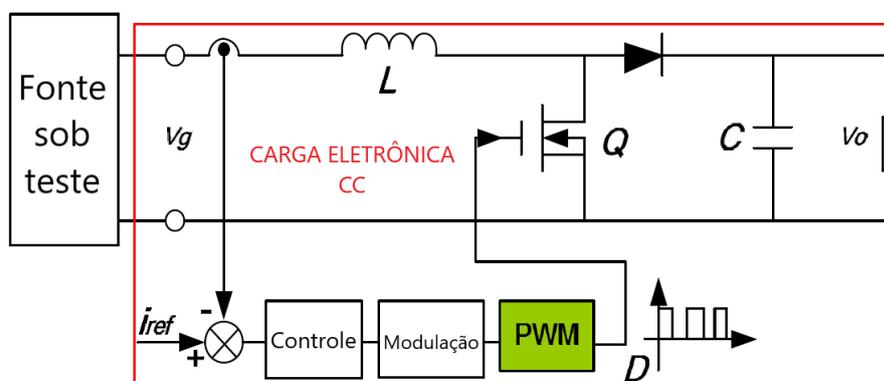
Segundo Bk Precision (2017), uma carga eletrônica CC pode dispor de quatro modos de operação conforme o modelo comercial ou artesanal que foi projetado, sendo eles, modo de corrente constante, modo de tensão constante, modo de resistência constante e modo de potência constante.

Devido à característica construtiva do conversor *Boost* em ter o indutor ligado em série com a entrada e se o seu valor de indutância for projetado para que ele permaneça em modo de condução contínua, ele se destaca como uma topologia mais adequada para ser usada como carga eletrônica, pois se comporta como uma fonte de corrente oferecendo e controlando de maneira eficiente essa variável de entrada (HU *et al.*, 2014).

Uma das técnicas para reduzir o *ripple* de corrente na entrada da carga eletrônica bem como demandar de um indutor com valor de indutância menor é a utilização de filtros LC. Algumas fontes que são colocadas em teste com a carga eletrônica não admitem grandes variações na sua corrente de saída. Um indutor de entrada apresenta característica de baixo ruído, reduzindo a distorção harmônica da corrente. O cálculo do indutor de entrada deve ser feito considerando um *ripple* que equilibre o dimensionamento do indutor (DE FARIA *et al.*, 2010).

A Figura 14 apresenta uma topologia de carga eletrônica com base no conversor CC-CC *Boost* proposto por Hu *et al.* (2014).

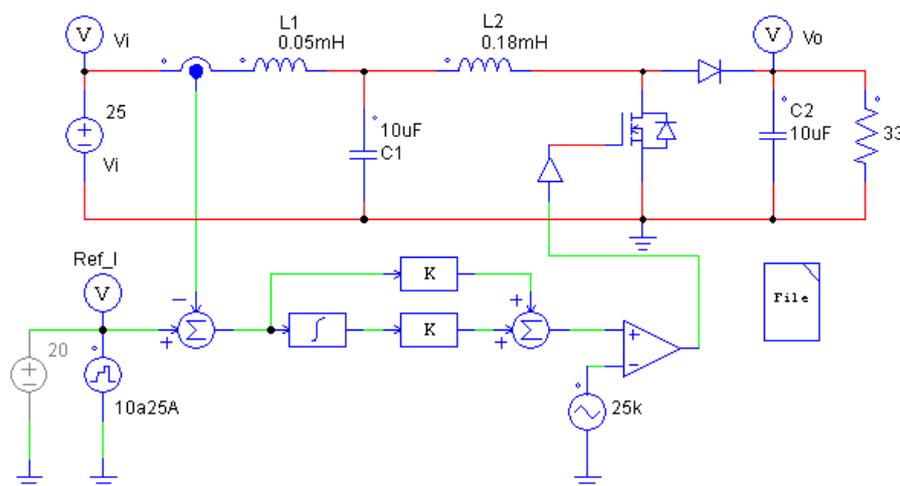
Figura 14 - Topologia de uma carga eletrônica baseada em um conversor *Boost*



Fonte: Hu *et al.* (2014, p. 37).

No seu trabalho, os autores apresentam um modelo de carga que trabalha para obter a corrente de entrada constante, utilizando um MOSFET para chaveamento, e sem filtro  $LC$  na entrada. Na Figura 15 é possível visualizar um modelo de carga eletrônica com base na topologia *Boost*, a qual será utilizada neste trabalho, onde destaca-se na sua entrada o filtro  $LC$  para amenizar o *ripple* de corrente.

Figura 15 – Circuito da carga eletrônica utilizada como processo neste trabalho



Fonte: Elaborado pelo autor.

Os conversores CC-CC permitem regular a tensão de saída com erro nulo em regime permanente e permitem em geral encontrar boas respostas transitórias em termos de tempo de acomodação e de *overshoot*. O projeto dos ganhos do PID pode ser feito por técnicas de controle clássico, como lugar das raízes e resposta em frequência. Entretanto, o projeto otimizado dos ganhos de PIDs ainda é um ponto de interesse, especialmente quando o conversor é sujeito a variações paramétricas (MATTOS *et al.*, 2021).

Neste sentido, uma sintonia adequada de um controlador permite uma boa relação de compromisso entre desempenho e robustez para o conversor operando em malha fechada.

## 2.2 Controladores PID

Åström e Hägglund (2001), explicam que controladores são equipamentos responsáveis em manter variáveis de um processo em valores pré-determinados, conhecidos como *set-point's*, a partir de um algoritmo geralmente proporcional a uma ou mais variáveis que são medidas em tempo real por um equipamento de controle

geralmente baseado em microprocessador. Um sistema de controle é o arranjo de componentes para que a planta tenha uma resposta com características pré-definidas, sendo em malha aberta (atuador manipula diretamente um processo) ou malha fechada (onde existe um sinal de retroação) (DORF; BISHOP, 2001).

O controlador PID (Proporcional, Integral e Derivativo) é ainda hoje predominante no meio industrial. Mais de 90% de todas as malhas existentes são do tipo PI / PID atingindo uma larga faixa de aplicações: controle de processos, drivers para motores, indústria automobilística, controladores de voo, pilotos automáticos, instrumentação industrial, entre outros (CAMPOS; TEIXEIRA, 2010). Sua popularidade está na simplicidade do ajuste de parâmetros para obter um bom desempenho e do fato deste algoritmo estar disponível em quase todos os equipamentos de controle da indústria. Dependendo do fabricante, podem existir algumas diferenças de implementação prática desse algoritmo, mas a essência do controlador PID é a mesma (ISERMANN, 1989). O controle PID tem uma estrutura simples e geralmente está disponível como um pacote. Para ter um bom desempenho com os problemas do processo industrial, o controlador deve ter valores ajustados e otimizados (NETO; GOMES, 2010).

“Em particular, quando o modelo matemático da planta não é conhecido e, métodos de projeto analítico não podem ser utilizados, controles PID se mostram os mais úteis” (OGATA, 2011). Ainda segundo o autor, o controle com retroalimentação, apesar de apresentar maior complexidade, possui algumas vantagens, entre elas a manutenção da qualidade do sinal de saída sem a necessidade de ajuste manual. Sua desvantagem é que é necessário ajustar os três parâmetros, o ganho proporcional ( $K_p$ ), o ganho integral ( $K_i$ ) e o ganho derivativo ( $K_d$ ), dificultando a sincronização. O projeto do controlador é exclusivo para a planta à qual será aplicado. Caso seja utilizado em outra planta com uma dinâmica diferente é necessário reprojeter o controlador.

A expressão que representa a estrutura clássica PID no tempo é dada pela equação (4):

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (4)$$

Aplicando a transformada de Laplace, podemos obter a função de transferência do controlador PID, em função das constantes de tempo de integração ( $T_i$ ) e do tempo derivativo ( $T_d$ ), como mostra a equação (5).

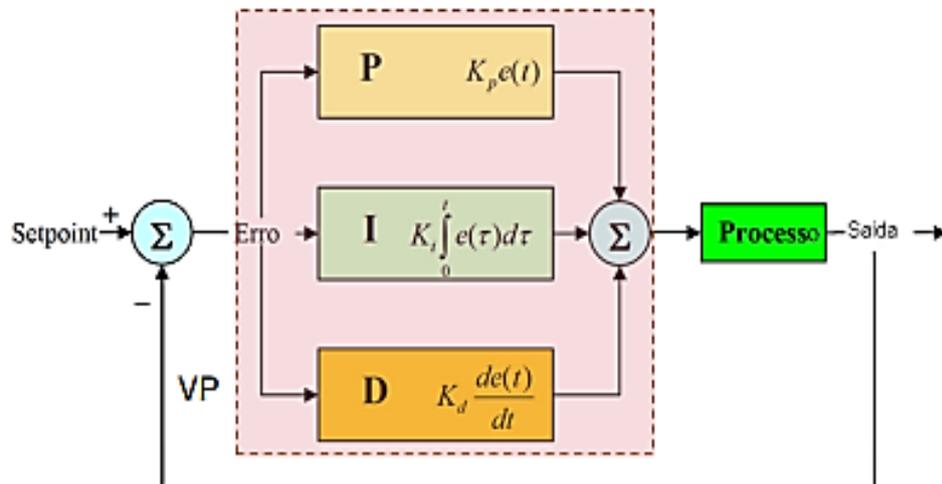
$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (5)$$

Há outras formas de se definir o controlador PID em forma discreta. Uma forma alternativa pode ser definida partindo-se da função de transferência em tempo contínuo, equação (4) para chegar na equação (6).

$$G(s) = K_p + K_i \frac{1}{s} + K_d s \quad (6)$$

A Figura 16, apresenta o diagrama de blocos para um controle PID de um processo a ser otimizado.

Figura 16 - Diagrama de blocos do controle PID



Fonte: Mehta *et al.* (2017).

A ação mais simples do controlador PID é a ação proporcional (P), que atua no erro entre a referência e a saída. O seu incremento diminui o erro em regime permanente, mas não o anula, diminui a margem de fase e desloca a curva de magnitude acelerando o sistema, porém dependendo do valor do ganho poderá provocar oscilações, aumento do *overshoot* e levar o sistema à instabilidade.

A ação integral (I), por sua vez, tem a função de corrigir o erro nulo em regime permanente, com a parcela integral “armazenadora de energia”, que armazenará a soma de todos os erros ao longo do tempo. Isto garante que o sinal de controle aplicado seja mantido em um valor constante proporcional a energia armazenada, possibilitando o erro nulo em regime permanente, porém com a introdução de um polo na origem o sistema tem sua ordem aumentada e conseqüentemente torna-se mais lento e instável.

A ação derivativa (D) se relaciona com a tendência de variação da função no decorrer do tempo. Esta ação se antecipa aos distúrbios ou perturbações na entrada, provocando uma reação mais rápida do sistema, acelerando a resposta transitória e diminuindo o *overshoot*. Não possui influência no regime permanente (OGATA, 2011).

Devido ao uso generalizado do controle PID em processos industriais, múltiplas técnicas têm sido propostas para ajustar os parâmetros proporcional, integral e derivativo, onde este problema é conhecido como o problema de sintonia. Ziegler e Nichols (1942) foram os criadores das regras de sintonia, sendo importantes pela simplicidade da aplicação, no entanto apresentavam limitações no desempenho atingido. Tornando necessário mais investigações acerca da sintonia dos controladores PID (NETO; GOMES, 2010).

A sintonia de um controlador é de suma importância visto que se trata da determinação dos parâmetros utilizados em um controlador PID. São justamente esses parâmetros que definem um bom controle de um processo contínuo (DIAS *et al.*, 2015).

O problema da sintonia ótima do controlador PID pode ser descrito da seguinte forma: Definido um índice de desempenho para o sistema a ser controlado, determinar os ganhos do controlador PID de maneira que, dentro do intervalo de tempo especificado, a resposta transitória do sistema de controle composto pelo processo e o controlador, seja satisfatória para minimizar os valores dentro de certas especificações de projeto. Assim, quanto melhor for o controlador, menor serão os valores destas medidas de desempenho (OLIVEIRA, 2020).

Quando soluções analíticas podem acarretar um custo elevado no desenvolvimento de um controlador PID, as sintonias baseadas em meta-heurísticas tais como: algoritmos genéticos (GA), enxame de partículas (PSO), redes neurais artificiais (RNAs) são utilizadas, com o objetivo de encontrar soluções de qualidade num tempo viável de determinado espaço de busca, com baixo custo de obtenção (HUANG; LAM, 1997).

### 2.2.1 Controlador PI

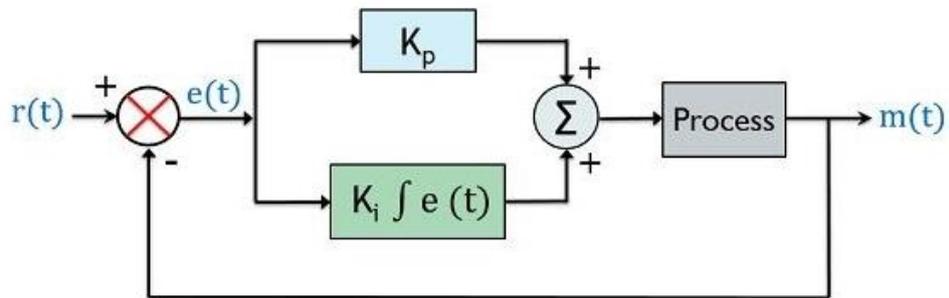
Este controlador é uma variante do controlador PID, sem a ação derivativa. é definido com base na sua função de transferência, representada pela equação (7).

$$G_c(s) = K_p \left( 1 + \frac{1}{T_i s} \right) \quad (7)$$

Ele é um compensador de atraso de fase, que atua em baixas frequências corrigindo a instabilidade do sistema ocasionada pela elevação do ganho para correção do erro em regime permanente. Ele adiciona um polo na origem em  $s = 0$  e um zero em  $s = -1/T_i$ . Logo, a característica do controlador PI é possuir ganho infinito na frequência nula. Para evitar que o sistema se torne instável, valores de  $K_p$  e  $T_i$  devem ser escolhidos cuidadosamente para garantir uma resposta temporal apropriada. Pode-se evitar ou diminuir muito o *overshoot* de uma resposta temporal, projetando-o de maneira adequada. Devido sua característica de filtro passa-baixa, atenuando as componentes de alta frequência do sinal, ele apresenta uma velocidade de resposta mais lenta. A introdução de controladores PI em um sistema de controle, faz o erro em regime permanente do sistema ficar extremamente reduzido sem afetar a estabilidade do sistema (OGATA, 2011).

O diagrama de blocos desse sistema é mostrado na Figura 17

Figura 17 - Diagrama de blocos de um sistema de controle com controlador PI

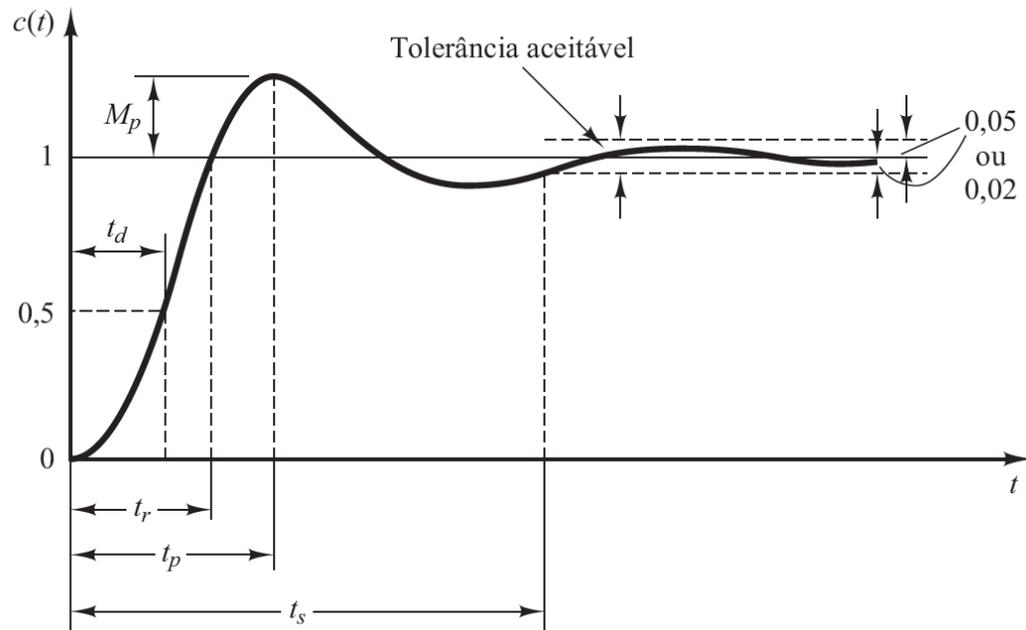


Fonte: Coach (2022).

### 2.2.2 Características das respostas transitórias de um sistema de controle

Segundo Ogata (2011), antes de atingir o regime permanente, a resposta transitória de um sistema de controle apresenta, frequentemente, oscilações amortecidas com características que medem seu desempenho. Estas especificações são detalhadas com base na Figura 18.

Figura 18 - Curva da resposta transitória em degrau unitário



Fonte: Ogata (2011, p. 155).

Tempo de subida ( $t_r$ ) - Tempo decorrido até a variável resposta atingir o valor de referência. Normalmente entre 10% a 90%;

Sobressinal ( $M_p$ ) - neste trabalho, será adotado para esta especificação a palavra *Overshoot* com o símbolo ( $\%UP$ ) - que significa o percentual de ultrapassagem que a variável resposta atingiu em relação ao valor de referência desejado para o regime estacionário, onde esse valor é dado pela razão entre eles.

Tempo de pico ( $t_p$ ) - instante de tempo no qual ocorre o maior valor na resposta

Tempo de Acomodação ( $t_s$ ) - tempo gasto para que a resposta do sistema estabilize e permaneça dentro de uma faixa pré-estabelecida (normalmente entre 2% e 5%) em torno da amplitude de entrada.

### 2.2.3 Índices de Desempenho

Segundo Dorf e Bishop (2001, p. 199), "Um índice de desempenho é uma medida quantitativa do desempenho de um sistema e é escolhido de modo que a ênfase seja dada às especificações de sistemas importantes". São métodos baseados em critérios de integração do erro, onde a curva de resposta completa em malha fechada é utilizada para encontrar os parâmetros de sintonia. Para a operação de sistemas de controle adaptativos modernos, se faz necessária a medida quantitativa

do desempenho de um sistema. Esta consideração da resposta completa é atingida quando os parâmetros do sistema são bem ajustados, de modo que o índice de desempenho atinja o menor valor possível (minimização).

A seguir serão apresentados os índices de desempenho (IAE, ITAE, ISE e o ITSE) dos controladores PID, baseados na integral do erro

Nas integrais, considera-se a integração desde o  $T = 0$ , ponto onde ocorre o início da perturbação, e o tempo  $T = \infty$ , que seria o tempo em que ocorre a estabilização ou tempo de acomodação ( $t_s$ ) da resposta final.

IAE - Integral do Erro Absoluto (*Integral of the Absolute Error*).

$$IAE = \int_0^T |e(t)| dt \quad (8)$$

Esse índice é bastante útil e simples de ser implementado, é equivalente a soma das áreas acima e abaixo do valor de referência. A integral do módulo do erro, considera erros com valor absoluto pequeno como ruídos. Em sistemas que são configurados com esse índice minimizado, como sistemas analógicos, possuem um amortecimento razoável e uma característica de resposta transitória satisfatória. Apresenta baixa seletividade e pesos iguais para erros independentes do tempo em que ocorrem (PAIVA, 2010).

ITAE - Integral do Erro Absoluto ponderada pelo Tempo (*Integral of Time multiplied by Absolute Error*).

$$ITAE = \int_0^T t|e(t)| dt \quad (9)$$

O critério ITAE baseia-se na ponderação do erro no tempo. Ele perdoa erros com tempo curto, reduzindo a contribuição do erro inicial, e penaliza erros que persistem por longos períodos do regime permanente. Quanto maior o tempo, maior o produto  $t|e(t)|$  (tempo multiplicado pelo módulo do erro). No início, apesar do erro ser maior, como há a ponderação no tempo, o termo somado tende a ser pequeno. Assim, quanto maior o tempo, maior o termo somado na integral, a depender do valor do erro (PAIVA, 2010).

ISE - Integral do Erro Quadrático (*Integral of the Squared of the Error*).

$$ISE = \int_0^T e^2(t)dt \quad (10)$$

Usado em sistemas superamortecidos e subamortecidos. Tem a tendência de penalizar erros maiores, que ocorrem no início da resposta, pois o erro é elevado ao quadrado, e para erros que ocorrem em tempos maiores, estes serão perdoados por serem menores. Em sistemas que são configurados com esse índice minimizado, tendem a ter uma resposta mais rápida, porém podem apresentar um *overshoot* elevado e uma resposta oscilatória com baixa estabilidade (PAIVA, 2010).

ITSE - Integral do Erro Quadrático ponderado pelo Tempo (*Integral of Time multiplied by the Squared Error*).

$$ITSE = \int_0^T te^2(t)dt \quad (11)$$

Usa o tempo como um valor ponderador do peso de cada erro, reduzindo a contribuição do erro inicial, e coloca ênfase nos erros mais atrasados. Ou seja, ele penaliza erros grandes para tempos elevados, e perdoados erros pequenos em tempos curtos (PAIVA, 2010).

### 2.3 Ferramentas de Otimização

Neste capítulo é apresentada uma revisão sobre as origens e características das meta-heurísticas como uma área de pesquisa que se insere no ramo das técnicas de Inteligência Artificial (IA) propostas neste trabalho, dentre as quais o algoritmo genético (GA) e a otimização por enxame de partículas (PSO) são utilizadas para encontrar uma solução otimizada na obtenção dos parâmetros das funções gaussianas para os ganhos de um controlador PID adaptativo.

Segundo Goldberg, M., Goldberg, E. e Luna (2016, p. 72), o conceito de meta-heurística é definido como: “Trata-se de uma arquitetura geral de regras que, formada a partir de um tema em comum, pode servir de base para o projeto de uma ampla gama de heurísticas computacionais.”

Os autores ainda comentam, que devido à dificuldade dos computadores e algoritmos atuais terem de achar uma resposta exata para problemas difíceis e de grande porte, é que nas últimas décadas foram concentrados grandes esforços em

pesquisa e desenvolvimento, no traçado de estratégias que aproximem as respostas de soluções eficientes para problemas difíceis e de grande porte.

Essas estratégias podem ser denominadas de meta-heurísticas, que nada mais são que procedimentos de alto nível em busca de soluções de alta qualidade para problemas difíceis, que varrem um espaço de busca examinando cada região, em busca de um mínimo global. Elas podem ser desenvolvidas arbitrariamente e buscar base em analogias como por exemplo, as meta-heurísticas bioinspiradas, que derivam suas regras para a formação de heurísticas da cópia de processos biológicos, como os GAs e PSOs.

### 2.3.1 Algoritmo Genético GA

O algoritmo genético (GA) é o mais conhecido algoritmo de otimização meta-heurística, desenvolvido por *John Holland* inspirado na evolução darwiniana da teoria da evolução das espécies, na qual a seleção natural é proposta (PAIVA, 2010).

São métodos de busca estocásticos que utilizam regras de probabilidade para realizar a procura num espaço de soluções que seja provável que se encontrem os pontos ótimos. Executando operações de seleção, *crossover* e mutação, o algoritmo desenvolve uma população de soluções candidatas em evolução. A técnica parte da ideia de que cada solução de um problema definido é codificada em um indivíduo e seus genótipos estão diretamente relacionados aos parâmetros livres do modelo selecionado. Uma função de aptidão (*fitness*) deve ser definida para medir o grau de adaptação do indivíduo ao ambiente, que é a expressão da função de custo a ser otimizada. O processo iterativo entre a função de aptidão e os operadores genéticos leva os indivíduos a mudar seu genótipo. Selecionando operadores bem projetados e parâmetros ideais, os GAs exibiram um alto grau de robustez em termos de encontrar soluções ideais para problemas de otimização difíceis. A robustez resulta do processo evolutivo ser capaz de avançar soluções que seriam muito difíceis de prever a priori, embora os GAs muitas vezes requeiram um grande número de avaliações de função. Este processo permite a exploração do espaço de busca pelo método, a fim de encontrar boas soluções (GOLDBERG, 1989).

Para o entendimento do texto sobre GA, faz-se necessário uma explanação sobre algumas terminologias que se assemelham da genética natural ou biológica e que serão resumidas no Quadro 1.

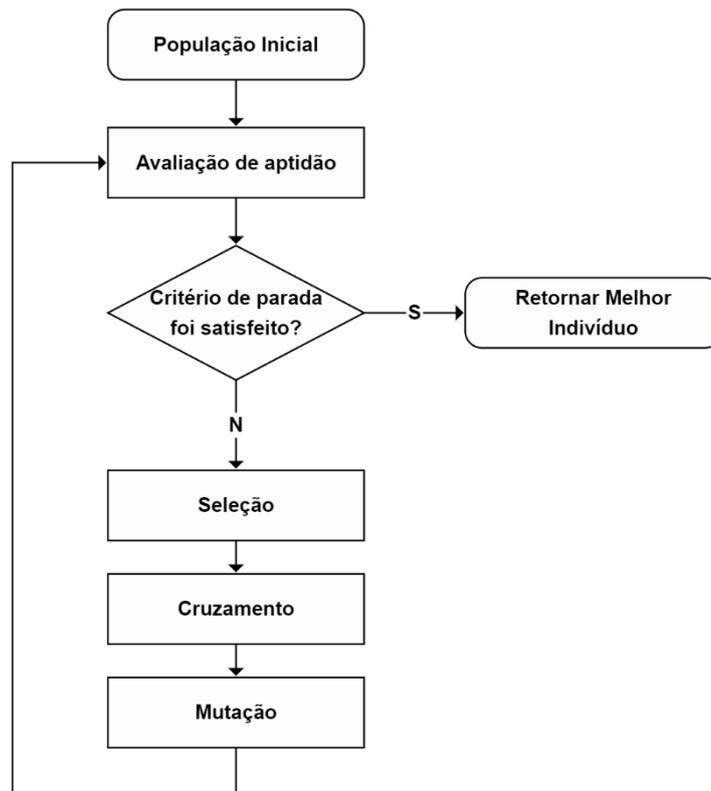
Quadro 1 – Relação entre Biologia e Problema de Otimização

<b>Biologia / Natureza</b>	<b>Problema de Otimização no GA</b>
Indivíduo – cromossomo	Solução viável do problema
População	Conjunto de soluções
Adequação ( <i>fitness</i> )	Valor (ou custo) da solução
Gene	Parte da solução
Cruzamento ( <i>crossover</i> ) / Mutação	Operadores de busca
Seleção natural	Seleção de soluções

Fonte: Adaptado de Goldberg, M., Goldberg, E. e Luna (2016, p. 102).

A Figura 19 apresenta um fluxograma do algoritmo genético (GA), detalhando suas etapas de execução.

Figura 19 - Fluxograma do Algoritmo Genético (GA)



Fonte: Adaptado de Souza, Valiati e Gómez (2012, p. 2604).

**População Inicial:** É criada aleatoriamente dentro de um espaço de busca definido, contendo indivíduos representados por cromossomos e que estão em constante evolução através de sucessivas gerações, podendo ser ou não, uma proposta de solução para o problema em estudo, através das regras de seleção e dos

operadores genéticos. A população é definida por um tamanho experimental, que consiga cobrir o espaço de busca sem demorar tempo excessivo de processamento computacional (GOLDBARG, M.; GOLDBARG, E.; LUNA, 2016).

**Avaliação de Aptidão:** Utiliza-se uma função de custo (*fitness*) para avaliar cada indivíduo da população dando pesos diferentes de acordo com sua adaptação na resolução do problema. Estes indivíduos são classificados entre aptos e inaptos, sendo os aptos selecionados a passarem para a reprodução na busca de um resultado ainda melhor, e os inaptos são descartados. Esse movimento ocorre a cada geração, até que um critério de parada seja atingido, como por exemplo, encontrar uma tolerância em torno da resposta esperada ao problema, um tempo de execução ou número máximo de iterações seja alcançado. Quando isso ocorre, o algoritmo entende que foi encontrado um indivíduo que melhor atende a função objetivo e apresenta seus valores, encerrando a busca. Caso o critério de parada não seja alcançado o algoritmo passa novamente por mais uma evolução pelos operadores genéticos em busca de um resultado melhor (GOLDBARG, M.; GOLDBARG, E.; LUNA, 2016).

**Seleção:** Nesse processo, é levado em conta a aptidão dos indivíduos, que no caso dos mais aptos apresentem maior probabilidade de serem selecionados e reproduzidos, passando seus genes para próxima geração. Essa seleção pode ser feita através de processos como a roleta, o torneio etc., onde dois indivíduos são selecionados para o cruzamento. A Figura 20 ilustra as etapas de seleção dos pais para reprodução de uma nova geração de filhos mais aptos (GOLDBARG, M.; GOLDBARG, E.; LUNA, 2016).

Figura 20 - Seleção dos Pais para Reprodução no GA

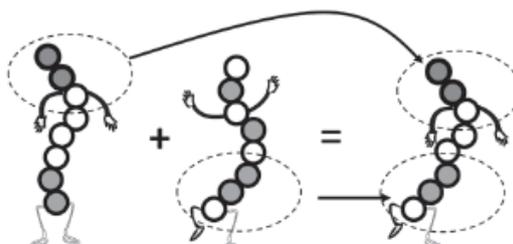


Fonte: Goldberg, M., Goldberg, E. e Luna (2016, p. 102).

**Cruzamento (*crossover*):** Nessa etapa uma nova geração de filhos é criada com base nas características genéticas advindas dos pais que passaram pelo processo de seleção. O processo de cruzamento pode acontecer por diversos métodos, no caso da representação binária o cruzamento acontece com troca de material genético dos

pais, conforme é ilustrado na Figura 21, onde é definido um ponto de corte nos seus cromossomos e feita uma mistura entre eles para geração de um novo indivíduo (GOLDBARG, M.; GOLDBARG, E.; LUNA, 2016).

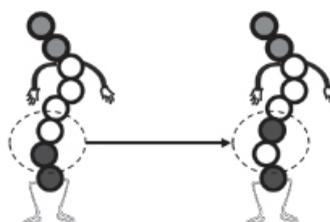
Figura 21 - DNA sendo misturado no cruzamento do GA



Fonte: Goldberg, M., Goldberg, E. e Luna (2016, p. 102).

**Mutação:** Devido ao grau de aleatoriedade do GA durante uma busca em um espaço pré-determinado, podem ocorrer situações de que no problema sejam apresentados espaços de busca como mínimos locais, fazendo com que o algoritmo fique preso nesses espaços convergindo para uma resposta ruim. Sendo assim a mutação permite modificar aleatoriamente um gene de alguns indivíduos de forma que se consiga um aumento na diversidade da população. A taxa de mutação normalmente é baixa cerca de 10% para ajudar na convergência, pois se a mutação do indivíduo for grande, pode ocorrer dele ser descartado nos próximos eventos de avaliação de aptidão, seleção e cruzamento (GOLDBARG, M.; GOLDBARG, E.; LUNA, 2016). A Figura 22 apresenta o efeito da mutação num cromossomo durante a execução do GA.

Figura 22 – Efeito da Mutação de um gene num cromossomo do GA



Fonte: Goldberg, M., Goldberg, E. e Luna (2016, p. 102).

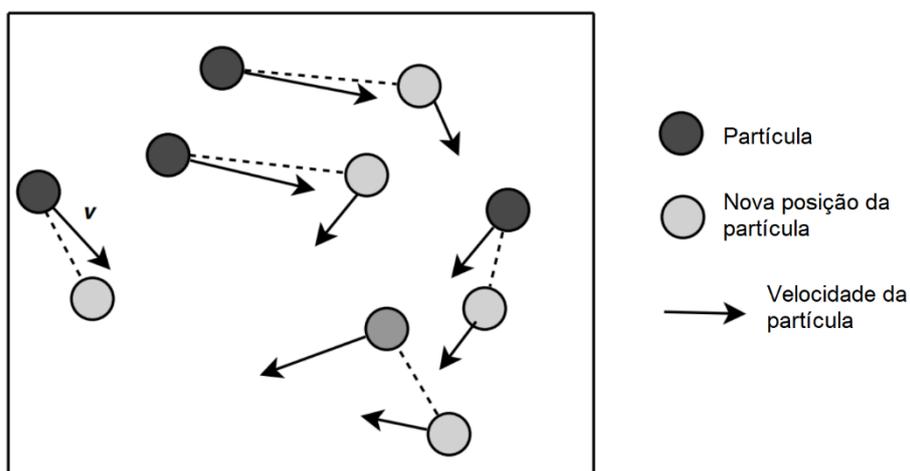
### 2.3.2 Algoritmo Enxame de Partículas PSO

Desenvolvido em 1995 por James Kennedy (psicólogo social) e Russell Eberhart (engenheiro eletricitista), PSO é um método de busca heurística com uma técnica de otimização estocástica robusta. Inspirado na interação social, no movimento e na inteligência de animais que vivem em grupos de enxames, como bandos de pássaros e cardumes de peixes. Utiliza da experiência social do grupo para solucionar um problema de forma coletiva (KENNEDY; EBERHART, 1995).

O PSO começa com um grupo (enxame) de partículas geradas aleatoriamente que se movem por um espaço, em busca da melhor solução. Cada partícula representa uma possível solução para o problema de otimização, e é caracterizada por duas propriedades principais: posição e velocidade. A posição de uma partícula é a avaliação da função de aptidão da otimização, que representa a distância das partículas da solução ótima ideal (zero em caso de minimização). Da mesma forma, a velocidade de cada partícula determina o movimento da partícula no espaço da solução (OUYANG; PANO, 2015).

A Figura 23, mostra um enxame de partículas com suas posições e velocidades associadas. A cada iteração, uma partícula se move de uma posição para outra no espaço de decisão, e cada posição é comparada entre as partículas e memorizada aquela que for a melhor solução para o problema. Com isso, o restante do enxame de partículas através dessa troca de informações aprende o melhor caminho para resolver o problema, como encontrar comida por exemplo (TALBI, 2009).

Figura 23 – Movimento de Partículas dentro de um espaço de busca

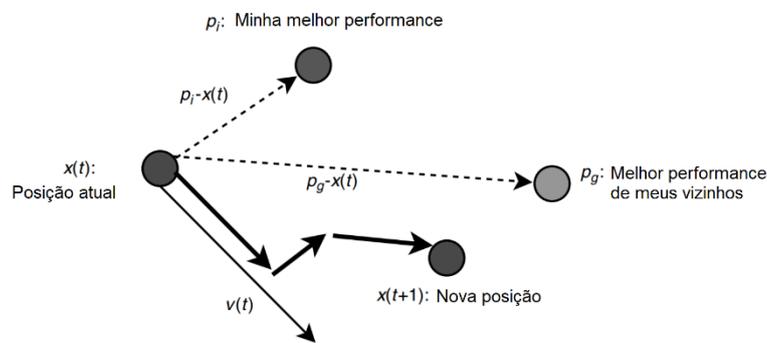


Fonte: Talbi (2009, p. 248).

O melhor valor de cada partícula é registrado na matriz  $p_{best}$  e chamado de melhor posição local ou experiência própria. A cada iteração, é feita uma comparação entre todas as posições de cada partícula e aquela que for melhor, é registrada na matriz global  $g_{best}$  que guarda toda a experiência do grupo, passando a conduzir o exame de partículas para uma solução de convergência coletiva (GANGISHETTI; SANDIPAMU, 2017).

O PSO consiste em acelerar cada partícula em direção às suas localizações  $p_{best}$  e  $g_{best}$ , com uma aceleração ponderada aleatória em cada iteração. A Figura 24 mostra o conceito de mudança de posição no espaço de busca.

Figura 24 - Modificação do ponto de busca no PSO



Fonte: Talbi (2009, p. 250).

As equações (12) da velocidade e (13) da posição descrevem o modelo matemático do PSO.

$$v_{ij}(t + 1) = wv_{ij}(t) + r_1c_1(p_{ij}(t) - x_{ij}(t)) + r_2c_2(g_j(t) - x_{ij}(t)) \quad (12)$$

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1) \quad (13)$$

Onde:

- $v_{ij}(t + 1)$  é a velocidade da partícula na próxima iteração;
- $x_{ij}(t + 1)$  é a posição da partícula na próxima iteração;
- $w$  é o coeficiente de inércia;
- $v_{ij}(t)$  é a velocidade atual da partícula;
- $r_1$  e  $r_2$  são dois números aleatórios no intervalo [0 a 1];
- $c_1$  e  $c_2$  são os coeficientes de aceleração;
- $p_{ij}(t)$  é a melhor solução individual ( $p_{best}$ );
- $x_{ij}(t)$  é a posição atual da partícula;
- $g_j(t)$  é a melhor solução global ( $g_{best}$ ).

Inicialmente quando Kennedy e Eberhart (1995) propuseram o algoritmo PSO, ele era mais simples e não tinha o coeficiente peso de inércia ( $w$ ) e os coeficientes de aceleração ( $c_1$  e  $c_2$ ) eram pesos fixos igual a 2 para obterem uma média igual a 1. Posteriormente Shi e Eberhart (1998) introduziram o peso de inércia na equação geral do PSO, onde observaram que numa faixa entre (0,9 e 1,2) desse parâmetro levava-se o algoritmo a atingir um ótimo global mais rápido.

O algoritmo PSO passou então, a ser estudado por muitos pesquisadores e comunidade acadêmica, que por não conhecerem a fundo seu funcionamento acabavam obtendo propriedades dinâmicas indesejáveis na busca por regiões ótimas de espaços de busca complexos. Neste contexto, buscando um entendimento mais claro, é que os autores Clerc e Kennedy (2002) propuseram um estudo mais aprofundado para explicar o funcionamento do algoritmo, envolvendo análise e sugestões de melhoria para o algoritmo original.

Os autores analisaram a trajetória de uma partícula conforme seu movimento no espaço de busca e apresentaram um conjunto de coeficientes para controlar as tendências de convergência do sistema, modificando o peso de inércia, sendo representados pela equação (14).

$$X = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (14)$$

A relação entre os coeficientes de constrição, utilizados para controlar as características dinâmicas do algoritmo é dada pela equação (15).

$$\varphi = \varphi_1 + \varphi_2 \geq 4 \quad (15)$$

Os valores sugeridos para os parâmetros  $k$  e  $\varphi$  são:  $k = 1$ ;  $\varphi_1 = \varphi_2 = 2,05$ .

Para tornar o algoritmo ainda mais robusto, calcula-se o fator de inércia e os coeficientes de aceleração, a fim de levar o algoritmo a convergir mais rapidamente e ser capaz de resolver um maior número de problemas. Sendo:  $w = X$ ;  $c_1 = X\varphi_1$  e  $c_2 = X\varphi_2$

Aplicando os valores sugeridos nas equações (14) e (15) obtém-se os valores para o fator de inércia e os coeficientes de aceleração:  $w = 0,7298$  e  $c_1 = c_2 = 1,4962$ .

O próximo capítulo descreve a metodologia utilizada neste trabalho.

### 3 METODOLOGIA

Neste capítulo é apresentada a metodologia proposta para o desenvolvimento deste trabalho, sendo detalhados os procedimentos de teste e validação do experimento.

Com o objetivo de comparar quantitativamente o resultado da otimização entre duas meta-heurísticas GA e PSO, na obtenção dos parâmetros para sintonia do controlador PID de cada conversor CC-CC, um do tipo *Buck* e outro do tipo *Boost* (na configuração de carga eletrônica), é que este trabalho apresenta as etapas do seu desenvolvimento.

#### 3.1 Ambiente de Simulação

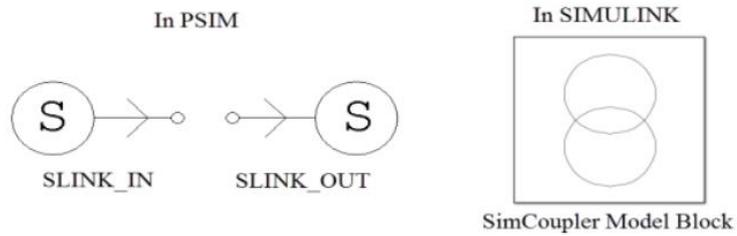
Para realização deste trabalho todas as simulações foram produzidas e realizadas com as ferramentas de software da *MathWorks*<sup>®</sup> (MATLAB<sup>®</sup>/Simulink<sup>®</sup>) e *Powesim*<sup>®</sup> (PSIM). O MATLAB<sup>®</sup> é um ambiente de programação e computação numérica, que foi utilizado para desenvolver os algoritmos e a função objetivo. O Simulink<sup>®</sup> é o ambiente de diagrama de blocos para simulação e *design* de sistemas de engenharia. Ele foi utilizado para desenvolver o circuito de controle PID de cada planta em malha fechada. Ambos na versão 7.12 (R2011a). Já através do *software* PSIM versão 9.0, foi feita a modelagem do circuito de potência das duas plantas (*Buck* e *Boost*), que se conectam por simulação compartilhada.

##### 3.1.1 Simulação compartilhada

Utilizando o módulo *SimCoupler* do PSIM, foi possível fazer o *link* entre PSIM e Matlab/Simulink<sup>®</sup> na simulação compartilhada.

Para realizar esta integração, foram utilizados três módulos conforme estão mostrados na Figura 25. No PSIM, através dos módulos *SLINK\_IN* e *SLINK\_OUT* foi possível receber e enviar respectivamente os sinais ao Simulink<sup>®</sup>, onde este por sua vez, utilizando o bloco do modelo *SimCoupler*, foi possível se conectar com a planta através das portas de entrada e saída, permitindo de forma complementar usar toda a capacidade desses dois *softwares* poderosos.

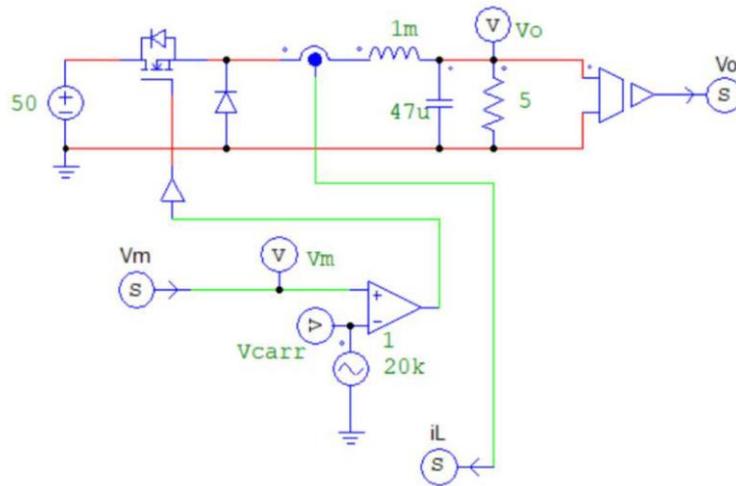
Figura 25 – Bloco *SimCoupler* do PSIM



Fonte: Elaborado pelo autor.

A Figura 26 mostra uma planta de exemplo de um conversor *Buck* no PSIM com os detalhes para entrada e saída dos sinais de controle.

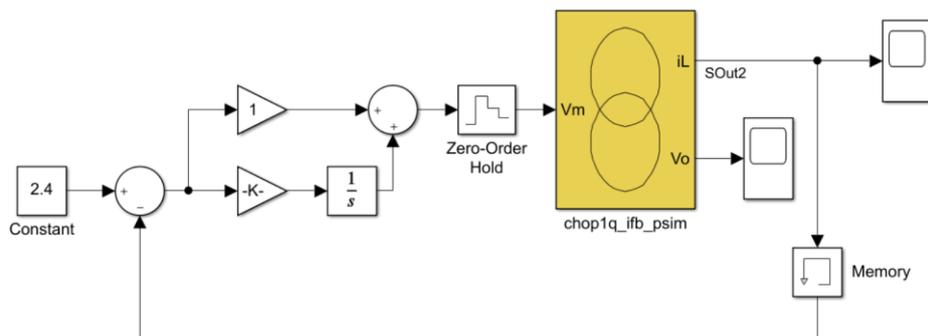
Figura 26 – Exemplo conexão de entrada e saída do módulo *SimCoupler* no PSIM



Fonte: Powersim (2020, p. 4).

A Figura 27 mostra um exemplo de malha de um sistema de controle, que foi interligada com o PSIM através do módulo *SimCoupler* dentro do Simulink®. O fundo do bloco fica em amarelo para se destacar dos demais blocos.

Figura 27 – Exemplo de ligação do módulo *SimCoupler* no Simulink®



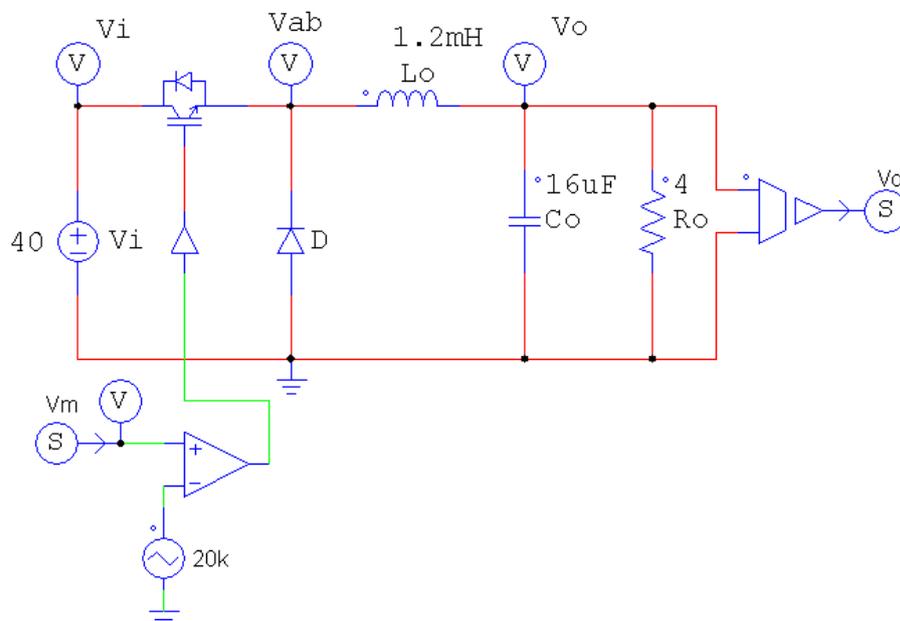
Fonte: Powersim (2020, p. 8).

Outros dois blocos foram adicionados na entrada e saída do *SimCoupler* para melhorar a comunicação entre os dois *softwares*. O bloco *Zero-Order Hold (ZOH)* foi introduzido na porta de entrada do bloco *SimCoupler* e programado seu tempo de amostragem para um valor igual ao passo de tempo do PSIM. Isso se fez necessário pois o tipo de resolução no *solver* do Simulink® foi configurado como passo variável. O bloco *Memory* foi utilizado para quebrar um *loop* algébrico e criar um atraso de passo de tempo, pois quando se utiliza o bloco do modelo *SimCoupler* em um sistema de *feedback* no Simulink®, ele pode fazer parte de um *loop* algébrico e algumas versões do Matlab/Simulink® não conseguem resolver o sistema contendo laços algébricos, enquanto outras conseguem, mas com desempenho degradado (POWERSIM, 2020).

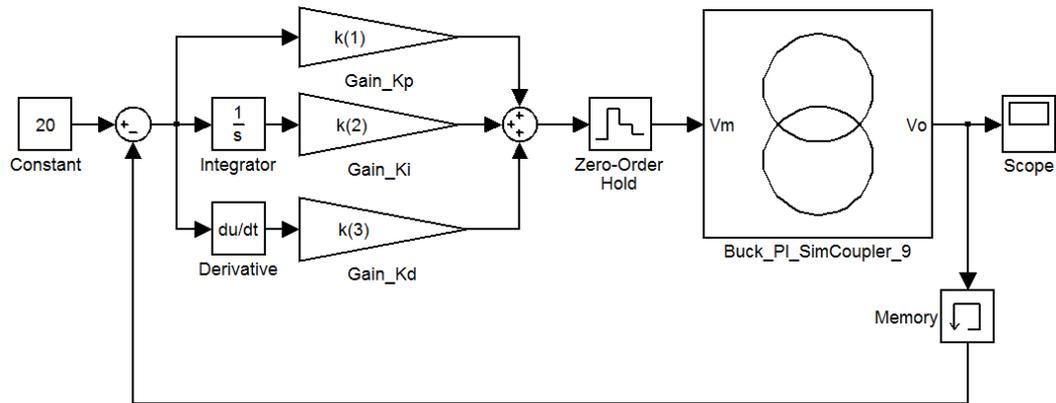
### 3.2 Planta do Conversor CC-CC tipo *Buck*

O circuito de potência e controle do conversor CC-CC *Buck* está representado pela Figura 28 e Figura 29.

Figura 28 - Circuito de potência no PSIM do conversor CC-CC *Buck*



Fonte: Elaborado pelo autor.

Figura 29 - Diagrama de Blocos no Simulink® do conversor CC-CC *Buck*

Fonte: Elaborado pelo autor.

O conversor *Buck* possui as seguintes características de projeto detalhadas na Tabela 1.

Tabela 1 – Parâmetros do Conversor Buck

Parâmetros	Valores
Tensão de entrada ( $V_i$ )	[35 a 50] V
Tensão de saída na carga ( $V_o$ )	20 V
Corrente de saída ( $I_o$ )	5 A
Frequência de chaveamento ( $f$ )	20 kHz
Resistência da Carga ( $R$ )	[3,4 e 5] ohms
Capacitor ( $C$ )	16 $\mu$ F
Indutor ( $L$ )	1,2 mH
Ondulação de corrente no indutor ( $\Delta I_L$ )	10% ( $I_o$ )
Ondulação de tensão no capacitor ( $\Delta V_C$ )	1% ( $V_o$ )

Fonte: Elaborado pelo autor.

O dimensionamento do indutor foi calculado pelas seguintes expressões:

$$\Delta I_L = \frac{V_o(1 - D)}{Lf} \quad (16)$$

$$\Delta I_L = 10\% I_o = \Delta I_L = 0,5 A \quad (17)$$

$$V_o = DV_i \quad (18)$$

$$D = \frac{V_o}{V_i} = \frac{20}{35} = 0,5714 \quad D = \frac{V_o}{V_i} = \frac{20}{50} = 0,4 \quad (19)$$

$$L = \frac{V_o(1-D)}{\Delta I_L f} = \frac{20(1-0,5714)}{0,5 \times 20 \times 10^3} = 857,2 \mu H \quad L = \frac{V_o(1-D)}{\Delta I_L f} = \frac{20(1-0,4)}{0,5 \times 20 \times 10^3} = 1,2 \text{ mH} \quad (20)$$

O indutor escolhido foi o de 1,2 mH por trabalhar com corrente abaixo de ( $\Delta I_L$ ) para qualquer um dos ciclos ativos, impedindo que o conversor operasse em modo descontínuo.

Para o dimensionamento do capacitor foi utilizado a seguinte expressão:

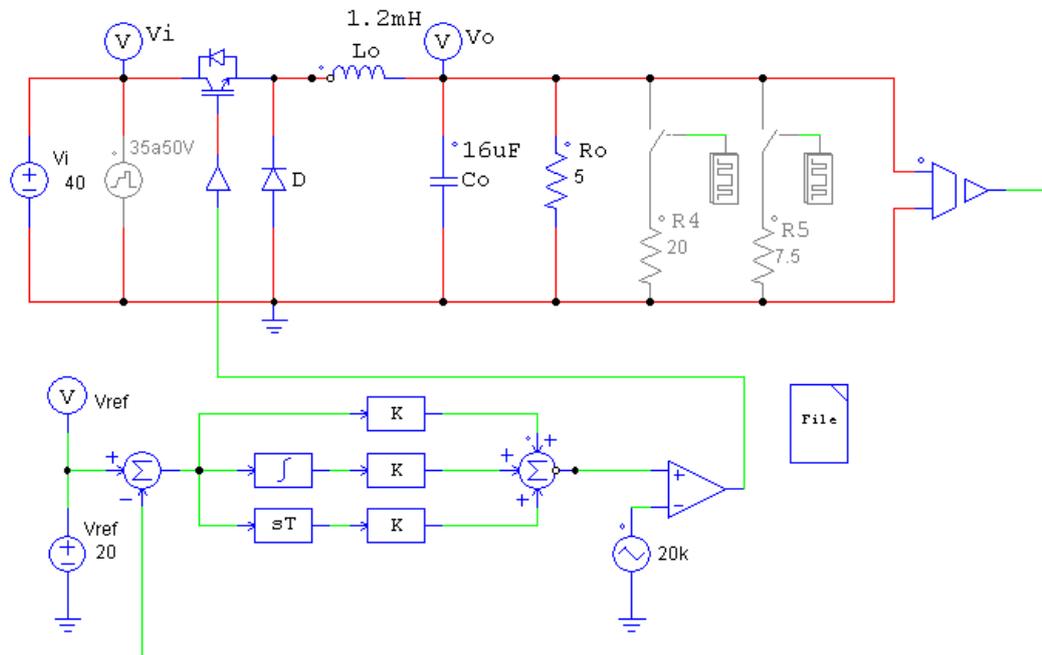
$$C_o \geq \frac{\Delta I_L}{8 \cdot f \cdot \Delta V_C} = \frac{0,5}{8 \cdot (20 \times 10^3) \cdot (1\% \cdot 20)} = 15,63 \mu F \quad (21)$$

O dimensionamento do capacitor foi projetado para que na frequência de chaveamento, apresentasse uma impedância de chaveamento muito menor que a da carga, deixando passar a ondulação em alta frequência e o valor CC ir direto para a carga.

Nas simulações o circuito de potência operou com tensão de entrada fixa em 40 V, com objetivo do circuito de controle em malha fechada, manter em média uma tensão contínua de 20 V na carga. Para manter esse valor, um sinal de controle vindo do PID foi comparado com uma onda triangular (portadora) operando numa frequência de 20 kHz, dando origem ao sinal PWM que fez o chaveamento do MOSFET. Um dos objetivos desse trabalho foi encontrar o melhor conjunto de parâmetros do controlador PID, que permitisse a possibilidade de variar a tensão de entrada e resistência da carga, mantendo o valor médio da tensão de saída, sem grandes alterações.

Para aferir se o conjunto dos melhores parâmetros encontrados pelas meta-heurísticas foram capazes de atender aos requisitos de projeto, um circuito de controle ilustrado pela Figura 30, foi modelado no PSIM, onde os parâmetros  $K_p$ ,  $K_i$  e  $K_d$  foram testados. Foi inserido um bloco rampa de tensão na entrada variando de 35 a 50 V.

Figura 30 – Circuito de teste dos parâmetros  $K_p$ ,  $K_i$  e  $K_d$  do conversor *Buck*

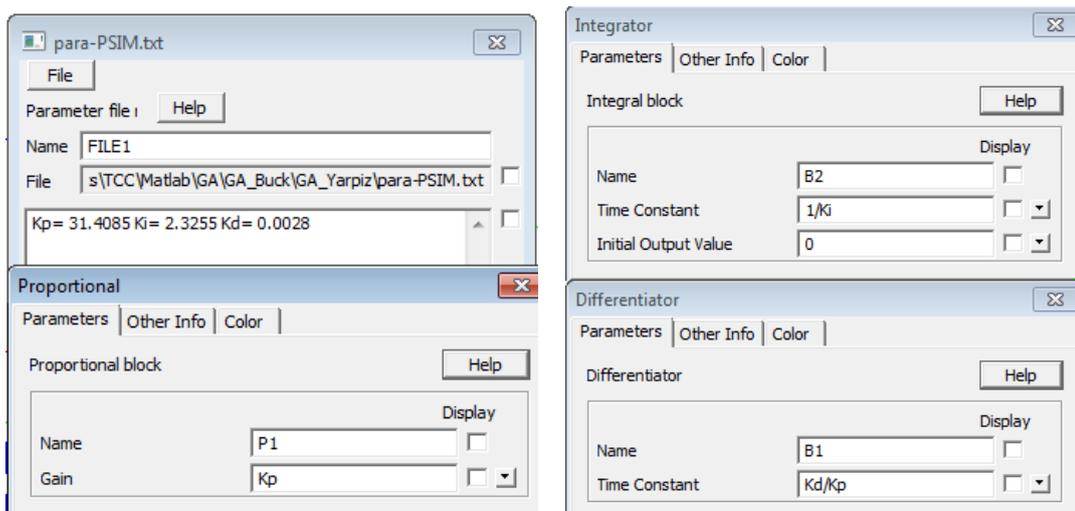


Fonte: Elaborado pelo autor.

O bloco *File*, pegou os valores de  $K_p$ ,  $K_i$  e  $K_d$  armazenados em um arquivo que foi gerado por um *script* no MATLAB®, e fez a atribuição em cada bloco do controlador.

A Figura 31 apresenta um exemplo de como os valores de ( $K_p$ ,  $K_i$  e  $K_d$ ) são inseridos no arquivo, e como eles foram atribuídos em cada bloco do PSIM.

Figura 31 - Parâmetros do Controlador no PSIM



Fonte: Elaborado pelo autor.

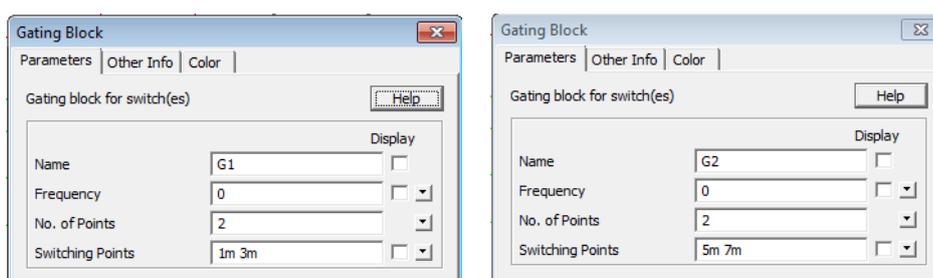
Foram adicionadas duas resistências em paralelo com a carga de 5 ohms, sendo uma de 20 ohms para gerar 4 ohms de resistência equivalente no circuito, e outra de 7,5 ohms, para uma resistência equivalente de 3 ohms.

Elas foram calculadas pela seguinte equação (22):

$$R_{eq} = \frac{R_1 R_2}{R_1 + R_2} \quad (22)$$

Para acionamento das chaves, foi adicionado um bloco para cada resistência chamado *gating block*, que permite o ajuste do tempo de acionamento da chave durante o regime permanente. A Figura 32 apresenta a configuração de cada intervalo de acionamento.

Figura 32 - Configuração do bloco *Gating Block* no PSIM



Fonte: Elaborado pelo autor.

### 3.3 Planta do conversor CC-CC tipo *Boost* como carga eletrônica

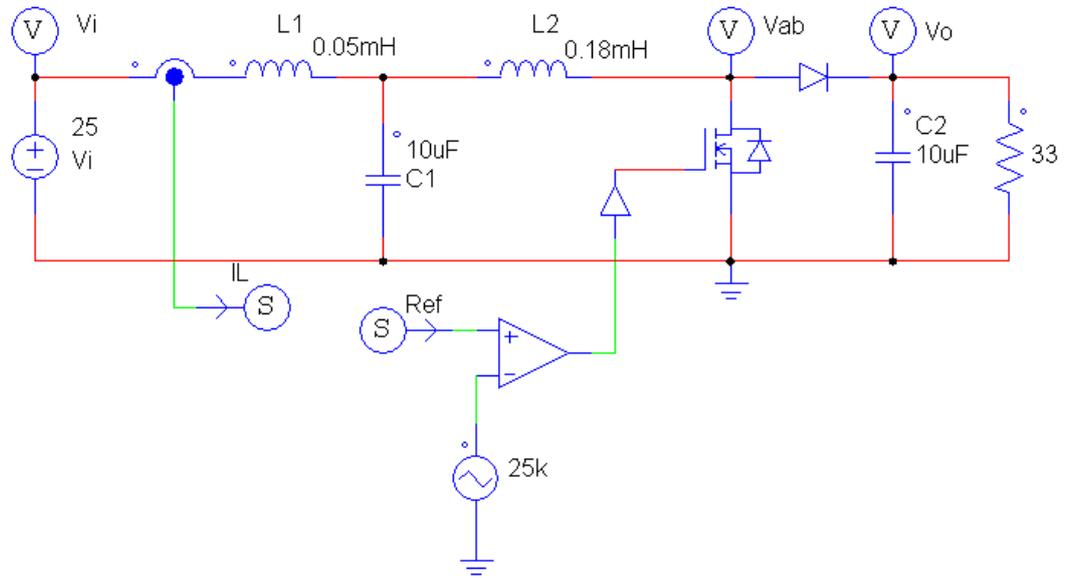
O circuito de potência da carga eletrônica foi modelado utilizando um conversor CC-CC do tipo *Boost* ou *Step-up*, que está representado na Figura 33. Ele foi escolhido por recomendação do orientador, e seu dimensionamento não será abordado neste trabalho. Seu princípio de funcionamento foi abordado no item 2.1.5 Carga Eletrônica CC.

As características dessa fonte chaveada são:

- Tensão de entrada variando de 5 a 25 V
- Corrente drenada pela carga eletrônica de 10 a 25 A.

Foi adotado para todas as simulações a tensão de entrada fixa em 25 V e o sinal da corrente de referência em 20 A.

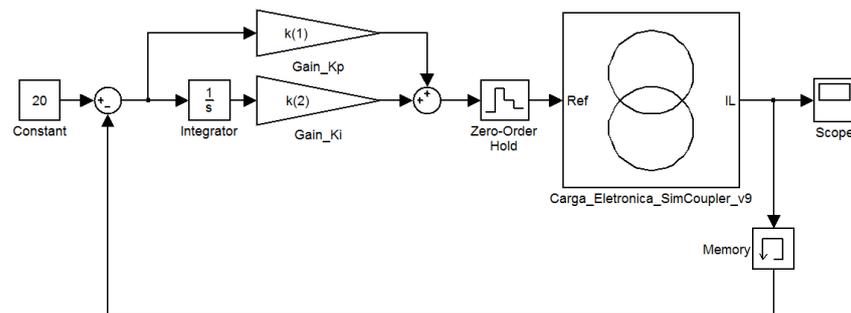
Figura 33 - Circuito de Potência do conversor *Boost* como Carga Eletrônica



Fonte: Elaborado pelo autor.

A Figura 34 apresenta o circuito de controle da carga eletrônica, que teve como escolha um controlador na configuração PI, combinado com as ações de controle proporcional (P) e integral (I), e definido com base na sua função de transferência, já representada pela equação (7). O Sinal de referência ficou fixado em 20 A.

Figura 34 - Diagrama de blocos do controlador PI para carga eletrônica

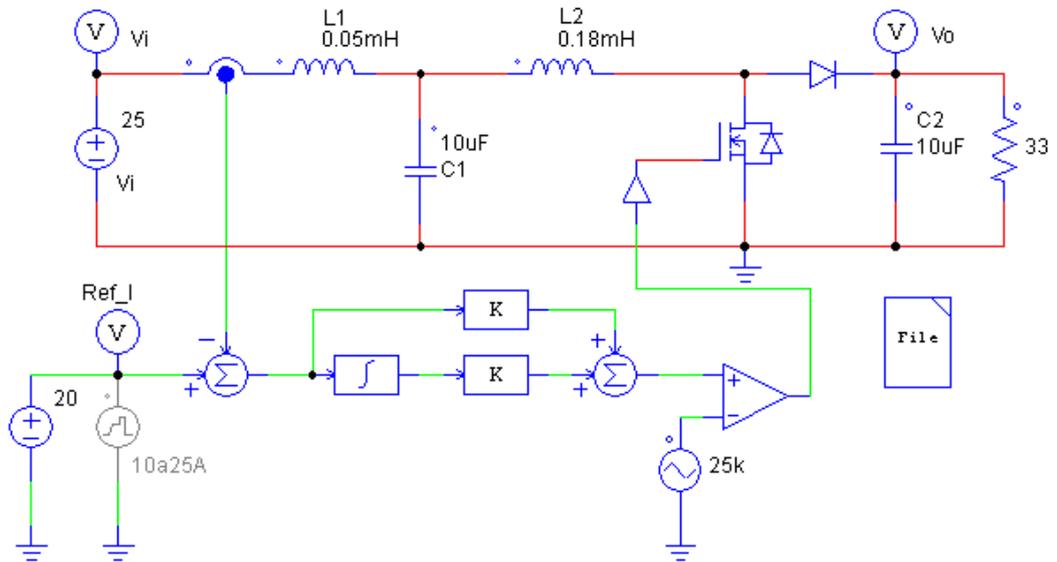


Fonte: Elaborado pelo autor.

Assim como no conversor *Buck*, também foi gerado um circuito de teste no PSIM para a carga eletrônica, onde foi possível variar a corrente da carga para uma determinada tensão de entrada, verificando se a fonte em teste conseguiu seguir o sinal de referência da rampa.

A Figura 35 ilustra o circuito de teste para carga eletrônica, onde de forma análoga aos parâmetros do controlador do circuito do conversor *Buck*, foi utilizado o bloco *File*, com a diferença de não ter a parcela derivativa.

Figura 35 - Circuito de teste para Carga Eletrônica



Fonte: Elaborado pelo autor.

### 3.4 Scripts para as Meta-heurísticas

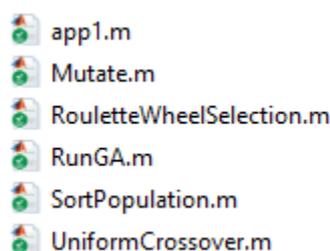
As funções utilizadas nas operações matemáticas de cada algoritmo, GA e PSO, foram fornecidas gratuitamente pelo *Yarpiz*, que tem como objetivo ser um recurso de códigos científicos e tutoriais acadêmicos e profissionais, visando especialmente as áreas de Inteligência Artificial, *Machine Learning*, Otimização de Engenharia, Pesquisa Operacional e Engenharia de Controle. Os códigos fonte fornecidos no *Yarpiz* são todos gratuitos para pesquisa e fins acadêmicos, e gratuitos para compartilhar e modificar. Os códigos estão disponíveis no site [www.yarpiz.com](http://www.yarpiz.com). (HERIS, 2015)

As funções do GA e PSO foram interligadas com outros *scripts* criados pelo autor no MATLAB®, assim como as funções objetivos de cada planta. Dentro do MATLAB® existe uma ferramenta (*toolbox*) pronta para otimização chamada *Optimization Tool - optimtool*. Nela é possível utilizar para resolver problemas de otimização a função *solver* do GA, que já vem com todas as opções disponíveis para seleção e ajuste do algoritmo. Versões mais recentes do MATLAB® já possuem uma função *solver* para o PSO além de outras meta-heurísticas. Foi dada preferência ao código fonte fornecido pelo *Yarpiz*, por ser um código aberto e estar disponível tanto para GA como para o PSO. Além disso, em caso de necessidade de ajuste na dinâmica do algoritmo, este permitiu fácil acesso.

### 3.4.1 Script do Algoritmo Genético GA

O algoritmo GA utilizado neste trabalho apresenta seis funções que serão detalhadas na sequência, e que podem ser observadas na Figura 36.

Figura 36 - Funções do algoritmo GA



Fonte: Elaborado pelo autor.

A função *app1* é o programa principal, que foi responsável por selecionar a função objetivo a ser executada, carregar os parâmetros do problema, fazer a passagem de parâmetros e a chamada do algoritmo, além de solicitar ao Simulink® a execução da simulação do sistema de controle. Nela também foi feito o controle da quantidade de simulações, a análise dos resultados e a medição total do tempo de execução do programa.

A função *RunGA* é o programa do algoritmo em si, que teve sua implementação feita através da escolha dos parâmetros genéticos. Ela se comunica com as demais funções trocando informações para a resolução do problema. A sequência de execução foi feita na seguinte ordem: Inicialização da população principal, seleção dos pais, cruzamento e reprodução, mutação dos descendentes, mesclagem da população principal com os filhos mutados, avaliação dessa nova população com uma classificação e seleção dos melhores candidatos para resolução do problema, e por último ela fez a verificação do critério de parada.

A função *UniformCrossover* é um operador genético que foi responsável por fazer o cruzamento de dois pais de uma população, e reproduzir dois filhos ou mais descendentes, de acordo com a configuração, onde estes herdaram as propriedades e características dos pais. A seleção dos genes dos pais a serem herdados pelos filhos foi feita de forma aleatória combinando uma operação matemática dada pela equação (23).

$$y_1 = ax_1 + (1 - a)x_2 \quad (23)$$

$$y_2 = \alpha x_2 + (1 - \alpha)x_1$$

Onde:

- $y_1$  e  $y_2$  são os filhos que herdaram os genes dos pais.
- $x_1$  e  $x_2$  são os pais que foram cruzados para geração dos filhos.
- $\alpha$  é um vetor de números aleatórios variando entre [0,1].

A função *Mutate* é um operador genético de mutação, que foi responsável por criar um desvio de genética numa parcela dos descendentes, aumentando com isso a capacidade exploratória do algoritmo em encontrar novas soluções e se desviar de mínimos locais.

A função *SortPopulation* foi responsável por classificar a população numa ordem decrescente dos melhores indivíduos para os piores, e no final de cada iteração fez a exclusão daqueles que apresentaram os piores desempenhos. Isso permitiu obter soluções melhores a cada geração.

A função *RouletteWheelSelection* fez a seleção da população de forma probabilística, onde simulou uma roleta dividida em fatias com área proporcional a aptidão da população de pais candidatos ao cruzamento. Ela atribuiu pesos aos melhores indivíduos, fazendo com que eles obtivessem maiores chances de seleção.

### 3.4.2 Script do Algoritmo Enxame de Partículas PSO

O desenvolvimento do algoritmo PSO utilizado neste trabalho foi feito utilizando apenas duas funções que são a PSO e a PSO2, detalhadas na sequência.

Dentre os dois modelos de programa PSO disponibilizados pelo *Yarpiz*, foi optado pelo PSO2, por já vir carregado com os coeficientes de constrição, utilizados para controlar as características dinâmicas do algoritmo. Ele foi o programa principal, igual como no GA, que teve como papel ser o responsável por selecionar a função objetivo de cada planta, carregar os parâmetros do problema, fazer a passagem de parâmetros e a chamada do algoritmo, além de solicitar ao Simulink® a execução da simulação do sistema de controle. Nele também foi feito o controle da quantidade de simulações, a análise dos resultados e a medição total do tempo de execução do programa.

A função PSO é o programa do algoritmo em si, que teve sua implementação feita através do mapeamento da melhor posição de um conjunto de partículas inseridas dentro de um espaço de busca de forma aleatória, visando encontrar a

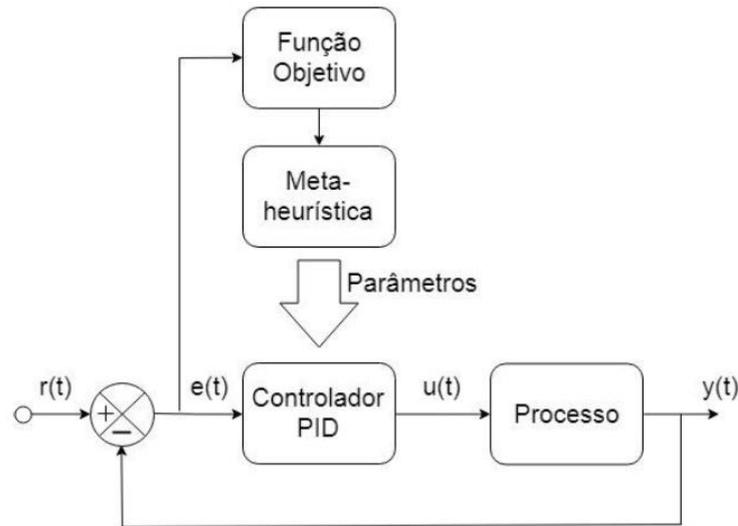
solução do problema proposto por cada planta. A sequência de execução feita por essa função, foi na seguinte ordem: criou-se uma estrutura contendo a população inicial que foi distribuída em posições aleatórias dentro do espaço de busca. Cada partícula foi inicializada e foram armazenadas a posição, velocidade, e o custo da função de cada uma. Cada partícula por padrão do PSO, têm a capacidade de gravar na memória seu próprio *record* como a melhor posição e o melhor custo, sendo sua melhor experiência desde o *start* do algoritmo até o momento da iteração. Isso também foi armazenado dentro da estrutura do PSO, que na sequência fez uma avaliação da posição com base no *global best* (melhores valores de todo o enxame), em seguida a seleção dos melhores candidatos para resolução do problema, além da verificação do critério de parada.

### 3.5 Parâmetros das Meta-Heurísticas

Para rodar os algoritmos, foi preciso carregar parâmetros iniciais que descrevessem a dinâmica do problema, como o número de variáveis e a delimitação do espaço de busca. Em seguida foi necessário informar os parâmetros e variáveis do algoritmo que estava sendo implementado, como número de iterações, o tamanho da população, entre outros. A escolha destes parâmetros, acarretou o desempenho do resultado da otimização. Segundo a literatura, uma parte difícil no projeto de meta-heurísticas é a definição de seus parâmetros. Estes podem ser escolhidos de forma empírica, executando várias vezes a meta-heurística com diferentes valores de parâmetros e tentar selecionar os melhores valores, ou de forma aproximada, quando se tem uma ideia da resposta do problema simulado com outras técnicas (TALBI, 2009).

A Figura 37 mostra através de um diagrama de blocos, a ação das meta-heurísticas em encontrar valores mínimos para as funções objetivo, permitindo assim, realizar a sintonia dos parâmetros de um controlador PID, num sistema de controle.

Figura 37 - Determinação dos parâmetros do PID através das meta-heurísticas



Fonte: Adaptado de Souza, Valiati e Gómez (2012, p. 2602).

No início do processo, o programa principal seleciona a função objetivo com base no índice de desempenho a ser executado, carrega os parâmetros do problema e do algoritmo, e executa a simulação. A cada iteração, a meta-heurística e a função objetivo trocam informações com o sistema de controle, através do Simulink®. O controlador recebe os coeficientes  $K_p$ ,  $K_i$  e  $K_d$ , e realiza a comunicação com o PSIM transmitindo o sinal de controle pela variável  $u(t)$ , que será comparada com o sinal da portadora em alta frequência, gerando o sinal PWM para o chaveamento do MOSFET. Com isso é realizada a simulação da parte de potência, onde o PSIM devolve ao Simulink® o valor médio do sinal do processo (planta) pela variável  $y(t)$ . Este valor é comparado com o valor de *set-point*  $r(t)$  (sinal de referência) e sua diferença gera um erro  $e(t)$  em regime permanente. Esse erro é avaliado pela função objetivo junto dos índices de desempenho, que solicitarão ao algoritmo a produção de novos parâmetros para o controlador PID. Com isso, o funcionamento do controlador será analisar o sinal de erro e corrigir o sinal que é transmitido para a entrada do processo (planta). O ciclo continua, até que as iterações terminem ou um critério de parada seja alcançado.

Para a escolha dos parâmetros de inicialização deste trabalho, foram utilizados valores aproximados para o espaço de busca, com base em simulações de outros trabalhos como o dos autores Varol e Bingul (2004) e de Souza (2021). Cada valor dentro do espaço de busca é um candidato a solucionar o problema, que será analisado pela função objetivo. Os demais parâmetros, inicialmente foram testados com valores *default* de cada algoritmo.

### 3.5.1 Teste das Meta-Heurísticas em uma planta com valores conhecidos

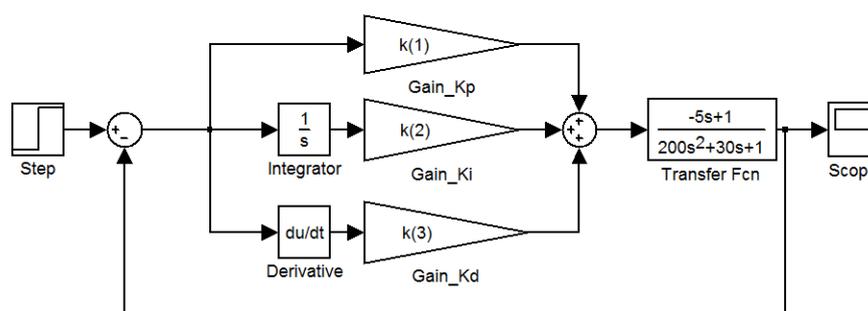
Para certificação de que os algoritmos estavam funcionando adequadamente, foi proposto um teste de desempenho, comparando-os com um trabalho dos autores Varol e Bingul (2004). Neste trabalho, os autores propuseram obter a sintonia de um controlador PID utilizando o algoritmo de otimização por colônia de formigas (ACO) combinado com três funções de custo, como os índices de desempenho (IAE) e (ISE), além de uma nova função chamada de erro baseado em referência com controle mínimo de esforço (RBEMCE). Eles fizeram comparações com técnicas de ajuste conhecidas, como *Ziegler-Nichols* (ZN), *Internal Model Control* (IMC) e *Iterative Feedback Tuning* (IFT). O processo de ajuste foi bem-sucedido pelo ACO e a função de custo RBEMCE, principalmente na presença de alto ruído, que na nossa comparação não foi submetida.

A planta do processo que foi estudada pelos autores está representada pela equação (24)

$$G(s) = \frac{(1 - 5s)}{(1 + 10s)(1 + 20s)} \quad (24)$$

Através do Simulink® foi criado o sistema de controle PID para a planta em estudo, que pode ser observada na Figura 38.

Figura 38 - Sistema de Controle PID Planta Varol e Bingul

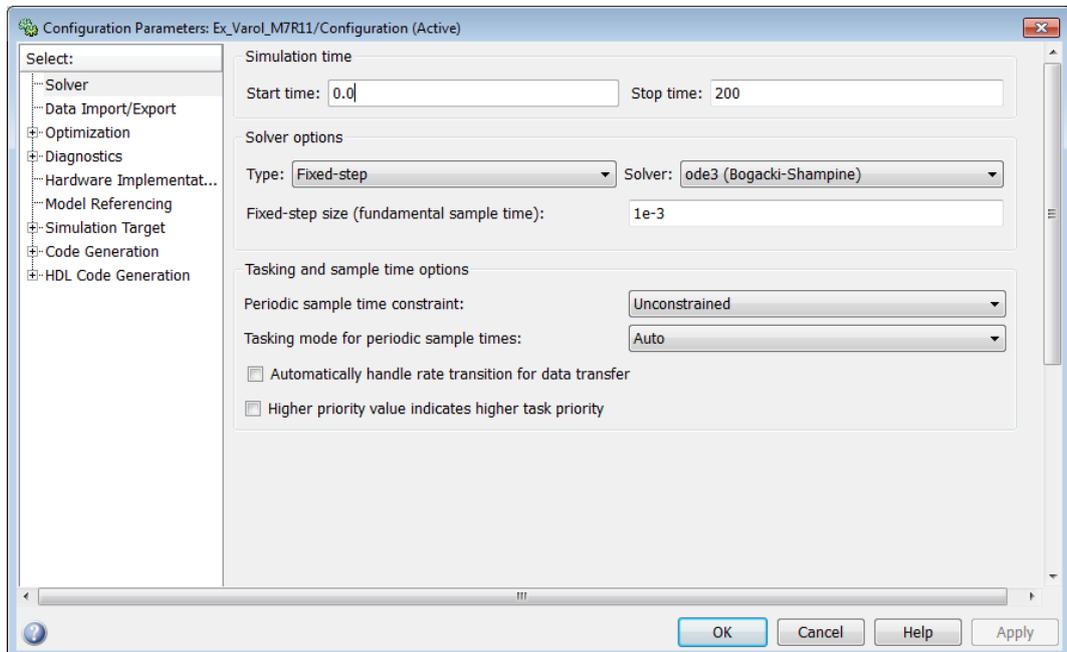


Fonte: Elaborado pelo autor.

A configuração dos parâmetros de simulação no Simulink®, bem como a escolha dos índices de desempenho IAE e ISE, tiveram como base o trabalho dos autores Varol e Bingul (2004). Tudo foi feito de forma similar para comparação do

desempenho das meta-heurísticas propostas nesse trabalho. A Figura 39 apresenta os parâmetros do sistema de controle PID da planta, onde observa-se que foi colocado um tempo de simulação de 200s, igual ao tempo utilizado na simulação dos autores.

Figura 39 - Parâmetros do sistema de controle PID planta Varol e Bingul

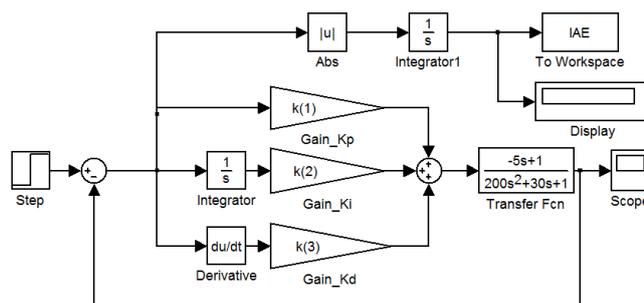


Fonte: Elaborado pelo autor.

Outros dois sistemas de controle foram adicionados com a inclusão dos índices de desempenho do erro de cada teste, para que com isso, fosse possível a comunicação com a função objetivo. A Figura 40 e Figura 41, mostram cada um deles. Estes índices foram montados com base nas equações (8) e (10) do item 2.2.3 Índices de Desempenho.

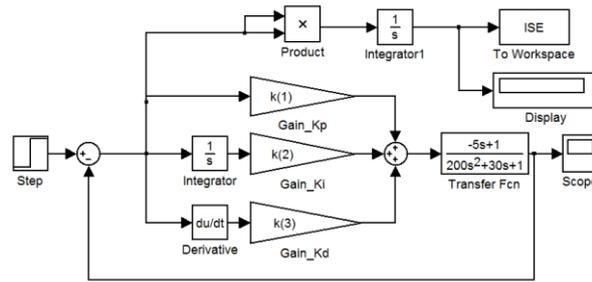
O bloco *Display* inserido na saída do integrador, serviu para visualizar o menor valor de custo passado para a função objetivo.

Figura 40 - Sistema de controle com o índice IAE



Fonte: Elaborado pelo autor.

Figura 41 - Sistema de controle com o índice ISE



Fonte: Elaborado pelo autor.

Para otimizar o desempenho do sistema de controle PID, os ganhos do controlador foram ajustados pelas meta-heurísticas, na tentativa de minimizar ao máximo os índices de desempenho.

A Figura 42, mostra como foi implementada a função *fitness* ou de custo do índice de desempenho da integral do erro absoluto (IAE), que foi utilizada tanto no GA, como no PSO.

Figura 42 - Função de Custo IAE

```

1  function cost = optimize_PID(k)
2  -     assignin('base','k',k);
3  -     sim('Ex_Var01_IAE_M7R11.mdl');
4  -     cost=IAE(length(IAE));
5  -     end

```

Fonte: Elaborado pelo autor.

A função custo IAE é análoga à ISE, mudando apenas a chamada do arquivo de simulação e o vetor da integral de erro. Dentro dela foi inserida a função *assignin* que significa atribuição, ela teve como função criar na 'base' (área de trabalho *Workspace* do MATLAB®) a variável 'k' um vetor de três posições  $k(1)$ ,  $k(2)$  e  $k(3)$  atribuindo a esse vetor valores de  $k$  utilizados dentro do PID, sendo respectivamente os valores de  $K_p$ ,  $K_i$  e  $K_d$ . Em seguida ela chama a simulação do modelo e carrega o valor total, gerado pela integral do erro absoluto do vetor IAE, como resultado da função objetivo. Ela é o elo de comunicação entre a meta-heurística e o modelo de simulação do controle PID.

A primeira meta-heurística testada nessa planta foi a do GA, e seus parâmetros podem ser visualizados no Quadro 2.

A fim de comparar os resultados, foi escolhido um espaço de busca que abrangesse soluções ao redor dos resultados dos autores, e mantidos os demais

critérios utilizados em seu trabalho, como o tamanho da população, número máximo de iterações e número máximo de simulações. A taxa de cruzamento e mutação foram mantidas como padrão do algoritmo.

Quadro 2 – Parâmetros do GA testados no trabalho de Varol e Bingul

<b>Parâmetro de Otimização</b>	<b>Valor</b>
Nº de variáveis do problema	3
Espaço de Busca Mínimo	[0 0 0]
Espaço de Busca Máximo	[5 1 25]
Tamanho da População	5
Nº máximo de iterações	1000
Nº de simulações	5
Taxa de crossover	0,9
Taxa de mutação	0,02

Fonte: Elaborado pelo autor.

No PSO, o Quadro 3 resume os parâmetros que foram utilizados nos testes. Como padrão para os testes iniciais, utilizou-se os coeficientes de restrição sem alteração de valores, conforme foram apresentados e definidos nas equações (14) e (15) do item 2.3.2 Algoritmo Enxame de Partículas PSO.

Quadro 3 - Parâmetros do PSO testados no trabalho de Varol e Bingul

<b>Parâmetro de Otimização</b>	<b>Valor</b>
Nº de variáveis do problema	3
Espaço de Busca Mínimo	[0 0 0]
Espaço de Busca Máximo	[5 1 25]
Tamanho da População	5
Nº máximo de iterações	1000
Nº de simulações	5
Fator $k$	1
Coefficiente de restrição - $\varphi_1$	2,05
Coefficiente de restrição - $\varphi_2$	2,05
Componente inercial - $x$	0,7298
Coefficientes de aceleração - $c_1$ e $c_2$	1,4962
Fator de amortecimento inercial - $w$	1

Fonte: Elaborado pelo autor.

### 3.5.2 Configurações do GA e PSO na planta do Conversor CC-CC *Buck*

Os parâmetros utilizados em comum tanto no GA como no PSO, para otimização da planta do Conversor CC-CC *Buck* estão resumidos no Quadro 4. No GA a taxa de cruzamento e a taxa de mutação ficaram as mesmas simuladas na planta de Varol e Bingul Quadro 2, assim como os coeficientes de constrição do PSO no Quadro 3.

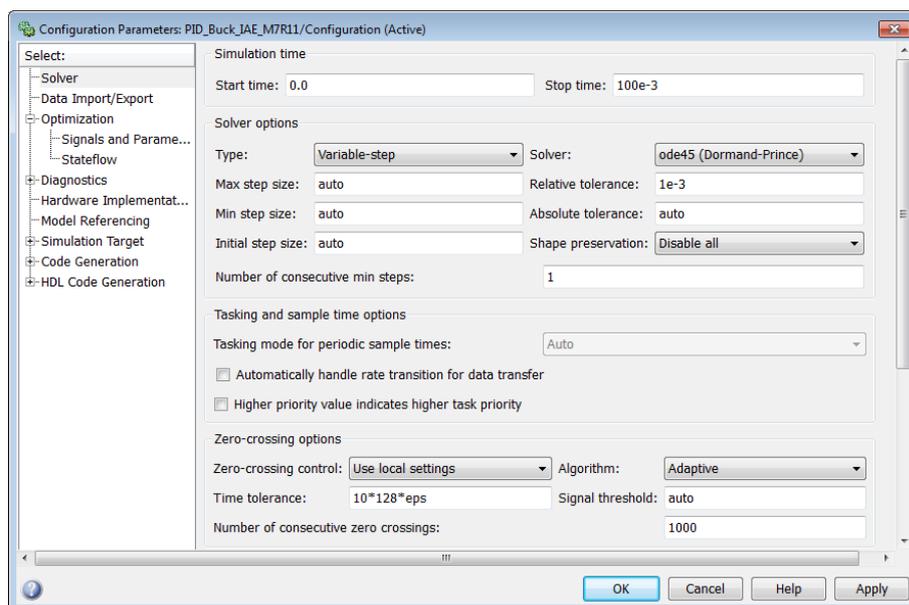
Quadro 4 - Parâmetros do GA e PSO simulados na planta do Conversor *Buck*

Parâmetro de Otimização	Valor
Nº de variáveis do problema	3
Espaço de Busca Mínimo	[0 0 0]
Espaço de Busca Máximo	[40 20 2]
Tamanho da População	5
Nº máximo de iterações	20
Nº de simulações	5

Fonte: Elaborado pelo autor.

Em todos os modelos, foram utilizadas as mesmas configurações de parâmetros do Simulink®, como tempo de simulação em 100 ms, *solver* do tipo *variable-step*, entre outros, conforme é mostrado na Figura 43.

Figura 43 – Parâmetros de simulação para o controle do conversor *Buck*



Fonte: Elaborado pelo autor.

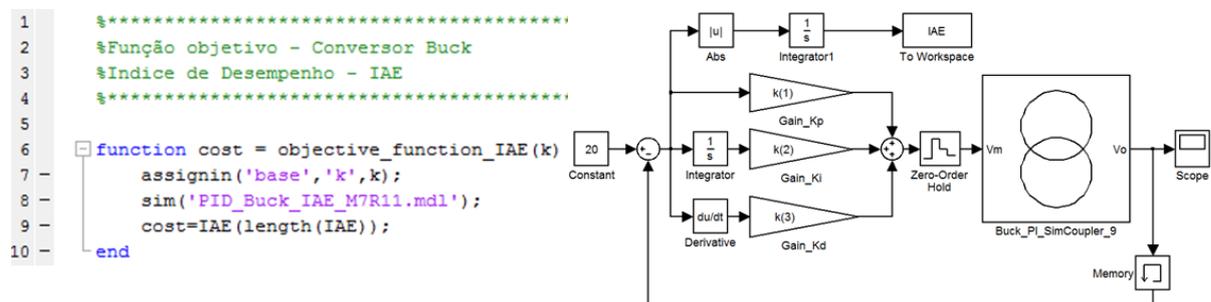
No programa principal de cada algoritmo, foi adicionado uma opção para selecionar o índice de desempenho aplicado como função objetivo, visando sintonizar o controlador do conversor. Cada função objetivo foi escrita em *script* separado, assim como, um modelo em diagrama de blocos foi desenvolvido no Simulink®. O sinal de referência para o controle, ficou fixado em 20 V, em todos os modelos.

A seguir, são apresentadas as funções objetivo e seus respectivos diagramas de blocos, que foram utilizados para otimizar os parâmetros PID do conversor *Buck*.

Para que seja possível representar um sistema com confiabilidade nas características do processo a ser otimizado, é fundamental que se escolha uma função *fitness* adequada. Optou-se então, por utilizar conforme descrito no item 2.2.3 Índices de Desempenho, os critérios baseados na integração do erro pelo tempo, que são o IAE, ITAE, ISE e ITSE, detalhados na sequência.

A Figura 44 apresenta o *script* da função objetivo, bem como o diagrama de blocos do índice IAE, que teve como base a equação (8) para o seu desenvolvimento.

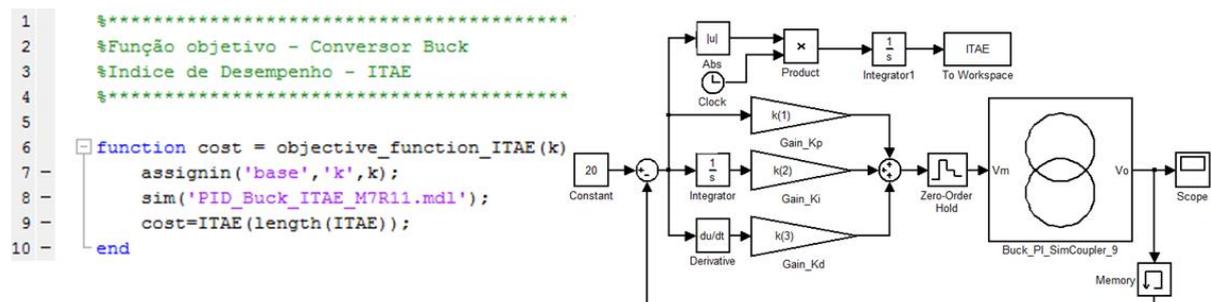
Figura 44 – Função objetivo e Índice de desempenho IAE para o conversor *Buck*



Fonte: Elaborado pelo autor.

O próximo índice é o ITAE, onde a Figura 45 apresenta o *script* da função objetivo, bem como o diagrama de blocos, que teve como base a equação (9).

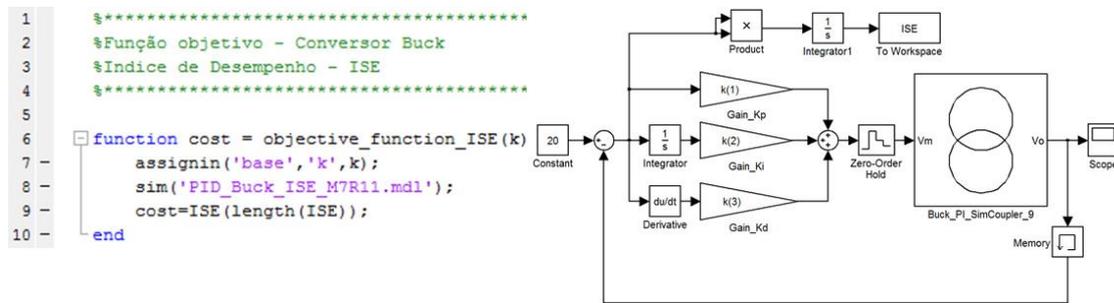
Figura 45 - Função objetivo e Índice de desempenho ITAE para o conversor *Buck*



Fonte: Elaborado pelo autor.

Em seguida é apresentado o índice ISE, onde a Figura 46 apresenta o *script* da função objetivo, bem como o diagrama de blocos, que teve como base a equação (10).

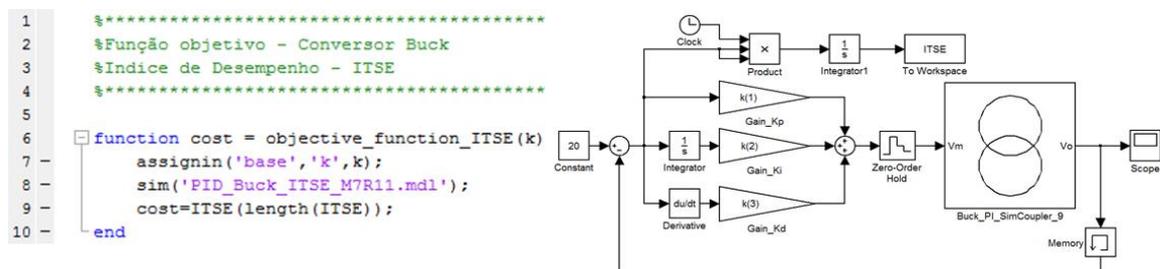
Figura 46 - Função objetivo e Índice de desempenho ISE para o conversor *Buck*



Fonte: Elaborado pelo autor.

Por último temos o índice ITSE, onde a Figura 47 apresenta o *script* da função objetivo, bem como o diagrama de blocos, que teve como base a equação (11).

Figura 47 - Função objetivo e Índice de desempenho ITSE para o conversor *Buck*



Fonte: Elaborado pelo autor.

### 3.5.3 Configurações do GA e PSO na planta do Conversor CC-CC *Boost*

Assim como na planta do conversor *Buck*, foram criadas as funções objetivo com base nos índices de desempenho, tendo seus *scripts* e modelos em diagrama de blocos criados em separado. Para todos os modelos, foram utilizadas as mesmas configurações de parâmetros do Simulink®. O tempo de simulação mais adequado para os testes foi de 25 ms. Nesta planta foi mais difícil obter as características da resposta transitória pelo Simulink® então, foi realizada a simulação no PSIM com os parâmetros obtidos, e feita a exportação dos dados da resposta transitória ao Simulink®, para posterior análise.

O Quadro 5 resume os parâmetros realizados com GA e PSO na planta *Boost*.

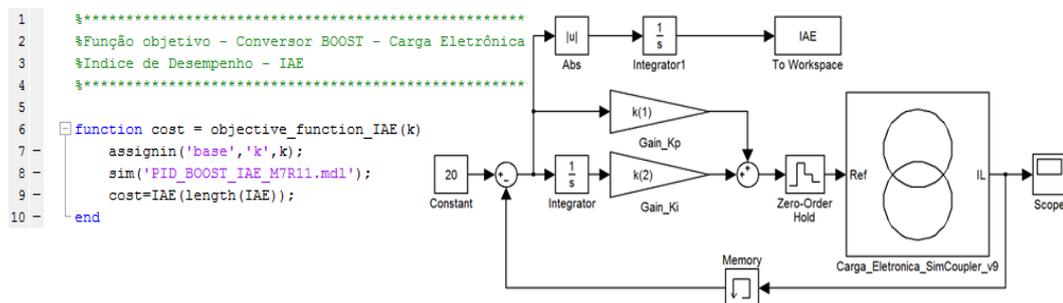
Quadro 5 - Parâmetros do GA e PSO simulados na planta do Conversor *Boost*

Parâmetro de Otimização	Valor GA	Valor PSO
Nº de variáveis do problema	2	2
Espaço de Busca Mínimo	[0 2500]	[0 2500]
Espaço de Busca Máximo	[0,01 4000]	[0,01 5000]
Tamanho da População	50	50
Nº máximo de iterações	50	50
Nº de simulações	10	10
Taxa de crossover	1	-
Taxa de mutação	0,02	-
Fator $k$	-	1
Coeficiente de restrição - $\varphi_1$	-	2,05
Coeficiente de restrição - $\varphi_2$	-	2,05
Componente inercial - $x$	-	0,7298
Coeficientes de aceleração - $c_1$ e $c_2$	-	1,4962
Fator de amortecimento inercial - $w$	-	1,5

Fonte: Elaborado pelo autor.

As funções objetivo e os índices de desempenho utilizados na planta do conversor CC-CC *Boost* como carga eletrônica, são detalhados na sequência.

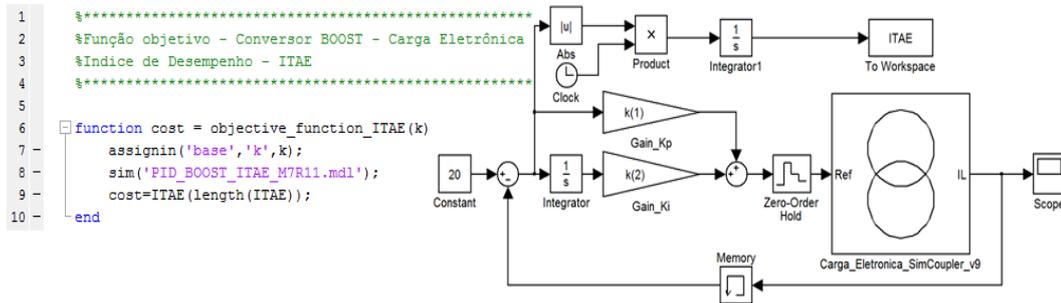
A Figura 48 apresenta o *script* da função objetivo, bem como o diagrama de blocos do índice IAE, que teve como base a equação (8) para o seu desenvolvimento.

Figura 48 – Função objetivo e Índice de desempenho IAE para o conversor *Boost*

Fonte: Elaborado pelo autor.

O próximo índice é o ITAE, onde a Figura 49 apresenta o *script* da função objetivo, bem como o diagrama de blocos, que teve como base a equação (9).

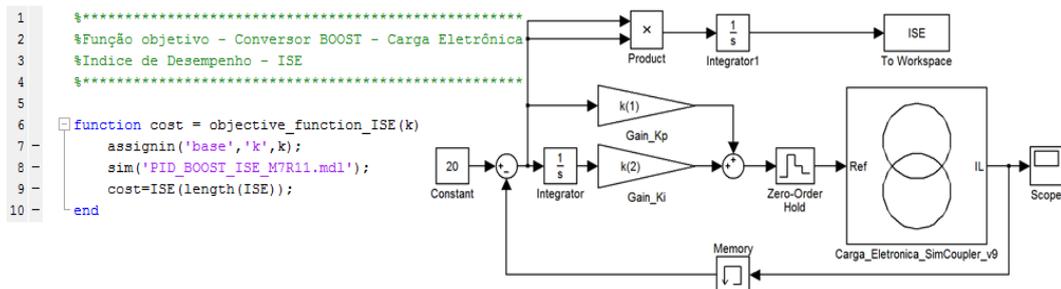
Figura 49 - Função objetivo e Índice de desempenho ITAE para o conversor *Boost*



Fonte: Elaborado pelo autor.

Em seguida é apresentado o índice ISE, onde a Figura 50 apresenta o *script* da função objetivo, bem como o diagrama de blocos, que teve como base a equação (10).

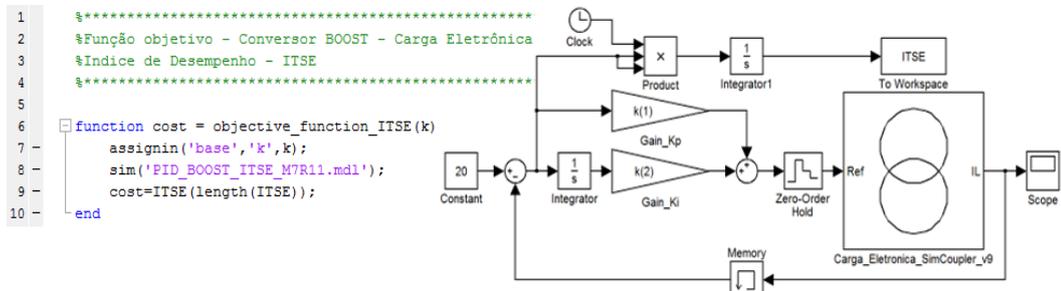
Figura 50 - Função objetivo e Índice de desempenho ISE para o conversor *Boost*



Fonte: Elaborado pelo autor.

Por último temos o índice ITSE, onde a Figura 51 apresenta o *script* da função objetivo, bem como o diagrama de blocos, que teve como base a equação (11).

Figura 51 - Função objetivo e Índice de desempenho ITSE para o conversor *Boost*



Fonte: Elaborado pelo autor.

Com os parâmetros inseridos em cada algoritmo, deu-se início as simulações, onde seus resultados serão abordados no próximo capítulo.

## 4 ANÁLISE DOS RESULTADOS E DISCUSSÃO

Este capítulo trata da análise dos resultados e discussão de cada etapa do trabalho, com o objetivo de mostrar se as meta-heurísticas conseguiram atender a demanda de sintonizar adequadamente os conversores CC-CC do tipo *Buck* e *Boost* como carga eletrônica, bem como saber qual teve melhor performance e qual índice de desempenho conseguiu minimizar melhor a função de custo.

A técnica utilizada na determinação da sintonia dos parâmetros PID para determinação dos coeficientes de  $K_p$ ,  $K_i$ ,  $K_d$  foi experimental por otimização computacional, pois não foi modelada a função de transferência de cada processo.

### 4.1 Resultado comparativo do GA e PSO na Planta de Varol e Bingul

Conforme descrito no item 3.5.1 Teste das Meta-Heurísticas em uma planta com valores conhecidos, foram utilizados os mesmos parâmetros dos autores para iniciar as primeiras rodadas de testes.

A Tabela 2 mostra o resultado dos valores atingidos no trabalho dos autores, comparando com os índices IAE e ISE.

Tabela 2 – Parâmetros comparativos do trabalho de Varol e Bingul.

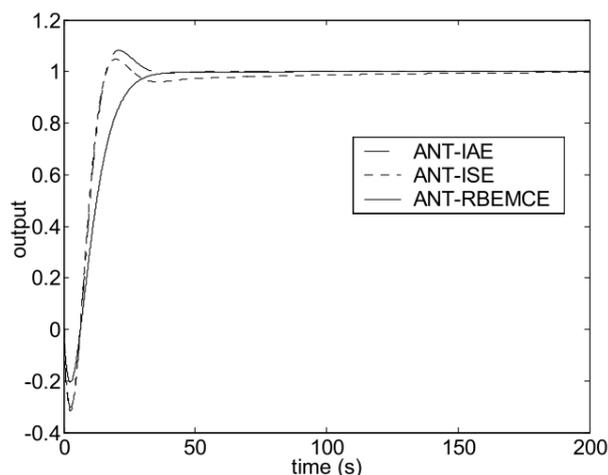
	Ant - IAE	Ant - ISE	Ant - RBMCE
$K_p$	4,606	4,7655	3,3358
$K_i$	0,0913	0,0725	0,0661
$K_d$	21,7854	22,23	21,7854
%UP	8,2	4,7	0
$t_s$	29,95	72,3	31,7
$t_r$	7,17	7,07	14,97

Fonte: Adaptado de Varol e Bingul (2004, p. 2156).

Conforme comentado pelos autores em seu artigo, utilizando o algoritmo de otimização por colônia de formigas e a função de custo RBMCE, eles obtiveram melhores resultados como *overshoot* em 0%, do que utilizando os índices de desempenho IAE e ISE, como pode ser visualizado na Tabela 2.

A Figura 52 mostra o desempenho de sua função objetivo frente aos demais índices testados.

Figura 52 - Curvas de desempenho das funções de custo de Varol e Bingul



Fonte: Varol e Bingul (2004, p. 2156).

A Tabela 3, mostra o desempenho do GA e PSO utilizando os mesmos índices IAE e ISE que Varol e Bingul utilizaram em seu trabalho, e também de forma comparativa foi testado com valores de  $K_p$ ,  $K_i$  e  $K_d$  cedidos por Souza, Valiati e Gómez (2012), que foram encontrados durante seus testes de desempenho para desenvolvimento de seu trabalho. Para cada simulação foram extraídas as respostas transitórias de controle, como *overshoot*, tempo de acomodação e tempo de subida. Além disso foi contabilizado o custo de cada função objetivo.

Tabela 3 – Comparativo GA e PSO com diferentes funções de custo

	GA-IAE	GA-ISE	PSO-IAE	PSO-ISE	Varol e Bingul	Souza <i>et al.</i>
$K_p$	3,5855	4,7471	2,4340	2,4411	3,3358	4,3299
$K_i$	0,0783	0,0840	0,0532	0,0661	0,0661	0,0856
$K_d$	18,2045	24,1321	20,7415	12,3942	21,7854	21,9300
%UP	2,7344	0,5585	0,0000	3,1977	0,0000	2,2678
$t_s$	28,1435	38,2489	56,9690	44,2362	31,7000	17,0380
$t_r$	10,4421	7,6790	29,2715	15,4364	14,9700	8,4005
<i>Custo</i>	12,7786	10,9086	19,6587	13,5877		

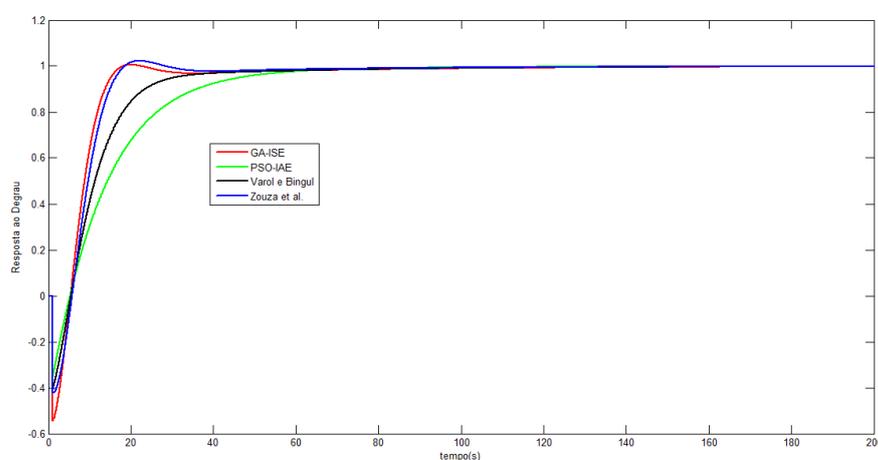
Fonte: Elaborada pelo autor.

O critério de avaliação foi feito da seguinte maneira: avaliação do menor *overshoot* (%UP), seguido do menor tempo de acomodação ( $t_s$ ), e por último o menor tempo de subida ( $t_r$ ). No algoritmo GA o índice de desempenho ISE, obteve melhor resposta que o IAE, e foi o que encontrou o menor custo da função objetivo entre os quatro índices. Já no algoritmo PSO o índice IAE foi o melhor entre os quatro testes

no quesito *overshoot*. Já no comparativo com o resultado dos autores, ele perdeu tanto no tempo de acomodação como no tempo de subida. Em relação aos resultados do teste dos autores Souza, Valiati e Gómez (2012), ele ganhou no *overshoot* mas perdeu nos demais tempos de acomodação e subida.

A Figura 53, mostra de forma comparativa as curvas de resposta ao degrau do sistema em malha fechada, com as melhores funções de custo obtidas pelo GA e PSO para sintonia dos controladores PID, frente os resultados dos autores.

Figura 53 - Respostas ao degrau com diferentes funções de custo



Fonte: Elaborada pelo autor.

#### 4.1.1 Novos arranjos de parâmetros para uma melhor performance

Embora a simulação anterior tenha sido satisfatória do ponto de vista de comparação, foram testados um novo conjunto de parâmetros para cada algoritmo, com o objetivo de melhorar ainda mais a resposta do sistema, visto que a quantidade da população era baixa para um número grande de iterações. O Quadro 6, apresenta apenas as mudanças que foram realizadas no tamanho da população e na quantidade de iterações.

Quadro 6 – Mudanças de parâmetros para nova simulação com GA e PSO

Parâmetro de Otimização	Valor
Tamanho da População	20
Nº máximo de iterações	50

Fonte: Elaborado pelo autor.

A Tabela 4 mostra o desempenho do GA e PSO com os novos parâmetros, comparados com as funções objetivo.

Tabela 4 – Resultado da simulação com população de 20 e 50 iterações

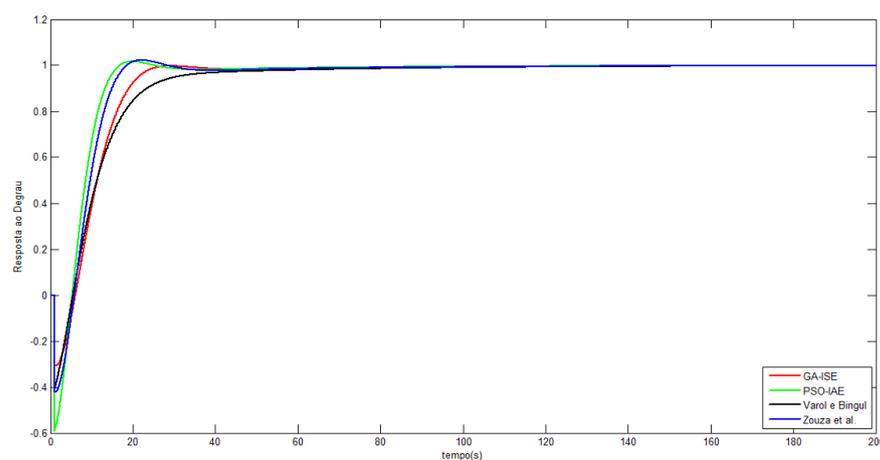
	GA-IAE	GA-ISE	PSO-IAE	PSO-ISE	Varol e Bingul	Souza <i>et al.</i>
$K_p$	4,8218	3,4952	4,8628	3,7885	3,3358	4,3299
$K_i$	0,0957	0,0741	0,0943	0,0768	0,0661	0,0856
$K_d$	23,9358	19,2494	24,7709	18,8592	21,7854	21,9300
%UP	3,4593	0,0000	1,7133	2,4825	0,0000	2,2678
$t_s$	21,3832	23,2039	15,1518	49,3510	31,7000	17,0380
$t_r$	7,0957	11,9051	7,3351	9,7392	14,9700	8,4005
Custo	12,3347	11,0608	12,7507	11,7238		

Fonte: Elaborado pelo autor.

Percebe-se uma melhora nos quatro testes com a redução dos tempos de acomodação, sendo a função de custo GA-ISE a vencedora no quesito do *overshoot* e a que alcançou o menor custo.

A Figura 54, mostra de forma comparativa as curvas de resposta ao degrau do sistema em malha fechada, com as melhores funções de custo obtidas pelo GA e PSO para sintonia do controlador PID, frente aos resultados dos autores.

Figura 54 – Resposta ao degrau com novo teste de GA e PSO x Varol e Bingul



Fonte: Elaborado pelo autor.

Como foram executadas cinco simulações para cada função objetivo, a Tabela 5 resume a média aritmética dos valores encontrados pelas meta-heurísticas para os três parâmetros: *overshoot* (%UP), tempo de assentamento ( $t_s$ ) e tempo de subida ( $t_r$ ),

onde os valores de  $K_p$ ,  $K_i$  e  $K_d$  foram mantidos daqueles que tiveram melhor desempenho.

Tabela 5 – Média de 5 simulações com 20 indivíduos e 50 iterações

	GA-IAE	GA-ISE	PSO-IAE	PSO-ISE	Varol e Bingul	Souza <i>et al.</i>
$K_p$	4,82184	3,49518	4,8628	3,78847	3,3358	4,3299
$K_i$	0,09565	0,07406	0,0943	0,07677	0,0661	0,0856
$K_d$	23,9358	19,2494	24,7709	18,8592	21,7854	21,93
%UP	4,9229	0,71468	0,7222	2,24541	0	2,2677
$t_s$	25,0096	22,5986	14,7255	25,1033	31,7	17,038
$t_r$	7,4579	11,8579	6,1631	9,19239	14,97	8,4005

Fonte: Elaborado pelo autor.

O resultado das médias de 5 simulações melhorou bastante o desempenho dos algoritmos quanto ao tempo de assentamento ( $t_s$ ) e tempo de subida ( $t_r$ ), com um pequeno *overshoot* para as funções GA-ISE e PSO-IAE, visto que só foram realizadas 50 iterações das 1000 que tinham sido feitas do primeiro teste. Como o objetivo era apenas verificar se os algoritmos estavam bem implementados, essa análise foi suficiente para a continuidade das simulações nas plantas dos conversores CC-CC *Buck* e *Boost*.

#### 4.2 Análise de desempenho do conversor CC-CC *Buck*

Com base no item 3.5.2 Configurações do GA e PSO na planta do Conversor CC-CC *Buck*, deu-se inícios às simulações. Como forma de comparar os resultados dos testes de desempenho dos algoritmos, foi coletado com Souza (2021), valores de um controlador do tipo PI, que foram projetados por ele de forma empírica para controlar o conversor *Buck*. No algoritmo GA, este por possuir mais variáveis de controle que o PSO, a técnica utilizada foi de variar parâmetros, identificando a resposta transitória e comparando-a com o resultado de Souza (2021). Foram realizadas cinco rodadas com cada função objetivo, e os parâmetros alterados foram, os limites do espaço de busca, o tamanho da população, o número de iterações, a taxa de cruzamento e a taxa de mutação, onde pode-se perceber que devido o comportamento do algoritmo ser estocástico, foi mais difícil e moroso determinar de forma assertiva a melhor configuração de parâmetros que apresentassem uma

resposta satisfatória para o problema. Tudo foi baseado em ajuste de parâmetros e análise da resposta. Algumas alterações como o tamanho da população acima de 40 indivíduos, exigiu um esforço computacional maior, aumentando bastante o tempo da simulação. Para o PSO, foram testados os limites do espaço de busca, tamanho da população, número de iterações e fator de amortecimento inercial, onde este parâmetro meche no ganho de velocidade da partícula a cada iteração, fazendo com que ela se mova mais rápido à procura de novas posições, fugindo de mínimos locais.

A Tabela 6, resume o resultado destas simulações apresentado aqueles índices que obtiveram os melhores desempenho para cada algoritmo.

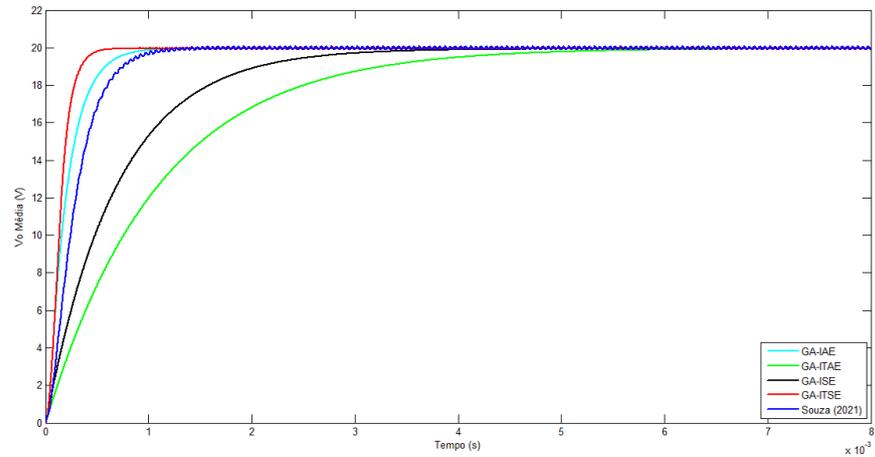
Tabela 6 – Resultados da otimização no conversor CC-CC *Buck*

	Critério	$K_p$	$K_i$	$K_d$	(%UP)	$t_s(s)$	$t_r(s)$	Vo Média (V)	Fitness Custo
Souza	Empírico	0,2500	666,6700	-	0,3781	0,0010	0,0005	19,9394	-
	IAE	25,6680	11,1037	0,0046	0,0000	0,0008	0,0004	19,9214	0,0079
GA	ITAE	29,6626	8,2737	0,0353	0,0000	0,0043	0,0024	19,7593	0,0001
	ISE	14,1837	18,3567	0,0101	0,0000	0,0028	0,0015	19,8171	0,1408
	ITSE	31,4085	2,3255	0,0028	0,0000	0,0004	0,0002	19,9377	7E-06
	IAE	40,0000	7,7148	0,0013	0,0000	2,7E-04	2E-04	19,9481	0,0052
PSO	ITAE	40,0000	18,7036	0,0262	0,0000	0,0025	0,0014	19,8556	0,0001
	ISE	25,9151	11,2925	0,0023	0,0000	0,0004	0,0002	19,9327	0,0423
	ITSE	18,1477	8,9434	0,0005	0,0000	2,6E-04	1,71E-04	19,9534	5E-06
	IAE	40,0000	7,7148	0,0013	0,0000	2,7E-04	2E-04	19,9481	0,0052

Fonte: Elaborado pelo autor.

Nos testes com GA, todos os índices de desempenho não apresentaram *overshoot* em relação ao sinal de referência. Dentre eles o índice que se destacou foi o ITSE, sendo melhor no tempo de assentamento  $t_s(s)$ , no tempo de subida  $t_r(s)$ , tendo a maior média de tensão de saída e obtendo o menor valor do custo da função. Em relação a comparação com o resultado dos testes de Souza (2021), ele também performou em todas as características de resposta transitória, perdendo apenas por 1,7 mV na média de tensão de saída, o que tecnicamente pode se dizer empatados neste quesito. Já nos testes com PSO, o índice ITSE foi melhor em todos os quesitos. A Figura 55, apresenta as respostas obtidas com GA para todos os índices em comparação com a resposta de Souza (2021).

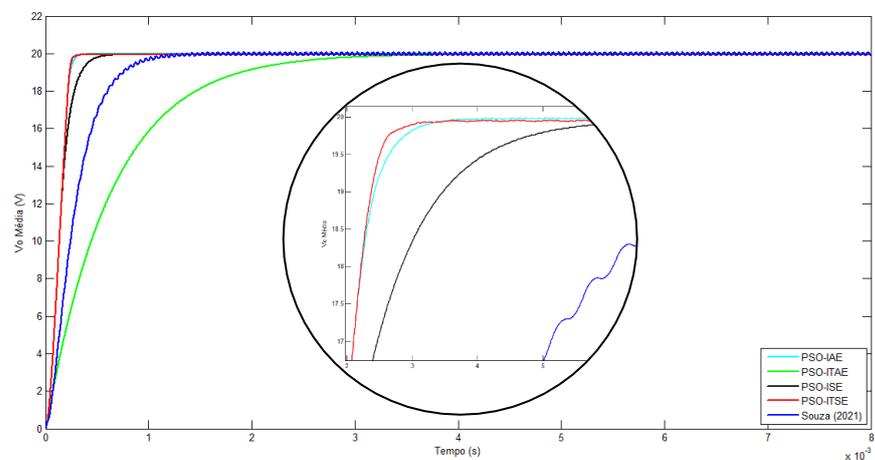
Figura 55 - Resposta do GA no conversor *Buck* comparando com Souza (2021)



Fonte: Elaborado pelo autor.

A Figura 56, apresenta as respostas obtidas com PSO para todos os índices em comparação com Souza (2021).

Figura 56 - Resposta do PSO no conversor *Buck* comparando com Souza (2021)



Fonte: Elaborado pelo autor.

Na Figura 56, foi dado um *zoom* para destacar melhor as curvas dos índices IAE e ITSE, pois conforme a Tabela 6 eles apresentaram características de resposta transitória muito parecidas, quase sobrepondo as linhas, o que podemos dizer que eles quase possuem a mesma estabilidade relativa.

A Tabela 7 apresenta a média das 5 melhores simulações de cada índice, onde foram mantidos os melhores valores de  $K_p$ ,  $K_i$  e  $K_d$ , incluindo o menor custo da função objetivo.

Tabela 7 – Média das 5 simulações na otimização do conversor CC-CC *Buck*

	Critério	$K_p$	$K_i$	$K_d$	(%UP)	$t_s(s)$	$t_r(s)$	$V_o$ Média (V)	Fitness Custo
Souza	Empírico	0,2500	666,6700	-	0,3781	0,0010	0,0005	19,9394	-
GA	IAE	25,6680	11,1037	0,0046	0,0070	0,0083	0,0047	19,5526	0,0079
	ITAE	29,6626	8,2737	0,0353	0,0011	0,0056	0,0031	19,6999	0,0001
	ISE	14,1837	18,3567	0,0101	0,0196	0,0109	0,0062	19,4201	0,1408
	ITSE	31,4085	2,3255	0,0028	1,1797	0,0023	0,0013	19,8497	7E-06
PSO	IAE	40,0000	7,7148	0,0013	3,8247	0,0009	0,0005	19,9195	0,0052
	ITAE	40,0000	18,7036	0,0262	0,0000	0,0054	0,0030	19,7086	0,0001
	ISE	25,9151	11,2925	0,0023	7,6250	0,0005	0,0002	19,9303	0,0423
	ITSE	18,1477	8,9434	0,0005	0,0000	0,0014	0,7622	19,8922	5E-06

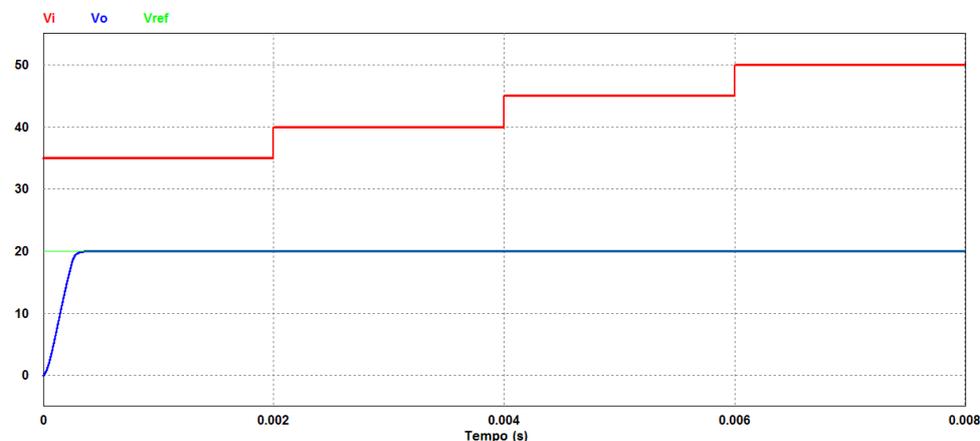
Fonte: Elaborado pelo autor.

A fim de testar o funcionamento dos parâmetros obtidos pelas meta-heurísticas, foi desenvolvido o circuito do conversor CC-CC *Buck* no PSIM, conforme pode ser visto na página 49 Figura 30, onde foi feita a análise de sua robustez.

#### 4.2.1 Teste de variação de tensão de entrada numa faixa de 35 a 50 V no *Buck*

Este teste foi feito uma rampa de tensão de entrada variando seus valores numa faixa de 35 a 50 V, onde o *set point* de referência foi configurado em 20 V e a carga em 4 ohms. O objetivo foi verificar se com a mudança de tensão de entrada o conversor conseguiria manter a tensão média na saída de acordo com a referência estipulada. A Figura 57 apresenta o teste realizado com o índice ITSE do PSO.

Figura 57 - Teste em rampa de 35 a 50 V com parâmetros do PSO-ITSE



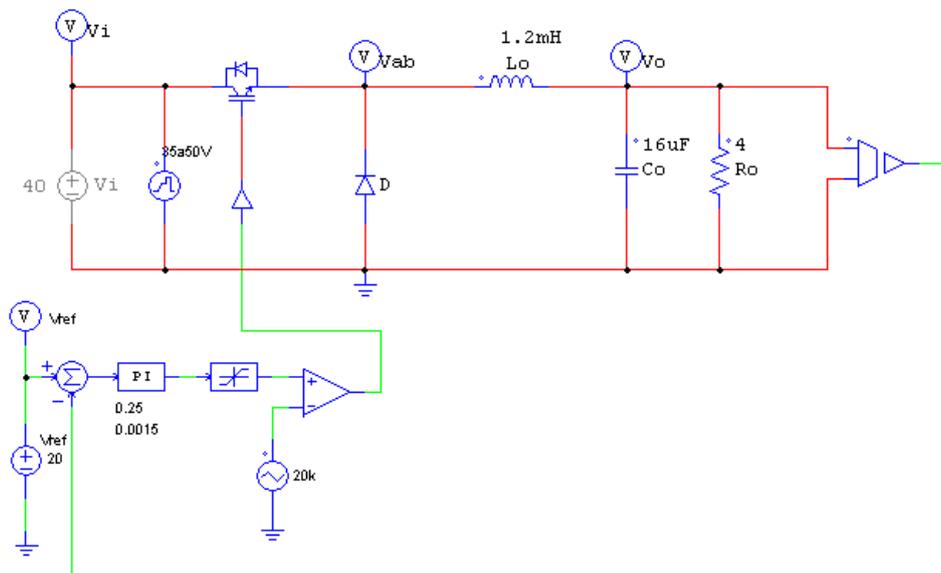
Fonte: Elaborado pelo autor.

No teste de rampa de tensão de entrada, todas as funções objetivo obtiveram êxito, sendo o índice ITSE do PSO aquele que atingiu mais rápido o sinal de referência com a tensão média de saída na faixa de 20 V.

Como comparativo, o mesmo teste foi submetido com os parâmetros de Souza (2021) na sintonia do controlador PID no conversor *Buck*.

O circuito utilizado nesse teste é apresentado na Figura 58, onde por ele ter ligado na saída do bloco PI um bloco limitador, foi necessário testar seus parâmetros nessa configuração.

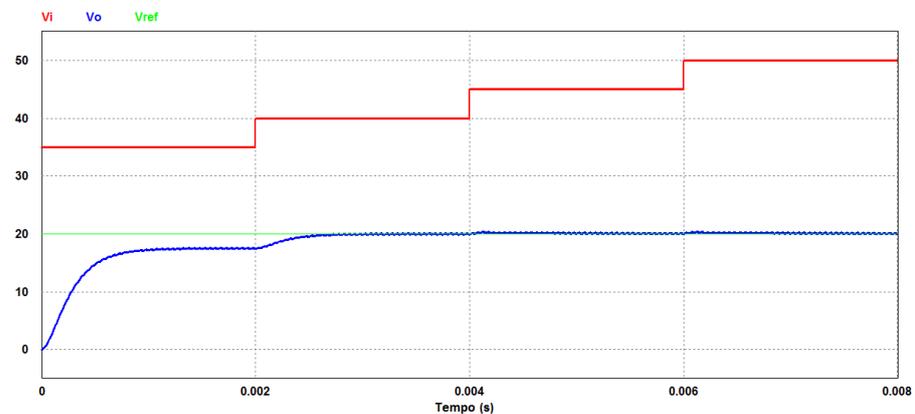
Figura 58 – Circuito de teste em rampa para os dados de Souza (2021)



Fonte: Elaborado pelo autor.

O resultado desta simulação está apresentado na Figura 59.

Figura 59 - Teste em rampa de 35 a 50 V com parâmetros de Souza (2021)



Fonte: Elaborado pelo autor.

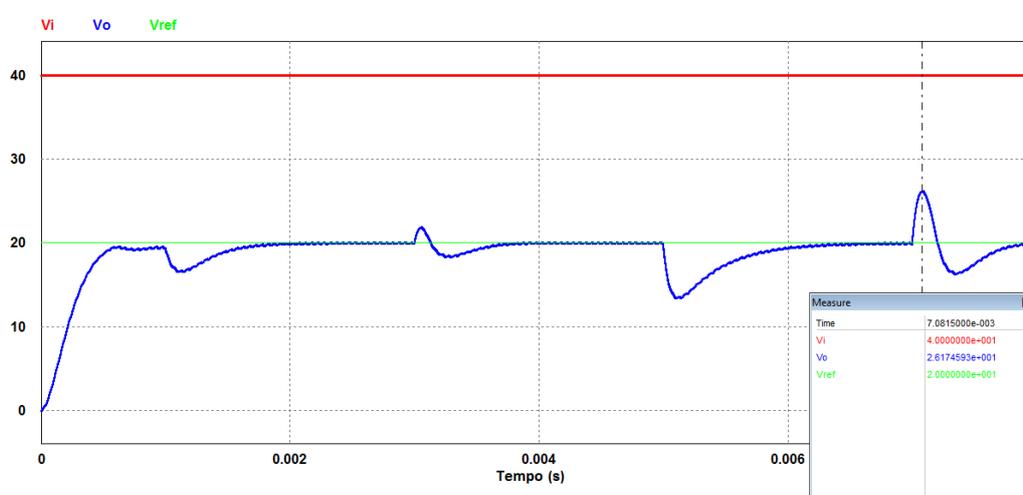
Observa-se que na tensão de entrada de 35 V, a tensão de saída não consegue alcançar a referência do *set point* de 20 V, e isso só se resolve quando ocorre a transição para a tensão de 40 V, onde a curva inicia o seguimento de referência corrigindo o erro em regime permanente permanecendo estável para as demais tensões.

#### 4.2.2 Teste no *Buck* com variação de resistência na carga de 3, 4 e 5 ohms

O próximo teste foi criar uma variação de resistência na carga (rampa) de 3, 4 e 5 ohms mantendo a tensão de entrada fixa em 40 V e a referência de tensão de saída em 20 V. O circuito que foi elaborado para esse teste está registrado na Figura 30.

A Figura 60 apresenta o teste de desempenho com os ganhos do controlador de Souza (2021), onde a resistência da carga partiu com 5 ohms da origem, e no tempo de (1 ms) recebeu uma nova resistência de 20 ohms em paralelo, totalizando uma resistência equivalente de 4 ohms, em seguida no instante de (3 ms), a resistência de 20 ohms é desconectada do circuito, voltando o mesmo a operar com 5 ohms, até que no instante de tempo de (5 ms) uma nova carga é conectada em paralelo com valor de (7,5 ohms), totalizando uma resistência equivalente de 3 ohms, o tempo continua até que a chave se abre no tempo (7 ms) e a carga é removida voltando o circuito a operar com 5 ohms de carga total.

Figura 60 - Teste de variação de resistência com parâmetros de Souza (2021)

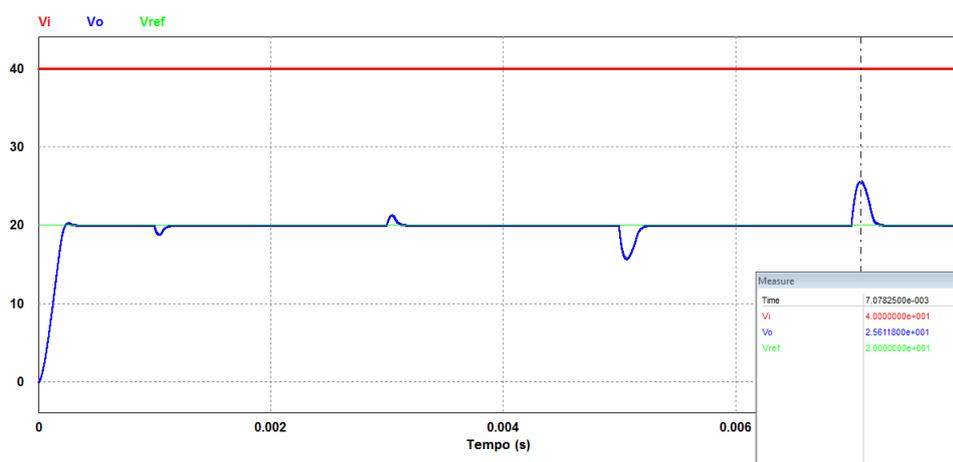


Fonte: Elaborado pelo autor.

Observou-se que a curva da tensão de saída, precisa de um tempo maior para acompanhar o sinal de referência, frente aos distúrbios inseridos no sistema. No final, quando a chave é aberta para desconexão da resistência paralelo, observou-se um *overshoot* de 30%.

Para fins de comparação, o mesmo teste foi feito com os parâmetros do índice ITSE do PSO, onde sua resposta é mostrada na Figura 61.

Figura 61 - Teste de variação de resistência com parâmetros do PSO-ITSE



Fonte: Elaborado pelo autor.

É possível observar que para os valores de resistência de 4 ohms onde o distúrbio ocorreu, a curva respondeu rapidamente, mas para a carga de 3 ohms, demonstrou percentual de ultrapassagem considerável. Foi registrado um *overshoot* de 28% na desconexão da resistência paralelo de 3 ohms.

#### 4.3 Análise de desempenho do conversor CC-CC Boost – Carga Eletrônica

Para o circuito da carga eletrônica, não foi possível utilizar o controlador do tipo PID, devido a média do sinal de corrente ter ficado muito baixa, não satisfazendo os testes de desempenho. Foi adotado então, o controlador na configuração PI, que apresentou melhor desempenho, permitindo encontrar bons parâmetros para sintonia.

Foram realizadas baterias de 10 simulações com cada função objetivo, e da mesma forma como no conversor *Buck*, mudanças nos parâmetros de cada algoritmo foram feitas para encontrar respostas mais eficientes, onde foi escolhido aquela que desempenhou melhor resultado no controle do conversor.

A Tabela 8 resume os resultados com cada função objetivo.

Tabela 8 – Resultados da otimização no conversor CC-CC *Boost*

	Critério	$K_p$	$K_i$	(%UP)	$t_s(s)$	$t_r(s)$	IL Média (A)	Fitness Custo
Souza	Empírico	0,0030	3333,3330	0,8102	0,0102	0,0060	19,5857	-
GA	IAE	0,0005	3671,4422	0,8572	0,0098	0,0087	19,6046	7,9524
	ITAE	0,0009	3810,3941	0,7115	0,0194	0,0168	18,1664	0,0072
	ISE	0,0008	3514,3738	0,7067	0,0235	0,0204	19,4992	18,9871
	ITSE	0,0006	3610,9850	0,7058	0,0299	0,0263	19,4999	0,4621
PSO	IAE	0,0009	4975,8098	0,7292	0,0144	0,0129	19,9762	0,7958
	ITAE	0,0009	4921,5929	0,7271	0,0145	0,0130	19,9764	0,0198
	ISE	0,0016	5000,0000	0,8080	0,0277	0,0077	19,9684	18,5843
	ITSE	0,0009	5000,0000	0,7628	0,0143	0,0128	19,9798	0,4604

Fonte: Elaborado pelo autor.

Mantendo os mesmos critérios de avaliação dos testes com o conversor *Buck*, foi escolhido com melhor desempenho, aquela função que conseguiu atingir nas características de resposta transitória, o menor *overshoot* (%UP), menor tempo de estabilização ( $t_s$ ) e menor tempo de subida ( $t_r$ ). Neste cenário avaliando o desempenho do algoritmo genético o índice ITSE foi o que apresentou o menor *overshoot* com 0,7058%, porém nos outros critérios sua curva foi a que teve o maior tempo de acomodação, assim como o maior tempo de subida, podendo ser visto na Figura 62 na cor *ciano*. Neste cenário adotou-se como melhor desempenho o índice IAE, que apresentou melhor tempo de subida, como também melhor tempo de acomodação. No comparativo com os resultados de Souza (2021), o IAE apresentou um *overshoot* um pouco menor com uma boa acomodação, porém não foi tão rápido. Já nos testes com PSO, tivemos três índices praticamente empatados (IAE, ITAE e ITSE) apresentando *overshoot* na faixa dos 0,7%, sendo assim, o critério de escolha foi para aquele que teve o menor tempo de subida e acomodação, sendo escolhido o ITSE.

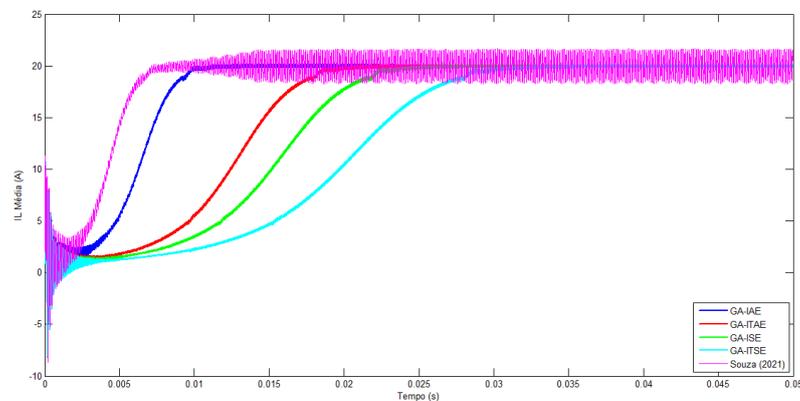
#### 4.3.1 Teste com tensão de entrada fixa em 25 V e referência de corrente em 20 A

Após a obtenção dos melhores valores para parametrização do controlador PI, partiu-se para o teste de desempenho da carga eletrônica no PSIM, onde foi simulado uma tensão de entrada de 25 V e um sinal de referência de corrente em 20 A, onde o objetivo era verificar qual das funções objetivos conseguia reproduzir os coeficientes de ganho proporcional e ganho integral, satisfazendo a sintonia do controlador, afim

de fazer a carga drenar essa corrente da fonte de entrada, mantendo o valor de referência, com a melhor resposta transitória.

A Figura 62, apresenta todas as curvas de resposta obtidas pelo GA, para a corrente de entrada da carga eletrônica, comparando o melhor resultado de cada função objetivo em relação ao resultado de Souza (2021). É possível perceber que o aumento da tensão de entrada de 25 V, ocasionou uma oscilação maior de corrente na curva do autor, devido a carga e descarga do indutor  $L_1$ , mas que na média ficou próxima dos 20 A do sinal de referência.

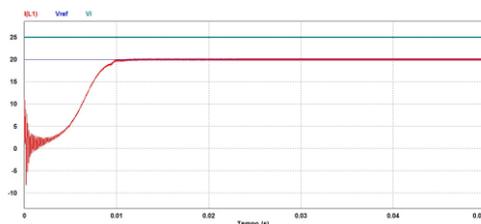
Figura 62 - Respostas do GA na carga eletrônica, com tensão de entrada em 25 V



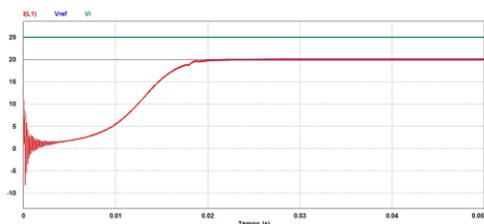
Fonte: Elaborado pelo autor.

A Figura 63 apresenta as curvas individuais de corrente drenada pela carga.

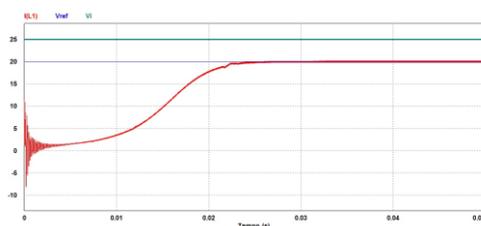
Figura 63 – Respostas individuais do GA para cada índice de desempenho



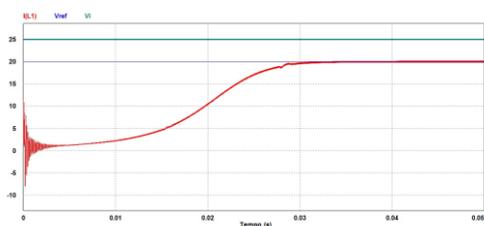
a) GA-IAE



b) GA-ITAE



c) GA-ISE

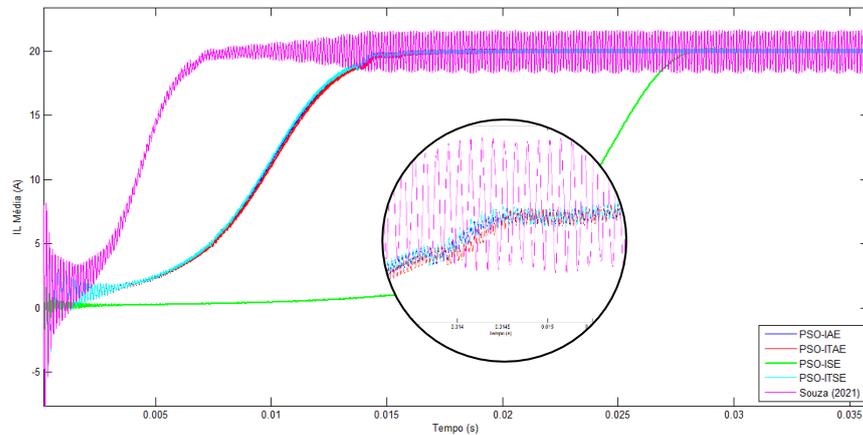


d) GA-ITSE

Fonte: Elaborado pelo autor.

A Figura 64, apresenta todas as curvas de resposta obtidas pelo PSO, para a corrente de entrada da carga eletrônica

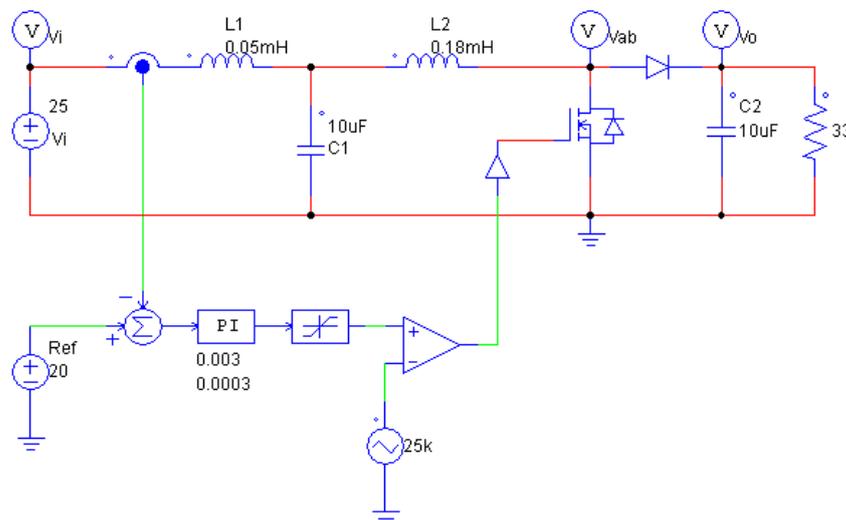
Figura 64 - Respostas PSO na carga eletrônica, com tensão de entrada em 25 V



Fonte: Elaborado pelo autor.

Para a reprodução da curva de resposta do autor Souza (2021), foi necessária uma modificação no circuito da Figura 35, em relação a construção do controlador PI, onde o autor utilizou em seu trabalho um bloco pronto do PI ligado a um bloco chamado *Limiter* que é um circuito limitador responsável por manter a razão cíclica do conversor entre os valores mínimos e máximos especificados. A Figura 65 apresenta o circuito de controle PI com seus respectivos valores de ganho proporcional e derivativo, assim como a conexão do bloco *Limiter*.

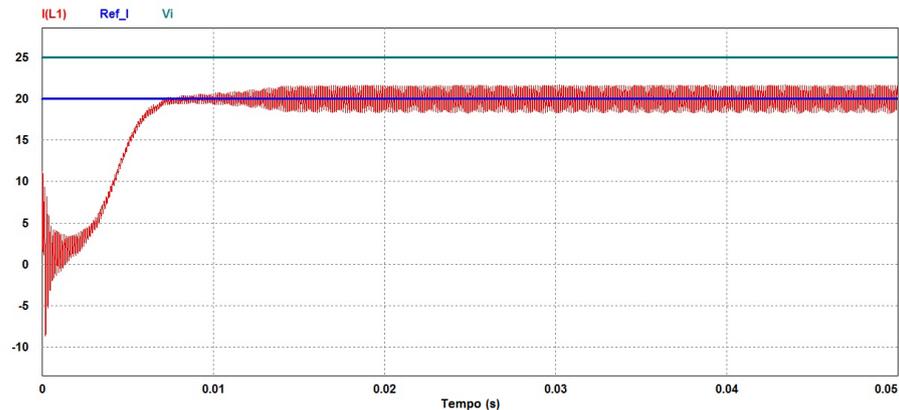
Figura 65 - Circuito carga eletrônica com controle PI em bloco



Fonte: Elaborado pelo autor.

Com este circuito foi extraída a curva de resposta, para os sinais da tensão de entrada em 25 V e sinal de referência de corrente em 20 A. A Figura 66, apresenta a corrente de entrada sendo drenada pela carga, atingindo em média o *set point*, junto dos sinais de tensão e de referência.

Figura 66 - Corrente drenada pela carga, na sintonia de Souza (2021)



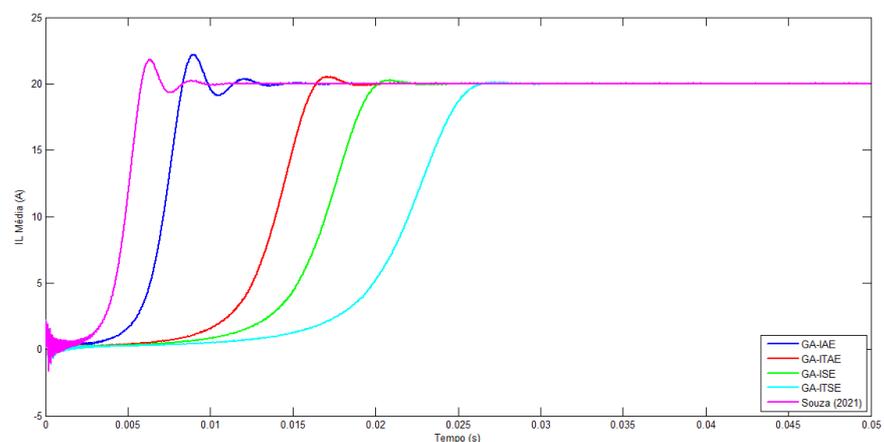
Fonte: Elaborado pelo autor.

#### 4.3.2 Teste com tensão de entrada em 5 V e referência de corrente em 20 A

Para execução deste teste, foi alterado o valor da tensão da fonte de entrada do circuito para 5 volts, onde o objetivo assim como no teste anterior, foi de atingir o valor de referência em 20 A com a melhor resposta possível.

A Figura 67 apresenta as curvas das respostas obtidas pelo GA, para a corrente drenada pela carga eletrônica, quando sua tensão de entrada é colocada em 5 V.

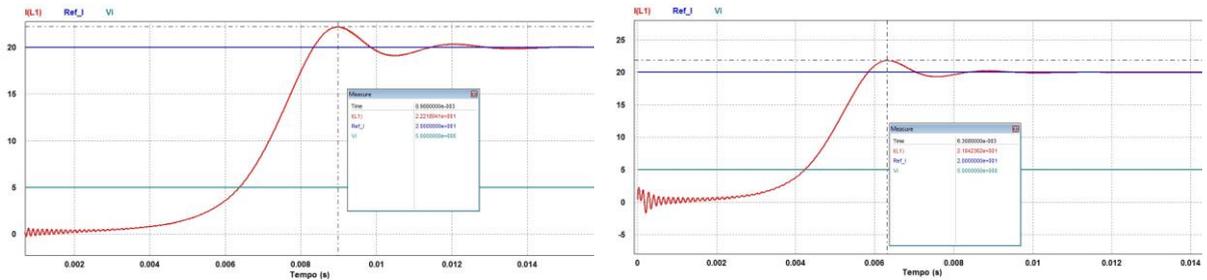
Figura 67 - Respostas do GA na carga eletrônica, com tensão de entrada em 5 V



Fonte: Elaborado pelo autor.

O índice GA-IAE escolhido no teste anterior, aqui demonstrou um *overshoot* de 11%, ele foi comparado com o resultado de Souza (2021), e seus detalhes são destacados na Figura 68.

Figura 68 – Comparativo de *overshoot* no teste de tensão de entrada de 5 V



a) GA-IAE *MP* de 11%

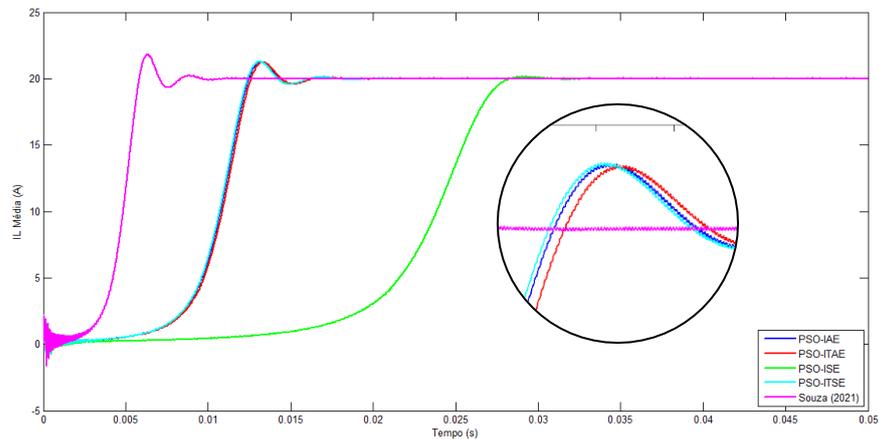
b) Souza (2021) *MP* de 9%

Fonte: Elaborado pelo autor.

Como o nível de *overshoot* do GA-IAE foi 2% acima do resultado do autor, decidiu-se manter o índice e verificar seu comportamento no próximo teste.

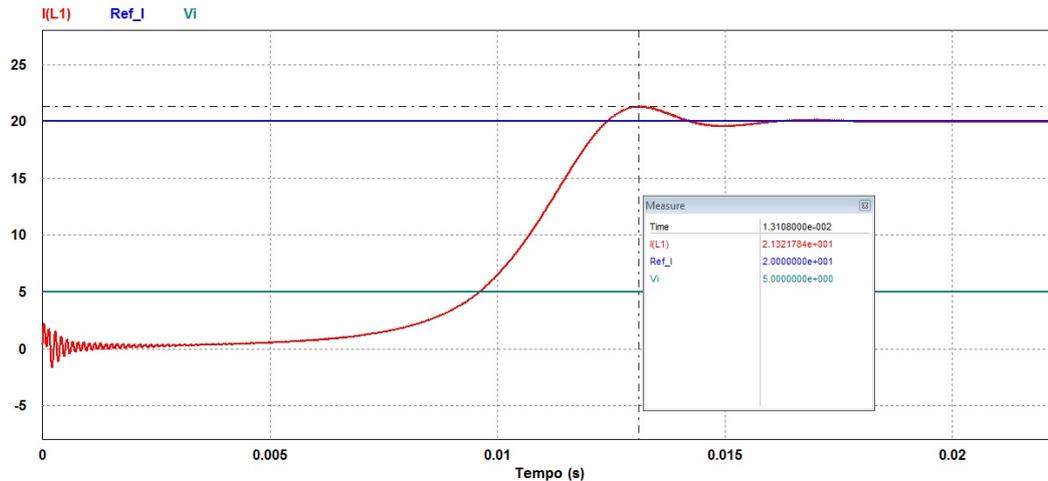
Assim também no PSO, foram realizados os testes com todos os índices de desempenho e seus resultados podem ser visualizados na Figura 69.

Figura 69 - Respostas do PSO na carga eletrônica, com tensão de entrada em 5 V



Fonte: Elaborado pelo autor.

Para o índice escolhido PSO-ITSE, este também apresentou *overshoot* junto dos outros dois índices (IAE e ITAE) que ficaram quase sobrepostos. Sua resposta é mostrada através da Figura 70.

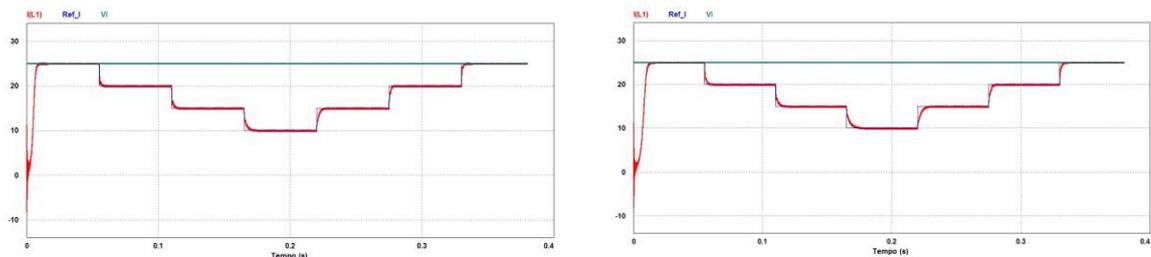
Figura 70 - Resposta PSO-ITSE com *overshoot* de 6%

Fonte: Elaborado pelo autor.

#### 4.3.3 Teste em rampa de corrente de 10 a 25 A com entrada fixa em 25 V

Conforme o circuito da Figura 35, foi utilizado um bloco chamado *Piecewise linear voltage source*, que permitiu a criação de uma rampa de sinais especificando um valor e um tempo a cada ponto. Foi escolhido variar o sinal de corrente de referência, na faixa de especificação da carga eletrônica (10 a 25 A), onde iniciou-se diminuindo a corrente de 25 A para 10 A e depois subindo dos 10 A para os 25 A, observando o comportamento da corrente da fonte, se ela conseguia seguir a referência. A tensão de entrada da fonte ficou fixa em 25 V. Os testes foram feitos apenas com os índices escolhidos, sendo para o GA o índice IAE, e para o PSO o índice ITSE. A Figura 71 mostra o resultado para o teste em rampa de forma comparativa, entre os dois índices.

Figura 71 – Teste de rampa de corrente na carga eletrônica com GA e PSO



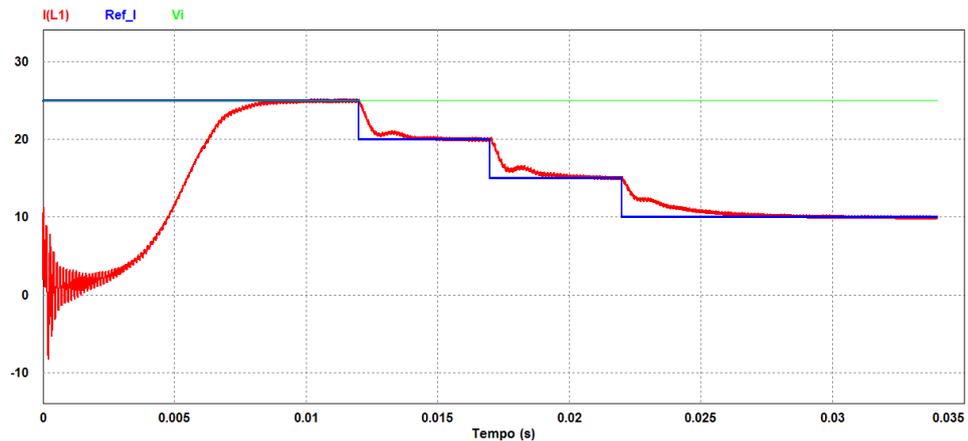
a) GA-IAE

b) PSO-ITSE

Fonte: Elaborado pelo autor.

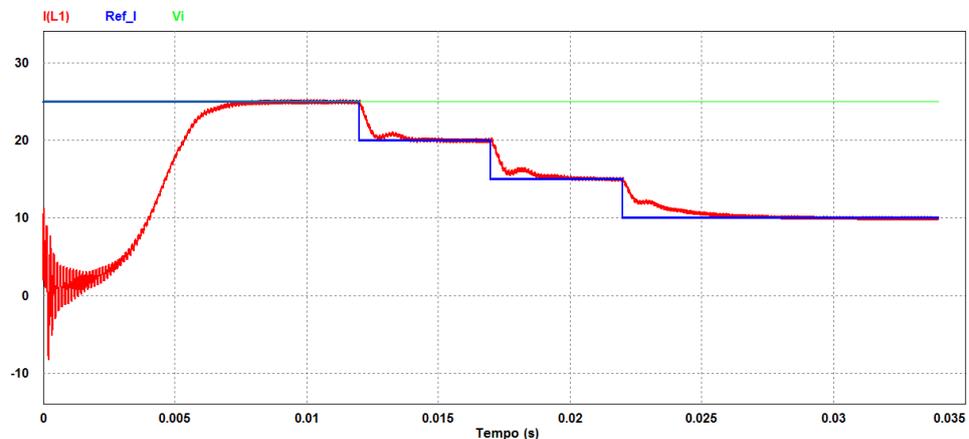
Para uma melhor visualização da Figura 71, foi aumentado a resolução de cada resultado e diminuído o tempo de comutação da rampa, onde foi feita uma simulação variando apenas o degraú de descida de 25 a 10 A, onde seu resultado pode ser observado na Figura 72 e Figura 73.

Figura 72 – Teste em rampa na carga eletrônica com parâmetros do GA-IAE



Fonte: Elaborado pelo autor.

Figura 73 - Teste em rampa na carga eletrônica com parâmetros do PSO-ITSE



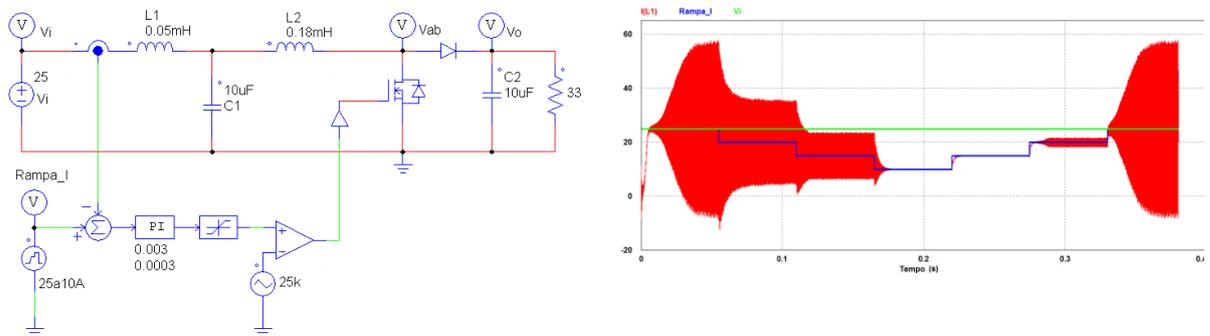
Fonte: Elaborado pelo autor.

Os resultados ficaram parecidos, mas podemos observar que o índice ITSE do PSO leva um tempo menor para acomodar a resposta.

Para o teste comparativo com a curva de desempenho do autor Souza (2021) foi adicionado ao circuito da Figura 65 o bloco *Piecewise linear voltage source*, para criação da mesma rampa de corrente, e ser testado as mesmas condições dos testes feitos com os índices de desempenho IAE e ITSE.

A Figura 74, apresenta o circuito utilizado para o teste e sua resposta, onde foram mantidos os parâmetros dos testes anteriores.

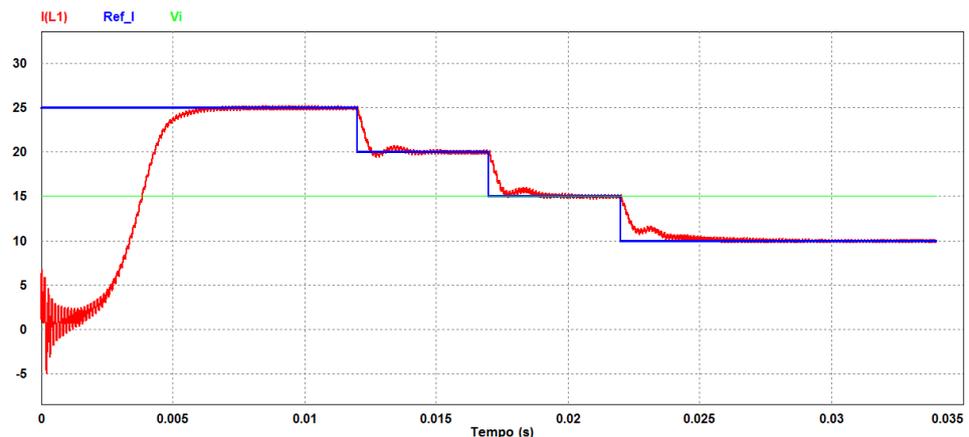
Figura 74 - Circuito de controle PI com rampa de corrente de 10 a 25 A



Fonte: Elaborado pelo autor.

Neste teste, foi possível observar que os parâmetros deste controlador não conseguiram manter a corrente drenada como os demais índices fizeram, causando um *ripple* elevado de corrente. Em outro momento, baixando a tensão de entrada para 15 V, permitiu que o conversor projetado acompanhasse as variações de corrente desejada, sem saturação. A Figura 75, representa essa mudança já com o ajuste da resolução para uma melhor visualização do tempo de acomodação.

Figura 75 – Resposta a rampa de corrente com tensão de entrada em 15 V



Fonte: Elaborado pelo autor.

Com isso encerra-se os testes e simulações dos conversores *Buck* e *Boost* como carga eletrônica, onde o índice vencedor foi o ITSE com o algoritmo enxame de partículas PSO.

## 5 CONCLUSÃO

Como considerações finais, neste trabalho foram desenvolvidas soluções utilizando do campo da inteligência artificial, as meta-heurísticas como os algoritmos genéticos (GA) e o enxame de partículas (PSO), a fim de encontrar a sintonia dos parâmetros de dois controladores, sendo um na configuração PI e outro na PID. O controlador PID foi utilizado num conversor CC-CC do tipo *Buck – Step Down* escolhido como uma das plantas a ser controlada. E o controlador PI foi utilizado num conversor CC-CC do tipo *Boost - Step up* configurado como uma carga eletrônica, como sendo a segunda planta a ser controlada. Funções objetivo foram criadas e índices de desempenho como (IAE, ITAE, ISE e ITSE) foram colocados em teste para ver qual encontraria o menor custo destas funções e que apresentassem as melhores respostas transitórias, como mínimo *overshoot*, mínimo tempo de acomodação e mínimo tempo de subida, deixando assim, o sistema mais robusto para reagir frente à distúrbios, provocados durante os testes de regime permanente. Foram realizadas comparações com outros trabalhos que permitiram concluir, que as técnicas abordadas durante o presente trabalho tiveram êxito, uma vez que foi possível sintonizar as duas plantas de forma satisfatória. O resultado da otimização para as duas plantas consagrou como vencedor o algoritmo por enxame de partículas (PSO), utilizando o índice de desempenho ITSE. Isso não quer dizer que o GA não tenha desempenhado um bom trabalho, e sim que de acordo com a avaliação do projetista, para cada situação um ou outro resultado, pode ser utilizado de acordo com a resposta que se busca. Nos resultados das simulações, o tempo de processamento computacional foi demasiadamente longo, principalmente por não se conhecer os limites do espaço de busca do problema, assim como as possíveis combinações de ajuste de cada algoritmo, levando a se trabalhar com um número grande de simulações.

Como sugestões para trabalhos futuros, seria importante para fins de automação da simulação, realizar uma função multiobjetivo que verificasse além do índice de desempenho, as respostas transitórias do processo durante a otimização, fazendo o algoritmo avaliar o candidato a solução com base em critérios desejados pelo projetista, como mínimo *overshoot* ou tempo de acomodação. Outra sugestão seria implementar as plantas utilizadas nesse trabalho e aplicar os parâmetros encontrados nos controladores PI e PID, comprovando a prática com a simulação.

## REFERÊNCIAS

AHMED, Ashfaq. **Eletrônica de Potência**. São Paulo: Pearson Prentice Hall, 2000.

ANGADI, Sachin; SAIKUMAR, Vakkala; SATYANARAYANA KUMARI, B. **A novel digital controller for microstepping stepper motor drive using FPGA for Solar Array Drive Assembly in satellites-A comparison With alternative schemes**. In: 2013 INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS (ICACCI) 2013, Anais... : IEEE, 2013.

ARRABAÇA, Devair A.; GIMENEZ, Salvador P. **Conversores de Energia Elétrica CC/CC para Aplicações em Eletrônica de Potência - Conceitos, Metodologia de Análise e Simulação**. 1ª ed. São Paulo: Editora Érica Ltda, 2014.

ÅSTRÖM, K. J.; HÄGGLUND, T. The future of PID control. **Control Engineering Practice**, [s. l.], v. 9, n. 11, p. 1163–1175, 2001.

BARBI, Ivo. **Modelagem de Conversores CC-CC Empregando Modelo Médio em Espaço de Estados**. Edição do ed. Florianópolis.

BK PRECISION, CORP. **Conceitos sobre carga eletrônica DC**. 2017. Disponível em: <<https://www.youtube.com/watch?v=OTJufSYGum8&t=23s>>. Acesso em: 8 maio. 2022.

CAMPOS, Mario Massa De; TEIXEIRA, Hebert Campos Gonçalves. **Controles típicos de equipamentos e processos industriais**. 2. ed. São Paulo: Edgar Blucher, 2010.

CLERC, Maurice; KENNEDY, James. **The particle swarm-explosion, stability, and convergence in a multidimensional complex space**. IEEE Transactions on Evolutionary Computation, [s. l.], v. 6, n. 1, p. 58–73, 2002.

COACH, Electronics. **Proportional Integral (PI) Controller**. 2022. Disponível em: <<https://electronicscoach.com/proportional-integral-controller.html>>. Acesso em: 23 maio. 2022.

DE FARIA, Julliano H. S. et al. **Conversor boost atuando como carga eletrônica a corrente controlada para fins de teste de uma célula a combustível**. 2010 9th IEEE/IAS International Conference on Industry Applications, INDUSCON 2010, [s. l.], n. December 2013, p. 1–6, 2010.

DIAS, Luciano Resende et al. **Uso de Algoritmo Genético na Otimização do Tempo de Sintonia de Controlador PID com dois Graus de Liberdade**. Linkania, [s. l.], v. 5, n. 1, p. 197–226, 2015.

DORF, Richard C.; BISHOP, Robert H. **Sistemas de Controle Modernos**. 8ª ed. Rio de Janeiro: LTC, 2001.

FRANKLIN, Gene F.; POWELL, J. David; EMANI-NAEINI, Abbas. **Sistemas de**

**Controle para Engenharia.** 6ª ed. Porto Alegre: Bookman, 2013.

GANGISHETTI, Srinivas; SANDIPAMU, Tarakalyani. **Tuning of PI Controllers by GA and PSO Techniques for Sensorless Vector Control of Induction Motor** †. National Conference on Recent Advancement in Electronics (NCRAE-2017), [s. l.], p. 39–52, 2017.

GOLDBARG, Marco Cesar; GOLDBARG, Gouveia Elizabeth; LUNA, Henrique Pacca Loureiro. **Otimização combinatória e meta-heurísticas: algoritmos e aplicações.** 1. ed. Rio de Janeiro: Elsevier, 2016.

GOLDBERG, David E. **Genetic algorithms in search, optimization, and machine learning.** Addison-We ed. Massachusetts: Company, Inc., 1989. v. 27

HERIS, Mostapha Kalami. **About Yarpiz.** 2015. Disponível em: <<https://yarpiz.com/about>>. Acesso em: 19 maio. 2022.

HU, Guozhen et al. **Constant Current Control of DC Electronic Load based on Boost Topology.** Electronics and Electrical Engineering, [s. l.], v. 20, n. 2, p. 36–39, 2014.

HUANG, W.; LAM, H. N. **Using genetic algorithms to optimize controller parameters for HVAC systems.** Energy and Buildings, [s. l.], v. 26, n. 3, p. 277–282, 1997.

ISERMANN, Rolf. **Digital Control Systems.** 2. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989.

KENNEDY, James; EBERHART, Russell. **Particle swarm optimization.** In: PROCEEDINGS OF ICNN'95 - INTERNATIONAL CONFERENCE ON NEURAL NETWORKS 1995, Perth, WA, Australia. Anais... Perth, WA, Australia: IEEE, 1995.

LIPING GUO; HUNG, John Y.; NELMS, R. M. **Evaluation of DSP-Based PID and Fuzzy Controllers for DC–DC Converters.** IEEE Transactions on Industrial Electronics, [s. l.], v. 56, n. 6, p. 2237–2248, 2009.

LUCAS, Ricardo. **Análise e implementação de estruturas de controle em dispositivo FPGA aplicadas a um conversor Buck.** 2015. UTFPR, Ponta Grossa, 2015.

MATTOS, EVERSON et al. **Um Procedimento de Projeto de Controladores PID para Conversores CC-CC com Validação em Hardware-in-the-Loop.** In: PROCEEDINGS OF THE 13TH SEMINAR ON POWER ELECTRONICS AND CONTROL (SEPOC 2021) 2021, Anais... : sepoc, 2021.

MEHTA, Nikunj et al. **Design of HMI Based on PID Control of Temperature.** International Journal of Engineering Research and, [s. l.], v. V6, n. 05, p. 117–120, 2017.

MELLO, Luiz Fernando Pereira De. **Projetos de Fontes Chaveadas Teoria e Prática.** 1ª ed. São Paulo: Editora Érica Ltda, 2011.

NETO, Accacio F. Santos; GOMES, José Francisco. **Controladores Pid : Introduzindo Inteligência Computacional no Controle Industrial**. Cobenge, [s. l.], p. 10, 2010.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 5. ed. ed. São Paulo: Pearson Education, Inc., 2011.

OLIVEIRA, Phelipe Wesley De. **Contribuições ao Problema de Sintonia Ótima de Controladores PID de Ordens Inteira e Fracionária via Otimização Metaheurística**. 2020. UFC - Universidade Federal do Ceará, Fortaleza, 2020.

OUYANG, Puren; PANO, Vangjel. **Comparative Study of DE, PSO and GA for Position Domain PID Controller Tuning**. Algorithms, [s. l.], v. 8, n. 3, p. 697–711, 2015.

PAIVA, Leonardo Silveira. **Aplicação de Algoritmos Genéticos para Sintonia de Controladores**. 2010. Instituto Superior de Engenharia do Porto, Porto, 2010.

POWERSIM, Psim. **Tutorial on How to Use the SimCoupler Module**. [s.l.: s.n.].

RASHID, Muhamad H. **Eletrônica de potência: dispositivos, circuitos e aplicações**. 4ª ed. São Paulo: Pearson Education do Brasil, 2014.

SHI, Yuhui; EBERHART, Russell. **A modified particle swarm optimizer**. In: 1998 IEEE INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION PROCEEDINGS. IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE (CAT. NO.98TH8360) 1998, Anais... : IEEE, 1998.

SOUZA, João Olegário de Oliveira De. **Conversores Chaveados**, São Leopoldo, 2021.

SOUZA, João Olegário de Oliveira De; VALIATI, João Francisco; GÓMEZ, Arthur Tórgo. **Metaheurísticas aplicadas a Sintonia de Controladores PID: Estudo de Casos**. Congresso Latino-Iberoamericano de Investigación Operativa, [s. l.], p. 2600–2611, 2012.

TALBI, El-Ghazali. **Metaheuristics: From Design to Implementation**. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2009.

VAROL, Huseyin Atakan; BINGUL, Zafer. **A new PID tuning technique using ant algorithm**. In: PROCEEDINGS OF THE 2004 AMERICAN CONTROL CONFERENCE 2004, Boston. Anais... Boston: IEEE, 2004.

ZAMBON, Vitor Biagiolli; VARGAS, Francisco J. Triveno; FLORIAN, Fabiana. **Simulação de conversor CC-CC tipo Buck usando Matlab-Simulink**. Brazilian Journal of Development, [s. l.], v. 6, n. 7, p. 50623–50632, 2020.