

Universidade Federal de Ouro Preto ICEA Departamento de Engenharia Elétrica



Simulação de elementos da eletrônica de potência usando Hardware-in-the-Loop

Gustavo Araújo de Souza

João Monlevade MG 2022 Gustavo Araújo de Souza

Simulação de elementos da eletrônica de potência usando Hardware-in-the-Loop

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau em Bacharel em Engenharia Elétrica.

Orientador: Welbert Alves Rodrigues

Coorientador: Renan Fernandes Bastos

João Monlevade - MG

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S729s	Souza, Gustavo Araújo de. Simulação de elementos da eletrônica de potência usando Hardware- in-the-Loop. [manuscrito] / Gustavo Araújo de Souza 2022. 69 f.: il.: color., tab
	Orientador: Prof. Dr. Welbert Alves Rodrigues. Coorientador: Prof. Dr. Renan Fernandes Bastos. Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia Elétrica.
	1. Circuitos eletrônicos. 2. Conversores de corrente elétrica. 3. Eletrônica de potência. 4. Microcontroladores. 5. Processamento eletrônico de dados em tempo real. I. Bastos, Renan Fernandes. II. Rodrigues, Welbert Alves. III. Universidade Federal de Ouro Preto. IV. Título.
	CDU 621.3

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DE OURO PRETO REITORIA INSTITUTO DE CIENCIAS EXATAS E APLICADAS DEPARTAMENTO DE ENGENHARIA ELETRICA



FOLHA DE APROVAÇÃO

Gustavo Araújo de Souza

Simulação de elementos da eletrônica de potência usando Hardware-In-the-Loop

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro Eletricista

Aprovada em 15 de Junho de 2022

Membros da banca

Dr - Welbert Alves Rodrigues - Orientador - UFOP Dr - Renan Fernandes Bastos - Co-Orientador - UFOP Dr - Marcelo Moreira Tiago - UFOP MSc - Thainan Santos Theodoro - UFOP

Welbert Alves Rodrigues, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 23/06/2022



Documento assinado eletronicamente por **Welbert Alves Rodrigues, PROFESSOR DE MAGISTERIO SUPERIOR**, em 23/06/2022, às 22:08, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do <u>Decreto nº 8.539, de 8 de outubro de 2015</u>.



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php? acao=documento conferir&id_orgao_acesso_externo=0, informando o código verificador 0350721 e o código CRC 0460D8D0.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.008311/2022-28

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000 Telefone: (31)3808-0818 - www.ufop.br

Agradecimentos

Primeiramente gostaria de agradecer a Deus por me abençoar em todos os momentos vividos até aqui.

Agradeço ao meu orientador Welbert por aceitar conduzir essa proposta de monografia e ao coorientador Renan, que tem me ajudado muito na caminhada até aqui.

A todos os meus professores do curso de engenharia elétrica da Universidade Federal de Ouro Preto pela excelência da qualidade técnica.

Aos meus pais Reinaldo e Valéria que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória e a minha namorada por estar comigo independente dos problemas.

E por fim as pessoas que participaram dos momentos de estudo e lazer que permitiram minha chegada até aqui, Gabriel, Adeilson, Lorran, Guilherme e Flávio.

Resumo

Os circuitos da eletrônica de potência devem ser validados por uma sequência de testes, que de preferência envolvam um baixo custo e uma alta segurança, dependendo de uma montagem física rápida para avaliação do seu comportamento prático. Tais procedimentos apresentam uma baixa segurança e um custo elevado, visto que qualquer erro de projeto pode causar uma perda significativa para a produção. Dentre as técnicas que são utilizadas para contornar esses problemas, a mais comum é o Hardware-In-the-Loop, que tem como objetivo permitir a validação de um sistema físico por meio de seu modelo simulado em tempo real, possibilitando a conexão da planta virtual com elementos físicos externos. Isso torna o processo mais benéfico em determinados casos, pois não é preciso construir um sistema real para a realização de testes específicos. Porém, ainda há dificuldades de expansão dessa implementação em algumas aplicações, sendo a maior delas o alto custo das plataformas digitais presentes no mercado. Por isso o objetivo do trabalho foi de utilizar um microcontrolador de baixo custo (Launchxl-f28379d) para simular os modelos de topologias comuns encontradas na literatura da eletrônica de potência. A metodologia abordada nessa monografia envolve a simulação nas duas configurações principais do HIL, para malha aberta e malha fechada, onde os conversores Buck, Boost, Buck-boost e o inversor monofásico são modelados por equações diferenciais e discretizados para a implementação no hardware digital escolhido. Para a configuração malha fechada o modelo do conversor buck é colocado em realimentação com um controlador PI, de forma a testar a iteração entre ambos. Os resultados experimentais obtidos de cada circuito são comparados com as respostas de um software computacional, sendo possível perceber as semelhanças e diferenças apresentadas em cada caso, que possibilitam uma base para a validação. O que se pode notar é que o resultado obtido para cada circuito em HIL seguiu o comportamento esperado perante a uma sequência de testes, deixando a contribuição da possibilidade de uso desse microcontrolador como simulador digital em tempo real de baixo custo para sistemas de complexidade compatível as suas especificações.

Palavras-chave: Launchxl-f28379d, Conversores, *Hardaware-In-the-Loop*, Simulação em tempo real.

Abstract

Power electronics circuits must be validated by a sequence of tests, which preferably involve low cost and high security, depending on a quick physical assembly to evaluate their practical behavior. Such procedures present a low safety and a high cost, since any design error can cause a significant loss to the production. Among the techniques that are used to circumvent these problems, the most common is Hardware-In-the-Loop, which aims to allow the validation of a physical system through its simulated model in real time, enabling the connection of the plant virtual with external physical elements. This makes the process more beneficial in certain cases, as it is not necessary to build a real system to perform specific tests. However, there are still difficulties in expanding this implementation in some applications, the biggest of which is the high cost of digital platforms on the market. Therefore, the objective of this work was to use a low-cost microcontroller (Launchxl-f28379d) to simulate the common topology models found in the power electronics literature. The methodology addressed in this monograph involves the simulation in the two main configurations of the HIL, for open loop and closed loop, where the Buck, Boost, Buck-boost converters and the single-phase inverter are modeled by differential equations and discretized for implementation in the chosen digital hardware. . For the closed loop configuration, the buck converter model is placed in feedback with a PI controller, in order to test the iteration between them. The experimental results obtained from each circuit are compared with the responses of a computational software, making it possible to perceive the similarities and differences presented in each case, which provide a basis for validation. What can be noticed is that the result obtained for each circuit in HIL followed the expected behavior before a sequence of tests, leaving the contribution of the possibility of using this microcontroller as a low-cost real-time digital simulator for systems of complexity compatible with the your specifications.

Keywords: Launchxl-f28379d, Converters, Hardaware-In-the-Loop, Real-time Simulation.

Lista de figuras

Figura 1 –	Principais configurações de arquitetura para o Hardware-in-The-Loop	2
Figura 2 –	Conversor <i>Buck</i> ideal	6
Figura 3 –	Comportamento ideal da corrente i_L no conversor <i>Buck</i>	7
Figura 4 –	Conversor <i>Boost</i> ideal	8
Figura 5 –	Comportamento da corrente i_L no conversor <i>Boost</i>	8
Figura 6 –	Conversor Buck-Boost ideal.	9
Figura 7 –	Comportamento da corrente i_L no conversor <i>Buck-Boost</i>	9
Figura 8 –	Inversor Monofásico ideal.	10
Figura 9 –	Comportamento do inversor monofásico para corrente e tensão de saída	11
Figura 10 –	Modulação PWM para onda senoidal	12
Figura 11 –	Esquema de malha fechada do controlador PI	13
Figura 12 –	Representação dos ganhos no controlador PI em malha fechada	14
Figura 13 –	Aproximações Backward e Forward para uma função	15
Figura 14 –	Configuração padrão do HIL em malha fechada.	18
Figura 15 –	Simuladores mais utilizados presentes no mercado, sendo: a) RTDS e b)	
	Typhoon HIL.	19
Figura 16 –	Microcontrolador f28379d da texas instruments.	21
Figura 17 –	Configuração do conversor <i>Buck</i> no modo: a) chave fechada e b) chave aberta.	23
Figura 18 –	Diagrama de blocos para algoritmo no microcontrolador LAUNCHXL	26
Figura 19 –	Esquema de ligação entre processadores.	27
Figura 20 –	Diagrama conexão PI na prática.	28
Figura 21 –	Plataforma <i>sisotool</i> representando os ganhos obtidos para o controlador PI.	31
Figura 22 –	PWM uniformemente amostrado com: a) Portadora Dente de serra, b) Porta-	
	dora Triangular.	32
Figura 23 –	Configuração do conversor <i>Boost</i> no modo: a) chave fechada e b) chave aberta.	33
Figura 24 –	Configuração do conversor <i>Buck-Boost</i> no modo: a) chave fechada e b) chave	
	aberta.	34
Figura 25 –	Topologia Inversor Monofásico ponte H.	35
Figura 26 –	Equivalente do Inversor Monofásico.	35
Figura 27 –	Esquema de representação do conversor <i>Buck</i> no PSIM	37
Figura 28 –	Comportamento de i_L para variação de entrada no conversor <i>Buck</i> em ambas	
	simulações, na qual a) Resultado PSIM e b) Resultado HIL	39
Figura 29 –	Comportamento de V_C para variação de entrada no conversor <i>Buck</i> em ambas	
	simulações, na qual a) Resultado PSIM e b) Resultado HIL	40
Figura 30 –	Corrente i_L Buck sobreposta para ambas simulações com $d = 0,75$	41
Figura 31 –	Tensão V_C Buck sobreposta para ambas simulações com $d = 0,75$	41

Figura 32 – Resultados para regime permanente do conversor $Buck$ em HIL mantendo d	
fixo em 0,75	42
Figura 33 – Resultados para regime permanente do conversor $Buck$ em HIL mantendo d	
fixo em 0,5	43
Figura 34 – Esquemático da implementação do controle PI no PSIM	44
Figura 35 – Comportamento do conversor <i>Buck</i> em malha fechada perante a uma referên-	
cia variante.	44
Figura 36 – Escala ampliada para controle de corrente no conversor <i>buck</i> com referência	
de 5 Amperes.	45
Figura 37 – Escala ampliada para controle de corrente no conversor <i>buck</i> com referência	
de 8 Amperes.	46
Figura 38 – Comportamento para carga Resistiva de 3 Ohms. Fonte: Do Autor	47
Figura 39 – Esquematico do conversor <i>Boost</i> no PSIM	47
Figura 40 – Comportamento de i_L para variação de entrada no conversor <i>Boost</i> em ambas	
simulações, na qual a) Resultado PSIM e b) Resultado HIL	48
Figura 41 – Comportamento de V_C para variação de entrada no conversor <i>Boost</i> em ambas	
simulações, na qual a) Resultado PSIM e b) Resultado HIL	49
Figura 42 – Corrente i_L Boost sobreposta para ambas simulações com $d = 0,75$	50
Figura 43 – Tensão V_C Boost sobreposta para ambas simulações com $d = 0,75$	50
Figura 44 – Resultados para regime permanente do conversor <i>Boost</i> em HIL mantendo d	
fixo em 0,75	51
Figura 45 – Resultados para regime permanente do conversor <i>Boost</i> em HIL mantendo d	
fixo em 0,25	51
Figura 46 – Esquemático do conversor <i>Buck-Boost</i> no PSIM	52
Figura 47 – Comportamento de i_L para variação de entrada no conversor <i>Buck-Boost</i> em	
ambas simulações, na qual a) Resultado PSIM e b) Resultado HIL	53
Figura 48 – Comportamento de V_C para variação de entrada no conversor <i>Buck-Boost</i> em	
ambas simulações, na qual a) Resultado PSIM e b) Resultado HIL	54
Figura 49 – Corrente i_L do <i>Buck-Boost</i> sobreposta para ambas simulações com $d = 0,75$.	55
Figura 50 – Tensão V_C do <i>Buck-Boost</i> sobreposta para ambas simulações com $d = 0,75$.	55
Figura 51 – Resultados para regime permanente do conversor <i>Buck-Boost</i> mantendo d	
fixo em 0,75	56
Figura 52 – Resultados para regime permanente do conversor <i>Buck-Boost</i> mantendo d	
fixo em 0,25	57
Figura 53 – Inversor Monofásico implementado no PSIM	57
Figura 54 – Resposta do inversor obtida pela simulação em tempo real	58
Figura 55 – Resposta do inversor obtida pela simulação em tempo real com ampliação de	
escala.	59
Figura 56 – Corrente i_{ac} do inversor sobreposta para ambas simulações	59

Figura 57 – Corrente i_{dc} do inversor sobreposta para ambas simulações	60
Figura 58 – F28379D LaunchPad Pin Out e Pin Mux Options - J1, J3	68
Figura 59 – F28379D LaunchPad Pin Out and Pin Mux Options - J4, J2	68
Figura 60 – F28379D LaunchPad Pin Out and Pin Mux Options - J5, J7	69
Figura 61 – F28379D LaunchPad Pin Out and Pin Mux Options - J8, J6	69

Lista de tabelas

Tabela 1	_	- Correspondente na frequência para cada aproximação apresentada (HAUGEN,	
		2010)	16
Tabela 2	_	Paramêtros utilizados no controle do conversor <i>buck</i>	30
Tabela 3	_	Parâmetros utilizados na implementação do conversor Buck	38
Tabela 4	_	Valores quantitativos para regime permanente do conversor <i>Buck.</i>	42
Tabela 5	_	Parâmetros utilizados na implementação do Conversor Boost	48
Tabela 6	_	Valores quantitativos para regime permanente do conversor <i>Boost.</i>	51
Tabela 7	_	Parâmetros utilizados na implementação do Conversor Buck-Boost	52
Tabela 8	_	Valores quantitativos para regime permanente do conversor <i>Buck-Boost</i>	56
Tabela 9	_	Parâmetros utilizados na implementação do inversor monofásico	58
Tabela 4 Tabela 5 Tabela 6 Tabela 7 Tabela 8 Tabela 9		Valores quantitativos para regime permanente do conversor Buck.Parâmetros utilizados na implementação do Conversor Boost.Valores quantitativos para regime permanente do conversor Boost.Valores quantitativos para regime permanente do conversor Boost.Parâmetros utilizados na implementação do Conversor Buck-Boost.Valores quantitativos para regime permanente do conversor Buck-Boost.Parâmetros utilizados na implementação do Conversor Buck-Boost.Valores quantitativos para regime permanente do conversor Buck-Boost.Parâmetros utilizados na implementação do inversor Buck-Boost.Parâmetros utilizados na implementação do inversor monofásico.	4 4 5 5 5 5

Lista de abreviaturas e siglas

HIL	Hardware-in-the-loop.	
PI	Proporcional integral.	
DSP	Processador digital de sinais, do inglês Digital signal processor.	
CC	Corrente contínua.	
CA	Corrente alternada.	
UPS	Fonte de alimentação ininterrupta, do inglês uninterruptible power supply.	
IGBT	Transistor bipolar de porta isolada, do inglês nsulated Gate Bipolar Transis- tor.	
PWM	Modulação por largura de pulso, do inglês Pulse Width Modulation.	
PID	Proporcional integral derivativo.	
ABS	Sistema de travagem antibloqueio, do inglês Anti lock braking system.	
ECU	Unidade de controle eletrônico, do inglês Electronic control unit.	
PV	Painel fotovoltaico, do inglês Photovoltaics.	
PHIL	Power-hardware-in-the-loop.	
RTDS	Real time digital simulator.	
FPGA	Arranjo de Portas Programáveis em Campo, do inglês <i>Field Programmable Gate Array</i> .	
GPIO	Entrada ou saída de uso geral, do inglês General purpose input/output.	
A/D	Analogico / Digital.	
CCS	Code compose studio.	
DAC	Saída digital-analógica, do inglês digital-to-analog converter.	
USB	Porta serial universal, do inglês universal serial bus.	

Lista de símbolos

d	Duty Cycle.
T_S	Período de amostragem ou de chaveamento.
i_L	Corrente que percorre o indutor.
V_0	Tensão de saída.
R	Resistência da carga.
V_S	Tensão de entrada.
Q's	Semicondutores do inversor.
V_{dc}	Tensão DC da fonte de entrada.
i_{ac}	Corrente AC no inversor.
R_L	Resistência indutiva.
t	Tempo.
τ	Constante de tempo (Indutância / Resistência).
i_{acmin}	Corrente AC mínima no inversor.
v_0	Tensão nos semicondutores do inversor.
i_0	Corrente nos semicondutores do inversor.
8	Variável de Laplace que representa frequência contínua.
G(s)	Planta do sistema no domínio da frequência contínua.
C(s)	Controlador em malha fechada no domínio da frequência contínua.
K_P	Ganho Proporcional.
K_I	Ganho Integral.
z	Variável que representa frequência discreta.
f_s	Frequência de amostragem.

Sumário

1	INTRODUÇÃO
- 11	Contextualização e Motivação
12	Objetivo geral
13	Objetivos específicos 3
1.5	Organização do Texto 4
1.1	orgunização do Texto
2	REVISÃO BIBLIOGRÁFICA
2.1	Conversores CC-CC
2.1.1	Conversor Buck
2.1.2	Conversor Boost
2.1.3	Conversor Buck-Boost
2.2	Inversor de Frequência
2.3	Controlador PI
2.4	Métodos de Discretização
2.5	Conceitos e aplicações do Hardware-In-the-Loop
2.6	A arquitetura Hardware-In-the-Loop para simulação em Tempo Real . 18
2.7	Plataformas de simulação em tempo real
2.8	Plataforma escolhida para simulação
3	MODELAGEM E DESENVOLVIMENTO
3.1	Conversor Buck em malha aberta
3.2	Controle PI em malha fechada do Conversor Buck
3.3	Conversor Boost em malha aberta
3.4	Conversor Buck-Boost em malha aberta
3.5	Inversor Monofásico em malha aberta
4	RESULTADOS E DISCUSSÕES 37
4.1	Análise Buck em malha aberta
4.2	Análise do Controlador PI no conversor Buck
4.3	Análise Boost em malha aberta 47
4.4	Análise Buck-Boost em malha aberta
4.5	Simulação do Inversor Monofásico em malha aberta
5	CONCLUSÃO
5.1	Sugestões para trabalhos futuros

	Referências	63
	Anexos	66
A.1	ANEXO A – ENDEREÇO VIA GITHUB	67 67
	ANEXO B – DEVICE PIN OUT LAUNCHXL-F28379D	68

1 Introdução

Este capítulo tem como objetivo realizar uma apresentação dos conceitos iniciais referentes a simulação em tempo real de circuitos eletrônicos de potência, abordando uma breve contextualização histórica atrelado as questões que levaram a motivação de sua escolha para a pesquisa desta monografia. Por fim são citados os objetivos gerais e específicos que definem os tópicos presentes no trabalho.

1.1 Contextualização e Motivação

A constante evolução da tecnologia no âmbito digital vem se tornando um importante assunto abordado em estudos de diferentes áreas, como a aeronáutica, a automotiva, a engenharia, entre outras. Devido a esse crescimento, empresas envolvidas no ramo digital tendem a focar suas atenções na busca por novos métodos para a produção e validação de seus sistemas de forma rápida e segura, envolvendo o menor custo possível.

Uma das áreas que utiliza desses ambientes digitais e encontra-se presente na engenharia elétrica é a eletrônica de potência, um campo constituído de três pilares: potência, eletrônica e controle (AHMED, 2008). Grande parte das suas aplicações estão voltadas ao projeto de circuitos conversores de energia, que converte a forma de potência CC em CA ou vice-versa entre fonte e carga (HART, 2011).

A parte de projeto desses circuitos deve envolver uma série de etapas, de forma organizada e segura, especialmente quando a produção está alinhada a uma larga escala. Tais etapas são adotadas a fim de dividir o processo completo em partes, sendo a primeira relacionada a definição dos valores dos componentes que compõem o sistema de forma teórica.

A segunda etapa geralmente está relacionada a implementação do circuito em *software* computacional (como PSIM e MATLAB), que são capazes de predizer o comportamento do sistema de forma ideal sem precisar de uma montagem física (HOSSEINPOUR und HAJIHOS-SEINI, 2009). Por fim, a última engloba a montagem final do sistema em *hardware*, na qual um conjunto de testes devem ser feitos com o protótipo para a validação final.

A etapa de simulação envolve os chamados testes de *software*, permitindo a implementação do modelo representativo do sistema de forma computacional, para uma futura análise dos dados como uma prévia dos resultados do sistema real. Essa representação em *software* possibilita a implementação de novas ideias, e ajuda a prevenir possíveis problemas práticos, sem precisar lidar com o sistema físico completo.

Entretanto, a simulação *off-line* não consegue realizar a iteração entre o modelo virtual e o mundo real, não sendo possível replicar de forma exata os equipamentos que constituem

o sistema, uma vez que os *softwares* de simulação utilizam sinais virtuais e elementos ideais. Sensores elétricos, por exemplo, são interessantes de se testar com o protótipo físico, uma vez que se deseja saber o comportamento prático do mesmo e não teórico, implicando que os projetos necessitem de uma implementação do protótipo em *hardware* para validação dos componentes de forma correta (RODRIGUES und OTHERS, 2018).

Uma das técnicas mais utilizadas atualmente por pesquisadores e engenheiros para contornar o problema dessa situação é denominada *Hardware-In-the-Loop* (HIL). O objetivo principal dessa técnica é realizar os testes de validação em *hardware* por meio de uma simulação em tempo real antes da montagem final do sistema, implicando em vários benefícios para a fase de testes.

No HIL um ambiente digital é criado com a possibilidade de inserção de componentes reais ao modelo virtual matemático do sistema. Essa possibilidade permite uma maior confiabilidade na simulação, onde os testes de *software* porem ocorrer inicialmente com a simulação em tempo real e à medida que os componentes vão sendo estabelecidos, realiza-se a inserção destes com o HIL para os testes de *hardware*.

Outra vantagem dessa técnica é a possibilidade de realizar testes que são impraticáveis com protótipos na vida real, como por exemplo, um curto-circuito em um equipamento de alta potência, que poderia prejudicar todo um sistema (BASTOS u. a., 2019). Montar um circuito físico apenas para testes também pode levar muito tempo e envolver um custo alto, além de que em casos de possíveis erros, todos os elementos poderão ser danificados.

A metodologia HIL apresenta duas configurações, sendo a primeira em malha aberta e a segunda em malha fechada, definido pela conexão do elemento externo, caso esse exista. A Figura 1 representa o esquemático das implementações.



a) Simulação em malha aberta

Figura 1 – Principais configurações de arquitetura para o Hardware-in-The-Loop.

Fonte: do autor.

b) Simulação em malha fechada

A configuração em malha aberta emula o modelo da planta simulado em tempo real com conexão a um elemento externo que não depende dos resultados simulados. Já a configuração em malha fechada refere ao modelo simulado comunicando de forma bidirecional a um elemento externo, como um controlador, sendo possível validar esse com base no funcionamento da planta (ANAKWA u. a., 2002).

Apesar dos grandes benefícios citados, há fatores que impedem a ampla utilização do HIL em ambientes simples, sendo o mais impactante relacionado ao custo elevado de simuladores digitais, que já apresentam estudos em alternativas mais baratas. Processadores em *Pentium* já são utilizados em algumas aplicações, possibilitando uma compra de um sistema completo envolvendo computador, softwares e dispositivos na faixa de \$4.000, um valor ainda alto de compra dependendo da aplicação (SILVA, 2008).

Dessa forma, se torna interessante encontrar uma plataforma de baixo custo capaz de realizar a simulação em tempo real para topologias comuns da eletrônica de potência de baixa complexidade. Este trabalho terá como foco a implementação do modelo proposto de cada topologia escolhida em um *hardware* microcontrolador de alto custo benefício, que funcionará como simulador digital em tempo real, onde seu comportamento será testado na configuração malha aberta e fechada.

Em malha fechada um controlador do tipo PI será projetado e implementado em outro microcontrolador, para se testar a iteração do controle digital com a topologia *Buck* simulada em tempo real. Os outros circuitos serão implementados apenas em malha aberta, sendo esses *Buck, Boost, Buck-Boost* e o inversor monofásico, que são encontrados em aplicações como instalações fotovoltaicas, fontes de alimentação ininterruptas (UPS) e fontes chaveadas.

O resultado que se espera é obter respostas experimentais do HIL que sejam semelhantes as esperadas de um circuito prático, e para isso o circuito passará por uma simulação *off-line* para se ter uma base de comparação entre os resultados, se tornando desnecessário a implementação em protótipo para validação da simulação no micro. Dessa forma, se o resultado for o esperado, qualquer circuito eletrônico simples poderá ser simulado em tempo real pelo microcontrolador escolhido.

1.2 Objetivo geral

O objetivo geral do trabalho é validar a simulação em tempo real dos sistemas *Buck, Boost, Buck-Boost* e inversor monofásico através de um microcontrolador de baixo custo, realizando as comparações necessárias entre resultado teórico e experimental.

1.3 Objetivos específicos

Os objetivos específicos da monografia são:

- Estudar e compreender arquitetura da simulação em tempo real aplicada à conversores eletrônicos de potência;
- Modelar os circuitos listados através de equações diferenciais e posteriormente digitaliza-las para implementação em forma de algoritmo no simulador.
- Implementar as equações que modelam os circuitos no microcontrolador que realizará a função de simulador digital em tempo real;
- Avaliar as respostas obtidas pelo simulador digital comparando-as com a simulação ideal realizada em software computacional comum, a fim de validar as equações modeladas e o uso do DSP para tais aplicações.
- Projetar e implementar um controlador do tipo PI em um *hardware* externo apropriado para o controle de corrente do conversor *Buck*, realizando a comunicação entre planta e controlador de forma real-virtual.

1.4 Organização do Texto

Este trabalho está organizado da seguinte forma:

- Capítulo 2: Apresenta a revisão bibliográfica, explicando os conceitos que abrangem o tema de acordo com as referências da literatura.
- Capítulo 3: Apresenta a modelagem e a implementação dos modelos propostos para a simulação em tempo real de malha aberta e fechada, detalhando também o projeto do controlador PI para o conversor *Buck*.
- Capitulo 4: Apresenta os resultados para a simulação em tempo real dos circuitos no microcontrolador junto a comparação com a resposta do *software* PSIM.

2 Revisão Bibliográfica

Este capítulo apresenta discussões necessárias a respeito da arquitetura HIL tendo como base trabalhos publicados por diferentes autores. Os circuitos definidos também serão abordados de forma teórica, contendo imagens e equações importantes para esse trabalho.

2.1 Conversores CC-CC

Desde os anos 60, várias topologias de conversores chaveados e lineares com o objetivo de aumentar ou diminuir um nível de tensão ou corrente foram propostas. Cada arranjo de configuração apresenta sua estrutura, seus méritos e suas limitações, e a escolha de uso deve depender da aplicação de atuação do mesmo (VIDOR, 2017).

O funcionamento de um conversor CC-CC tem como base a utilização de elementos armazenadores de energia, tais como indutores e capacitores, em que ambos têm a função de transferir a potência de entrada para a saída de acordo com a posição de uma chave localizada no começo do circuito. Essa chave na prática é substituída por um elemento semicondutor capaz de realizar a mesma função, acionando uma parte do sistema em uma alta frequência (HART, 2011).

Para a comutação, um sinal externo é gerado de forma a acionar o elemento semicondutor trabalhando nas áreas de corte e condução. Quando o sinal estiver em nível alto, a chave se fecha (depende da polaridade) e a transferência de potência acontece da fonte para os armazenadores, já em nível baixo a transferência acontece dos elementos para a carga, fornecendo uma saída de tensão dependente do capacitor (AHMED, 2008).

O ajuste que controla o valor da tensão de saída é chamado *Duty Cycle*, que representa a fração entre o tempo que a chave será ligada e o período completo de chaveamento. Nessa monografia, os elementos semicondutores que constituem os conversores escolhidos serão considerados ideais, sendo sua potência absorvida igual a zero, constituindo uma transferência completa de potência entre fonte e carga.

Para entender melhor como se relaciona o *Duty Cycle*, a referência (GONZADA, 2018) deve ser consultada, a fim de explicar como funciona o sinal PWM (*Pulse Width Modulation*) aplicado aos elementos semicondutores dos circuitos implementados.

2.1.1 Conversor Buck

O conversor *Buck* é um conversor do tipo CC-CC, cuja função é disponibilizar um valor de tensão menor ou igual ao fornecido em sua entrada (HART, 2011). A Figura 2 apresenta a topologia *Buck*, que conta com dois armazenadores de energia na configuração ideal.







Para analisar o funcionamento do sistema, antes é necessário supor que algumas condições sejam satisfeitas, sendo elas (HART, 2011):

- O circuito deve funcionar de forma estável.
- A corrente no indutor não pode atingir valores iguais ou menores que zero, estando em um estado chamado modo de condução contínua.
- Os componentes devem ser ideais.
- O valor do capacitor precisa ser alto suficiente para manter a tensão de saída constante.

O princípio de funcionamento do sistema consiste na avaliação do *status* da chave semicondutora. No primeiro modo de operação a chave pode ser considerada fechada em um período definido de $d \cdot T$ (a expressão representa o valor de *Duty Cycle* x Período de comutação), implicando em um fluxo de potência da fonte de alimentação para os armazenadores, aumentando a corrente do indutor e do capacitor. Quando a chave é aberta por um período (1 - d)T, o fluxo começa a ocorrer dos armazenadores para a carga, fazendo com que a corrente que passa pelo indutor diminua.

O comportamento de corrente no indutor é ilustrado na Figura 3. De acordo com o acionamento da chave é possível notar um formato triangular de onda, na qual a corrente cresce até um certo valor e decresce pela mudança de estado.



Figura 3 – Comportamento ideal da corrente i_L no conversor *Buck*.

Fonte: (HART, 2011).

O valor médio de i_L depende diretamente da carga conectada, podendo ser calculada de acordo com:

$$i_L = \frac{V_S}{R} \tag{2.1}$$

A tensão de saída V_S apresenta uma forma pulsante, sendo o valor médio menor ou igual a de V_0 na entrada, como mostrado na equação (2.2).

$$V_0 = V_S \cdot d \tag{2.2}$$

2.1.2 Conversor Boost

O conversor *Boost* possui uma estrutura similar ao do conversor *Buck*, porém com o objetivo de fornecer na saída um valor de tensão maior ou igual ao fornecido em sua entrada (HART, 2011). As mesmas suposições devem ser feitas para se analisar o circuito.

A Figura 4 apresenta a topologia ideal do conversor *Boost*, também composto por dois armazenadores de energia.

Figura 4 – Conversor Boost ideal.



Fonte: (HART, 2011).

A analise deve ser feita para os dois estados da chave, agindo de forma idêntica a topologia *Buck*, como pode ser observado na Figura 5.

Figura 5 – Comportamento da corrente i_L no conversor *Boost*.



Fonte: (HART, 2011).

Com a chave fechada, a corrente no indutor cresce até um determinado nível máximo em regime permanente, limitado pelo valor do indutor e pela frequência. Já com a chave aberta, o valor de corrente começa a decair, atingindo seu valor mínimo, onde esse comportamento se torna periódico.

O valor médio da tensão de saída pode ser relacionado com o valor de entrada seguindo a equação a seguir.

$$V_0 = \frac{V_S}{(1-d)}$$
(2.3)

2.1.3 Conversor Buck-Boost

O conversor *Buck-Boost* engloba as duas topologias apresentadas num mesmo circuito, capaz de fornecer valores de tensão maiores ou menores que ao valor de tensão aplicado na entrada. A Figura 6 apresenta a topologia desse conversor, que apesar de englobar as outras duas topologias, permite sua configuração com apenas dois armazenadores de energia.

Figura 6 – Conversor Buck-Boost ideal.



Fonte: (HART, 2011).

Assim como os outros conversores, seu comportamento segue um padrão similar e deve ser considerado as condições iniciais descritas anteriormente para a análise de seu funcionamento, que pode ser ilustrado pela Figura 7.

Figura 7 – Comportamento da corrente i_L no conversor *Buck-Boost*.



Fonte: (HART, 2011).

Quando a chave é fechada, o diodo se polariza reversamente, impedindo que a corrente de entrada percorra a carga, fazendo com que a corrente do indutor cresça até o valor máximo. Quando a chave é aberta, o diodo é polarizado diretamente, implicando que a corrente do indutor seja descarregada na carga. Dessa forma, a corrente i_L assume uma forma triangular com um período igual ao de chaveamento.

A tensão de saída é encontrada pela relação da equação (2.4), onde o valor de *d* define se a tensão de saída será maior, igual ou menor do que a de entrada.

$$V_0 = -V_S \cdot \frac{d}{(1-d)}$$
(2.4)

A expressão indica que o sinal de tensão da saída irá apresentar uma polaridade oposta à aplica na entrada.

2.2 Inversor de Frequência

O inversor de frequência monofásico é um circuito largamente utilizado na eletrônica de potência, que se deve a diversos fatores como a expansão no uso de energia fotovoltaica. O objetivo do circuito inversor é fornecer uma potência CA a partir de uma alimentação CC, realizando a conversão do tipo CC-CA.

Esses dispositivos são empregados em diversas aplicações na eletrônica de potência, tais como em unidades de motor de velocidade ajustável, fontes de alimentação ininterrupta (UPS), e execução de aparelhos CA de baterias automotivas (HART, 2011).

A Figura 8 apresenta a topologia para o inversor monofásico ideal, chamada ponte completa, na qual as chaves *Q* representam cada elemento semicondutor de entrada (geralmente IGBT's) responsável pelo acionamento do circuito de acordo com o sinal aplicado.

Figura 8 – Inversor Monofásico ideal.



Fonte: (HART, 2011).

O princípio de funcionamento depende do sinal aplicado ao conjunto de chaves. As chaves Q1 e Q4, assim como Q3 e Q2 não podem ser fechadas ao mesmo tempo, implicando que o sistema de controle para acionamento necessite de uma metodologia impedindo que esse

caso ocorra. No momento inicial, as chaves Q1 e Q2 são fechadas, fornecendo o valor de entrada $+V_{dc}$ para a carga.

Depois de um tempo determinado pela frequência de chaveamento, essas chaves se abrem, e as outras (Q3 e Q4) se fecham fornecendo uma tensão $-V_{dc}$ a carga. No período completo, a tensão e corrente na carga seguirão o formato de onda quadrada, já suficiente para algumas cargas CA.

Uma maneira de tornar a corrente da carga próxima a característica senoidal é acrescentar um indutor em série com a carga, atuando como um filtro. Para o período em que a onda de tensão é positiva, a corrente se comporta de forma crescente, de acordo com (2.5).

$$i_{ac}(t) = \frac{V_{dc}}{R_L} + (i_{acmin} - \frac{V_{dc}}{R_L})e^{-t/\tau}$$
(2.5)

Já para o período em que a tensão é negativa, a corrente do indutor decresce, representado na equação:

$$i_{ac}(t) = -\frac{V_{dc}}{R_L} + (i_{acm\acute{a}x} - \frac{V_{dc}}{R_L}) - e^{(t - \frac{T}{2})/\tau}$$
(2.6)

Essas equações ditam o comportamento da saída do inversor com o filtro R_L . A Figura 9 apresenta o resultado da corrente da tensão de saída com o filtro RL de saída, onde v_0 e i_0 representam a tensão e corrente na carga.

Figura 9 – Comportamento do inversor monofásico para corrente e tensão de saída.



Esse formato ainda não é o suficiente para aplicações que necessitam de uma característica alternada, como motores síncronos e a conexão com a rede. Diante disso, uma metodologia diferente pode ser utilizada para obter uma saída próxima ao requerido pela carga. A modulação por largura de pulso, do inglês *Pulse width modulation* (PWM) é capaz de controlar a tensão ou corrente de saída através da comparação entre onda senoidal e onda portadora de alta frequência, calculando o *Duty Cycle* correspondente para obter o formato desejado.

A frequência da onda portadora define a frequência de chaveamento (comutação) e a onda senoidal de referência modela a largura do pulso. O sinal PWM é alimentado ao conjunto de semicondutores assumindo valor 0 para chave desligada e 1 para ligada. A Figura 10 mostra o comportamento da modulação PWM de acordo com uma referência senoidal.





Fonte: (HART, 2011).

A onda portadora comumente é do tipo triangular simétrica ou dente de serra. Enquanto a referência senoidal for maior que a portadora triangular, o resultado de d é saturado em um nível positivo, no momento que a referência passa a ser menor, o resultado de d pode ser saturado em negativo ou em zero.

2.3 Controlador PI

Os controladores são projetados para realizar a função de controle em uma ou mais variáveis de um sistema, estabelecendo uma relação direta de comparação entre a saída a ser controlada e a referência ajustada (OGATA, 1998).

No mercado existem diversos tipos de controladores, sendo os mais comuns do tipo compensador por atraso de fase (DE PAULA und GUARACY, 2016), compensador por avanço de fase (YAMANAKA, 2021) e controladores PID (Proporcional integral derivativo) ou PI, onde cada um deles apresentando suas características e limitações.

Em relação aos outros controladores, os do tipo PID são mais utilizados na indústria, sendo possível utilizar sua derivação PI para foco desse trabalho, devido a sua estrutura simples

e reduzida e a sua baixa complexidade, ideal para implementação em equipamentos digitais simples (DE SOUZA BAETA u. a., 2015).

A Figura 11 ilustra a configuração em malha fechada onde o controlador PI representado por C(s) age sobre a planta G(s), ambas modeladas no domínio da frequência.



Figura 11 – Esquema de malha fechada do controlador PI.

O controlador recebe como parâmetro de entrada um sinal de erro, gerado a partir da diferença entre a saída a ser controlada e o valor de referência definido. Dessa forma, o objetivo é forçar essa diferença a atingir um valor próximo a zero em um tempo considerável, onde ambos os fatores dependerão dos ganhos projetados.

O sistema PI resulta da combinação dos fatores proporcional (P) e integral (I) em um único bloco de compensação, onde cada um tem uma determinada ação na planta. A ação proporcional multiplica o sinal de erro (diferença entre valor de referência e saída) por uma constante K_P , de forma que o resultado desse produto seja usado como sinal de correção (PINHEIRO, 2009).

A ação integral (K_I) produz uma saída proporcional ao erro acumulado, com objetivo de forçar a saída seguir um padrão igual a referência, zerando o erro em regime permanente, em contrapartida essa ação pode produzir uma oscilação na resposta por conta do aumento de ordem que ela influencia no sistema, como pode ser visto na equação (2.7), que relaciona os ganhos com a função em *s* para o controlador *Cs*.

$$Cs(s) = K_P + \frac{K_I}{s} \tag{2.7}$$

Os ganhos que representam as ações do PI podem ser vistos de forma representativa na Figura 12.

Fonte: (CASTRO, 2020).



Figura 12 – Representação dos ganhos no controlador PI em malha fechada.

Adaptado de: (CASTRO, 2020).

Existem diversas maneiras para obtenção dos ganhos K_P e K_I , porém como o projeto do controlador em si não é o foco do trabalho, esses ganhos serão encontrados a partir de métodos de sintonia computacional. Outros métodos via cálculo podem ser encontrados em (DORF und BISHOP, 2001).

O *software* utilizado será o *MATLAB*, onde é possível obter os ganhos através da ferramenta *SISOTOOL* com o método *PI tuning*, fornecendo a função de transferência do sistema e parâmetros como frequência de corte e margem de fase. O sistema escolhido para implementação teste do controle junto a simulação em tempo real é o conversor *Buck*.

2.4 Métodos de Discretização

As equações de modelo dos sistemas propostos serão obtidas na forma contínua inicialmente, implicando que qualquer operação feita através delas deverá estar no domínio contínuo. Entretendo, equações nesse domínio requerem uma velocidade impossível de atingir, visto que o tempo entre os dados são infinitesimais, tornando necessário o uso de uma transformação de domínio para a implementação na forma digital.

Um modelo em tempo contínuo é capaz de descrever a dinâmica do sistema em qualquer instante de tempo, já um modelo em tempo discreto descreve a dinâmica do sistema apenas em instantes de tempo pré-estabelecidos, e igualmente espaçados no tempo, com período Ts (ECKHARD und CAMPESTRINI, 2016).

Para a discretização de operações matemáticas como integral e derivada, existem métodos de aproximações capazes de mapear o plano contínuo s de Laplace para o plano z de frequência discreta. Os principais métodos encontrados nas literaturas de processamento de sinais são: *Forward, Backward difference* e Tustin.

Cada aproximação apresenta sua particularidade de implementação, como será descrito posteriormente. A Figura 13 apresenta o gráfico de uma função f(t) contínua no tempo, onde os dados são amostrados pela aproximação de *Backward* e *Forward*.



Figura 13 - Aproximações Backward e Forward para uma função.

Fonte: (HAUGEN, 2010).

A amostra atual é representada por x(k), a anterior por x(k-1) e a posterior por x(k+1), todas elas espaçadas por um período de amostragem T_S . A aproximação de *Forward Difference* pode ser deduzida através da propriedade de derivação, utilizando a amostra futura, representada como:

$$\frac{dx(t)}{dt} = \frac{x(k+1) - x(k)}{Ts}$$
(2.8)

Desenvolvendo a integral da área da figura pela definição onde $\frac{d}{dt} \int e(t)dt = e(t)$, encontra-se a equação de aproximação disponibilizada na tabela 1. Esse método apresenta a vantagem de fácil implementação, porém pode levar uma planta estável em domínio continuo para uma planta instável no domínio discreto (SONG und LU, 2014).

O segundo método é o *Backward difference*, que utiliza do mesmo conceito do *Forward*, porém trabalha com operações nas amostras de instantes passados, como pode ser visto na equação (2.9), representando a taxa de variação na função f(t).

$$\frac{dx(t)}{dt} = \frac{x(k) - x(k-1)}{Ts}$$
(2.9)

Através da propriedade da integral junto a transformada *z*, encontra-se a equação equivalente presente na tabela de discretização. Esse método é muito utilizado por ser simples e trabalhar com dados passados, facilitando na estabilidade da transformação, apesar da leve limitação sobre o desempenho dinâmico do sistema de controle (CIOBOTARU und TEODORESCU, 2006).

Por fim, o método de Tustin é baseado numa aproximação, onde a integral é interpretada como a área entre o a função e o eixo do tempo, e esta área é aproximada por pequenos trapézios (HAUGEN, 2010). Sua implementação é mais complexa computacionalmente que as outras e

pode apresentar erros quando o sistema a ser discretizado possui frequências de ressonância, porém o método obtém uma ótima representação tanto em sua dinâmica quanto na estabilidade do sistema discretizado(CIOBOTARU und TEODORESCU, 2006).

A Tabela 1 mostra a relação entre o plano s e o plano z para cada método.

Tabela 1 – Correspondente na frequência para cada aproximação apresentada (HAUGEN, 2010).

Método de Discretização	Equivalência em z
Forward Euler	$s = \frac{z-1}{Ts}$
Backward Euler	$s = \frac{z-1}{zTs}$
Tustin	$s = \frac{2(z-1)}{Ts(z+1)}$

A aproximação escolhida para uso nesta monografia é a de *backward*, comumente usada em filtros digitais e em controladores industriais, cuja complexidade é compatível com o processador usado para a simulação em tempo real (HAUGEN, 2010). Dessa forma os modelos dos conversores e do controlador podem ser escritos no domínio contínuo de Laplace, e serem discretizado com a aproximação de *backward* para o domínio z, apresentando um espaçamento finito T_S .

2.5 Conceitos e aplicações do Hardware-In-the-Loop

A técnica HIL (*Hardware-In-the-loop*) refere-se a implementação de um ambiente de simulação digital em tempo real, cujo objetivo é realizar testes em *hardware* sem o protótipo do sistema físico. Nessa técnica, alguns dos elementos podem ser reais e não simulados, tornando a técnica mais segura e de baixo custo em determinados casos (SILVA, 2008).

Uma possibilidade de redução do tempo para o desenvolvimento de um projeto é acrescentar essa técnica a etapa dois de projeto (simulação *off-line* comum), sendo possível complementar os resultados obtidos por *software*. Assim, a simulação em tempo real de malha aberta pode ser feita simulando o modelo virtual de todos os componentes e, à medida que cada um vai sendo definido, pode-se substituir seus respectivos modelos matemáticos por elementos físicos.

O conceito de *Hardware-In-the-Loop* não é novo, e de acordo com (ISERMANN und OTHERS, 1999) provavelmente teve seu primeiro uso na área da aeronáutica, onde as primeiras simulações eram usadas para testes de voo em tempo real. Os instrumentos presentes na estrutura do avião eram testados de acordo com a comunicação entre a cabine de pilotagem (*cockpit*) e um simulador, onde essa cabine recebia os sinais elétricos e/ou hidráulicos do simulador através de atuadores. Os atuadores, por sua vez, permitiam a movimentação do *cockpit* de acordo com os comandos realizados pelo comandante, de forma que os pilotos poderiam ser treinados da maneira mais segura e com condições próximas ao mundo real.

As primeiras gerações usavam controladores de tubo analógicos para as simulações de movimento, que posteriormente foram substituídas por computadores analógicos e de processamento. Com o passar do tempo, os processadores foram se tornando mais eficientes, e junto a questão da grande demanda de produtos em diversas áreas, o conceito de HIL se expandiu, como por exemplo para aplicações no uso automotivo, na qual o principal motivo da utilização é devido à dificuldade de se modelar alguns componentes, como molas e amortecedores.

Através da simulação por *hardware*, é possível simular movimentos de suspensão, excitação de rodas e outros testes dinâmicos para carros, que permitem não só o treinamento de pilotos como o teste de elementos presentes nos veículos (DROSDOL und PANIK, 1985).

Na literatura, pode-se encontrar diferentes estudos em diversas áreas onde a tecnologia HIL é explorada.

- (HANSELMANN, 1993) abordam em seu artigo a simulação de freios ABS (Sistema de freios antitravamento, em inglês *Anti lock Braking System*), um sistema que proporciona benefícios como a não derrapagem do veículo e, consequentemente, o aumento da estabilidade em condições de frenagens de emergência, que foi implementado na Alemanha, onde os testes foram realizados em condições de inverno e verão, no laboratório, sem o carro físico. O sistema ABS foi simulado em um ECU (*Electronic Control Unit* ou unidade de controle eletrônico) onde os componentes hidráulicos eram reais ;
- (SCHODER und OTHERS, 2013) desenvolvem discutem o potencial que o teste de *Hardware-In-the-Loop* pode desempenhar tanto em testes reais quanto em experimentos de laboratório, e mostra os resultados obtidos durante testes em escala de Megawatts de uma topologia de conversor CA-CC;
- Em (TERLIP und OTHERS, 2012), uma metodologia é proposta e executada para caracterizar e modelar inversores para estudos de integração de rede usando Power *Hardware-Inthe-Loop*;
- Em (JUNG, 2015) é apresentado uma dinâmica eletrotérmica para uma célula fotovoltaica (PV) que combina parâmetros elétricos e térmicos para emular PV com precisão para simulação em tempo real do tipo *Power-Hardware-In-The-Loop* (PHIL)

Nota-se que apesar das desvantagens citadas anteriormente como o alto custo das plataformas existentes no mercado e a dificuldade em ajustar algumas configurações internas dos simuladores, essa técnica é usada amplamente em diversas áreas e até mesmo em ambientes de pesquisas, contribuindo para a evolução dos equipamentos industriais que utilizam de ambientes digitais.

2.6 A arquitetura Hardware-In-the-Loop para simulação em Tempo Real

A arquitetura de uma simulação HIL pode apresentar diferentes configurações, dependendo do propósito da implementação. A Figura 14 apresenta a arquitetura tradicional de um sistema HIL, onde é utilizado a configuração malha fechada ilustrando um controlador digital controlando um processo implementado em um simulador digital.



Figura 14 – Configuração padrão do HIL em malha fechada.

Adaptado de: (KARPENKO und OTHERS, 2009).

A configuração tradicional é composta por elementos reais (sistema embarcado, sensores e atuadores) e pela planta implementada através da simulação em tempo real (KARPENKO und OTHERS, 2009). O sistema embarcado ilustrado representa, geralmente um controlador, que recebe e envia sinais do processo através de sensores e atuadores. Os sensores, por sua vez, realizam a medição da variável de interesse e os atuadores atuam sobre a planta de acordo com o comando recebido.

Essa configuração depende das características do sistema que se deseja simular. Em alguns casos por exemplo, os elementos reais realizam apenas a comunicação unidirecional, enviando sinais ao processo sem precisar ler sua resposta, por exemplo.

O sistema em tempo real é implementado no *Target*, um dispositivo onde será executado o software da planta gerado pelo computador host, que é o elemento responsável por processar um conjunto de equações que modelam o processo matematicamente.

Segundo (LAPLANTE und OVASKA, 2012), há três conceitos para definir sistemas de tempo real. O primeiro diz que um sistema de tempo real é um sistema computacional que deve satisfazer à limitação dada pela restrição do tempo de resposta, e o segundo conceito diz que um sistema de tempo real é tal que a correção lógica se baseia tanto na exatidão dos resultados quanto na sua acurácia temporal. (KOPETZ, 2011), define que um sistema de tempo real é definido como um sistema computacional cujo correto funcionamento da estrutura não depende apenas dos

resultados lógicos da computação realizada, mas também do tempo físico que esses resultados são gerados de acordo com os eventos de entrada.

Considerando o exemplo de arquitetura apresentado, o computador *Host* será o elemento utilizado para programar o *target*, que irá simular o processo. A função do *Host* é realizar uma interface entre programador e equipamento, através de softwares conhecidos no mercado, como por exemplo o MATLAB. Esses aplicativos permitem que os algoritmos sejam desenvolvidos a partir da utilização de blocos representando o modelo do sistema com facilidade, diferente de se programar um *target* com linguagem de baixo nível, que apesar de ser possível ajustar registradores importantes, necessita de um estudo prévio da arquitetura.

2.7 Plataformas de simulação em tempo real

Há diversas plataformas disponíveis no mercado para realização da simulação em tempo real em ambiente digital. A Figura 15 mostra duas das mais famosas utilizadas em aplicações especificas.

Figura 15 – Simuladores mais utilizados presentes no mercado, sendo: a) RTDS e b) Typhoon HIL.



Fonte: (RTDS, 2017) e (TYPHOON HIL, 2018).

Cada uma delas apresenta sua particularidade em estrutura e capacidade de processamento. O hardware apresentado na Figura 15 (a) é do fabricante RTDS (*Real Time Digital Simulator*), desenvolvida inicialmente por volta de 1993, que conta com vários modelos de diferentes características. Sua estrutura é formada de forma escalonável por meio de armários (*racks*), possibilitando que o sistema possa ser simulado por partes independentes (HERNÁNDEZ und CANESIN, 2015). Cada chassi do modelo NovaCor montado em *racks* apresenta um processador POWER8TM de última geração da IBM, contendo 10 núcleos rodando a 3,5 GHz. Este poderoso processador *multicore* torna esse modelo mais rápido e capaz do que o *hardware* de processamento de versões anteriores da marca (RTDS, 2017).

A plataforma da direita apresentada na Figura 15(b) representa o Typhoon HIL, desenvolvido inicialmente em 2008 e líder de mercado em soluções de *Hardware-In-the-Loop* para projeto, teste e validação de microrede e sistemas de controle de eletrônica de potência. Apesar de suas vantagens, o equipamento ainda apresenta um custo alto, aproximadamente \$9.00 por 4 *racks* (TYPHOON HIL, 2018).

Cada uma dessas plataformas é indicada para uma aplicação especifica, o NovaCor por exemplo é mais indicado a aplicações eletrônicas de potência em geral. Ambas apresentam uma estrutura escalável, mas cada *rack* individualmente do Typhoon apresenta maior desempenho, porém o RTDS possui a capacidade de alocação de mais desses.

Como já mencionado, plataformas de simulação em tempo real já são comuns no mercado da eletrônica de potência, porém, apesar de suas vantagens de aplicação, o custo ainda é um fator considerável. Para mitigar esse problema, existem alguns trabalhos acadêmicos que abordam o desenvolvimento de uma plataforma com custo menor, como a apresentada em (MINA u. a., 2016), que utiliza de FPGAs (Arranjo de Portas Programáveis em Campo, em inglês *Field Programmable Gate Array*) para a simulação.

O objetivo da monografia será utilizar um microcontrolador LAUNCHXL-F28379D para simulação de sistemas eletrônicos de potência que trabalham em uma faixa considerável de frequência, tendo um custo menor e uma função semelhante aos simuladores apresentados, onde seja possível alterar parâmetros de entrada e obter resultados na saída em um baixo período de tempo. Os critérios de sua escolha serão descritos a seguir.

2.8 Plataforma escolhida para simulação

Desejando utilizar uma plataforma de baixo custo para a simulação em tempo real, primeiro é preciso analisar alguns conceitos dos modelos escolhidos para que a plataforma sirva para a aplicação desejada. A frequência de simulação, dada por $f_{sim} = 1/T_{Step}$, onde T_{Step} é o período entre as amostras, deve ser consideravelmente maior que a frequência de entrada, seja ela analógica ou digital (BASTOS u. a., 2019). Caso seja analógica, o sistema deve obedecer ao conceito do teorema da amostragem.

Esse conceito é proveniente de Nyquist-Shannon, cuja definição diz que um sinal analógico limitado em banda pode ser representado corretamente de forma discreta através de uma amostragem, onde a frequência f_s seja duas vezes maior que a frequência presente no sinal a ser discretizado (OPPENHEIM, 2010). Caso o sinal de entrada seja um PWM digital com frequência fixa, a frequência de simulação deve ter um valor de no mínimo 100 vezes maior que a entrada (BASTOS u. a., 2019), para que a representação seja a mais fiel possível, sendo possível detectar uma alteração na entrada de forma rápida. Por exemplo, se houver uma entrada PWM no simulador com uma frequência de 10 kHz, a simulação deverá ocorrer em no mínimo 1 MHz.

O tempo de simulação depende da complexidade do código implementado, implicando em uma possibilidade de frequência maior caso o circuito seja mais simples. O tempo de execução do algoritmo deve ser de, no máximo, 70% do período de amostragem para manter uma margem de segurança.

A facilidade de programação também influencia na escolha, pois muitas vezes um processador exige uma linguagem de baixo nível para suas configurações, tornando o trabalho de implementação mais complexo e demorado. Assim, a plataforma proposta para a simulação em tempo real dos circuitos dessa monografia, dispõe de um Microcontrolador LAUNCHXL-F28379D, mostrado na Figura 16 e seu software gratuito para programação.



Figura 16 – Microcontrolador f28379d da texas instruments.

Fonte: (INSTRUMENTS, 2017).

Sua compra está disponível por um valor aproximado de \$33.79 (INSTRUMENTS, 2017), um custo bem inferior comparado as plataformas apresentadas, sendo possível assim realizar as implementações sem dificuldades em adquirir o produto.

Esse Microcontrolador contém uma unidade de microcontrolador de ponto flutuante de 32 bits, projetada para aplicações de controle de malha fechada, como acionamentos de motores industriais, inversores solares, detecção e processamento de sinal. Além disso, essa unidade conta com um processamento Dual-Core, que fornecem 200 MHz de desempenho para processamento de sinal em cada núcleo, permitindo a implementação de equações independentes em núcleos diferentes caso necessário.

Com 4 conectores de 20 pinos representando GPIO's (*General Purpose Input/Output*), o DSP possui 16 entradas ADC (analógicas convertidas para digital) de 16/12 bits e duas saídas DAC (digital para analógica) de 12 bits, que são disponibilizadas de acordo com o datasheet.
Além de ter seu custo e tamanho reduzidos (145,9 gramas), o simulador da família Delphino utiliza o software CCS (*Code Compose Studio*) para a programação em linguagem C padrão, de forma gratuita e complexidade baixa ao computador *Host*.

O CCS possui também uma gama de exemplos disponíveis que trabalham a configuração dos registradores para PWM, assim dependendo da aplicação é possível apenas adaptar um algoritmo já pronto, sem se preocupar tanto com configurações internas do DSP, contando também com a opção debug caso um erro ocorra.

Dessa forma, esse microcontrolador é a escolha ideal de baixo custo para a aplicação descrita nesse trabalho, pois os modelos para as topologias podem ser resumidos em duas variáveis de interesse, sendo possível utilizar os dois DAC's disponíveis, além de apresentar a frequência de simulação necessária para uma escala de até 10 kHz para a entrada. Assim a simulação em tempo real será realizada tendo seus resultados medidos através de um osciloscópio.

Esse capitulo abordou uma breve revisão entre os circuitos a serem simulados, além de apresentar alguns trabalhos que utilizam de simulação HIL. Por fim, foram citados os motivos para a escolha do microcontrolador como simulador digital, pois esse apresenta um alto custo benefício devido a sua velocidade de processamento e seu custo.

3 Modelagem e Desenvolvimento

Este capítulo apresenta a metodologia adotada para a modelagem dos circuitos escolhidos. Além disso, são apresentados os procedimentos realizados para cada implementação na plataforma de simulação em tempo real, seguindo os conceitos apresentados na revisão bibliográfica.

3.1 Conversor Buck em malha aberta

O primeiro estágio para a simulação de um determinado sistema em tempo real é obter as equações que descrevem seu comportamento de forma dinâmica. Para o conversor *Buck*, a modelagem inicial depende de análises compreendendo suas variáveis de interesse para cada modo de operação, de forma a representar cada estado individualmente.

A Figura 17 ilustra a topologia padrão do *Buck* para seus dois modos de funcionamento, onde R_L simboliza a resistência parasita causada pela não idealidade do indutor.

Figura 17 - Configuração do conversor Buck no modo: a) chave fechada e b) chave aberta.



Adaptado de: (HART, 2011).

Todas as topologias de conversores modelados nessa monografia seguem um padrão de operação em estado estável e em modo de condução contínua, permitindo que as equações deduzidas possam reproduzir o comportamento do conversor diante das características desejadas.

O modelo representativo para o primeiro estado de operação pode ser obtido considerando a chave fechada por um período $d \cdot T$, onde d simboliza o valor de *Duty Cycle* e T o período completo de chaveamento. As equações (3.1) e (3.2) são encontradas a partir da aplicação das leis de Kirchoff para tensão e corrente, considerando o diodo como um elemento ideal.

$$-V_S + R_L \cdot i_L(t) + \frac{L \cdot di_L(t)}{dt} + V_C(t) = 0$$
(3.1)

$$i_L(t) = \frac{C \cdot dV_C(t)}{dt} + \frac{V_C(t)}{R}$$
(3.2)

em que:

 V_S = Tensão de alimentação CC aplicada na entrada;

 V_C = Queda de tensão no capacitor;

 i_L = Corrente que circula pelo indutor;

 $R, L \in C =$ Resistor, indutor, capacitor;

As variáveis de estado em conversores CC-CC devem ser escolhidas de acordo com a aplicação, sendo as grandezas em torno dos armazenadores de energia as mais importantes para projeto. A corrente no indutor e a tensão no capacitor são as mais utilizadas, apresentando um resultado variantes no tempo.

Assim, i_L e V_C podem ser definidas como variáveis de estado, possibilitando o isolamento de suas respectivas derivadas para integração computacional futura, conforme:

$$\frac{di_L(t)}{dt} = -\frac{V_C(t)}{L} + \frac{V_S}{L} - \frac{R_L \cdot i_L(t)}{L}$$
(3.3)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} + \frac{i_L(t)}{C}$$
(3.4)

Para o segundo modo, a chave deve ser considerada aberta por um período (1 - d)T, seu modelo de equações pode ser encontrado de maneira semelhante ao modo chave fechada, sendo possível obter as equações (3.5) e (3.6) em função das derivadas de i_L e V_C .

$$\frac{di_L(t)}{dt} = -\frac{V_C(t)}{L} - -\frac{R_L \cdot i_L(t)}{L}$$
(3.5)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} + \frac{i_L(t)}{C}$$
(3.6)

Observando os termos de ambos os modos, é possível notar uma semelhança que permita a escrita de uma equação geral que relaciona os dois estados da chave. Essa expressão geral deve ser escrita em função do *Duty Cycle*, como mostra as equações a seguir, reduzindo as duas configurações em um equivalente.

$$\frac{di_L(t)}{dt} = -\frac{V_C(t)}{L} + \frac{V_S \cdot d}{L} - \frac{R_L \cdot i_L(t)}{L}$$
(3.7)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} + \frac{i_L(t)}{C}$$
(3.8)

Como o modelo está descrito em torno das derivadas definidas, é preciso utilizar a integral com objetivo de encontrar o valor real da variável desejada. Essa operação deve ser discretizada para implementação no LAUNCH escolhido, uma vez que aparelhos digitais trabalham com um tempo T_S entre as amostras, sendo impossível realizar o processamento no tempo contínuo.

Para encontrar essa representação discreta, a aproximação *backward* deve ser aplicada a expressão no domínio de Laplace, como é mostrado nas equações a seguir.

$$sIL(s) = \frac{V_C(s)}{L} + \frac{V_S \cdot d}{L} - \frac{R_L \cdot i_L(s)}{L}$$
(3.9)

$$sV_c(s) = \frac{V_C(s)}{R \cdot C} + \frac{i_L(s)}{C}$$
(3.10)

Em seguida, a variável s deve ser substituída pelo equivalente no domínio z e transformada para o espaço amostral k, de forma a obter uma relação final para integração de i_L e V_C , como apresentado nas equações (3.11) e (3.12).

$$i_L(k) = i_L(k-1) + \frac{di_L(k)}{dk}T_s$$
(3.11)

$$V_C(k) = V_C(k-1) + \frac{dV_C(k)}{dk}T_s$$
(3.12)

No formato computacional, os valores atuais dependem das variáveis de estado calculadas nas amostras anteriores. Dessa forma, substituindo o tempo t pelo equivalente k no modelo contínuo, obtém-se o modelo discreto a ser implementado no LAUNCH, representado nas equações (3.13) e (3.14).

$$\frac{di_L(k)}{dk} = -\frac{V_C(k-1)}{L} + \frac{V_S \cdot d}{L} - \frac{R_L \cdot i_L(k-1)}{L}$$
(3.13)

$$\frac{dV_C(k)}{dk} = -\frac{V_C(k-1)}{R \cdot C} + \frac{i_L(k-1)}{C}$$
(3.14)

Essas equações discretas possibilitam a implementação do modelo digital do conversor CC-CC no microcontrolador escolhido (LAUNCHXL), utilizando o *software* CCS para implementação do código da simulação. O algoritmo utilizado está presente no repositório *GitHub*, cujo *link* encontra-se disponível no Anexo A dessa monografia, e segue o diagrama representado na Figura 18.



Figura 18 – Diagrama de blocos para algoritmo no microcontrolador LAUNCHXL.

Fonte: (BASTOS u. a., 2020).

O processamento no simulador digital acontece por meio de interrupções causadas por uma frequência ajustável internamente, que deve depender da aplicação desejada. As equações do conversor *Buck* são processadas em um tempo relativamente curto, devido a sua baixa complexidade ($T_{process} < 10 \ \mu s$), permitindo um ajuste para frequência de interrupção de até 500 kHz, mantendo a margem de segurança (70% maior que o processamento das equações).

Esse fator implica em um limite para frequência de entrada que deve obedecer a relação de amostragem caso seja um sinal contínuo, e caso seja um sinal PWM digital, o processamento deve ser 100 vezes maior que a frequência de comutação, para uma boa atualização dos resultados de saída dependendo dos eventos de entrada.

Os testes para verificação do comportamento HIL ocorreram a partir de uma bancada que envolvia 2 microcontroladores. O primeiro (Esp 32) tem objetivo de emitir um sinal PWM para a entrada do segundo (LAUNCHXL), que por sua vez tem finalidade de funcionar como simulador em tempo real. A saída do Esp 32 é alimentada na entrada do LAUNCHXL com o parâmetro de *Duty Cycle*, que se deve ao fato de i_L e V_C serem dependentes do valor de *d*, sendo possível testar as variações de comportamento alterando a forma do PWM.

A Figura 19 ilustra a conexão entre a saída digital do pino D2 do Esp 32, carregando o sinal PWM, com a entrada digital j5-50 do LAUNCH Delfino.



Figura 19 – Esquema de ligação entre processadores.

SISTEMA EM MALHA ABERTA



O algoritmo da simulação em tempo real é carregado no microcontrolador delfino e o código do PWM no Esp 32. O Microcontrolador escolhido para simular digitalmente o sistema processa sinais adquiridos pela entrada GPIO digital e disponibiliza os resultados das variáveis de estado em suas saídas DAC, nos pinos j7-70 (V_C) e j3-30 (i_L), conforme mostra o Anexo B. Ambas as saídas são conectadas a um osciloscópio para visualização e medição das grandezas utilizadas para teste, com os resultados apresentados no capitulo 4.

3.2 Controle PI em malha fechada do Conversor Buck

O desenvolvimento apresentado para a implementação das equações dinâmicas do conversor *Buck* se manteve restrito a simulação em tempo real de malha aberta, onde não há presença de realimentação entre o simulador e o elemento real (Esp 32). Entretanto, uma das aplicações mais utilizadas para a metodologia HIL é a configuração em malha fechada, que possibilita a conexão de elementos reais que se comunicam de forma bidirecional com o *hardware* que irá simular a planta.

Essa conexão permite observar o comportamento que o elemento externo causa no modelo implementado dentro do simulador digital, sem que seja necessário lidar com o sistema real para realização de testes dinâmicos. Assim, torna-se interessante abordar a aplicação de um elemento controlador em malha fechada com o conversor *Buck*, tendo o objetivo de avaliar a influência do controlador PI no modelo proposto.

A Figura 20 ilustra o diagrama esquemático da estrutura implementada entre o controlador digital e a planta simulada em tempo real do conversor *Buck*, escolhido por ser um circuito trivial na eletrônica de potência. O Esp 32 foi utilizado para realizar as funções de leitura dos dados de

 i_L provenientes do LAUNCH e geração do PWM, cujo valor de largura de pulso foi definido por um controlador de corrente PI embarcado no micro.







Os ganhos do controlador PI podem ser encontrados de forma computacional pelo método de sintonia do *sisotool*, sendo necessário modelar o sistema em torno de um ponto de operação, para relacionar a saída de corrente i_L a ser controlada com a entrada de *Duty Cycle*, conforme:

$$G(s) = \frac{i_L(s)}{d(s)} \tag{3.15}$$

O modelo de pequenos sinais permite predizer o comportamento das saídas no equipamento quando ocorrem variações de baixa frequência na razão cíclica ou na fonte de entrada, sendo preciso lineariza-la em torno de um ponto de operação (FÁBIO, 2014).

Para encontrar essa representação do conversor *Buck*, é necessário escrever as equações propostas no modo chave fechada e aberta de forma matricial em espaço de estados, ponderando as matrizes de acordo com d, como representado nas equações (3.16) e (3.17) (BASTOS, 2013).

$$\dot{x} = [A_1d + A_2(1-d)]x + [B_1d + B_2(1-d)] \cdot V$$
(3.16)

$$\dot{y} = [C_1 d + C_2 (1 - d)]x + [D_1 d + D_2 (1 - d)] \cdot V$$
(3.17)

No qual as matrizes A, B, C e D podem ser encontradas nas expressões modeladas em espaço de estado seguindo a forma $\dot{x} = A_1x + B_1u$ e $y = C_1x$ para chave fechada. Associando os termos, torna-se possível obter:

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dV_C}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-r_L}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ V_C \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V$$
(3.18)

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ V_C \end{bmatrix}$$
(3.19)

Para chave aberta, as equações de modelo seguem o padrão anterior, podendo ser escritas na forma $\dot{x} = A_2 x + B_2 u$ e $y = C_2 x$, conforme mostra as equações (3.20) e (3.21).

$$\begin{bmatrix} \frac{di_L}{dt} \\ \frac{dV_C}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-r_L}{L} & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} i_L \\ V_C \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} V$$
(3.20)

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} i_L \\ V_C \end{bmatrix}$$
(3.21)

Nota-se que para o conversor *Buck*:

$$A_1 = A_2; \quad B_1 = B_2; \quad C_1 = C_2; \quad D_1 = D_2 = 0;$$

As deduções completas podem ser encontradas em (BASTOS, 2013) e serão apresentadas de forma resumida a seguir. Supondo uma pequena perturbação aplicada em (3.16) para separar as componentes CC e CA, os sinais podem ser decompostos seguindo o padrão das equações a seguir.

$$x = X + \hat{x} \tag{3.22}$$

$$i_L = I_L + i_L \tag{3.23}$$

$$d = D + \hat{d} \tag{3.24}$$

onde X representa a parte continua e \hat{x} a parte alternada. Desconsiderando a perturbação na entrada V e analisando apenas a parte oscilatória em torno do ponto médio, a equação (3.16) pode ser reescrita na forma expressa:

$$\dot{\hat{x}} = AX + BV + A\hat{x} + \left[(A_1 - A_2)x + (B_1 - B_2)V \right] \cdot \hat{d}$$
(3.25)

O mesmo procedimento pode ser realizado para se encontrar o correspondente na saída, obtendo como resultado a equação (3.26).

$$\hat{i}_L = C\hat{x} + [(C_1 - C_2)X] \cdot \hat{d}$$
(3.26)

Dessa forma, para obter a função de transferência representada em pequenos sinais, aplica-se a transformada de Laplace nas equações oscilantes (3.25) e (3.26) e relacionando ambas, conforme é mostrado:

$$G(s) = \frac{\hat{i}_L(s)}{\hat{d}(s)} = C[sI - A]^{-1}[(A_1 - A_2)X + (B_1 - B_2)V + (C1 - C_2)X]$$
(3.27)

As matrizes A, $B \in C$ devem ser substituídas pelas representações encontradas nas equações de espaço de estado em cada modo de funcionamento do conversor *Buck*. As variáveis dos parâmetros internos dos componentes foram definidas de acordo com (HART, 2011), sendo os valores disponibilizados na Tabela 2.

Parâmetro	Valor	
Indutor	$400 \ \mu H$	
Capacitor	$100 \ \mu \mathrm{F}$	
Tensão de entrada	50 Volts	
Resistor (R)	5 Ω	
Resistor (R_L)	$100 \text{ m}\Omega$	
Frequência de Chaveamento	2 kHz	

Tabela 2 – Paramêtros utilizados no controle do conversor buck

Realizando a substituição das equações com os valores dos componentes definidos, obtémse através de métodos computacionais a função de transferência 3.28 modelada em pequenos sinais.

$$G(s) = \frac{12500s + 2500000}{s^2 + 200s + 255000}$$
(3.28)

No *software MATLAB*, a ferramenta *sisotool* tem objetivo de obter os ganhos $K_I e K_P$ ao se informar a margem de fase, a frequência de corte do controlador e a função de transferência em pequenos sinais. Para o conversor *Buck*, a margem de fase foi definida como 65 graus e a frequência de corte como 10 vezes menor a de chaveamento, para respeitar a dinâmica do sistema, gerando os ganhos mostrados na equação abaixo.

$$Cs(s) = 0,0194684 + \frac{11,452}{s}$$
(3.29)

A Figura 21 apresenta o ambiente *sisotool* na qual foram definidos os parâmetros internos e obtido os ganhos do controlador PI.

Figura 21 - Plataforma sisotool representando os ganhos obtidos para o controlador PI

ID Tuning	×
Compensator	
$C = 38.048 \times \frac{(1 + 0.0019s)}{s}$	
▼ Select Loop to Tune	
LoopTransfer_C 💌	
Add New Loop	
Specifications	
Tuning method: Robust response time 💌	
Controller Type: P P I PI PD PID Design with first order derivative filter	
Design mode: Frequency -	
≪ 16.3 Bandwidth (rad/s) 1.63e+03 1200 €	
0 Phase Margin (deg) 90 Parameters	

Fonte: do autor.

A função encontrada relacionando os ganhos do controlador Cs(s) é representada no domínio contínuo de Laplace, tornando necessário a discretização para implementação digital. Esse processo pode ser feito pela aproximação *backward*, mostrado nas equações 3.30, 3.31 e 3.32, onde e(k) representa o valor de erro e Int(z) a operação integral.

$$(z-1)Int(z) = K_I e(z)T_S z$$
 (3.30)

$$Int(k) = Int(k-1) + K_I e(k) T_s$$
 (3.31)

$$C(k) = Int(k-1) + K_I e(k)T_s + K_P e(k)$$
(3.32)

A modulação PWM no controlador Esp 32 utiliza do formato dente de serra para a onda portadora de alta frequência, implicando em algumas particularidades que se diferenciam da onda tradicional triangular. Para a portadora triangular simétrica, as interrupções de amostragem ocorrem na parte central da onda modulante, como mostra a Figura 22(b), gerando uma amostra próxima a média do sinal de leitura.



Figura 22 – PWM uniformemente amostrado com: a) Portadora Dente de serra, b) Portadora Triangular.

Fonte: (BUSO und MATTAVELLI, 2015).

onde c(t) representa a onda portadora, V_{M0} o sinal PWM e m(t) o sinal amostrado. Já para a portadora dente de serra da Figura 22(a), as interrupções ocorrem na borda de transição da onda modulante, implicando em uma amostragem próximo aos picos do sinal de leitura, gerando um *offset* com metade do valor pico a pico na corrente controlada (BUSO und MATTAVELLI, 2015).

Dessa forma, a corrente i_L triangular de saída no conversor *Buck* simulado não será controlada a partir do seu valor médio pela referência e sim em seu valor mínimo. A arquitetura HIL malha fechada foi implementada apenas para o conversor *Buck*, mas para os demais casos, a mesma metodologia.

3.3 Conversor Boost em malha aberta

A análise para modelagem do conversor *Boost* é similar ao *Buck* e deve compreender ambos os modos de funcionamento. A Figura 23 apresenta a topologia nas duas configurações para o conversor a ser modelado.



Figura 23 – Configuração do conversor Boost no modo: a) chave fechada e b) chave aberta.

Adaptado de: (HART, 2011).

No primeiro modo, a chave é considerada fechada pelo período $d \cdot T$, onde é possível pelas leis de Kirchoff encontrar as relações das equações (3.33) e (3.34), que descrevem o comportamento das derivadas nas variáveis de interesse.

$$\frac{di_L(t)}{dt} = \frac{V_S}{L} - \frac{R_L \cdot i_l(t)}{L}$$
(3.33)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} \tag{3.34}$$

Para o segundo modo, a chave deve ser considerada aberta por um período (1 - d)T, possibilitando a dedução do conjunto das equações:

$$\frac{di_L(t)}{dt} = \frac{V_S}{L} - \frac{V_C(t)}{L} - \frac{R_L \cdot i_l(t)}{L}$$
(3.35)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} + \frac{i_L(t)}{C}$$
(3.36)

Relacionando as equações de ambos os modos através da variável d, é possível discretizar o modelo com a aproximação de *backward* e encontrar as expressões (3.37) e (3.38).

$$\frac{di_L(k)}{dk} = \frac{V_S}{L} - \frac{R_L \cdot i_l(k-1)}{L} - \frac{V_C(k-1) \cdot (1-d)}{L}$$
(3.37)

$$\frac{dV_C(k)}{dk} = -\frac{V_C(k-1)}{R \cdot C} + \frac{i_L(k-1) \cdot (1-d)}{C}$$
(3.38)

Essas equações descrevem a dinâmica seguida pelo conversor *Boost* no modelo discreto, sendo possível implementa-las através de algoritmos no simulador digital em tempo real, seguindo as restrições necessárias para simulação em malha aberta.

3.4 Conversor Buck-Boost em malha aberta

Analogamente as outras implementações, o conversor *Buck-Boost* apresenta uma topologia de uma adaptação dos dois conversores anteriores, apresentando modos de operação que dependem da posição de sua chave, da forma ilustrada pela Figura 24.

Figura 24 – Configuração do conversor *Buck-Boost* no modo: a) chave fechada e b) chave aberta.



Adaptado de: (HART, 2011).

Quando a chave permanece fechada por um período de tempo $d \cdot T$, o circuito é modelado de acordo com as equações:

$$\frac{di_L(t)}{dt} = \frac{V_S \cdot d}{L} - \frac{R_L \cdot i_L(t)}{L}$$
(3.39)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} \tag{3.40}$$

A partir do momento em que a chave se abre por um período (1 - d)T, a análise do circuito se altera, formando as equações (3.41) e (3.42), encontradas de acordo com a Leis de Kirchoff.

$$\frac{di_L(t)}{dt} = \frac{V_S \cdot d}{L} + \frac{V_C(t)}{L} - \frac{R_L \cdot i_L(t)}{L}$$
(3.41)

$$\frac{dV_C(t)}{dt} = -\frac{V_C(t)}{R \cdot C} - \frac{i_L(t)}{C}$$
(3.42)

Essas equações devem ser reescritas por meio do intermediário d para descrever o comportamento completo do sistema, independente do modo de operação, obtendo as expressões na forma discreta.

$$\frac{di_L(k)}{dk} = \frac{V_S \cdot d}{L} + \frac{V_C(k-1) \cdot (1-d)}{L} - \frac{R_L \cdot i_L(k-1)}{L}$$
(3.43)

$$\frac{dV_C(k)}{dk} = -\frac{V_C(k-1)}{R \cdot C} - \frac{i_L(k-1) \cdot (1-d)}{C}$$
(3.44)

3.5 Inversor Monofásico em malha aberta

A configuração de inversor monofásico escolhida para modelagem é a mais comum, chamada ponte H, contendo quatro chaves semicondutoras para o comando do sistema. A Figura 25 ilustra a topologia em ponte H, que é alimentada por uma fonte V_{dc} e acoplada a um filtro RL na saída, a fim de descartar as componentes de alta frequências geradas na corrente CA.

Figura 25 – Topologia Inversor Monofásico ponte H.



Adaptado de: (BASTOS u. a., 2019).

Para a simulação em tempo real, o sinal de acionamento das chaves será proveniente do *Duty Cycle*, implementado via PWM de forma digital no Esp 32. A modelagem das chaves pode ser obtida observando o comportamento do conjunto, sendo possível subtituir a ponte de semicondutores por (2d - 1).

O sistema monofásico pode ser redesenhado da forma equivalente mostrada na Figura 26, onde foi adicionado uma resistência r_{on} para representar a não idealidade da ponte H de semicondutores.





Adaptado de: (BASTOS u. a., 2019).

Utilizando o equivalente da Figura 26 e aplicando a Lei de Kirchoff para as tensões, é possível encontrar a relação da equação (3.45) representando a malha do circuito.

$$(2d-1)V_{dc}(t) = R_{on} \cdot i_{ac(t)} + L \frac{di_{ac}(t)}{dt} + R_L \cdot i_{ac}(t) + v_{ac}(t)$$
(3.45)

Discretizando por meio da aproximação *backward* para a implementação em *hardware* digital, obtém-se uma nova equação.

$$(2d-1)V_{dc}(k) = R_{on} \cdot i_{ac}(k-1) + L \frac{di_{ac}(k-1)}{dk} + R_L \cdot i_{ac}(k-1) + v_{ac}(k)$$
(3.46)

A corrente DC de entrada pode ser encontrada através da modelagem da ponte H, resultando na expressão 3.47.

$$i_{dc}(k) = i_{ac}(k)(2d-1) \tag{3.47}$$

A equação 3.48 mostra a forma isolada para a derivada de corrente AC de saída, definida como variável de interesse.

$$\frac{di_{ac}(k-1)}{dk} = \frac{V_{dc}}{L} - \frac{R_L \cdot i_{ac}(k-1)}{L} - \frac{R_{on} \cdot i_{ac}(k-1)}{L} - \frac{v_{ac}(k-1)}{L}$$
(3.48)

Como o inversor apresenta saída alternada, o PWM que gera o *Duty Cycle* deve assumir uma forma diferente das anteriores. Nesse caso será utilizado a modulação senoidal, como apresentado na revisão bibliografica, na qual a onda de referência assume forma senoidal e é modelada por um sinal triangular de alta frequência (portadora). A frequência fundamental da onda de saída é igual a frequência da referência (BACON und OTHERS, 2011).

O atual capitulo apresentou como se deu a modelagem dos sistemas e a implementação no *hardware* escolhido para a simulação em tempo real, junto a algumas particularidades a serem seguidas para uma boa representação da resposta final.

4 Resultados e Discussões

Este capítulo apresenta os resultados práticos obtidos com a simulação em tempo real pelo LAUNCHXL, utilizando o osciloscópio TK C011290 como equipamento de medição. As discussões realizadas são baseadas nas comparações dos resultados práticos com os simulados pelo *software* PSIM, a fim de se criar uma correlação entre ambos.

4.1 Análise Buck em malha aberta

As equações que modelam o conversor *Buck* em tempo discreto foram implementadas no LAUNCHXL-F28379D, com o objetivo inicial de realizar a simulação em tempo real de malha aberta. Os sinais gerados pelo DAC do LAUNCH foram adquiridos usando um osciloscópio digital (Tektronix, modelo C011290). Esses sinais serão comparados aos obtidos através de simulações realizadas usando o *software* PSIM

O esquemático da representação implementada no *software* simulador é mostrado na Figura 27.



Figura 27 – Esquema de representação do conversor Buck no PSIM.

Fonte: do autor.

Os valores internos ajustados para cada componente foram adaptados de (HART, 2011) e estão disponibilizados na Tabela 3.

Valor
500 kHz
$2 \ \mu \ s$
$400 \ \mu H$
$100 \ \mu F$
50 volts
5 Ω
$100 \ \Omega$
2 kHz

Tabela 3 –	Parâmetros	utilizados na	implementaci	ão do	conversor <i>Buck</i>
I u o o i u o	1 unumou ob	utilizado int	, impromontaç	uo uo	

Apesar da simulação computacional apresentar as dificuldades citadas anteriormente, ainda sim o PSIM permite descrever o comportamento do sistema simulado com resultados próximos aos ideais e um ajuste de passo temporal menor que a simulação em tempo real, que se assemelhe ainda mais a um circuito físico. Isso permite que os resultados da simulação em tempo real possam ser comparados aos obtidos pelo PSIM para se obter uma comparação que valide a simulação do LAUNCHXL.

Os parâmetros de saída para cada simulação devem ser definidos de acordo com as variáveis de interesse, $i_L \in V_C$, respeitando a organização dos pinos DAC no LAUNCHXL para a simulação em tempo real. O osciloscópio conectado é capaz de adquirir uma sequência de dados, permitindo a comparação do resultado gráfico do simulador digital com o computacional.

Pelo fato da simulação via PSIM emular as características ideais de um conversor *Buck*, alguns testes podem ser realizados para verificar as semelhanças entre os métodos. O primeiro deles surge devido a uma variação de *Duty Cycle* aplicado na entrada, onde o comportamento obtido de i_L é mostrado na Figura 28.





Fonte: do autor.

O valor de *d* foi alternado periodicamente entre 0,75, 0,5 e 0,25, entre um período de 120 ms para cada variação. Os dados destacados na cor vermelha da Figura 28 (a) representam a saída de corrente i_L para a simulação via *software*, respeitando um passo temporal de 100 η segundos. Já os dados em azul na Figura 28 (b) simbolizam a resposta obtida na saída do simulador experimental, em um passo de tempo ajustado para 2 μ segundos.

Ao serem submetidos a uma variação de *Duty Cycle* periodica na entrada, os gráficos obtidos aparentam ter resultados similares, onde no momento em que o processador detecta a mudança na entrada, o valor de saída é atualizado para i_L , respeitando o tempo de simulação e apresentando um estado transitório que se estabiliza no mesmo valor para ambos os métodos. Para a saída de V_C , a mesma variação foi aplicada, apresentando o resultado da Figura 29.

Figura 29 – Comportamento de V_C para variação de entrada no conversor *Buck* em ambas simulações, na qual a) Resultado PSIM e b) Resultado HIL.





Para ambas as saídas, é possível perceber a semelhança visual entre o comportamento obtido pelo simulador computacional e o digital, apresentando uma reação transitória e se estabilizando no valor médio até a próxima mudança. Para obter uma correlação mais detalhada a respeito, novas análises devem ser realizadas para estabelecer diferenças e semelhanças.

A resposta encontrada pode ser separada em estado transitório e regime permanente, uma vez que o sistema segue sua dinâmica dependente do *Duty Cycle*. Mantendo o valor de *d* fixo na entrada, ambos os resultados podem ser sobrepostos para as simulações realizadas, como mostra a Figura 30.



Figura 30 – Corrente i_L Buck sobreposta para ambas simulações com d = 0,75.



A escala de corrente obtida pelo resultado experimental foi multiplicada por um fator de 25 (normalização de corrente). Preservando os parâmetros aplicados para obter i_L sobreposto, o mesmo procedimento deve ser realizado para a tensão V_C , obtidos graficamente pela Figura 31.





Diante das Figuras encontradas para as saídas definidas, é possível notar que no estado transiente, i_L e V_C seguem o padrão parecido pra ambas as simulações, sendo necessário uma ampliação de escala para destacar suas diferenças. A resposta apresenta um máximo sobressinal

que é atingido no mesmo valor em ambos os métodos, junto a uma futura estabilização que ocorre em um valor médio.

As pequenas divergências ocorrem devido a diferença entre o passo temporal para cada simulação, a resolução do DAC de saída no HIL (elemento não presente na simulação PSIM), ruído causado por interferências externas e as dificuldades do aparelho medidor detectar pequenas variações, apresentando uma resolução pior para a saída V_C , pela presença de *ripples* em pequenas amplitudes.

Para o regime permanente, o sistema permanece estável alternando seu valor de i_L e V_C em torno de um ponto médio dependente dos parâmetros de entrada. Os valores médios obtidos por ambas simulações estão disponibilizados na Tabela 4, para dois valores distintos de d, possibilitando a comparação de forma quantitativa.

Tabela 4	– Va	lores	quantitativos	para	regime	permanente	do	conversor	Buck	k.
----------	------	-------	---------------	------	--------	------------	----	-----------	------	----

Parâmetro	Duty Cycle	Valor PSIM (médio)	Valor Medido (fator de conversão)
V_C	0,75	36,75 V	35,6 V (x16,66)
i_L	0,75	7,34 A	7,33 A (x8,33)
V_C	0,5	24.73 V	23,5 V (x16,66)
i_L	0,5	4.88 A	4,925 A (x8,33)

Mantendo a entrada d fixa no valor de 0,75, os resultados em regime permanente podem ser obtidos após a espera de um tempo para estabilização (cerca de 0,2 segundos), como mostrado na Figura 32.

Figura 32 – Resultados para regime permanente do conversor *Buck* em HIL mantendo d fixo em 0,75.



Fonte: do autor.

Alterando a entrada d para 0,5, uma nova resposta pode ser obtida, com o comportamento gráfico presente na Figura 33.



Figura 33 – Resultados para regime permanente do conversor *Buck* em HIL mantendo d fixo em 0,5.

Fonte: do autor.

Após a análise dos resultados, conclui-se que a simulação a simulação em tempo real representou adequadamente o padrão esperado de um conversor *Buck*. Os valores médios quantitativos mostraram diferenças mínimas entre as simulações (máximo de 5%), assim como os dados gráficos vistos de forma sobreposta, onde as divergências se devem a motivos já discutidos.

É importante ressaltar que o modelo do conversor *Buck* é de simples complexidade, o que implica na possibilidade de ajuste para a frequência de interrupção em 500 kHz, limitando a frequência do PWM de entrada para aproximadamente 5 kHz, podendo se tornar uma representação pior quando esse limite for superado. Assim, o resultado apresentado foi capaz de validar a simulação em tempo real do modelo proposto em tempo discreto implementada no *hardware* de baixo custo.

4.2 Análise do Controlador PI no conversor Buck

O controlador PI foi projetado no *sisotool* e implementado de forma digital no Esp 32, onde uma de suas saídas fornece o sinal PWM equivalente para o processamento do microcontrolador LAUNCHXL, com as mesmas equações da configuração malha aberta.

Por simplicidade, optou-se por controlar a corrente i_L do indutor para o conversor *buck*. A representação dos componentes que envolvem o processo foi esquematizada no PSIM, como mostra a Figura 34.





Fonte: do autor.

A Figura 35 apresenta o comportamento do sistema em tempo real. Os sinais foram monitorados pelo osciloscópio para o controle da corrente i_L onde a referência se altera em um período próximo a 600 ms.

Figura 35 – Comportamento do conversor *Buck* em malha fechada perante a uma referência variante.



Fonte: do autor.

A corrente de saída i_L , representada na cor azul escura, segue um comportamento esperado de um controle para conversores, onde sua frequência representa o chaveamento e o seu valor mínimo sua referência ajustada. O sinal verde *cyano* representa o PWM proveniente do Esp 32. Nota-se um pequeno transiente até o valor médio atingir o esperado, que se deve ao tempo de resposta do sistema e do controlador. Para esse teste, a variação na corrente de referência se deu nos valores de 8 e 5 A, com mudança periódica. O resultado obtido pela simulação em tempo real pode ser comparado ao do PSIM, mantendo uma referência de corrente fixa em 5 A, como mostra a Figura 36.







A resposta mostra que a corrente do conversor implementado pelo HIL estabiliza em um valor mínimo de 5 A em regime permanente. O controlador foi submetido a uma alteração de referência para 8 Amperes, onde a corrente i_L do sistema seja forçada a atingir o valor mínimo da onda triangular de acordo com o ajuste. Os resultados são mostrados na Figura 37.







A amostragem da leitura realizada pelo Esp é feita com a portadora no formato dente de serra, refletindo em uma saída controlada no valor mínimo de i_L e não no valor médio, como seria em uma amostragem com forma triangular.

Uma maneira de verificar a eficácia do controlador projetado é variando a carga de saída do conversor, pois independente desse valor, a corrente i_L deve permanecer com seu valor próximo ao de referência. Esse teste é apresentado na Figura 38, onde a resistência do *buck* foi alterada para 3 Ω mantendo o valor de referência em 8 A.



Figura 38 - Comportamento para carga Resistiva de 3 Ohms. Fonte: Do Autor.

Fonte: do autor.

Por meio das figuras apresentadas, é possível afirmar que o controlador projetado exerce a função de manter o valor mínimo da corrente de saída triangular próximo a referência. Devido aos ruídos causados pela implementação prática, o erro pode não zerar completamente, gerando uma pequena inconstância nos valores medidos, porém em aplicações menos precisas esse efeito pode ser desconsiderado. Dessa forma, o controlador PI digital implementado no módulo Esp conseguiu ser validado sem a necessidade da montagem de um conversor *Buck* com elementos reais, facilitando o processo de teste.

4.3 Análise Boost em malha aberta

Para os próximos conversores implementados via simulação em tempo real, os procedimentos adotados serão os mesmos, alterando apenas as equações que descrevem o circuito. Seguindo essa ideia, o conversor *Boost* foi implementado no *software* PSIM para realização de comparações de acordo com o esquemático da Figura 39.



Figura 39 – Esquematico do conversor Boost no PSIM.

Fonte: do autor.

Os valores internos de cada componente para esse sistema foram adaptados de (HART, 2011) e estão disponibilizados na Tabela 5.

Parâmetro	Valor
Frequência de interrupção HIL	500 kHz
Período de simulação HIL	$2 \ \mu s$
Indutor	$300 \ \mu H$
Capacitor	$400 \ \mu F$
Tensão de entrada	20 V
Resistor R	$10 \ \Omega$
Resistor R_L	0,1 Ω
Frequência de Chaveamento	10 kHz

Tabela 5 – Parâmetros utilizados na implementação do Conversor Boost.

Aplicando uma variação de *Duty Cycle* na entrada a cada 150 milissegundos, é possível obter o resultado para as saídas definidas, como a corrente i_L mostrado pela Figura 40 nas duas simulações.





Fonte: do autor.

Os dados provenientes do PSIM estão destacados na cor vermelha, na parte superior da Figura, e os dados do LAUNCH na cor azul. O passo de simulação em tempo real se manteve de 2 μ s, assim como o da simulação via software em 100 ns. A variável de tensão V_C pode ser obtida da mesma forma, sendo representada pela Figura 41.

Figura 41 – Comportamento de V_C para variação de entrada no conversor *Boost* em ambas simulações, na qual a) Resultado PSIM e b) Resultado HIL.



Fonte: do autor.

Ambas as simulações apresentam resultados similares, por isso é necessário novamente realizar outros testes que possam englobar o detalhamento para regime transitório e permanente com o *Duty Cycle* fixo. Para isso, o circuito foi submetido a entrada d no valor de 0,75, de forma a obter a Figura 42 para a sobreposição de ambas as simulações de i_L .





Preservando os parâmetros do circuito, é possível realizar o mesmo teste para a tensão V_C , mostrado na Figura 43.





Fonte: do autor.

O fator de multiplicação para comparação das duas respostas depende do valor de normalização implementado no código. A resposta do conversor *Boost* para corrente e tensão obteve uma reprodução pior que a do conversor *Buck*, que se deve a resolução do DAC, uma vez que frequências maiores são mais complexos de se reproduzir. Porém, mesmo com essa mudança, o resultado ainda seguiu um padrão similar entre ambas as simulações. O ajuste de escala mostrado no canto superior direito aponta a correlação entre as respostas.

Para a análise em regime permanente mantendo a entrada fixa, o resultado deve ser obtido após a espera do tempo de estabilização. Os valores medidos estão dispostos na Tabela 6 para dois valores de d, permitindo a comparação quantitativa de ambos os métodos.

Parâmetro	Duty Cycle	Valor PSIM (médio)	Valor Medido (fator de conversão)
V_C	0,75	68 V	65,6 V (x33,33)
i_L	0,75	2,7 A	2,3 A (x23,33)
V_C	0,25	26,25 V	23,6 V (x33,33)
i_L	0,25	3,57 A	3,5 A (x23,33)

Tabela 6 – Valores quantitativos para regime permanente do conversor *Boost*.

O comportamento das variáveis i_L (azul) e V_C (verde) junto ao PWM de entrada na simulação em tempo real é mostrado na Figura 44, obtidos via osciloscópio.

Figura 44 – Resultados para regime permanente do conversor *Boost* em HIL mantendo d fixo em 0,75.



Fonte: do autor.

Alterando o valor de *Duty Cycle* para 0,25 e mantendo os parâmetros internos, obtém-se novos resultados, presentes na Figura 45.

Figura 45 – Resultados para regime permanente do conversor *Boost* em HIL mantendo d fixo em 0,25.



Fonte: do autor.

As diferenças dos valores se devem a pequena imprecisão na medição do osciloscópio. A simulação em tempo real obteve uma representação fiel ao esperado de um conversor *Boost*, apresentando alguns erros devido a ruídos e a frequência de amostragem maior que a do PSIM, sendo um fator desconsiderado em algumas aplicações.

4.4 Análise Buck-Boost em malha aberta

A exemplo dos anteriores, as equações do conversor *Buck-Boost* foram implementadas no LAUNCH e no *software* PSIM, para efeito de comparação, como mostra o esquemático da Figura 46.



Figura 46 – Esquemático do conversor Buck-Boost no PSIM.

Fonte: do autor.

Os valores dos componentes para teste foram adaptados de (HART, 2011) e podem ser observados na Tabela 7.

Parâmetro	Valor
Frequência de interrupção HIL	500 kHz
Período de simulação HIL	$2 \ \mu s$
Indutor	2 mH
Capacitor	8 mF
Tensão de entrada	24 V
Resistor R	5 Ω
Resistor R_L	0,1 Ω
Frequência de Chaveamento	2 kHz

Tabela 7 – Parâmetros utilizados na implementação do Conversor Buck-Boost.

Para testar o comportamento do sistema, foi aplicado a mesma variação de *Duty Cycle* na entrada das outras topologias. O resultado obtido nas duas simulações é apresentado na Figura 47 para a corrente i_L .





Fonte: do autor.

A mesma variação deve ser aplicada na entrada do LAUNCHXL e da simulação computacional para obter a saída de tensão V_C , como mostrado na Figura 48, separando ambos os resultados obtidos. Figura 48 – Comportamento de V_C para variação de entrada no conversor *Buck-Boost* em ambas simulações, na qual a) Resultado PSIM e b) Resultado HIL.





Visualmente, para ambas as variáveis de interesse, o resultado segue um padrão similar, tanto na transição quanto na resposta final. Ao variar a entrada, a saída sofre um transiente até se estabilizar no valor respectivo de corrente e tensão, que depende do valor de PWM.

Para visualizar melhor o comportamento de ambas as simulações, os resultados obtidos de forma gráfica foram sobrepostos e suas escalas ajustadas, de forma a obter a Figura 49.

Figura 49 – Corrente i_L do *Buck-Boost* sobreposta para ambas simulações com d = 0,75.



Fonte: do autor.

Os resultados sobrepostos entre si mostram o comportamento semelhante das duas simulações, o que implica em uma boa capacidade de representação do modelo para a corrente do indutor. Para uma análise completa, os resultados de V_C também foram sobrepostos de forma a observar correlações entre as respostas, como na Figura 50.

Figura 50 – Tensão V_C do *Buck-Boost* sobreposta para ambas simulações com d = 0,75.



Fonte: do autor.

A tensão de saída V_C para essa topologia se difere das outras, pois a polaridade é invertida em relação a tensão de alimentação, por isso foi preciso realizar alguns ajustes no fator de escala da resposta experimental, para que os resultados possam ser sobrepostos. Com a imagem final, é possível notar que os dados são semelhantes, mas apresentam diferenças devido à dificuldade de se obter as variações no caso de tensão, além disso o ruido atrapalha na medição, tornando as respostas minimamente divergente entre os métodos.

Em regime permanente, os resultados se estabilizam em torno de um valor médio, que

pode ser calculado a fim de realizar comparações e validar a representação na simulação em tempo real. Os valores calculados para o circuito de forma ideal são mostrados na Tabela 8 junto aos valores obtidos por medição.

Paramêtro	Duty Cycle	Valor PSIM (médio)	Valor Medido (fator de conversão)
V_C	0,75	-53,8 V	-54 V (x-40)
i_L	0,75	44,5 A	43,7 A (x53,33)
V_C	0,25	-7,75 V	-7,5 V (x-40)
i_L	0,25	2,14 A	2,2 A (x53,33)

Tabela 8 – Valores quantitativos para regime permanente do conversor Buck-Boost.

O comportamento em regime permanente para o sistema em tempo real com entrada de *Duty Cycle* fixa em 0,75 é apresentado na Figura 51, obtida via osciloscópio.

Figura 51 – Resultados para regime permanente do conversor *Buck-Boost* mantendo d fixo em 0,75.



Fonte: do autor.

Alterando o valor de entrada para um d = 0,25, as grandezas de tensão e corrente na saída apresentam seu valor médio reduzido e uma forma de onda diferente, como mostra a Figura 52.





Fonte: do autor.

Com auxílio de todos os testes representados pelas figuras apresentadas para o sistema *Buck-Boost*, é possível concluir que o modelo implementado na simulação em tempo real se mostrou uma representação fiel do conversor real, tanto na resposta transiente quanto na estabilização, validando o modelo descrito para essa topologia.

4.5 Simulação do Inversor Monofásico em malha aberta

O inversor monofásico foi implementado no LAUNCH para simulação em tempo real e no software PSIM para fins de comparação entre os resultados. A Figura 53 apresenta o esquemático da topologia no *software*.



Figura 53 – Inversor Monofásico implementado no PSIM.

Fonte: do autor.

Os valores dos componentes para teste foram adaptados de (HART, 2011) e podem ser observados na Tabela 9.
Parâmetro	Valor
Frequência de interrupção HIL	500 kHz
Período de simulação HIL	$2 \ \mu s$
Indutor	3,8 mH
Tensão de entrada	20 V
Resistor R_L	13,8 Ω
Resistor R_{on}	0,1 Ω
Frequência de Chaveamento	1,8 kHz

T 1 1 0			• • •	~ 1	•	64 .
Tahela 9 -	Parametros	utilizados na	implements	acao do	1nversor	monotasico
	1 arametros	utilizados na	implement	içao ao	111101301	monorasico.

A entrada de *Duty Cycle* na simulação em tempo real recebeu um PWM na forma senoidal, gerado pelo ESP 32, com o objetivo de obter uma senoide de frequência fundamental em 60 Hz na corrente de saída. Analogamente, essa representação também foi implementada via PSIM.

A Figura 54 mostra o resultado obtido ao se conectar as pontas medidoras do osciloscópio nos pinos de saída do LAUNCH.



Figura 54 – Resposta do inversor obtida pela simulação em tempo real.

Fonte: do autor.

As cores verde, azul e roxo representam a corrente de entrada i_{dc} , corrente de saída i_{ac} e o PWM senoidal respectivamente. A resposta segue um padrão já esperado de um inversor, onde a corrente de saída i_{ac} apresenta um formato dependente da senoide de referência, ajustada com uma frequência fundamental de 60 Hz. As pequenas variações se devem a alta frequência de chaveamento, que dita o acionamento dos diodos em um período de 55 milissegundos, afetando também na forma de i_{dc} , cuja onda apresenta um comportamento similar à de um retificador.

Através de uma ampliação na escala de tempo, essas pequenas variações nas correntes decorrente da frequência de chaveamento se tornam mais perceptíveis, como mostra a Figura 55.



Figura 55 – Resposta do inversor obtida pela simulação em tempo real com ampliação de escala.

Fonte: do autor.

A sobreposição entre os dados encontrados na simulação em tempo real e no PSIM pode ser feita para uma análise mais detalhada e conclusiva, possibilitando estabelecer uma relação direta entre as grandezas encontradas de ambos os métodos. A Figura 56 apresenta a sobreposição para i_{ac} encontrado em ambas as simulações para o mesmo período de tempo.



Figura 56 – Corrente i_{ac} do inversor sobreposta para ambas simulações.

A Figura 56 mostra que as duas simulações seguem um padrão similar na forma senoidal, com uma frequência fundamental de 60 Hz. As pequenas distorções se devem ao passo diferente de simulação e aos ruídos decorrentes do medidor. Para a corrente de entrada i_{dc} essa sobreposição também foi realizada, como mostra a Figura 57.

Fonte: do autor.







Para a corrente de entrada, i_{dc} apresentou um formato similar em ambas as simulações, de forma semelhante a uma corrente contínua de um circuito retificador, padrão nessa configuração de semicondutores, com pequenas exceções decorrentes ao mesmo motivo da corrente i_{ac} . Porém, apesar dessas pequenas divergências, o modelo simulado em tempo real de malha aberta se mostrou uma ótima representação do inversor monofásico, sendo possível utilizar dele para realizar outras simulações futuras com conexão de elementos reais, como controladores e sensores.

5 Conclusão

O presente trabalho mostrou como se constitui a arquitetura HIL na prática, utilizando topologias comuns na eletrônica de potência para realizar a simulações em tempo real de malha aberta e fechada. Os sistemas a serem simulados passaram por estágios de modelagem por equações diferenciais, discretização por *backward* e implementação digital, onde todos os processos foram apresentados e discutidos.

A aproximação *backward* se mostrou eficaz e capaz de reproduzir as equações contínuas na forma desejada, se tornando um algoritmo de simples implementação necessário ao *hardware* de baixo custo. A implementação de malha aberta possibilitou a observação do comportamento dessa equações modeladas perante a uma entrada de *Duty Cycle* externa, proveniente do módulo Esp 32.

A resposta encontrada foi comparada com a simulação *off-line* do PSIM para as saídas de interesse, apresentando um comportamento parecido em ambas. Entre alguns dos benefícios que a simulação via *hardware* apresentou, podem ser citados a redução no custo de memória do computador *Host* em comparação com o PSIM e a possibilidade de conexão com um elemento externo para acionamento do conversor.

Os resultados foram analisados para estado transitório e regime permanente, onde pode ser notado pequenas divergências de comparação devido ao ruído do medidor e da diferença entre passo de simulação do microcontrolador e do PSIM, fator que pode ser desconsiderado de acordo com a finalidade dos testes.

O algoritmo foi implementado de forma que a frequência de interrupção ocorra em um evento de PWM interno do LAUNCH, implicando em um limite de resolução DAC de saída e em um limite de frequência de entrada.

Em malha fechada, o HIL foi capaz de validar o controlador PI projetado utilizando o modelo virtual do conversor Buck, sem precisar do sistema físico completo, abordando alguns testes para verificar o comportamento dinâmico do processo. Os resultados se mostraram coerentes com a expectativa, e confirmaram que o controlador atingiu seu objetivo de forma eficaz.

Com isso, foi possível notar que a utilização do HIL é muito importante principalmente em ambientes de pesquisa, visto que nem sempre há possibilidade para montagem de um circuito físico apenas para testes. A principal contribuição dessa monografia é mostrar que o microcontrolador LAUNCH é uma ótima opção para simulação em tempo real de circuitos com baixa complexidade, sendo necessário obter apenas o modelo do circuito escolhido e estabelecendo uma relação entre complexidade e frequência de simulação, de forma que o processamento seja feito para obter resultados fielmente representados.

5.1 Sugestões para trabalhos futuros

Algumas sugestões para seguimento do tema desse trabalho podem ser citadas, como:

- Implementação de topologias mais complexas no simulador LAUNCHXL, como conversores bidirecionais ou isolados.
- Simulação e controle em malha fechada para o inversor conectado à rede;
- Utilização de outros processadores mais poderosos para testes da simulação com frequência de chaveamento acima de 50 kHz.

Referências

AHMED, ASHFAQ: Eletrônica de potência. Pearson Education do Brasil, 2008 1, 5

ANAKWA, WINFRED K.; ROCA, HP; LOPEZ, JOSE; MALINOWSKI, ALEKSANDER: Environments for rapid implementation of control algorithms and hardware-in-the-loop simulation. In: *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02* Bd. 3 IEEE (Veranst.), 2002, S. 2288–2293 3

BACON, CAMPANHOL ; OTHERS: Análise comparativa das técnicas spwm e svm aplicadas a um inversor de tensão trifásico. In: *UNOPAR Científica Ciências Exatas e Tecnológicas* 10 (2011), Nr. 1 36

BASTOS, RENAN F. ; FUZATO, GUILHERME H. ; AGUIAR, CASSIUS R. ; NEVES, RODOLPHO V. ; MACHADO, RICARDO Q.: Model, design and implementation of a low-cost HIL for power converter and microgrid emulation using DSP. In: *IET Power Electronics* 12 (2019), Nr. 14, S. 3833–3841 2, 20, 21, 35

BASTOS, Renan F. ; SILVA, Fernando B. ; AGUIAR, Cassius R. ; FUZATO, Guilherme ; MACHADO, Ricardo Q.: Low-cost hardware-in-the-loop for real-time simulation of electric machines and electric drive. In: *IET Electric Power Applications* 14 (2020), Nr. 9, S. 1679–1685 26

BASTOS, RENAN F.: Sistema de gerenciamento para carga e descarga de baterias (chumboácido) e para busca do ponto de máxima potência gerada em painéis fotovoltaicos empregados em sistemas de geração distribuída, Universidade de São Paulo, Dissertation, 2013 28, 29

BUSO, SIMONE ; MATTAVELLI, PAOLO: Digital control in power electronics. In: *Synthesis Lectures on Power Electronics* 5 (2015), Nr. 1, S. 1–229 32

CASTRO, FELIPE FUSCALDI D.: *PROJETO DE CONTROLADORES PI ROBUSTO PARA PROCESSOS DE MOAGEM EMPREGANDO DESIGUALDADES MATRICIAIS LINEARES*. OURO PRETO, Universidade Federal de Ouro Preto, Diplomarbeit, 2020. – 83 S 13, 14

CIOBOTARU, MIHAI ; TEODORESCU, ET. A.: A new single-phase PLL structure based on second order generalized integrator. In: 2006 37th IEEE Power Electronics Specialists Conference IEEE (Veranst.), 2006, S. 1–6 15, 16

DE PAULA, CAIO F. ; GUARACY, ET A.: SINTONIA ANALÍTICA DE CONTROLADO-RES AVANCO/ATRASO DE FASE EM TEMPO DISCRETO ATRAVÉS DO LUGAR DAS RAÍZES. (2016) 12

DE SOUZA BAETA, ÉRIC JOSÉ ; DA SILVA, CEZAR G. ; DO CARMO SILVA, MARC´LIO: ESTRATÉGIA DE CONTROLE PREFERENCIAL EM PLANTAS DE BENEFICIAMENTO DE MINÉRIO DE FERRO: APLICAÇÕES PRÁTICAS E BENEFÍCIOS. (2015) 13

DORF, RICHARD C. ; BISHOP, ROBERT H.: Sistemas de Controle Modernos, 8ª Edição. In: *Editora LTC, Rio de Janeiro* (2001) 14 DROSDOL, JOHANNES ; PANIK, FERDINAND: The Daimler-Benz driving simulator a tool for vehicle development. In: *SAE Transactions* (1985), S. 981–997 17

ECKHARD, DIEGO; CAMPESTRINI, LUC'OLA: Análise do uso de modelos discretizados para identificação de modelos de biorreatores anaeróbicos. In: *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics* 4 (2016), Nr. 1 14

FÁBIO, JOSÉ F.: *Técnicas de controle aplicadas aos conversores CC-CC*, Universidade Tecnológica Federal do Paraná, B.S. thesis, 2014 28

GONZADA, ARTHUR S. O.: Estudo de técnica de modulação por largura de pulso (PWM) aplicado a inversores trifásicos. (2018) 5

HANSELMANN, H: Hardware-in-the loop simulation as a standard approach for development, customization, and production test of ECU's / SAE Technical Paper. 1993. – Forschungsbericht 17

HART, DANIEL W.: *Power electronics*. Tata McGraw-Hill Education, 2011 1, 5, 6, 7, 8, 9, 10, 11, 12, 23, 30, 33, 34, 37, 48, 52, 57

HAUGEN, FINN: Discretization of simulator, filter, and PID controller. In: Tech. Teach, http://www.mic-journal.no/PDF/ref/Haugen2010.pdf (2010) xi, 15, 16

HERNÁNDEZ, FRANK ALBERTO I. ; CANESIN, CARLOS A.: Simulação em tempo real de sistemas de distribuição de energia elétrica utilizando-se estruturas com descrição de hardware em software., Sao Paulo State University, Brazil, Dissertation, 2015 19

HOSSEINPOUR, F ; HAJIHOSSEINI, H: Importance of simulation in manufacturing. In: *World Academy of Science, Engineering and Technology* 51 (2009), Nr. 3, S. 292–295 1

INSTRUMENTS, TEXAS: *LAUNCHXL-F28379D Overview User's Guide*. <https://www.ti.com/lit/ug/sprui77c/sprui77c.pdf?ts=1653567375938>. 2016. – original document from SPRUI77C 68, 69

INSTRUMENTS, TEXAS: C2000 Delfino MCU F28379D LaunchPadTM development kit. https://www.ti.com/tool/LAUNCHXL-F28379D>. 2017. – Acessado em: 25 março 2021 21

ISERMANN ; OTHERS: Hardware-in-the-loop simulation for the design and testing of engine-control systems. In: *Control Engineering Practice* 7 (1999), Nr. 5, S. 643–653 16

JUNG, JEE-HOON: Power hardware-in-the-loop simulation (PHILS) of photovoltaic power generation using real-time simulation techniques and power interfaces. In: *Journal of Power Sources* 285 (2015), S. 137–145 17

KARPENKO; OTHERS: Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic actuators in a flight control application. In: *Mechatronics* 19 (2009), Nr. 7, S. 1067–1077 18

KOPETZ, HERMANN: *Real-time systems: design principles for distributed embedded applications.* Springer Science & Business Media, 2011 18

LAPLANTE, PHILLIP A.; OVASKA, SEPPO J.: *Real-Time Systems Design and Analysis: Tools for the Practitioner.* Wiley-IEEE Press, 2012 18

MINA, J ; FLORES, Z ; LÓPEZ, E ; PÉREZ, A ; CALLEJA, J-H: Processor-in-the-loop and hardware-in-the-loop simulation of electric systems based in FPGA. In: *2016 13th International Conference on Power Electronics (CIEP)* IEEE (Veranst.), 2016, S. 172–177 20

OGATA, KATSUHIKO: Engenharia de Controle Moderno, 3a. edição. In: *Editora. LTC. Rio de Janeiro* (1998) 12

OPPENHEIM, ALAN V.: Sinais e sistemas. Prentice-Hall, 2010 20

PINHEIRO, JOSÉ A.: *Desenvolvimento de um controlador PID para aplicação em uma mesa angular rotativa*, Universidade de São Paulo, Dissertation, 2009 13

RODRIGUES, WELBERT A. ; OTHERS: Estudo de um transformador eletrônico baseado em topologia de conversores modulares para aplicação em sistemas de microrredes. (2018) 2

RTDS: *NovaCor, a new generation of simulation hardware for the RTDS Simulator.* Disponível em: https://adaptaust.com.au/wp-content/uploads/2017/08/NovaCor-Features-20170413. pdf>. Acesso em: 14 abril 2021. 2017 19, 20

SCHODER, KARL ; OTHERS: Commissioning of MW-scale Power Hardware-in-the-Loop interfaces for experiments with AC/DC Converters. In: *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society* IEEE (Veranst.), 2013, S. 5364–5367 17

SILVA, HILGAD MONTELO D.: *Simulação com hardware in the loop aplicada a veículos submarinos semi-autônomos.*, Universidade de São Paulo, Dissertation, 2008 3, 16

SONG, LIN ; LU, ET. A.: A comparative study on Tustin rule based discretization methods for fractional order differentiator. In: 2014 4th IEEE International Conference on Information Science and Technology IEEE (Veranst.), 2014, S. 515–518 15

TERLIP, DANIEL V.; OTHERS: A methodology for characterizing and modeling inverters for grid integration studies using Power Hardware-in-the-Loop. In: 2012 IEEE Power and Energy Society General Meeting IEEE (Veranst.), 2012, S. 1–5 17

TYPHOON HIL: *Home Page - Typhoon HIL*. Disponível em: <https://www.typhoon-hil. com/>. Acesso em: 14 abril 2021. 2018 19, 20

VIDOR, EVERTON; RIGO NATANIEL; PINHEIRO JOSÉ R.: Análise de Conversores por Modelos de Similaridade com Fontes. In: *10th Seminar on Power Electronics and Control* (2017) 5

YAMANAKA, HUGO F.: CONTROLE DE UM SISTEMA VIGA-HÉLICE USANDO UM COMPENSADOR DE AVANÇO E ATRASO DE FASE. In: *CADERNO DE RESUMOS* (2021) 12 Anexos

ANEXO A – Endereço via GitHub

A.1 Repositório dos códigos implementados no TCC

<https://github.com/gustavoAS1/HIL.git>

ANEXO B – Device Pin Out LAUNCHXL-F28379D

Mux Value					J3		Mux Value				
x	2	1	0	Pin	Pin	0	Alt Function	2	X		
			3.3V	1	21	5V					
			GPIO32	2	22	GND					
	SCIRXDB		GPIO19	3	23	ADCIN14	CMPIN4P				
	SCITXDB		GPIO18	4	24	ADCINC3	CMPIN6N				
			GPIO67	5	25	ADCINB3	CMPIN3N				
			GPIO111	6	26	ADCINA3	CMPIN1N				
SPICLKA ⁽¹⁾			GPIO60	7	27	ADCINC2	CMPIN6P				
			GPIO22	8	28	ADCINB2	CMPIN3P				
		SCLA	GPIO105 ⁽²⁾	9	29	ADCINA2	CMPIN1P				
		SDAA	GPIO104 ⁽²⁾	10	30	ADCINA0	DACOUTA				

Figura 58 – F28379D LaunchPad Pin Out e Pin Mux Options - J1, J3.

Fonte: (INSTRUMENTS, 2016).

Figura 59 – F28379D LaunchPad Pin Out and Pin Mux Options - J4, J2.

Mux Value					J2				
x	2	1	0	Pin	Pin	0	1	2	x
		EPWM1A	GPI00	40	20	GND			
		EPWM1B	GPI01	39	19	GPIO61			
		EPWM2A	GPIO2	38	18	GPIO123			SD1_C1 ⁽¹⁾
		EPWM2B	GPIO3	37	17	GPIO122			SD1_D1 ⁽¹⁾
		EPWM3A	GPIO4	36	16	RST			
		EPWM3B	GPIO5	35	15	GPIO58			SPISIMOA ⁽¹⁾
		OUTPUTXBAR1	GPIO24	34	14	GPIO59			SPISOMIA ⁽¹⁾
OUTPUTXBAR7 ⁽¹⁾			GPIO16	33	13	GPIO124			SD1_D2 ⁽¹⁾
			DAC1	32	12	GPIO125			SD1_C2 ⁽¹⁾
			DAC2	31	11	GPIO29 ⁽²⁾			OUTPUTXBAR6 ⁽¹⁾

Fonte: (INSTRUMENTS, 2016).

Mux Value				J5	J7		Mux Value			
x	2	1	0	Pin	Pin	0	Alt Function	2	x	
			3.3V	41	61	5V				
			GPIO95	42	62	GND				
SCIRXDC ⁽¹⁾			GPIO139	43	63	ADCIN15	CMPIN4N			
SCITXDC ⁽¹⁾			GPIO56	44	64	ADCINC5	CMPIN5N			
			GPIO97	45	65	ADCINB5				
			GPIO94	46	66	ADCINA5	CMPIN2N			
SPICLKB ⁽¹⁾			GPIO65	47	67	ADCINC4	CMPIN5P			
			GPIO52 ⁽²⁾	48	68	ADCINB4				
SCLB ⁽¹⁾			GPIO41 ⁽²⁾	49	69	ADCINA4	CMPIN2P			
SDAB ⁽¹⁾			GPIO40 ⁽²⁾	50	70	ADCINA1	DACOUTB			

Figura 60 – F28379D LaunchPad Pin Out and Pin Mux Options - J5, J7.

Fonte: (INSTRUMENTS, 2016).

Figura 61 – F28379D LaunchPad Pin Out and Pin Mux Options - J8, J6.

Mux Value					.16	Mux Value			
x	2	1	0	Pin	Pin	0	1	2	x
		EPWM4A	GPIO6	80	60	GND			
		EPWM4B	GPIO7	79	59	GPIO66			
		EPWM5A	GPIO8	78	58	GPIO131			SD2_C1 ⁽¹⁾
		EPWM5B	GPIO9	77	57	GPIO130			SD2_D1 ⁽¹⁾
		EPWM6A	GPIO10	76	56	RST			
		EPWM6B	GPIO11	75	55	GPIO63			SPISIMOB ⁽¹⁾
OUTPUTXBAR3 ⁽¹⁾			GPIO14	74	54	GPIO64			SPISOMIB ⁽¹⁾
OUTPUTXBAR4 ⁽¹⁾			GPIO15	73	53	GPIO26			SD2_D2 ⁽¹⁾
			DAC3	72	52	GPIO27			SD2_C2 ⁽¹⁾
			DAC4	71	51	GPIO25			OUTPUTXBAR2 ⁽¹⁾

Fonte: (INSTRUMENTS, 2016).