



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

José Augusto Arbuseri

**Retificador PFC unidirecional dual-boost com redução de modo comum
utilizando semicondutores de GaN**

Florianópolis
2023

José Augusto Arbuseri

**Retificador PFC unidirecional dual-boost com redução de modo comum
utilizando semicondutores de GaN**

Dissertação submetida ao Programa de Pós-Graduação
em Engenharia Elétrica da Universidade Federal de Santa
Catarina para a obtenção do título de mestre em Eletrô-
nica de Potência e Acionamento Elétrico.
Orientador: Prof. Samir Ahmad Mussa, Dr.
Coorientador: Prof. Marcelo Lobo Heldwein, Dr.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Arbugeri, José Augusto

Retificador PFC unidirecional dual-boost com redução de modo comum utilizando semicondutores de GaN / José Augusto Arbugeri ; orientador, Samir Ahmad Mussa, coorientador, Marcelo Lobo Heldwein, 2023.

134 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2023.

Inclui referências.

1. Engenharia Elétrica. 2. Retificador. 3. PFC. I. Ahmad Mussa, Samir. II. Lobo Heldwein, Marcelo. III. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Engenharia Elétrica. IV. Título.

José Augusto Arbuseri

**Retificador PFC unidirecional dual-boost com redução de modo comum
utilizando semicondutores de GaN**

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Marcio Silveira Ortmann, Dr.Eng.

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina - IFSC.

Prof. Telles Brunelli Lazzarin, Dr. Eng.

Universidade Federal de Santa Catarina - UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Eletrônica de Potência e Acionamento Elétrico.

Coordenação do Programa de
Pós-Graduação

Prof. Samir Ahmad Mussa, Dr.
Orientador

Florianópolis, 2023.

Este trabalho é dedicado aos meus colegas de classe e aos
meus queridos pais.

AGRADECIMENTOS

Começo expressando minha profunda gratidão à minha família. A minha mãe, pai e irmãos por todo apoio durante essa minha jornada.

À minha noiva, Natalha Roberta Back, meu alicerce diário: sua força, carinho e incentivo têm sido fundamentais. Por cada decisão tomada, cada obstáculo enfrentado, ela esteve ao meu lado, sendo não apenas um suporte, mas uma fonte de inspiração.

Um reconhecimento especial ao Dr. Samir, meu orientador. Nossa parceria no INEP se estende para além das páginas desta dissertação e serei sempre grato pelas oportunidades, desafios e aprendizados que compartilhamos. Agradeço também à banca, cujas críticas e sugestões foram essenciais para elevar a qualidade deste trabalho.

Aos colegas do INEP, agradeço pela rica troca de experiências e ideias. Cada conversa, cada debate, enriqueceu não apenas este projeto, mas também minha trajetória profissional.

Por fim, expresso minha gratidão aos colegas da Reivax. Os últimos dois anos foram uma jornada de novas experiências, desafios e aprendizados, e sou grato por cada momento compartilhado com vocês.

*"Quando a educação não é libertadora,
o sonho do oprimido é ser o opressor."
Paulo Freire, 1968.*

RESUMO

Este trabalho propõe uma nova topologia de conversor de correção do fator de potência (PFC) monofásico operando em alta frequência de comutação, utilizando transistores eGaN (nitreto de gálio) e diodos SiC (carbeto de silício). A topologia é composta por quatro indutores que podem ser acoplados, com o objetivo de reduzir o número de componentes magnéticos na estrutura. O trabalho aborda as principais etapas de operação do conversor, analisando os esforços nos semicondutores, modelando as perdas de energia e desenvolvendo um modelo de controle para o conversor. Além disso, foi realizado o desenvolvimento de um protótipo do conversor para validar a estrutura proposta, juntamente com um protótipo de controle baseado em FPGA (Field-Programmable Gate Array). Por meio de experimentos e testes, foram obtidos resultados que comprovam o funcionamento adequado do conversor e a eficácia das malhas de controle implementadas. O trabalho contribui para o avanço da área de conversores de PFC, apresentando uma nova topologia com componentes de alta eficiência e explorando a redução de componentes magnéticos. Além disso, os resultados experimentais validam o desempenho do conversor e validam a eficácia do controle desenvolvido.

Palavras-chave: Retificador ativo. eGaN. Alta frequência. FPGA.

ABSTRACT

This work proposes a new topology for a single-phase Power Factor Correction (PFC) converter operating at high switching frequency, using eGaN transistors (gallium nitride) and SiC diodes (silicon carbide). The topology consists of four inductors that can be coupled together, aiming to reduce the number of magnetic components in the structure. The work explores the main operating stages of the converter, analyzing the stresses on the semiconductors, modeling energy losses, and developing a control model for the converter. Additionally, a converter prototype was developed to validate the proposed structure, along with an FPGA-based control prototype (Field-Programmable Gate Array). Through experiments and tests, results were obtained that confirm the proper functioning of the converter and the effectiveness of the implemented control loops. This work contributes to the advancement of PFC converter technology by introducing a new topology with high-efficiency components and exploring the reduction of magnetic components. Furthermore, the experimental results validate the converter's performance and the developed control approach.

Keywords: Active rectifier. eGaN. High frequency. FPGA.

LISTA DE FIGURAS

Figura 1 – Retificador PFC clássico , imagem de (HULIEHEL; LEE; CHO, 1992)	19
Figura 2 – Estrutura bridgeless derivadas das famílias de conversores: buck, boost, buck-boost, cuk e Sepic , imagem de (CHEN, Z.; DAVARI; WANG, 2020).	21
Figura 3 – Família de conversores bridgeless de 3 níveis, imagem de (LANGE <i>et al.</i> , 2015)	22
Figura 4 – Topologias com mitigação da tensão de modo comum, imagem de (SIU; HO, 2016)	22
Figura 5 – Inversor com baixa tensão de modo comum, imagem de (MUNARETTO; HELDWEIN, Marcelo L., 2018)	23
Figura 6 – Retificador com baixa tensão de modo comum, imagem de (GU <i>et al.</i> , 2013)	23
Figura 7 – Controle por valores médios, imagem de (CHEN, Z.; DAVARI; WANG, 2020).	23
Figura 8 – Controle por self-control, imagem de (LANGE; HELDWEIN, Marcelo Lobo, 2011).	23
Figura 9 – Controle por one-cycle-control, imagem de (JAPPE; MUSSA, Samir Ahmad, 2011).	25
Figura 10 – Conversor retificador proposto.	27
Figura 11 – Etapas de operação do conversor proposto. Na parte superior é apresentados as etapas para o semiciclo positivo e na inferior para o semiciclo negativo	28
Figura 12 – Principais forma de onda do conversor proposto. O primeiro gráfico exhibe as formas de onda da moduladora e da portadora. O segundo o sinal PWM do interruptor S_1 enquanto o terceiro apresenta o sinal PWM do interruptor S_2 . O quinto gráfico apresenta as tensões de entrada e saída do conversor. O sexto exhibe as correntes nos indutores L_1 e L_3 do conversor, bem como a corrente de entrada. O sétimo apresenta as correntes nos diodos S_1 e S_2 do conversor. Por fim, o oitavo exhibe as correntes nos diodos D_1 e D_2 do conversor.	29
Figura 13 – Valor de modulação em função do tempo e do índice do modulação α	31
Figura 14 – Corrente normaliza em função do tempo e do índice do modulação α	32
Figura 15 – Ondulação de tensão em função do tempo e da potência.	33
Figura 16 – Calculo de corrente RMS para o período da rede e no período de comutação.	34
Figura 17 – Corrente média e RMS dos semicondutores normaliza, demonstradas em função do índice de modulação α	39

Figura 18 – Comparação entre o modelo comutado do conversor e resultados de simulação na entrada do controle ativo. Na figura são apresentados a tensão de saída, tensão de entrada e corrente de entrada do conversor.	50
Figura 19 – Comparação entre o modelo comutado e médio do conversor e resultados de simulação, em regime permanente senoidal. Na figura são apresentados a tensão de saída, tensão de entrada e corrente de entrada do conversor.	50
Figura 20 – Modelo equivalente referente a um semiciclo da rede do conversor para análise da tensão de modo comum.	52
Figura 21 – Circuito equivalente para análise da tensão de modo comum no conversor	52
Figura 22 – Resultado tensão de modo comum para o conversor considerando L1 igual a L2.	53
Figura 23 – Resultado tensão de modo comum para o conversor considerando L2 igual a zero.	54
Figura 24 – Metodologia para projeto do indutor do conversor.	56
Figura 25 – Informações do núcleo utilizado fornecido pelo fabricante do núcleo Amosense.	57
Figura 26 – Modelo simplificado de tensão de condução do diodo utilizado. Imagem obtida em (AG, 2019).	60
Figura 27 – Tensão de condução dos diodos ao longo do ciclo da rede, e perdas instantâneas.	61
Figura 28 – Distribuição de perdas nos elementos do conversor	62
Figura 29 – Diagrama de blocos da estratégia de controle utilizada no conversor PFC.	63
Figura 30 – Diagrama de blocos do SOGI-PLL, que consiste na união da estrutura SOGI-QSG e SRF-PLL	64
Figura 31 – Estratégia de modulação utilizada do conversor proposto. Consiste no modulador PWM tradicional, com um circuito de habilitação dos interruptores referente a cada ciclo da rede.	65
Figura 32 – Resultados práticos da modulação implementada.	65
Figura 33 – A estratégia de amostragem do conversor implementada no DSP. A_isr PWM é disparada a cada 5 eventos do modulador pwm, nela é iniciada a conversão do ADC e disparo da tarefa do CLA. O fim da conversão do ADC gera uma interrupção que roda a rotina de controle e atualiza os valores de modulação, finalizando as rotinas de controle.	66
Figura 34 – A estratégia de amostragem do conversor implementada no DSP. Resultados práticos das temporizações das rotinas implementadas apresentados na Figura 33.	67

Figura 35 – A figura mostra os resultados da simulação do controle PFC apresentado. A figura consiste em seis partes que mostram as seguintes variáveis: (1) V_{in} (não filtrado), V_{in} (filtrado) e V_q (Eixo quadratura do SOGI-PLL); (2) V_{out} (não filtrado), V_{out} (filtrado) e V_{out} ref; (3) I_{in} (não filtrado) e I_{in} (filtrado); (4) S_p , $\langle S_p \rangle_{ts}$, S_n e $\langle S_n \rangle_{ts}$; (5) w_t e V_{pll} ; e (6) I_{out}	68
Figura 36 – Esquema ilustrativo da implementação do Hardware-in-the-Loop (HIL) nos dois CPU e CLA (Control Law Accelerator) do DSP (Digital Signal Processor) da Texas Instruments.	70
Figura 37 – Resultado de entrada de operação do conversor PFC, na parte inferior a corrente de entrada e na parte superior a tensão saída. A figura mostra o momento em que o controle é acionado, saindo do modo de retificador passivo para PFC ativo.	71
Figura 38 – A figura mostra um degrau carga de 50 % para 100 % e de 100 % para 50 %, na parte inferior a corrente de entrada e na parte superior a tensão saída.	71
Figura 39 – Na figura é apresentado o conversor em regime permanente senoidal. No canal 1 é apresentado a tensão de saída do conversor, Canal 2 a entrada e por fim canal 3 a tensão de entrada do conversor.	72
Figura 40 – Destaque na ondulação de tensão da saída V_{out} . É possível observar a componente em 120 Hz da ondulação do barramento, junto com a componente em alta frequência de comutação, 375 kHz	72
Figura 41 – Destaque na ondulação de corrente da entrada I_{in} . É possível observar a frequência da ondulação do indutor é de 375 kHz, que representa a frequência de comutação.	73
Figura 42 – O resultado para duas sequencias de degrau de carga. O primeiro degrau de 500 % para 100 % e segundo degrau de 100 % para 50 %, onde 100 % representa a carga nominal. É apresentado a corrente de saída, tensão de saída, tensão de entrada e corrente de entrada.	73
Figura 43 – O resultado para duas sequencias de degrau de carga. O primeiro degrau de 500 % para 100 % e segundo degrau de 100 % para 50 %, onde 100 % representa a carga nominal. É apresentado a tensão de saída, tensão de entrada e corrente no indutor L_1 e L_3 , respectivamente do semi ciclo positivo e negativo.	74
Figura 44 – Resposta ao degrau filtro passa baixa, do modelo Contínuo e modelo discretizado	86
Figura 45 – Resposta ao degrau filtro passa banda, do modelo Contínuo e modelo discretizado	88

Figura 46 – Resposta ao degrau filtro rejeita faixa, do modelo Contínuo e modelo discretizado	90
Figura 47 – Resposta ao degrau filtro passa alta, do modelo Contínuo e modelo discretizado	92
Figura 48 – Resposta ao degrau filtro G_1 , do modelo contínuo e modelo discretizado.	104
Figura 49 – Resposta ao degrau filtro G_2 , do modelo contínuo e modelo discretizado.	106
Figura 50 – Protótipo conversor PFC, visualização 3D real e alitum	123
Figura 51 – Protótipo do Conversor PFC: Um protótipo desenvolvido em dois layers, que integra as partes de potência, instrumentação e controle em seu layout. A imagem exibe o protótipo completo do conversor PFC, destacando sua estrutura em camadas e a distribuição dos componentes de potência, instrumentação e controle pelos layers.	124
Figura 52 – Protótipo de Gate Drive: Um protótipo desenvolvido com base nas recomendações do fabricante Gan Systems, conforme descrito no documento (GAN SYSTEMS, 2022). Este protótipo de gate drive incorpora uma fonte isolada e pulsos isolados. A imagem ilustra o resultado do projeto do protótipo gate drive.	125
Figura 53 – Protótipo do Módulo de Controle FPGA: Um protótipo desenvolvido pelo autor utilizando o FPGA Cyclone IV e conectores compatíveis com o LaunchPad da Texas Instruments. Este módulo de controle FPGA oferece recursos para processamento e controle de dados, como: ADC, PWM, SPI entre outros recursos que podem ser implementado via descrição de hardware (VHDL).	126
Figura 54 – Protótipo do sensor de corrente CFS1000: Um protótipo desenvolvido pelo autor utilizando o sensor CFS1000 da Sensitec. É evidenciado as dimensões da PCB na imagem.	127
Figura 55 – Software fornecido pelo fabricante para projeto da PCB do sensor de corrente CFS1000. Fonte: (SENSITEC, s.d.)	128
Figura 56 – Forno construído para solda das PCBs	129
Figura 57 – Controle de temperatura usando preditor de smith	131
Figura 58 – Página 1 do aplicativo de monitoração implementado em python.	132
Figura 59 – Página 2 do aplicativo de monitoração implementado em python.	132
Figura 60 – Página 3 do aplicativo de monitoração implementado em python.	133

LISTA DE TABELAS

Tabela 1 – Especificações dos parâmetros para cálculo e simulação.	38
Tabela 2 – Equações dos esforços dos semicondutores do conversor.	39
Tabela 3 – Especificações dos parâmetros para cálculo e simulação.	40
Tabela 4 – Tensão de modo comum para diferentes condições de $L1$ e $L2$	54
Tabela 5 – Parâmetros de entrada para o projeto do conversor	55
Tabela 6 – Valores de k_1 , k_2 e k_3 em diferentes condições de operação	59
Tabela 7 – Valores de k_4 , k_5 e k_6 em diferentes condições de operação.	59
Tabela 8 – Equações dos 4 tipos de filtros de segunda ordem	82

LISTA DE ABREVIATURAS E SIGLAS

CCM	<i>Continuous Conduction Mode</i>
eGaN	<i>Enhancement-mode gallium-nitride</i>
FPGA	<i>Field-programmable gate array</i>
PFC	<i>Power factor correction</i>
PWM	<i>Pulse Width Modulation</i> (Modulação por largura de pulso)

LISTA DE SÍMBOLOS

S_1	Interruptor GaN que opera no semiciclo positivo
S_2	Interruptor GaN que opera no semiciclo negativo
L_1	Indutor superior que opera no semiciclo positivo
L_3	Indutor superior que opera no semiciclo negativo
D_1	Diodo superior que opera no semiciclo positivo
D_2	Diodo superior que opera no semiciclo negativo
D_5	Diodo inferior que opera no semiciclo positivo
D_6	Diodo inferior que opera no semiciclo negativo
L_2	Indutor inferior que opera no semiciclo positivo
L_4	Indutor inferior que opera no semiciclo negativo

SUMÁRIO

1	INTRODUÇÃO	19
1.1	CONTEXTUALIZAÇÃO DO PROBLEMA	19
1.2	REVISÃO BIBLIOGRÁFICA	19
1.2.1	Revisão de topologias com PFC ativo monofásicos	19
1.3	REVISÃO DE TÉCNICAS DE CONTROLES APLICADOS RETIFICADORES PFC MONOFÁSICOS	21
1.3.1	Controle por self-control (Auto-controle)	23
1.3.2	Controle por one-cycle-control	24
1.3.3	Controle Robusto baseado no Self Control	24
1.4	OBJETIVOS DA DISSERTAÇÃO	24
2	TOPOLOGIA DO CONVERTOR RETIFICADOR PROPOSTO	27
2.0.1	Definindo sinal de modulação	30
2.1	DIMENSIONAMENTO DOS ELEMENTOS PASSIVOS: INDUTOR	30
2.2	DIMENSIONAMENTO DOS ELEMENTOS PASSIVOS: CAPACITOR	33
2.3	ESFORÇOS NOS SEMICONDUTORES	34
2.3.1	Esforços nos transistores	35
2.3.1.1	Corrente RMS transistores S1 e S2	35
2.3.2	Esforços nos diodos retificadores	35
2.3.2.1	Corrente RMS Diodos Retificadores	35
2.3.2.2	Corrente Média Diodos Retificadores	36
2.3.2.3	Corrente RMS Diodos D_3 e D_4	36
2.3.2.4	Corrente Média Diodos D_3 e D_4	36
2.3.3	Máxima tensão de bloqueio nos semicondutores	37
2.3.3.1	Corrente RMS capacitor	37
2.3.3.2	Máxima tensão sobre capacitor	37
2.3.3.3	Corrente RMS Indutores	38
2.4	ESFORÇOS DOS SEMICONDUTORES EM RELAÇÃO AO ÍNDICE DE MODULAÇÃO	38
2.5	VALIDAÇÕES DAS EQUAÇÕES DOS ESFORÇOS	38
3	MODELAGEM DO RETIFICADOR PROPOSTO	41
3.1	DESCRIÇÃO DAS MATRIZES DO CONVERTOR	42
3.1.1	Matriz de impedância e tensão sobre os enrolamentos	42
3.1.2	Equações dos estados topológicos	43
3.1.2.1	Análise para o semiciclo positivo da rede	44

3.1.2.2	Análise para o semiciclo negativo da rede	45
3.1.3	Modelo médio	45
3.1.4	Ponto de equilíbrio	46
3.1.5	Linearização	47
3.1.6	Obtendo as funções de transferência do modelo linearizado	47
3.1.7	Validação do modelo comutado e modelo médio	48
3.2	MODELAGEM DA TENSÃO DE MODO COMUM	51
4	PROJETO DO CONVERSOR	55
4.1	PROJETO DO INDUTOR ACOPLADO	55
4.1.1	Definindo o núcleo	55
4.1.2	Dimensionando os condutores do indutor	57
4.1.3	Perdas no núcleo dos indutores	58
4.1.3.1	Perda de comutação	59
4.1.4	Perdas nos diodos	60
4.1.5	Rendimento do conversor	61
5	CONTROLE	63
5.1	ALGORITMO PLL	63
5.2	MODULAÇÃO DO CONVERSOR	64
5.3	ESTRATEGIA DE AMOSTRAGEM	65
5.4	RESULTADOS DE SIMULAÇÃO	66
6	RESULTADOS	69
6.1	RESULTADOS HIL	69
6.2	RESULTADOS EXPERIMENTAIS DO CONVERSOR PROPOSTO	71
7	CONCLUSÃO	75
7.1	TRABALHOS FUTUROS	76
	REFERÊNCIAS	77
	APÊNDICE A – APÊNDICE A	82
A.1	FILTROS SEGUNDA ORDEM DIGITAIS	82
A.1.1	Filtro passa baixa (LPF)	84
A.1.2	Filtro passa banda (BPF)	85
A.1.3	Filtro rejeita faixa (BRF)	87
A.1.4	Filtro rejeita faixa (HPF)	89
A.1.5	Biblioteca em C dos filtros discretizado	91
A.2	IMPLEMENTAÇÃO SOGI PLL	102
A.2.1	Filtro $G_1(s)$, gerador em fase	102
A.2.2	Filtro $G_2(s)$, gerador em quadratura	104

A.2.3	Biblioteca em C: SOGI PLL	105
A.3	ROTINA ISRPWM	116
A.4	ROTINA DE INTERRUÇÃO (ISR ADC)	117
A.5	ROTINA DE CONTROLE	120
A.6	ROTINA DE FILTROS CLA	121
	APÊNDICE B – APÊNDICE B	123
B.1	PROTÓTIPO CONVERSOR	123
	APÊNDICE C – APÊNDICE C	129
C.1	REFLOW-OVEN (FORNINHO)	129
C.2	TÉCNICA DE CONTROLE	130
C.3	SOFTWARE DE CONTROLE	131
C.4	APLICATIVOS USANDO PYTHON	131

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

O aumento de cargas não lineares trouxeram problemas quanto a qualidade de energia. Assim padrões internacionais como, IEC61000-3-2 e IEEE519, surgiram para normatização da qualidade da energia. Conversores de potência estáticos, cada vez mais vem sendo estudados e usados para correção de fator de potência de forma ativa. Com o avanço das tecnologias de semicondutores, os conversores cada vez mais são projetados com alta frequência de comutação, com objetivo de diminuir o tamanho dos magnéticos e por consequência diminuir o tamanho dos conversores.

Além disso, a cada ano surgem novas topologias, cada uma com alguma finalidade, como exemplo: minimizar tensão de modo comum, alto rendimento, boa distribuição de perdas, estruturas com isolação, entre outras. Assim esse trabalho, propõe uma estrutura nova para correção de fator de potência com baixa tensão de modo comum empregando semicondutores de *wide-band-gap*.

Por fim esse trabalho explora novas tecnologias em semicondutores de potência GaN (nitreto de gálio) para aplicação em eletrônica de potência, podendo operar em frequências de comutação maiores comparados às tecnologias atuais, assim podendo aumentar a densidade de potência dos conversores, reduzindo tamanho e custos.

1.2 REVISÃO BIBLIOGRÁFICA

1.2.1 Revisão de topologias com PFC ativo monofásicos

O aumento por demanda de energia com alto fator de potência e baixa distorção harmônica da corrente, exigidas pelas normas, como a IEC61000-3-2, criaram a necessidade do aumento de topologias de conversores com correção de fator de potência. A Figura 1, mostra a estrutura amplamente usada na industria, a estrutura é descrita em (RAO *et al.*, 2008a).

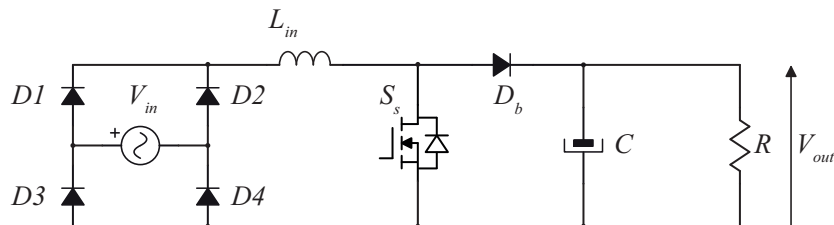


Figura 1 – Retificador PFC clássico, imagem de (HULIEHEL; LEE; CHO, 1992)

Apesar da simplicidade e robustez da estrutura da Figura 1, algumas aplicações necessitam de características que essa estrutura não possui, como exemplos:

- Isolação (SINGH *et al.*, 2011)
- Baixa tensão de modo comum (SIU; HO, 2016; GU *et al.*, 2013; MUNARETTO; HELDWEIN, Marcelo L., 2018; LANGE *et al.*, 2015)
- Alto rendimento (CHEN, Z.; DAVARI; WANG, 2020)
- Alta potência e boa distribuição de perdas (LANGE *et al.*, 2015).

O trabalho (SINGH *et al.*, 2011), mostra algumas famílias de conversores isolados obtidos a partir dos conversores buck, boost e buck-boost. Esses conversores com transformadores em alta frequência, tem bastante aplicação em fontes AC-DC de eletrônicos em geral, normalmente tendo aplicações em baixa potência, de fração de kW até alguns kW, como apresentado em (SINGH *et al.*, 2011).

As topologias bridgeless tem como principal características o alto rendimento, devido a troca dos diodos lentos por semicondutores controlados, como MOSFET e GaN. No caso específico do GaN se obtém a vantagem é ainda maior devido as perdas mínimas por recuperação reversa nas topologias totem-pole (LIU *et al.*, 2016).

Trabalhos recentes (CHEN, Z.; DAVARI; WANG, 2020) mostram metodologia para derivação de novas topologias de conversores bridgeless, baseado nas topologias básicas: buck, boost, buck-boost, cuk e Sepic, como pode ser visto na Figura 2. Ainda dentro dos conversores bridgeless o trabalho (LANGE *et al.*, 2015), propõem uma família de conversores bridgeless de 3 níveis, que podem ser justificadas em uso com potências mais elevadas.

O trabalho (SIU; HO, 2016), faz uma revisão de algumas topologias com mitigação da tensão de modo comum, as topologias são apresentadas na Figura 4. O trabalho considera somente as topologias PFC bridgeless.

O trabalho proposto na dissertação é baseado nas topologias (MUNARETTO; HELDWEIN, Marcelo L., 2018) e (GU *et al.*, 2013). Onde em (MUNARETTO; HELDWEIN, Marcelo L., 2018) é apresentando uma topologia inversora com baixa tensão de modo comum. Em (GU *et al.*, 2013), é apresentado uma topológica bidirecional onde são usados 4 indutores sendo 2 deles acoplados magneticamente em um único núcleo. Onde um núcleo é usado no semiciclo positivo da rede e outro no semiciclo negativo da rede, ver figure 6. No trabalho (MUNARETTO; HELDWEIN, Marcelo L., 2018) é empregado dois núcleos, como visto na Figura 5. A topologia proposta neste trabalho propõe o uso de 4 indutores acoplados em um único núcleo, a topologia será apresentado na próxima seção.

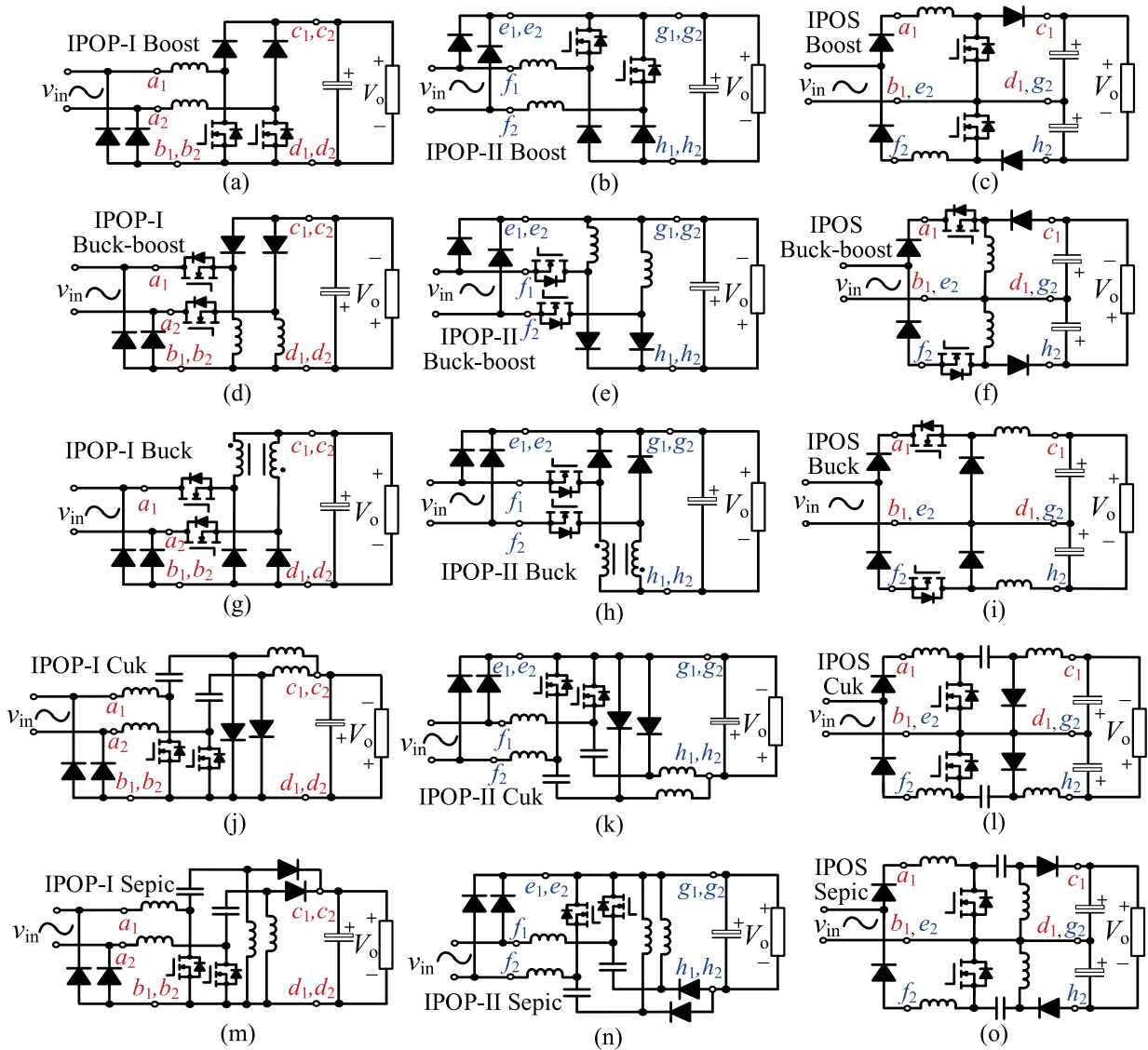


Figura 2 – Estrutura bridgeless derivadas das famílias de conversores: buck, boost, buck-boost, cuk e Sepic , imagem de (CHEN, Z.; DAVARI; WANG, 2020).

1.3 REVISÃO DE TÉCNICAS DE CONTROLES APLICADOS RETIFICADORES PFC MONOFÁSICOS

Controle por valores médios

A técnica de controle por valores médio já bem consolidada na literatura seu trabalhos relacionados desde os mais antigos até mais recentes com variações da versão clássica, pode ser visto no trabalhos (RAO *et al.*, 2008a; HULIEHEL; LEE; CHO, 1992; FIGUERES *et al.*, 2007; RAO *et al.*, 2008b; CHEN, H.-C., 2008). Para implementar o controle por valores médios são necessárias duas malhas de controle, uma de corrente e outra de tensão. A malha de

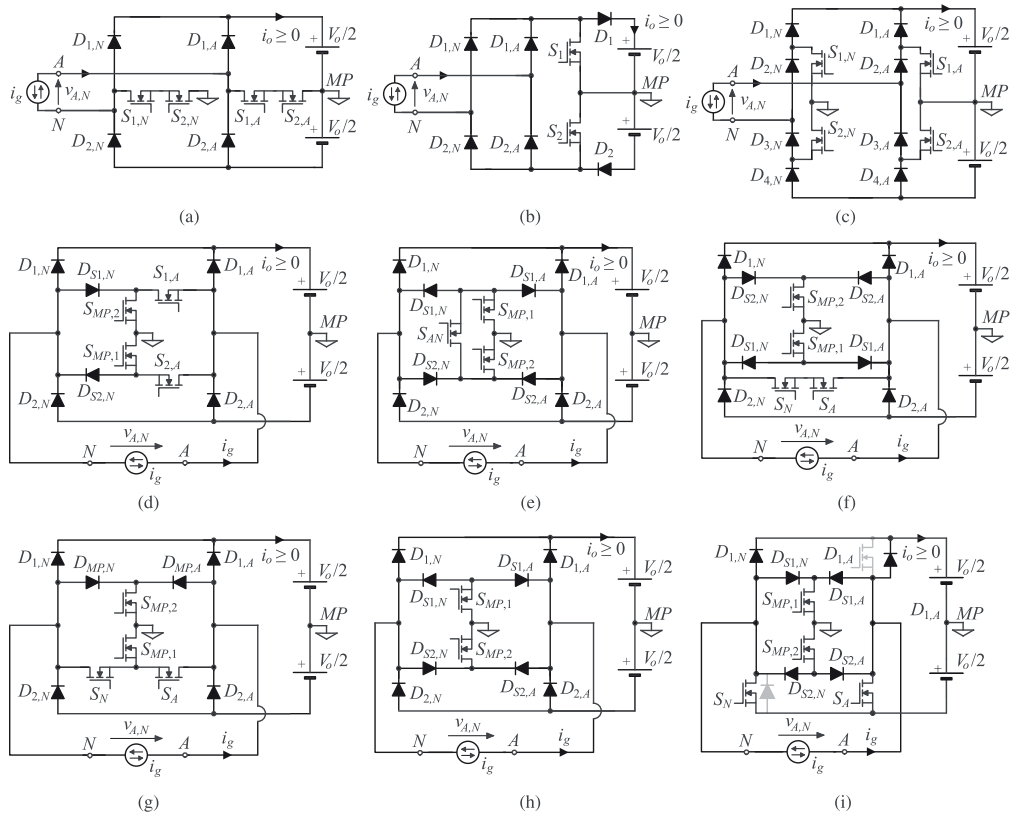


Figura 3 – Família de conversores bridgeless de 3 níveis, imagem de (LANGE *et al.*, 2015)

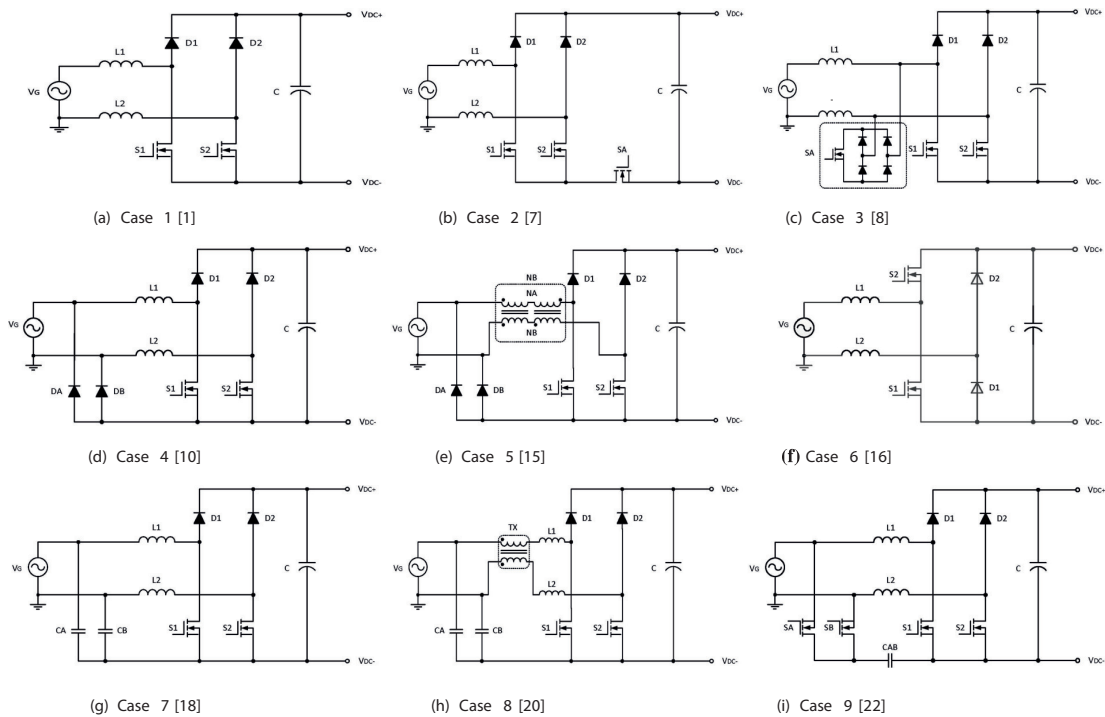


Figura 4 – Topologias com mitigação da tensão de modo comum, imagem de (SIU; HO, 2016)

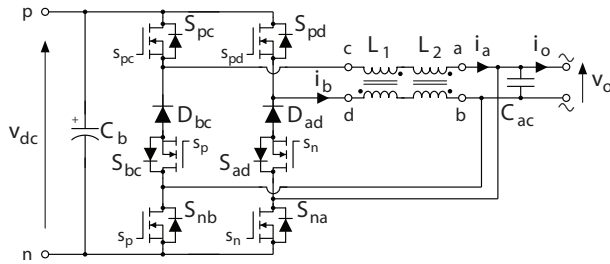


Figura 5 – Inversor com baixa tensão de modo comum, imagem de (MUNARETTO; HELDWEIN, Marcelo L., 2018)

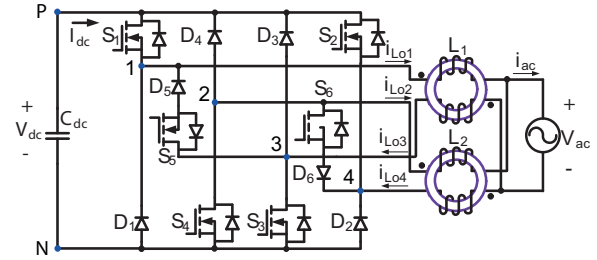


Figura 6 – Retificador com baixa tensão de modo comum, imagem de (GU et al., 2013)

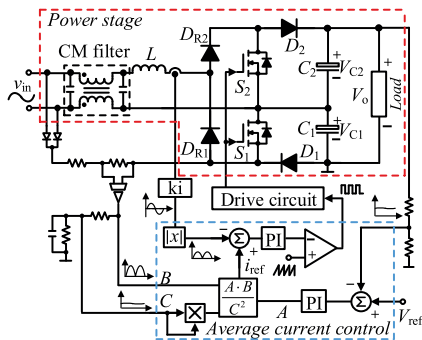


Figura 7 – Controle por valores médios, imagem de (CHEN, Z.; DAVARI; WANG, 2020).

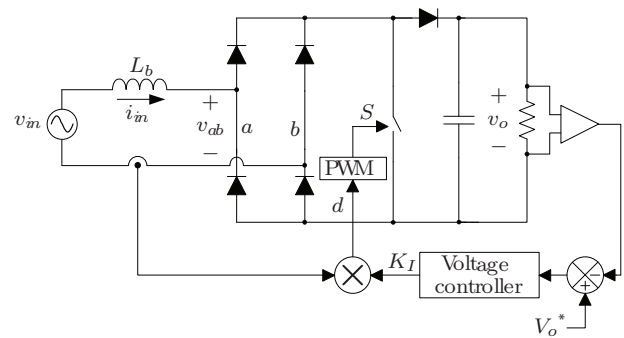


Figura 8 – Controle por self-control, imagem de (LANGE; HELDWEIN, Marcelo Lobo, 2011).

corrente tem a função de reproduzir no indutor uma corrente retificada de 120Hz sincronizada com a tensão de entrada, a fim de se obter fator de potência elevado na entrada do sistema. Para tal, é necessário que a malha de corrente seja rápida o suficiente para reproduzir a corrente sem grandes distorções. Já a malha de tensão tem o objetivo de manter a tensão de saída do conversor constante, ajustando-a quando ocorrerem variações de carga. Portanto, a malha de tensão deve ser suficientemente lenta para que exista um desacoplamento dinâmico em relação à malha de controle da corrente. A Figura 7 apresenta o esquema de controle do retificador.

1.3.1 Controle por self-control (Auto-controle)

A estratégia autocontrole é uma técnica mais recente, os trabalhos encontrados na literatura são (BORGONOVO, D. et al., 2005; LANGE; HELDWEIN, Marcelo Lobo, 2011; BORGONOVO, Deivis; MUSSA, Samir Ahmad, 2008; BORGONOVO, Deivis; HELDWEIN, Marcelo L.; MUSSA, Samir A., 2008). A estratégia de controle para o PFC usada é chamada de autocontrole (BORGONOVO, D. et al., 2005), este controle é muito similar ao controle

clássico por valores médios, a não ser por pequenas simplificações. A topologia do controle pode ser vista na Figura 8. Para a malha de corrente a estratégia consiste em definir o razão cíclica proporcional a amostras da corrente de entrada, assim essa estratégia remove a redundância do controle clássico onde a corrente de entrada e tensão são necessários. Algumas vantagens dessa estratégia comparada a clássica são melhor robustez e maior simplicidade de implementação, além de menor susceptibilidade a ruído e um start mais suave do conversor (BORGONOVO, D. *et al.*, 2005).

1.3.2 Controle por one-cycle-control

A técnica de controle OCC é recente e surgiu inicialmente usando moduladores analógicos. Seu uso com moduladores digitais são pouco abordado e muito pouco explorado. Os trabalhos mais relevantes relacionada a técnica OCC aplicado a PFC são mostrados em (JAPPE; MUSSA, Samir Ahmad, 2011; SCHWARZ, 1976; SMEDLEY; CUK, 1995; LAI; SMEDLEY, 1998; GHODKE *et al.*, 2009). A técnica OCC, ver Figura 9, consiste em “obrigar” a corrente de entrada a ser proporcional a tensão de entrada com objetivo de manter a impedância de entrada resistiva.

1.3.3 Controle Robusto baseado no Self Control

A técnica autocontrole tem muitos benefícios comparado ao controle clássico (valores médios), entretanto a técnica autocontrole torna-se instável operando com baixas cargas. A proposta dos trabalhos (JAPPE; MUSSA, Samir Ahmad; COUTINHO, 2011; ROSCHILD; COUTINHO; SOUZA, 2015) é usar teoria de Lyapunov, para aumentar o alcance de carga que o autocontrole consegue alcançar, porém nenhum desses trabalhos apresentaram resultados experimentais.

1.4 OBJETIVOS DA DISSERTAÇÃO

Este trabalho tem como contribuição realizar o estudo e modelagem da topologia retificado proposta. O trabalho propõem a implementação controle base em um FPGA/DSP com uma modulação de largura de pulso de alta resolução (HRPWM). E também o desenvolvimento de um protótipo operando com alta frequência de comutação, baseado em tecnologia GaN e SiC, a fim de validar o funcionamento do conversor. O trabalho é dividido em capítulos e cada capítulo aborda um tema.

O capítulo 2, explora a topologia de um conversor retificador, abordando a definição do sinal de modulação. Este capítulo foca no dimensionamento dos elementos passivos, como indutores e capacitores. Também discute os esforços aplicados aos semicondutores, tanto em

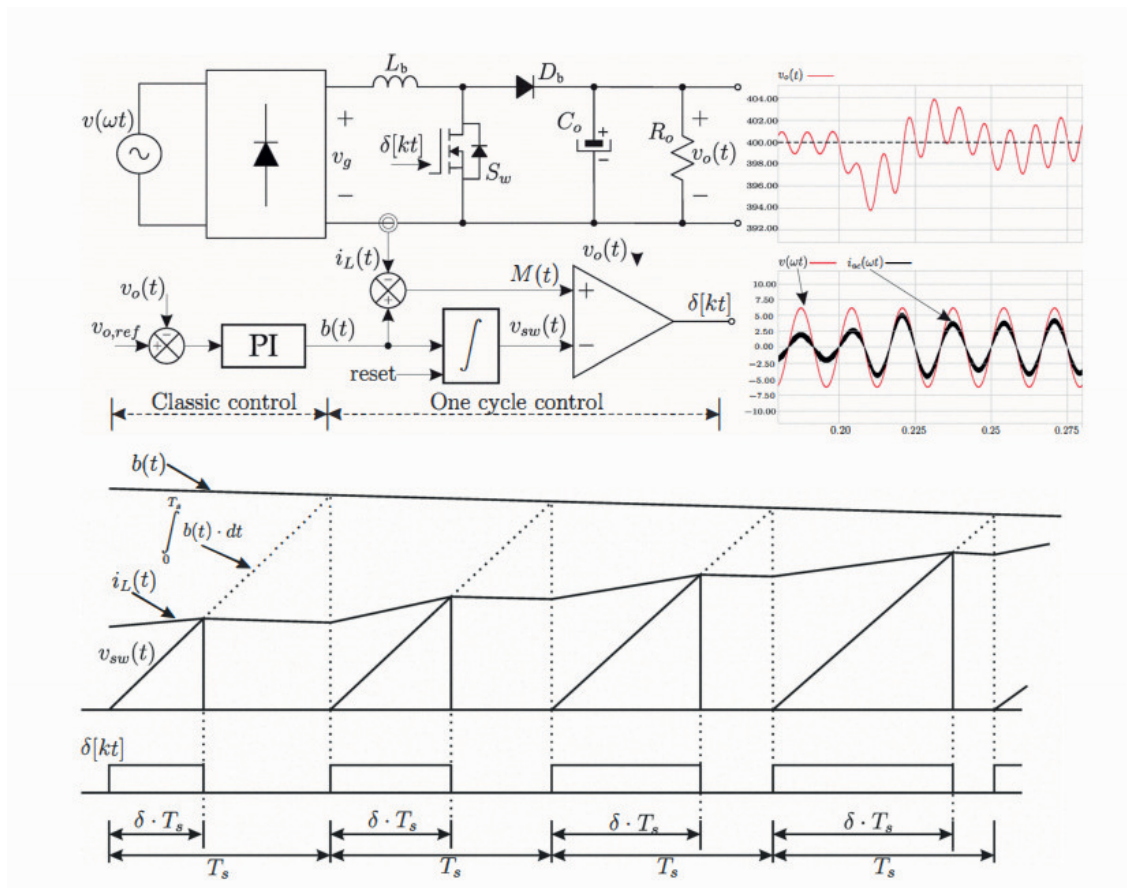


Figura 9 – Controle por one-cycle-control, imagem de (JAPPE; MUSSA, Samir Ahmad, 2011).

transistores quanto em diodos retificadores, e relaciona esses esforços ao índice de modulação. Conclui validando as equações de esforços.

O capítulo 3, dedica-se à modelagem do retificador proposto. Apresenta descrições das matrizes do conversor, abrangendo a matriz de impedância, equações dos estados topológicos, e análises semicíclicas da rede. A seção detalha o modelo médio, pontos de equilíbrio, linearização e a validação entre modelos. Finaliza com a modelagem da tensão de modo comum.

O capítulo 4, aborda o projeto do conversor, com foco especial no projeto do indutor acoplado. Detalha o processo de definição do núcleo, dimensionamento dos condutores, perdas no núcleo, perda de comutação, perdas em diodos e rendimento do conversor.

O capítulo 5, mostra os aspectos de controle do conversor retificador. Aborda o algoritmo PLL, a modulação do conversor, estratégias de amostragem e, por fim, mostra resultados de simulação.

O capítulo 6, apresenta os resultados obtidos, incluindo resultados HIL (Hardware In

Loop) e os resultados experimentais do conversor proposto em prática. Por fim, a conclusão, reflete sobre os resultados e descobertas do trabalho, resumindo as principais contribuições e concluindo a pesquisa. Também sugere possíveis trabalhos futuros na área.

2 TOPOLOGIA DO CONVERSOR RETIFICADOR PROPOSTO

O conversor proposto é apresentado na Figura 10. Na figura é apresentado quatro variações da topologia, a topologia consiste em quatro indutores, dois interruptores e seis diodos. Na topologia os diodos retificadores (D_1 , D_2 , D_5 e D_6 .) operam em alta frequência. E cada interruptor possui um diodo em serie para prevenir condução reversa no ciclo da rede em qual não está operando.

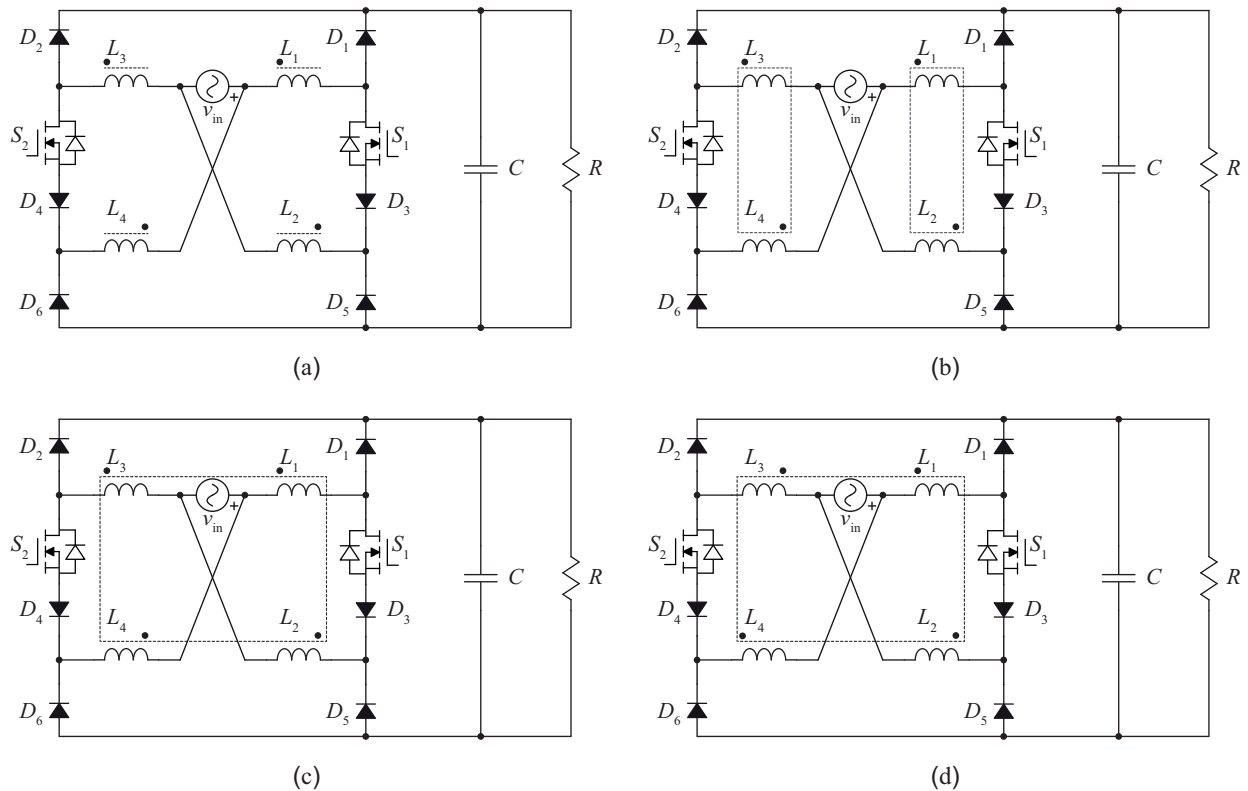


Figura 10 – Conversor retificador proposto.

Na Figura 10 (a) representa a topologia com os quatro indutores desacoplados entre si. Na Figura 10 (b) representa a topologia os indutores acoplados dois a dois, sendo que L_1 , L_2 , operam no semiciclo positivo e L_3 , L_4 no semiciclo negativo, essa topologia é apresentada em (GU *et al.*, 2013). Na Figura 10 (c) e (d) representa a topologia com todos os indutores acoplados, sendo a diferença entre eles, a orientação dos enrolamentos. Em (c) é orientado de tal forma que o fluxo magnético no indutor é positivo no semiciclo positivo e negativo no semiciclo negativo, já em (d) o fluxo magnético é sempre positivo.

Para entender o funcionamento é apresentado as etapas de operação na Figura 11. O conversor possui seis etapas de operação sendo duas delas no modo de condução descontinuo, que não serão apresentados nesse trabalho, pois o projeto do conversor foi realizado

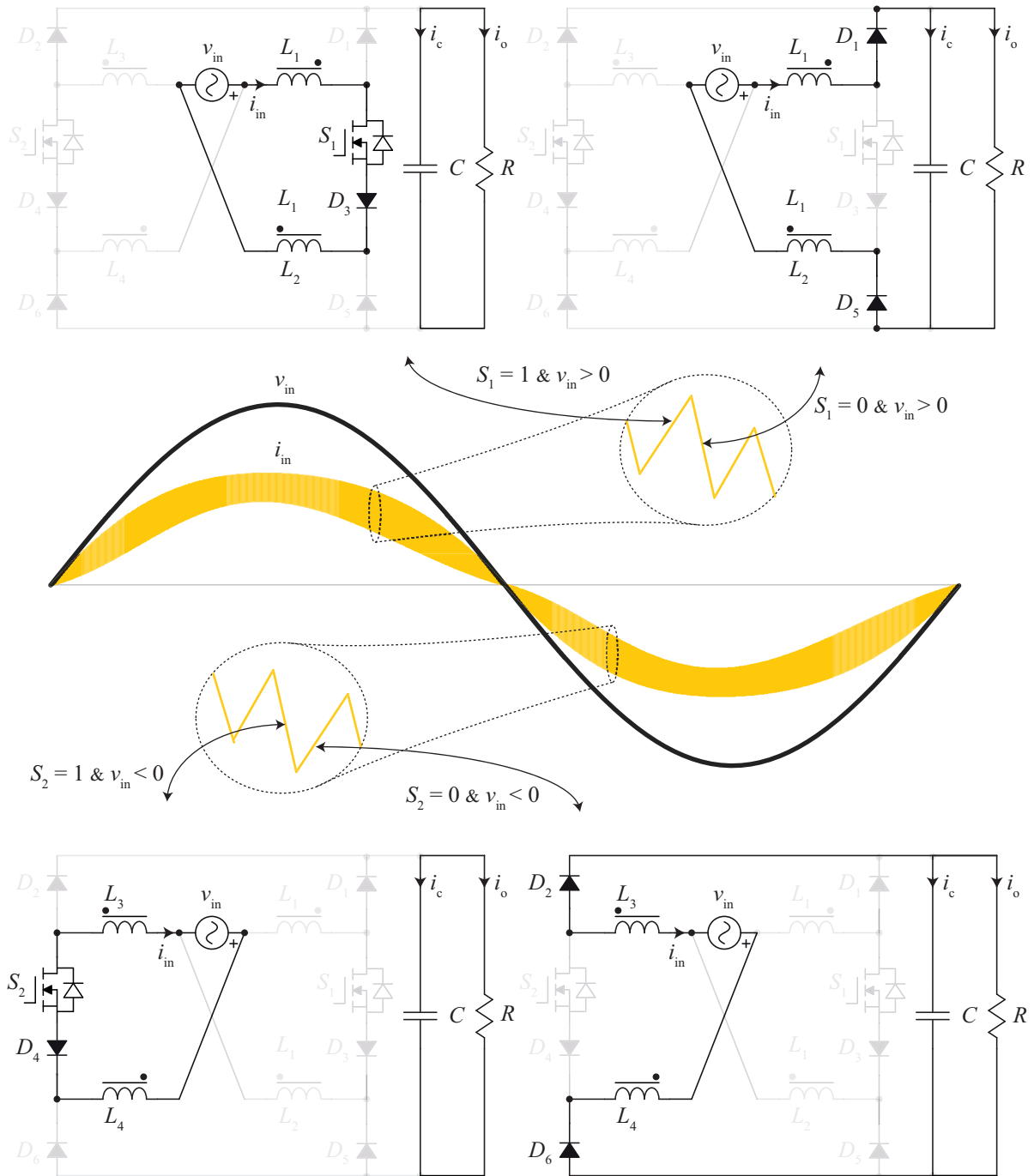


Figura 11 – Etapas de operação do conversor proposto. Na parte superior é apresentados as etapas para o semiciclo positivo e na inferior para o semiciclo negativo

considerado modo de condução contínuo. Na Figura 11 parte superior é possível visualizar as duas etapas de operação para o semiciclo positivo, na primeira quando S_1 é igual a 1, o interruptor conduz e armazena energia nos indutores, e nas segunda etapa quando S_1 é

igual a 0 a energia é transferida para a saída. O mesmo ocorre no semiciclo negativo com o interruptor S_2 como pode ser visto na parte inferior do gráfico.

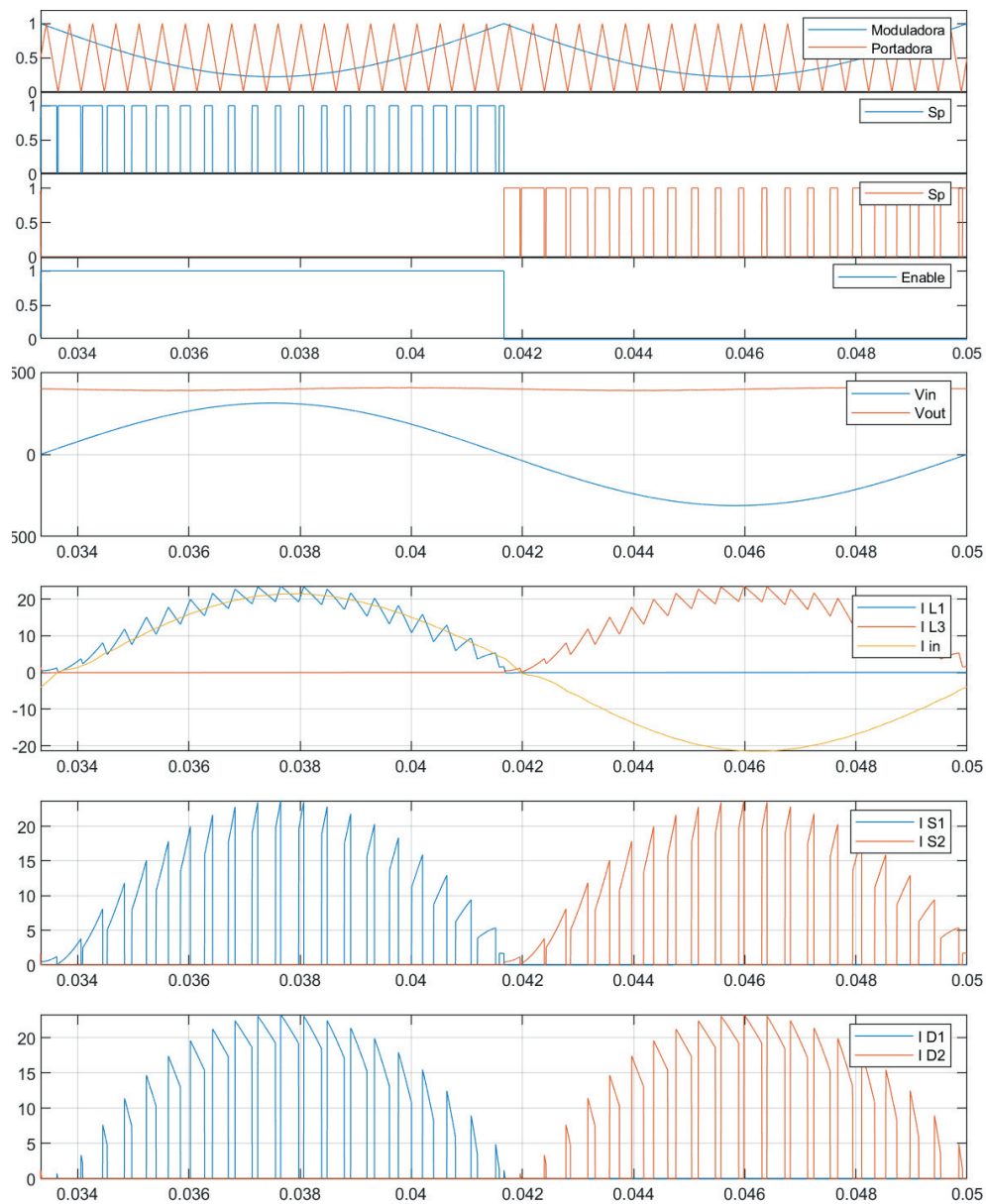


Figura 12 – Principais forma de onda do conversor proposto. O primeiro gráfico exibe as formas de onda da moduladora e da portadora. O segundo o sinal PWM do interruptor S_1 enquanto o terceiro apresenta o sinal PWM do interruptor S_2 . O quinto gráfico apresenta as tensões de entrada e saída do conversor. O sexto exibe as correntes nos indutores L_1 e L_3 do conversor, bem como a corrente de entrada. O sétimo apresenta as correntes nos diodos S_1 e S_2 do conversor. Por fim, o oitavo exibe as correntes nos diodos D_1 e D_2 do conversor.

Na Figura 12, pode ser visto algumas das principais formas de onda do conversor. O primeiro gráfico exibe as formas de onda da moduladora e da portadora. O segundo o sinal PWM do interruptor S_1 enquanto o terceiro apresenta o sinal PWM do interruptor S_2 . O quinto gráfico apresenta as tensões de entrada e saída do conversor. O sexto exibe as correntes nos indutores L_1 e L_3 do conversor, bem como a corrente de entrada. O sétimo apresenta as correntes nos diodos S_1 e S_2 do conversor. Por fim, o oitavo exibe as correntes nos diodos D_1 e D_2 do conversor.

2.0.1 Definindo sinal de modulação

Para o conversor Boost operando em modo de condução contínua CCM o ganho estático é dado por:

$$\frac{V_o}{V_{in}} = \frac{1}{1-D} \quad (1)$$

No caso de um conversor CC-CC onde a tensão de entrada é constante, com um valor de razão cíclica fixo obtém-se uma tensão constante na saída. No caso do PFC a tensão de entrada do conversor é a rede elétrica e possui formato senoidal logo sua tensão é dada por:

$$v_{in} = V_p \sin(\omega t) \quad (2)$$

onde V_p é a tensão de pico da rede. Substituindo (2) em (1) obtemos:

$$d(\omega t) = 1 - \frac{V_p}{V_o} \sin(\omega t) \quad (3)$$

e o índice de modulação é definido como:

$$\alpha = \frac{V_p}{V_o}. \quad (4)$$

É possível notar que para cada instante de tempo existe um valor de modulação $d(\omega t)$ para que a tensão de saída se mantenha constante, como pode ser visto na Figura 13.

2.1 DIMENSIONAMENTO DOS ELEMENTOS PASSIVOS: INDUTOR

Para projetar o indutor temos as duas componentes principais na corrente, uma em baixa frequência referente a tensão retificada, 120 Hz, e outra referente a frequência de comutação que possui componentes em alta frequência. Deve-se levar em conta a máxima ondulação em alta frequência desejada para o projeto do indutor.

A tensão retificada sobre o indutor L_{in} é dada por:

$$v_{in} = V_p \sin(\omega t) \quad (5)$$

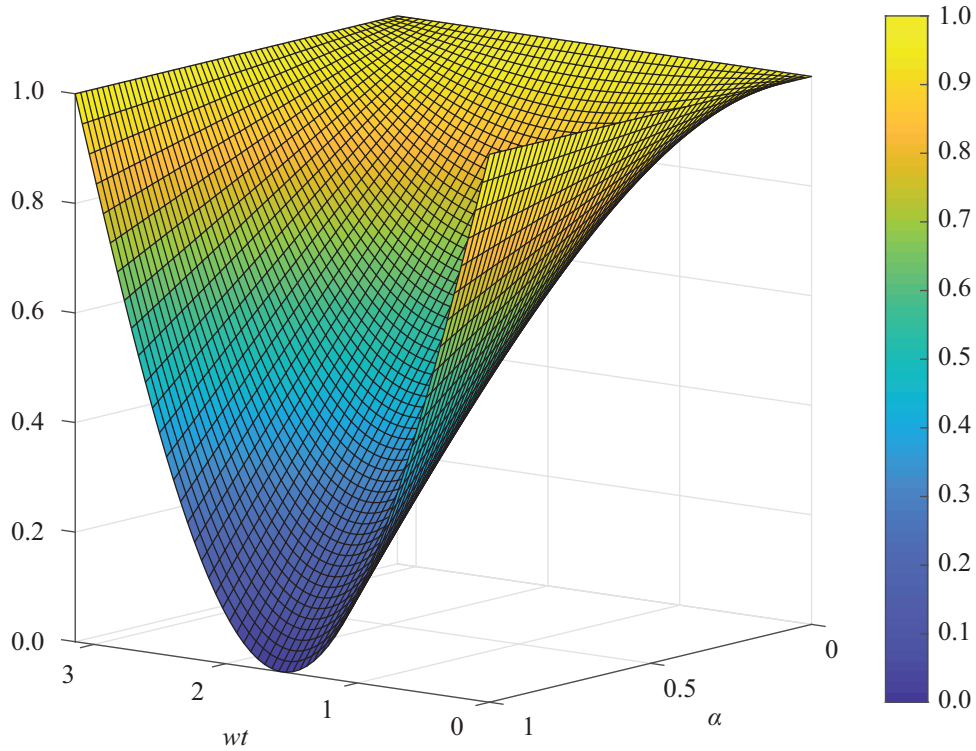


Figura 13 – Valor de modulação em função do tempo e do índice do modulação α .

onde L_{in} é igual a:

$$L_{in} = \begin{cases} L_1 + L_2, & v_{in} \geq 0 \\ L_3 + L_4, & v_{in} < 0 \end{cases} \quad (6)$$

Quando o interruptor S_1 está em condução tem-se a tensão v_{in} sobre o indutor L_{in} e a partir da equação volt-amper dado pela Equação (7)

$$v_{in} = L_{in} \frac{di_L}{dt} \quad (7)$$

Para um período de comutação o tempo de condução do interruptor S_1 é o próprio valor da razão cíclica multiplicado pelo período de comutação T_S .

$$\Delta t = d(wt) T_S \quad (8)$$

Para o período de comutação a Equação (7) pode ser reescrita como:

$$v_{in} = L_{in} \frac{\Delta I_L}{\Delta t} \quad (9)$$

Substituindo (5) e (8) em (9) tem-se:

$$d(wt)\sin(wt) = \frac{\Delta I_L L_{in}}{V_p T_s} \quad (10)$$

Substituindo a função da razão cíclica demonstrada em (3) em (10).

$$\sin(wt) - \frac{V_p}{V_o} \sin^2(wt) = \frac{\Delta I_L L_{in}}{V_p T_s} \quad (11)$$

O termo esquerdo da igualdade pode ser chamado de corrente normalizada, dado por $\overline{\Delta I_L}$

$$\overline{\Delta I_L} = \sin(wt) - \frac{V_p}{V_o} \sin^2(wt) \quad (12)$$

Assim como $D(\theta)$ tem dependência com a relação de parâmetro de projeto $\frac{V_p}{V_o}$ a corrente normalizada $\overline{\Delta I_L}$ também tem influência, a Figura 13 mostra a influência na ondulação para diferentes valores de $\frac{V_p}{V_o}$.

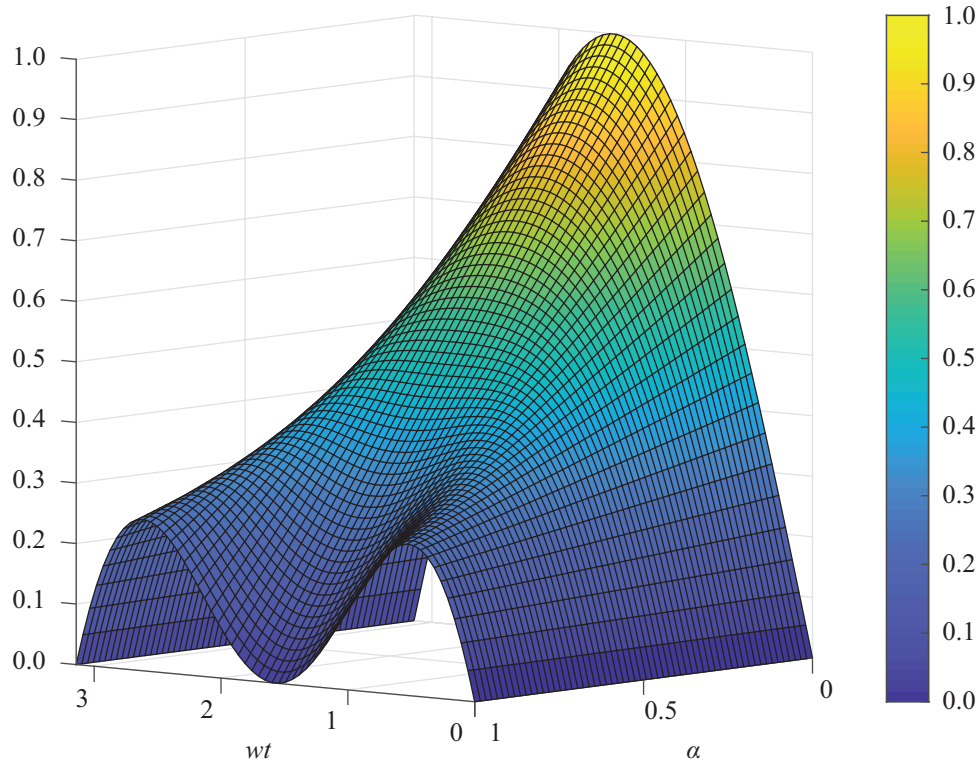


Figura 14 – Corrente normaliza em função do tempo e do índice do modulação α .

Assim, determina-se a indutância para o máximo valor da corrente normalizada $\overline{\Delta I_{L_{MAX}}}$

$$L_{in} = \frac{\overline{\Delta I_{L_{MAX}}} V_p}{\Delta I_L f_s} \quad (13)$$

2.2 DIMENSIONAMENTO DOS ELEMENTOS PASSIVOS: CAPACITOR

Uma das principais preocupações ao projetar um conversor PFC monofásico é minimizar a ondulação de 120Hz na saída do barramento DC. Essa ondulação tem origem na variação da corrente de entrada da rede elétrica e a capacidade limitada do capacitor do barramento. A redução da ondulação é importante para garantir que a saída DC possua uma ondulação aceitável, o que pode ser crítico para muitos sistemas eletrônicos.

$$C = \frac{P_o}{2\pi F_r V_o \Delta V_o} \quad (14)$$

Onde P_o é a potência de saída do sistema, F_r é a frequência da rede elétrica, V_o é a tensão de saída e ΔV é a variação de tensão permitida no barramento.

A ondulação percentual no capacitor é dada por:

$$\Delta V_o(\%) = 100 \frac{P_o}{2\pi F_r C V_o^2}$$

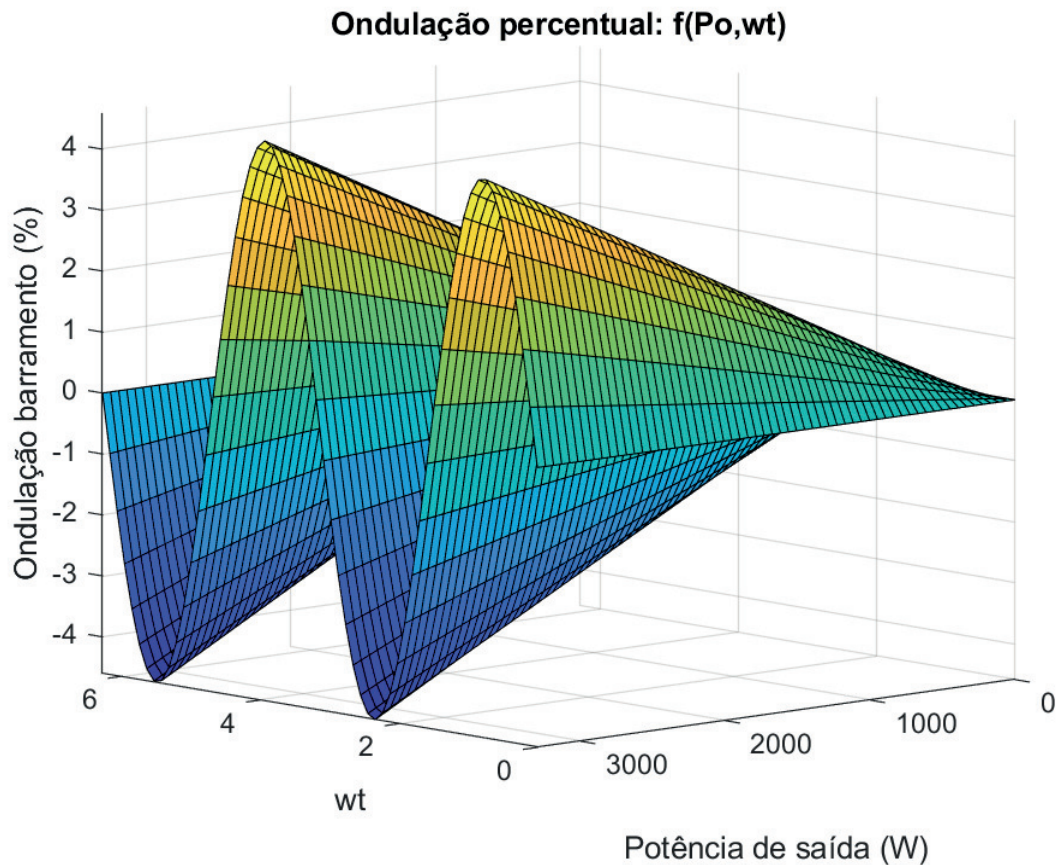


Figura 15 – Ondulação de tensão em função do tempo e da potência.

2.3 ESFORÇOS NOS SEMICONDUTORES

Para obter a corrente RMS nos componentes, a corrente RMS é calculada primeiro no período de comutação:

$$\langle I_x \rangle_{T_s}^{RMS} = \sqrt{\frac{1}{T_s} \int_0^{T_s} [i_x]^2 dt_s} \quad (15)$$

em sequencia no período da rede elétrica:

$$\langle I_x \rangle_{T_r}^{RMS} = \sqrt{\frac{1}{T_r} \int_0^{T_r} [\langle I_x \rangle_{T_s}^{RMS}]^2 dt_r} \quad (16)$$

Como pode ser visto na Figura 16. Para facilitar a análise dos esforços, será adotado que os

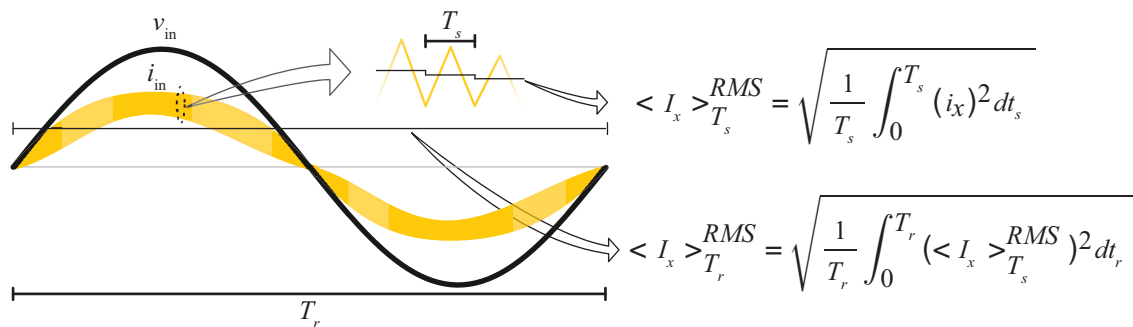


Figura 16 – Cálculo de corrente RMS para o período da rede e no período de comutação.

componentes assumem o valor médio, no período de comutação, da corrente do indutor:

$$i_{in} = I_p \sin(\omega t) \quad (17)$$

Para obter as equação com parametros em função de variáveis de projeto, substitui-se a corrente de pico em função de potencia. Sabe-se que desprezando as perdas nos componetes e considerando um alto fator de potencia tem-se:

$$\begin{aligned} P_o &= \langle I_{in} \rangle_{T_r}^{RMS} \langle V_{in} \rangle_{T_r}^{RMS} \\ &= \frac{V_p I_p}{2} \end{aligned} \quad (18)$$

e a corrente é representado por:

$$I_p = \frac{2P_o}{V_p} \quad (19)$$

2.3.1 Esforços nos transistores

2.3.1.1 Corrente RMS transistores S1 e S2

Aplicando as equações (15) e (16), obtém-se a corrente RMS no transistor:

$$\langle I_S \rangle_{T_s}^{RMS} = \sqrt{\frac{1}{T_s} \int_0^{dT_s} [I_p \sin(wt)]^2 dt_s} \quad (20)$$

resultando em,

$$\langle I_S \rangle_{T_s}^{RMS} = I_p \sin(wt) \sqrt{1 - \alpha \sin(wt)} \quad (21)$$

Aplicando a equação (21) em (16):

$$\langle I_x \rangle_{T_r}^{RMS} = \sqrt{\frac{1}{2\pi} \int_0^\pi [I_p \sin(wt) \sqrt{1 - \alpha \sin(wt)}]^2 dt_r} \quad (22)$$

resultando em,

$$\langle I_S \rangle_{T_r}^{RMS} = \frac{2P_o}{V_p} \sqrt{\frac{3\pi - 8\alpha}{12\pi}} \quad (23)$$

2.3.2 Esforços nos diodos retificadores

2.3.2.1 Corrente RMS Diodos Retificadores

Aplicando as Equações (15) e (16), a corrente RMS no diodo é:

$$\langle I_{D_1} \rangle_{T_s}^{RMS} = \sqrt{\frac{1}{T_s} \int_{dT_s}^{T_s} [I_p \sin(wt)]^2 dt_s} \quad (24)$$

resultando em,

$$\langle I_S \rangle_{T_s}^{RMS} = I_p \sin(wt) \sqrt{\frac{2\alpha}{3\pi}} \quad (25)$$

Aplicando a Equação (24) em (16):

$$\langle I_{D_1} \rangle_{T_r}^{RMS} = \sqrt{\frac{1}{2\pi} \int_0^\pi [I_p \sin(wt) \sqrt{\frac{2\alpha}{3\pi}}]^2 dt_r} \quad (26)$$

resultando em,

$$\langle I_{D_1} \rangle_{T_r}^{RMS} = \frac{P_o \alpha}{2V_p} \quad (27)$$

2.3.2.2 Corrente Média Diodos Retificadores

Aplicando o mesmo procedimento utilizado para obter a corrente corrente RMS, obtemos a corrente media no período de comutação:

$$\langle I_{D_1} \rangle_{T_s}^{Mean} = \frac{1}{T_s} \int_{dT_s}^{T_s} I_p \sin(wt) dt_s \quad (28)$$

resultando em,

$$\langle I_{D_3} \rangle_{T_s}^{Mean} = I_p \sin(wt) [\alpha \sin(wt)] \quad (29)$$

Para o período da rede:

$$\langle I_{D_3} \rangle_{T_r}^{Mean} = \frac{1}{2\pi} \int_0^\pi [I_p \sin(wt) \sqrt{\frac{2\alpha}{3\pi}}]^2 dt_r \quad (30)$$

$$\langle I_{D_3} \rangle_{T_r}^{Mean} = \frac{P_o \alpha}{2V_p} \quad (31)$$

2.3.2.3 Corrente RMS Diodos D_3 e D_4

A corrente RMS nos diodo D_3 e D_4 são as mesmas que as dos transistores S1 e S1 já que estão ligado em série:

$$\langle I_{D_3} \rangle_{T_s}^{RMS} = I_p \sin(wt) \sqrt{1 - \alpha \sin(wt)} \quad (32)$$

e

$$\langle I_{D_3} \rangle_{T_r}^{RMS} = \frac{2P_o}{V_p} \sqrt{\frac{3\pi - 8\alpha}{12\pi}} \quad (33)$$

2.3.2.4 Corrente Média Diodos D_3 e D_4

Aplicando o mesmo procedimento utilizado para obter a corrente corrente RMS, obtemos a corrente media no período de comutação:

$$\langle I_{D_3} \rangle_{T_s}^{Mean} = \frac{1}{T_s} \int_0^{dT_s} I_p \sin(wt) dt_s \quad (34)$$

resultando em,

$$\langle I_{D_3} \rangle_{T_s}^{Mean} = I_p \sin(wt) [1 - \alpha \sin(wt)] \quad (35)$$

Para o período da rede:

$$\langle I_{D_3} \rangle_{T_r}^{Mean} = \frac{1}{2\pi} \int_0^\pi I_p \sin(wt) [1 - \alpha \sin(wt)] dt_r \quad (36)$$

$$\langle I_{D_3} \rangle_{T_r}^{Mean} = \frac{P_o}{2V_p} \left(\frac{4 - \alpha\pi}{4\pi} \right) \quad (37)$$

2.3.3 Máxima tensão de bloqueio nos semicondutores

Para os semicondutores é definido como a máxima tensão a tensão do barramento. E é definida pela Equação (38).

$$V_{D_{1,max}} = V_o + \frac{\Delta V_o}{2} \quad (38)$$

Como existe elementos em série durante a comutação, na pratica não podemos definir a tensão em cada elementos devido as incertezas dos elementos parasitas. Um semiconductor pode assumir toda a tensão de bloquei durante a comutação enquanto outro assume zero, dessa forma garantimos que todos os componestes suporte a máxima tensão do barramento de tensão.

2.3.3.1 Corrente RMS capacitor

Primeiramente para o capacitor de barramento utilizando a Equação (15), para o semiciclo positivo quando S1 estão conduzindo a corrente pelo capacitor é I_o e quando S1 está bloqueado a corrente é $I_o - I_{D_1}$, assim para o período de comutação tem-se:

$$\langle I_C \rangle_{T_s}^{RMS} = \sqrt{\frac{1}{T_s} \left(\int_0^{dT_s} (I_p \sin(\omega t))^2 dt_s + \int_{dT_s}^{T_s} (I_o - I_p \sin(\omega t))^2 dt_s \right)} \quad (39)$$

resolvendo,

$$\langle I_C \rangle_{T_s}^{RMS} = \frac{P_o}{V_o} \sqrt{1 - 4 \sin^2(\omega t) \left(1 - \frac{\sin(\omega t)}{\alpha} \right)} \quad (40)$$

para o período da rede aplica-se ((16))

$$\langle I_C \rangle_{T_r}^{RMS} = \sqrt{\frac{1}{\pi} \int_0^{\pi} \frac{P_o^2}{V_o^2} \left(1 - 4 \sin^2(\omega t) \left(1 - \frac{\sin(\omega t)}{\alpha} \right) \right) d\omega t} \quad (41)$$

simplificando

$$\langle I_C \rangle_{T_r}^{RMS} = \frac{P_o}{V_o} \sqrt{\frac{16 - 3\alpha\pi}{3\alpha\pi}} \quad (42)$$

2.3.3.2 Máxima tensão sobre capacitor

A máxima tensão aplicada sobre o capacitor de barramento é dado pelo valore médio da tensão mais metade da ondulação de tensão como mostra a Equação ((43))

$$V_{C_{max}} = V_o + \frac{\Delta V_o}{2} \quad (43)$$

2.3.3.3 Corrente RMS Indutores

Para os indutores L1, L2, L3 e L4 os esforços de corrente são os mesmos, assim a corrente no período de comutação para indutor L1:

$$\langle I_{L1} \rangle_{T_s}^{RMS} = I_p \sin(\omega t) \quad (44)$$

para o período da rede aplica-se ((16))

$$\langle I_{L1} \rangle_{T_r}^{RMS} = \sqrt{\frac{1}{2\pi} \int_0^\pi (I_p \sin(\omega t))^2 d\omega t} \quad (45)$$

simplificando

$$\langle I_{L1} \rangle_{T_r}^{RMS} = \frac{P_o}{V_p} \quad (46)$$

2.4 ESFORÇOS DOS SEMICONDUTORES EM RELAÇÃO AO ÍNDICE DE MODULAÇÃO

Conforme apresentado na Tabela 2, que exhibe as equações de corrente RMS e média baseadas nos parâmetros de projeto do conversor, é possível representar a corrente normalizada em função do índice de modulação α . Essa representação pode ser visualizada na Figura 17. A corrente representada pelo interruptor S_1 é válida também para o diodo série.

2.5 VALIDAÇÕES DAS EQUAÇÕES DOS ESFORÇOS

Com o objetivo de validar as equações propostas, utilizou-se simulações para comparações. Para as simulações foram usados os parâmetros do conversor mostrados na Tabela 1.

Tabela 1 – Especificações dos parâmetros para cálculo e simulação.

Grandeza	Valor
P_o	3300 W
V_o	380 V
V_p	320 V
f_s	375 kHz
f_r	60 Hz
ΔI_L	30 %
ΔV_o	5 %

Com os parâmetros apresentados em 1, utilizando as equações teóricas, resumidas na Tabela 2 é possível determinar os valores teóricos e simulados para comparação, onde os resultados são apresentados em 3.

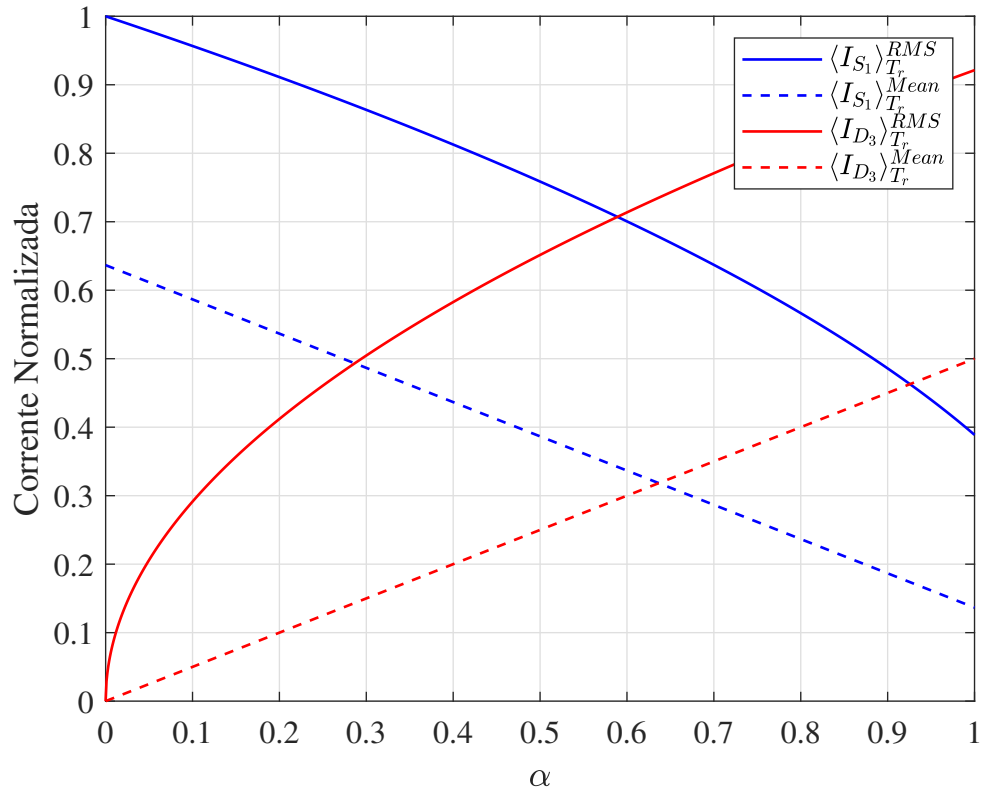


Figura 17 – Corrente média e RMS dos semicondutores normaliza, demonstradas em função do índice de modulação α

Tabela 2 – Equações dos esforços dos semicondutores do conversor.

Descrição	Equação
Corrente RMS transistores	$\langle I_S \rangle_{T_r}^{RMS} = \frac{2P_o}{V_p} \sqrt{\frac{3\pi - 8\alpha}{12\pi}}$
Corrente RMS diodos retificadores	$\langle I_{D_1} \rangle_{T_r}^{RMS} = \frac{2P_o}{V_{ip}} \sqrt{\frac{2\alpha}{3\pi}}$
Corrente média diodos retificadores	$\langle I_{D_1} \rangle_{T_r}^{Mean} = \frac{P_o\alpha}{2V_{ip}}$
Corrente RMS diodos série	$\langle I_{D_3} \rangle_{T_r}^{RMS} = \frac{2P_o}{V_p} \sqrt{\frac{3\pi - 8\alpha}{12\pi}}$
Corrente média diodos série	$\langle I_{D_3} \rangle_{T_r}^{Mean} = \frac{P_o}{2V_p} \left(\frac{4 - \alpha\pi}{4\pi} \right)$
Corrente RMS capacitor	$\langle I_C \rangle_{T_r}^{RMS} = \frac{P_o}{V_o} \sqrt{\frac{16 - 3\alpha\pi}{3\alpha\pi}}$

Tabela 3 – Especificações dos parâmetros para cálculo e simulação.

Grandeza	Teórico	Simulado	Erro (%)
$\langle I_S \rangle_{T_r}^{RMS}$	5.507 A	5.532 A	0.448
$\langle I_S \rangle_{T_r}^{Mean}$	2.223	2.212 A	0.496
$\langle I_{D_1} \rangle_{T_r}^{RMS}$	5.507 A	5.532 A	0.448
$\langle I_{D_1} \rangle_{T_r}^{Mean}$	2.223	2.212 A	0.496
$\langle I_{D_3} \rangle_{T_r}^{RMS}$	8.849	8.718	1.493
$\langle I_{D_3} \rangle_{T_r}^{RMS}$	4.418	4.342	1.747
$\langle I_C \rangle_{T_r}^{RMS}$	8.753	8.994	2.750

3 MODELAGEM DO RETIFICADOR PROPOSTO

A modelagem de um conversor PFC (Power Factor Correction) é um processo fundamental no desenvolvimento de sistemas de alimentação de energia, já que não é possível operar sem um controle em malha fechada no caso do conversor operando em modo de condução contínua. A modelagem adequada permite o desenvolvimento de estratégias de controle e a otimização do desempenho dinâmico do conversor PFC.

Baseado nas etapas de operação descritas no capítulo anterior será descrito as equações por espaço de estados do conversor com os 4 indutores acoplados.

A modelagem por espaço de estados é uma técnica comum e poderosa para modelar sistemas dinâmicos, incluindo sistemas elétricos, mecânicos, térmicos e de controle. Nesta técnica, o sistema é descrito por meio de equações diferenciais ou diferenças em termos de variáveis de estado, entradas e saídas. Essas equações descrevem como as variáveis de estado mudam ao longo do tempo em resposta às entradas e ao estado atual do sistema. O modelo é descrito por:

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)\end{aligned}\tag{47}$$

onde,

- $\mathbf{y}(t)$ - o vetor das variáveis de estado de tamanho n
- $\mathbf{y}(t)$ - as saídas desejadas de tamanho q
- $\dot{\mathbf{x}}(t)$ - o vetor da derivada das variáveis de estado de tamanho n
- $\mathbf{u}(t)$ - as entradas do modelo p
- \mathbf{A} - Matriz de estados tamanho $n \times n$
- \mathbf{B} - Matriz de entrada tamanho $n \times p$
- \mathbf{C} - Matriz de saída tamanho $q \times n$
- \mathbf{D} - Matriz de realimentação tamanho $q \times p$

O modelo de espaço de estados é útil para analisar e projetar sistemas de controle, uma vez que permite que o sistema seja representado de forma compacta. Além disso, o modelo de espaço de estados pode ser facilmente implementado em software para simulação e projeto de controle em tempo real, o qual foi o caso desse trabalho.

3.1 DESCRIÇÃO DAS MATRIZES DO CONVERSOR

3.1.1 Matriz de impedância e tensão sobre os enrolamentos

A matriz de impedância do indutor acoplado pode ser descrita por:

$$\mathbf{v}_L = \begin{bmatrix} L_1 & M_{12} & M_{13} & M_{14} \\ M_{21} & L_2 & M_{23} & M_{24} \\ M_{31} & M_{32} & L_3 & M_{34} \\ M_{41} & M_{42} & M_{43} & L_4 \end{bmatrix} \quad (48)$$

onde,

- L_1, L_2, L_3, L_4 - São as indutâncias próprias de cada enrolamento
- M_{xy} - São as indutâncias mutuas referente x para y

Para o semiciclo positivo sabe-se que corrente circula somente em L_1 e L_2 :

$$\begin{bmatrix} i_{L1}(t) \\ i_{L2}(t) \\ 0 \\ 0 \end{bmatrix}, \forall x > 0 \quad (49)$$

no caso do semiciclo negativos em L_3 e L_4 :

$$\begin{bmatrix} 0 \\ 0 \\ i_{L3}(t) \\ i_{L4}(t) \end{bmatrix}, \forall x < 0 \quad (50)$$

e as tensões sobre os enrolamentos é dado por:

$$\mathbf{v}_L = \mathbf{z}_L \cdot \begin{bmatrix} \dot{i}_{L1} \\ \dot{i}_{L2} \\ \dot{i}_{L3} \\ \dot{i}_{L4} \end{bmatrix} = \begin{bmatrix} L_1 \dot{i}_{L1} + M_{12} \dot{i}_{L2} + M_{13} \dot{i}_{L3} + M_{14} \dot{i}_{L4} \\ M_{21} \dot{i}_{L1} + L_2 \dot{i}_{L2} + M_{23} \dot{i}_{L3} + M_{24} \dot{i}_{L4} \\ M_{31} \dot{i}_{L1} + M_{32} \dot{i}_{L2} + L_3 \dot{i}_{L3} + M_{34} \dot{i}_{L4} \\ M_{41} \dot{i}_{L1} + M_{42} \dot{i}_{L2} + M_{43} \dot{i}_{L3} + L_4 \dot{i}_{L4} \end{bmatrix} \quad (51)$$

Para o semiciclo positivo da rede tem-se:

$$\mathbf{v}_{Lp} = \begin{bmatrix} L_1 \dot{i}_{L1} + M_{12} \dot{i}_{L2} \\ M_{21} \dot{i}_{L1} + L_2 \dot{i}_{L2} \\ M_{31} \dot{i}_{L1} + M_{32} \dot{i}_{L2} \\ M_{41} \dot{i}_{L1} + M_{42} \dot{i}_{L2} \end{bmatrix} \quad (52)$$

e para o semiciclo negativo da rede tem-se:

$$\mathbf{v}_{Ln} = \begin{bmatrix} M_{13}\dot{i}_{L3} + M_{14}\dot{i}_{L4} \\ M_{23}\dot{i}_{L3} + M_{24}\dot{i}_{L4} \\ L_3\dot{i}_{L3} + M_{34}\dot{i}_{L4} \\ M_{43}\dot{i}_{L3} + L_4\dot{i}_{L4} \end{bmatrix} \quad (53)$$

3.1.2 Equações dos estados topológicos

Inicialmente vamos definir os estados topológicos a partir das funções de comutação dos interruptores S_1 e S_2 e dos diodos, para os interruptores

$$\begin{aligned} p &= t \bmod t_s \\ s_p &= \begin{cases} 0.5(1 + \text{sgn}(p - d)), \forall v_{in} > 0, \\ 0, \forall v_{in} < 0, \end{cases} \\ s_n &= \begin{cases} 0, \forall v_{in} > 0, \\ 0.5(1 + \text{sgn}(p - d)), \forall v_{in} < 0, \end{cases} \\ d_b &= \begin{cases} 1, \forall v_{in} > 0, \\ 0, \forall v_{in} < 0, \end{cases} \end{aligned} \quad (54)$$

As equações são descritas por:

$$\begin{aligned} d_b(v_{in}) &= d_b(L_1\dot{i}_{L1} + M_{12}\dot{i}_{L2} - v_c(1 - s_p)) \\ d_b(v_{in}) &= d_b(L_2\dot{i}_{L2} + M_{21}\dot{i}_{L1} - v_c(1 - s_p)) \\ (1 - d_b)(v_{in}) &= (1 - d_b)(L_3\dot{i}_{L3} + M_{34}\dot{i}_{L4} - v_c(1 - s_p)) \\ (1 - d_b)(v_{in}) &= (1 - d_b)(L_4\dot{i}_{L4} + M_{43}\dot{i}_{L3} - v_c(1 - s_p)) \\ C\dot{v}_c + \frac{v_c}{R} &= -d_b(\dot{i}_{L1}((s_p - 1))) - (1 - d_b)(\dot{i}_{L2}((s_n - 1))) \end{aligned} \quad (55)$$

Observando nas Equações descritas em (55), nota-se que linha 1 e 2 são linearmente dependentes, já que o estado da corrente no indutor L_1 é mesma do L_2 , o mesmo ocorre para as linha 3 e 4. Para continuar a modelagem é representado a corrente 1 e 2 por i_p e a corrente 3 e 4 por i_n , assim chegamos em:

$$\begin{aligned} d_b(v_{in}) &= d_b(L_1\dot{i}_p + L_2\dot{i}_p + M_{21}\dot{i}_p + M_{12}\dot{i}_p - v_c(1 - s_p)) \\ (1 - d_b)(v_c) &= (1 - d_b)(L_3\dot{i}_n + L_2\dot{i}_n + M_{43}\dot{i}_n + M_{34}\dot{i}_n - v_c(1 - s_p)) \\ C\dot{v}_c + \frac{v_c}{R} &= d_b(\dot{i}_p((s_p - 1))) - (1 - d_b)(\dot{i}_n((s_n - 1))) \end{aligned} \quad (56)$$

3.1.2.1 Análise para o semiciclo positivo da rede

Para o semiciclo positivo temos que d_B é definido como 1

$$A_{p0} = \begin{bmatrix} 0 & 0 & -\frac{1}{L_1+L_2+M_{12}+M_{21}} \\ 0 & 0 & 0 \\ \frac{1}{C} & 0 & -\frac{1}{CR} \end{bmatrix} \quad B_{p0} = \begin{bmatrix} \frac{1}{L_1+L_2+M_{12}+M_{21}} \\ 0 \\ 0 \end{bmatrix} \quad (57)$$

e para S_p igual a 1:

$$A_{p1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{CR} \end{bmatrix} \quad B_{p1} = \begin{bmatrix} \frac{1}{L_1+L_2+M_{12}+M_{21}} \\ 0 \\ 0 \end{bmatrix} \quad (58)$$

O modelo completo é dado por:

$$\dot{x} = \sum_{i=1}^N (A_i x + B_i u) \cdot h_i \quad (59)$$

onde h_i é:

$$h_i = \begin{bmatrix} s_p & 1 - s_p \end{bmatrix} \quad (60)$$

Aplicando a equação (59), nos estados descritos obtém-se:

$$A_{p1} = \begin{bmatrix} 0 & 0 & -\frac{1-s_p}{L_1+L_2+M_{12}+M_{21}} \\ 0 & 0 & 0 \\ \frac{1-s_p}{C} & 0 & -\frac{1}{CR} \end{bmatrix} \quad B_{p1} = \begin{bmatrix} \frac{1}{L_1+L_2+M_{12}+M_{21}} \\ 0 \\ 0 \end{bmatrix} \quad (61)$$

Reescrevendo por espaço de estados temos:

$$\begin{bmatrix} \dot{i}_p \\ \dot{i}_n \\ \dot{v}_o \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{1-s_p}{L_1+L_2+M_{12}+M_{21}} \\ 0 & 0 & 0 \\ \frac{1-s_p}{C} & 0 & -\frac{1}{CR} \end{bmatrix} \begin{bmatrix} i_p \\ i_n \\ v_o \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1+L_2+M_{12}+M_{21}} \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} v_{in} \end{bmatrix} \quad (62)$$

e para a saída:

$$y_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_p \\ i_n \\ v_o \end{bmatrix} \quad (63)$$

3.1.2.2 Análise para o semiciclo negativo da rede

Para o semiciclo positivo temos que d_B é definido como 0, e seguindo os mesmos passos anteriores chegamos em:

$$\begin{bmatrix} \dot{i}_p \\ \dot{i}_n \\ \dot{v}_o \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{1-s_n}{L_3+L_4+M_{34}+M_{43}} \\ 0 & \frac{1-s_n}{C} & -\frac{1}{CR} \end{bmatrix} \begin{bmatrix} i_p \\ i_n \\ v_o \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_3+L_4+M_{34}+M_{43}} \\ 0 \end{bmatrix} [v_{in}] \quad (64)$$

e para a saída:

$$y_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_p \\ i_n \\ v_o \end{bmatrix} \quad (65)$$

3.1.3 Modelo médio

O modelo médio é uma técnica utilizada para simplificar a análise de sistemas dinâmicos com comutação. Em vez de considerar as variações instantâneas causadas pela comutação, o modelo médio busca uma representação simplificada do sistema, onde as comutações são representadas por valores médios no período de comutação.

Para obter o médio no período de comutação aplica-se:

$$\frac{d}{dt} \langle \vec{x} \rangle \triangleq \mathbf{A}_{avg}(t) \langle \vec{x} \rangle + \mathbf{B}_{avg}(t) \langle \vec{u} \rangle \quad (66)$$

onde,

- $\mathbf{A}_{avg}(t) = \sum_{i=1}^N \langle \mathbf{A}_i h_i \rangle$
- $\mathbf{B}_{avg}(t) = \sum_{i=1}^N \langle \mathbf{B}_i h_i \rangle$

Aplicando o operador médio para as matrizes comutadas apresentadas nas equações (62) e (64):

$$\begin{bmatrix} \dot{\langle i_p \rangle} \\ \dot{\langle i_n \rangle} \\ \dot{\langle v_o \rangle} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -\frac{1-d}{L_1+L_2+M_{12}+M_{21}} \\ 0 & 0 & 0 \\ \frac{1-d}{C} & 0 & -\frac{1}{CR} \end{bmatrix} \begin{bmatrix} \langle i_p \rangle \\ \langle i_n \rangle \\ \langle v_o \rangle \end{bmatrix} + \begin{bmatrix} \frac{1}{L_1+L_2+M_{12}+M_{21}} \\ 0 \\ 0 \end{bmatrix} [\langle v_{in} \rangle] \quad (67)$$

e para o semiciclo negativo:

$$\begin{bmatrix} \dot{\langle i_p \rangle} \\ \dot{\langle i_n \rangle} \\ \dot{\langle v_o \rangle} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{1-d}{L_3+L_4+M_{34}+M_{43}} \\ 0 & \frac{1-d}{C} & -\frac{1}{CR} \end{bmatrix} \begin{bmatrix} \langle i_p \rangle \\ \langle i_n \rangle \\ \langle v_o \rangle \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_3+L_4+M_{34}+M_{43}} \\ 0 \end{bmatrix} [\langle v_{in} \rangle] \quad (68)$$

3.1.4 Ponto de equilíbrio

O ponto de equilíbrio em variáveis de estado refere-se a um estado estacionário de um sistema dinâmico no qual as variáveis de estado não apresentam variação ao longo do tempo. Em outras palavras, é um ponto em que as derivadas das variáveis de estado em relação ao tempo são igualadas a zero.

Em um sistema descrito por equações de espaço de estados, o ponto de equilíbrio ocorre quando as derivadas das variáveis de estado em relação ao tempo são nulas. Matematicamente, isso pode ser expresso como:

$$\dot{\mathbf{x}} = \mathbf{0} \quad (69)$$

onde, $\dot{\mathbf{x}}$ representa o vetor das derivadas das variáveis de estado em relação ao tempo e $\mathbf{0}$ é um vetor nulo. Resolver essa equação permite encontrar os valores específicos das variáveis de estado que correspondem ao ponto de equilíbrio.

Conhecer o ponto de equilíbrio em variáveis de estado é importante para compreender o comportamento do sistema em regime permanente, analisar sua estabilidade e projetar controladores adequados. É uma etapa fundamental na modelagem e análise de sistemas dinâmicos.

Para determinar os valores em regime aplicamos a seguinte condição:

$$\begin{bmatrix} \dot{i}_p \\ \dot{i}_n \\ \dot{v}_o \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (70)$$

Aplicando (70) em (62):

$$\begin{bmatrix} \frac{V_{IN}}{L_3+L_4+M_{34}+M_{43}} - \frac{V_O(1-D)}{L_3+L_4+M_{34}+M_{43}} = 0 \\ 0 = 0 \\ -\frac{V_{IN}}{CR} + \frac{I+P(1-D)}{C} = 0 \end{bmatrix} \quad (71)$$

isolando os pontos de operação:

$$\begin{bmatrix} I_P \\ I_N \\ V_O \end{bmatrix} = \begin{bmatrix} \frac{V_{IN}}{R(1-D)^2} \\ 0 \\ \frac{V_{IN}}{1-D} \end{bmatrix} \quad (72)$$

aplicando o mesmo para o semiciclo negativo:

$$\begin{bmatrix} I_P \\ I_N \\ V_O \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{V_{IN}}{R(1-D)^2} \\ \frac{V_{IN}}{1-D} \end{bmatrix} \quad (73)$$

3.1.5 Linearização

Nesta seção, será abordada a linearização do modelo do sistema PFC. Para realizar a linearização, é necessário encontrar o Jacobiano da função $\langle \vec{f} \rangle$ em relação às variáveis i_p , v_o , v_{in} e d .

O Jacobiano é uma matriz que contém as derivadas parciais das funções em relação às variáveis de interesse. No caso do sistema PFC, o Jacobiano é representado pelas matrizes $A_{j\bar{p}}$ e $B_{j\bar{p}}$, conforme mostrado na equação (74). A matriz $A_{j\bar{p}}$ contém as derivadas das equações em relação às variáveis de estado, enquanto a matriz $B_{j\bar{p}}$ contém as derivadas em relação às entradas.

Além disso, as matrizes $C_{j\bar{p}}$ e $D_{j\bar{p}}$, apresentadas na equação (75), são utilizadas para definir as relações entre as saídas e as variáveis de estado e entradas do sistema linearizado.

Essa linearização é importante para simplificar o modelo do sistema e permitir a aplicação de técnicas de controle e análise mais tradicionais, considerando-o como um sistema linear. A partir dessas matrizes, é possível obter informações valiosas sobre o comportamento do sistema em torno de um ponto de operação específico, facilitando o projeto e a implementação de estratégias de controle eficientes para o conversor PFC.

Encontrando o Jacobiano de $\langle \vec{f} \rangle$ em relação aos termos: $\langle i_p \rangle$, $\langle v_o \rangle$, $\langle v_{in} \rangle$ e d

$$A_{j\bar{p}} = \begin{bmatrix} 0 & 0 & -\frac{1-d}{L_1+L_2+M_{12}+M_{21}} \\ 0 & 0 & 0 \\ \frac{1-d}{C} & 0 & -\frac{1}{CR} \end{bmatrix} \quad B_{j\bar{p}} = \begin{bmatrix} \frac{1}{L_1+L_2+M_{12}+M_{21}} & \frac{V_o}{L_1+L_2+M_{12}+M_{21}} \\ 0 & 0 \\ 0 & I_p \end{bmatrix} \quad (74)$$

e as matrizes C e D

$$C_{j\bar{p}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad D_{j\bar{p}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (75)$$

3.1.6 Obtendo as funções de transferência do modelo linearizado

A partir do sistema linearizado podemos obter as funções de transferência do sistema seguindo os passos descritos abaixo.

Dado um sistema dinâmico linear completo:

$$\begin{aligned} \frac{\partial \vec{D}\mathbf{x}}{\partial t} &= \mathbf{A}\vec{D}\mathbf{x} + \mathbf{B}\vec{D}\mathbf{u} \\ \vec{D}\mathbf{y} &= \mathbf{C}\vec{D}\mathbf{x} + \mathbf{D}\vec{D}\mathbf{u} \end{aligned} \quad (76)$$

As funções de transferência entre entradas e saídas do modelo são dadas por:

$$\vec{Y}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\vec{U}(s) + \mathbf{D}\vec{U}(s) \quad (77)$$

Aplicando a equação (77) nas matrizes linearizadas obtém-se o conjunto de funções de transferência

$$\frac{\tilde{Y}(s)}{\tilde{U}(s)} = \begin{bmatrix} \frac{\tilde{I}_p(s)}{\tilde{V}_{jn}(s)} & \frac{\tilde{I}_p(s)}{\tilde{D}(s)} \\ \frac{V_o(s)}{\tilde{V}_{in}(s)} & \frac{\tilde{D}(s)}{\tilde{D}(s)} \end{bmatrix} = \begin{bmatrix} G_{i_p v_{in}} & G_{i_p d} \\ G_{V_o v_{in}} & G_{V_o d} \end{bmatrix} \quad (78)$$

resultando na matriz abaixo.

$$\begin{bmatrix} G_{i_p v_{in}} \\ G_{i_p d} \\ G_{V_o v_{in}} \\ G_{V_o d} \\ G_{V_o i_p} \end{bmatrix} = \begin{bmatrix} \frac{C R s + 1}{R(L_1 + L_2 + M_{12} + M_{21}) C s^2 + (L_1 + L_2 + M_{12} + M_{21}) s + R(1-D)^2} \\ \frac{V_o + R I_p - R I_p d + C R V_o s}{R(L_1 + L_2 + M_{12} + M_{21}) C s^2 + (L_1 + L_2 + M_{12} + M_{21}) s + R(1-D)^2} \\ -\frac{R(d-1)}{R(L_1 + L_2 + M_{12} + M_{21}) C s^2 + (L_1 + L_2 + M_{12} + M_{21}) s + R(1-D)^2} \\ -\frac{R(V_o d - I_p + (L_1 + L_2 + M_{12} + M_{21}) I_p s)}{R(L_1 + L_2 + M_{12} + M_{21}) C s^2 + (L_1 + L_2 + M_{12} + M_{21}) s + R(1-D)^2} \\ \frac{G_{V_o d}}{G_{i_p d}} = -\frac{R(V_o d - V_o + (L_1 + L_2 + M_{12} + M_{21}) I_p s)}{V_o + R(L_1 + L_2 + M_{12} + M_{21}) - R I_p d + C R V_o s} \end{bmatrix} \quad (79)$$

3.1.7 Validação do modelo comutado e modelo médio

Neste item, será realizada a validação dos modelos comutado e médio do conversor. O objetivo é simular as equações de espaço de estados obtidas e validar esses modelos. Para isso, será utilizada a técnica de discretização pelo método Forward-Euler. Esse método consiste em aproximar as equações diferenciais por diferenças finitas, permitindo a simulação dos modelos de forma discreta. Através dessa validação, será possível verificar se os modelos são capazes de reproduzir o comportamento desejado do conversor e se estão adequados para a análise do seu desempenho.

Para simular as equações por espaço de estados obtidas, com a finalidade de validar os modelos, usou-se discretização pelo método Forward-Euler, dado por

$$\frac{\partial f(t)}{\partial t} = \lim_{\Delta T \rightarrow 0} \frac{f(t + \Delta T) - f(t)}{\Delta T} \quad (80)$$

Aplicando a discretizada (80) em (47) obtém-se:

$$\begin{aligned} x[n+1] &= \mathbf{A}_d x[n] + \mathbf{B}_d u[n] \\ y[n] &= \mathbf{C}_d x[n] + \mathbf{D}_d u[n] \end{aligned} \quad (81)$$

onde,

- $\mathbf{A}_d = \mathbf{I}_n + \mathbf{A}T_s$
- $\mathbf{B}_d = \mathbf{B}T_s$
- $\mathbf{C}_d = \mathbf{C}$
- $\mathbf{D}_d = \mathbf{D}$

Abaixo pode é apresentado implementação em código Matlab da simulação do modelo por espaço de estados:

```

1  for i = 1:length(t)-1
2      % Modulador PWM
3      if t(i) >= 1/60
4          d = 1 - Kic * dx_c(1,i);
5          s(i) = d > Portadora(i);
6      else
7          d = 0;
8          s(i) = d > Portadora(i);
9      end
10
11     % Modelo comutado
12     dx_c(1:N,i+1) = dx_c(1:N,i) + dt*( (Ac_1*s(i) + Ac_0*(1-s(i)
13         ))*(dx_c(1:N,i)) + Bs*V_in(i) );
14     % Representacao do modelo diodo
15     if dx_c(1,i+1) <= 0
16         dx_c(1,i+1) = 0;
17     end
18     % Modelo medio
19     dx_m(1:N,i+1) = dx_m(1:N,i) + dt*( (As)*(dx_m(1:N,i)) + Bs*
20         V_in(i) );
21 end

```

A partir do código é possível comparar o modelo obtido como resultados de simulação. O código em MATLAB apresentado realiza a simulação do modelo discretizado das equações diferenciais do conversor PFC. Ele utiliza um loop for para iterar sobre o vetor de tempo e implementa um modulador PWM para determinar o duty cycle do conversor. Em seguida, são atualizadas as variáveis de estado do modelo comutado e do modelo médio do conversor. O código também inclui uma verificação para garantir que o valor da variável de estado não seja negativo, representando o comportamento do diodo no modelo. Essas etapas permitem simular o comportamento do conversor PFC ao longo do tempo, considerando o modulador

PWM, os modelos comutado e médio. Na Figura 18 é apresentada formas de onda de tensão de entrada, tensão de saída e corrente do modelo, a esquerda são apresentando os modelos das equações obtidas e na direita resultados de simulação.

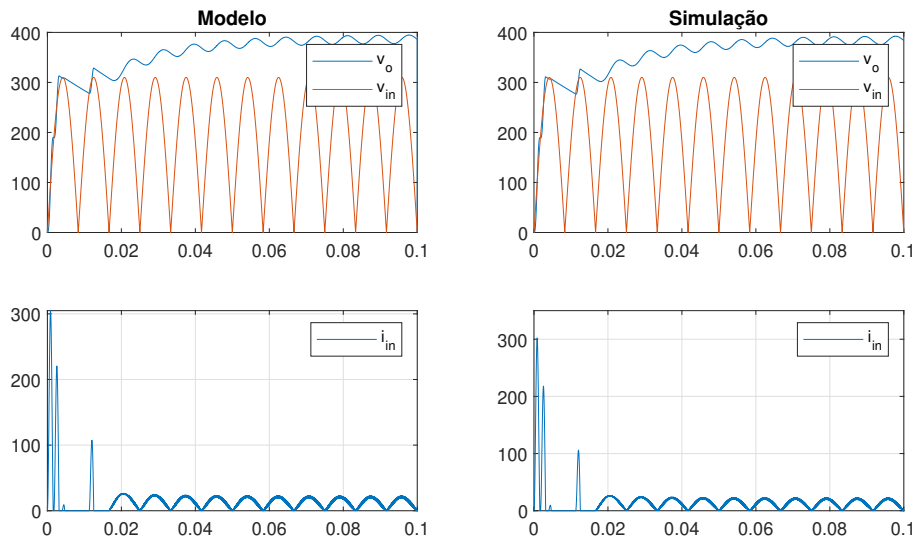


Figura 18 – Comparação entre o modelo comutado do conversor e resultados de simulação na entrada do controle ativo. Na figura são apresentados a tensão de saída, tensão de entrada e corrente de entrada do conversor.

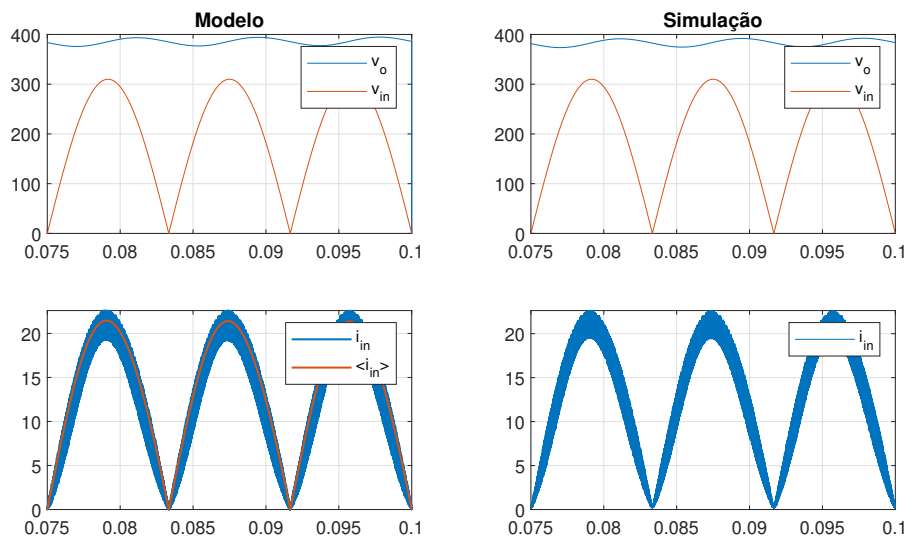


Figura 19 – Comparação entre o modelo comutado e médio do conversor e resultados de simulação, em regime permanente senoidal. Na figura são apresentados a tensão de saída, tensão de entrada e corrente de entrada do conversor.

Na Figura 19, é apresentado a operação em regime permanente senoidal. Na Figura 18, mostrando o conversor em regime permanente senoidal, é possível observar a corrente do modelo comutado e corrente do modelo médio.

3.2 MODELAGEM DA TENSÃO DE MODO COMUM

Os conversores com correção do fator de potência (PFC) são fundamentais para otimizar a eficiência e qualidade de energia em sistemas de eletrônica de potência. No entanto, um problema recorrente em alguns conversores é a geração de tensões de modo comum. A tensão de modo comum refere-se à tensão presente entre um ponto de referência (geralmente o terra) e o ponto médio entre as duas linhas de alimentação. Em outras palavras, é a média das tensões entre o terra e cada uma das linhas.

A tensão de modo comum nos conversores PFC surge devido a várias razões, incluindo: Transientes rápidos de comutação dos semicondutores, acoplamentos capacitivos entre o circuito e o chassi ou entre componentes individuais ou Assimetrias nos trajetos de corrente e disposição dos componentes.

A presença de tensões de modo comum pode causar vários problemas: Interferência Eletromagnética (EMI): A tensão de modo comum pode se converter em correntes de modo comum, que são uma fonte significativa de interferência eletromagnética radiada e conduzida. Estresse em componentes: Componentes como capacitores podem ser submetidos a tensões não previstas durante o projeto, diminuindo sua vida útil.

Problemas de Segurança: A tensão de modo comum pode levar a correntes indesejadas que flutuam através do chassi ou terra, criando potenciais problemas de segurança.

Diversas técnicas podem ser adotadas para minimizar os problemas associados à tensão de modo comum em conversores PFC: Design de Layout: Um design de layout cuidadoso, evitando loops grandes e minimizando os trajetos de corrente de comutação, pode ajudar a reduzir a tensão de modo comum. Filtragem: Filtros de modo comum, geralmente compostos por indutores e capacitores, podem ser inseridos na entrada do conversor para reduzir as correntes de modo comum. Para análise da tensão de modo comum, tomamos os modelo mostrado na Figura 20. Na Figura é apresentado o circuito do conversor referente a semiciclo positivo, a análise é válida para os dois ciclos considerando a simetria do converso.

Analisando as etapas de operação, é possível expressar as tensões nos pontos **A** e **B** do circuito com base na tensão do barramento e em função de comutação do interruptor. As expressões para as tensões são:

$$V_{AO} = \frac{V_o}{2}(1 - s_1(t)) \quad (82)$$

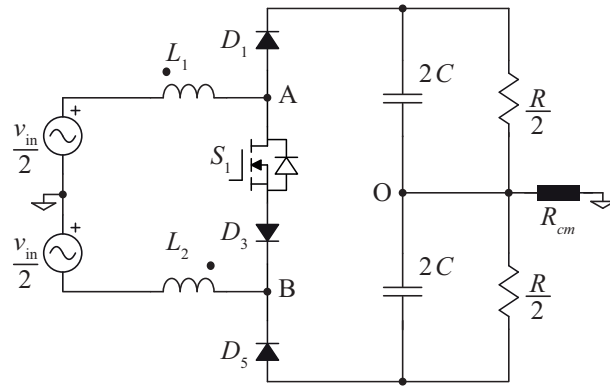


Figura 20 – Modelo equivalente referente a um semiciclo da rede do conversor para análise da tensão de modo comum.

$$V_{BO} = -\frac{V_o}{2}(1 - s_1(t)) \quad (83)$$

Com base nessas equações, podemos representar o circuito equivalente conforme ilustrado na Figura 21.

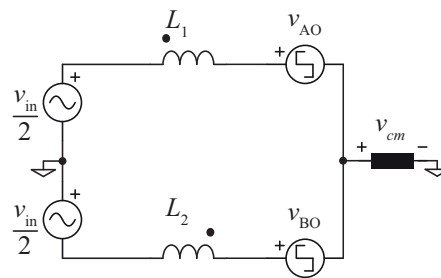


Figura 21 – Circuito equivalente para análise da tensão de modo comum no conversor

Utilizando o princípio da superposição, é possível decompor a tensão de modo comum em duas componentes. A componente de baixa frequência pode ser representada pela expressão:

$$v_{in} \cdot \frac{L1 \parallel L2}{2(L1 - L2)} \quad (84)$$

e a componente em alta frequência pode ser descrita como:

$$\left(\frac{V_o}{2L1} - \frac{V_o}{2L2}\right) \frac{L1L2}{L1 + L2}. \quad (85)$$

Considerando a simetria do circuito, sendo L1 e L2 idênticos, podemos notar que as componentes são zero tanto para alta frequência quanto para baixa frequência. A Figura 22 e 23 mostram respectivamente o resultado onde a indutância L1 e L2 são iguais, resultando em uma simetria do conversor e tensão de modo comum zero, já na segunda onde L2 é igual zero

podemos notar o aparecimento de uma tensão de modo comum diferente de zero, tanto para componente de baixa frequência quanto para alta frequência. Para resumir os resultados são apresentado na Tabela 4.

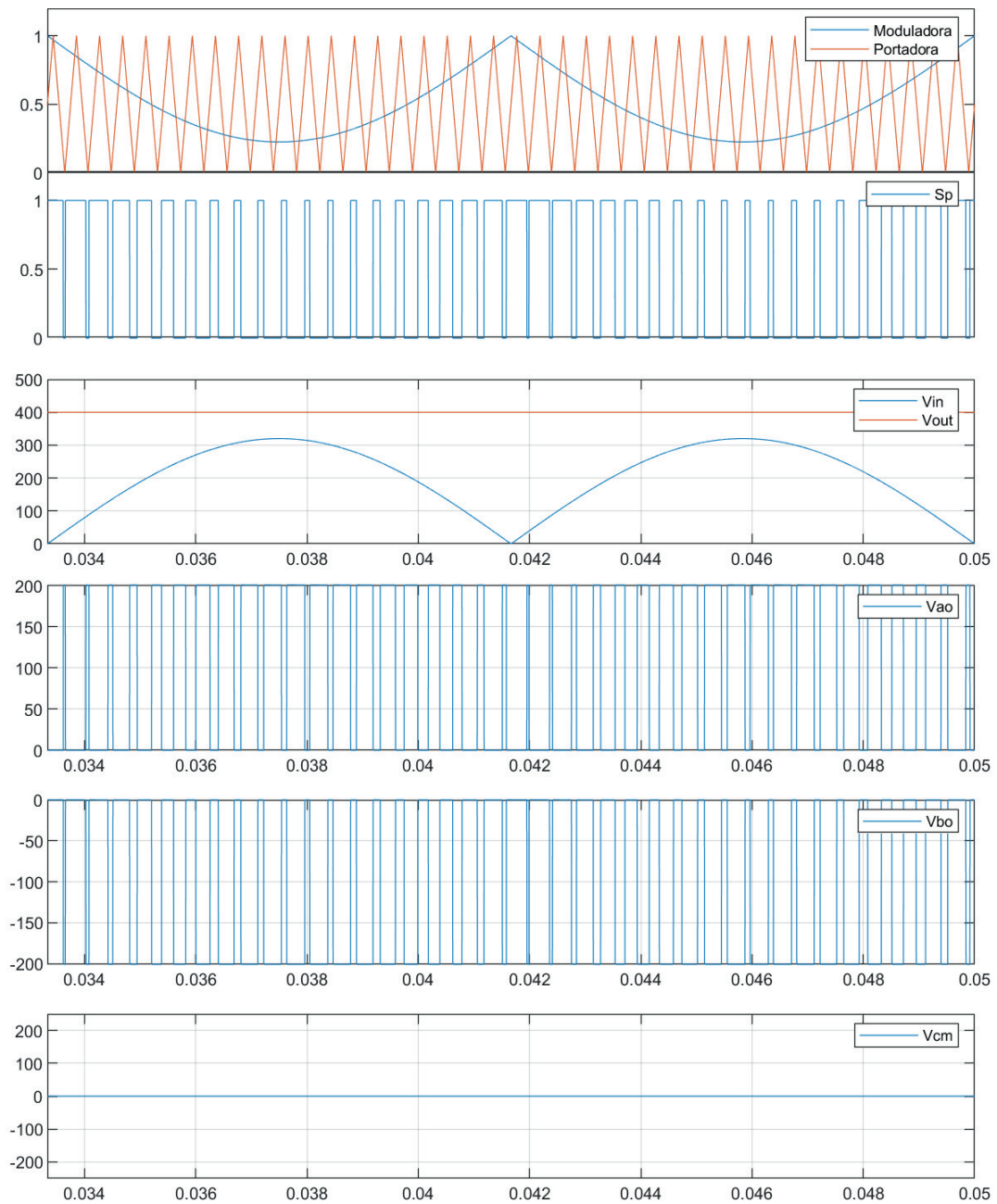


Figura 22 – Resultado tensão de modo comum para o conversor considerando $L1$ igual a $L2$.

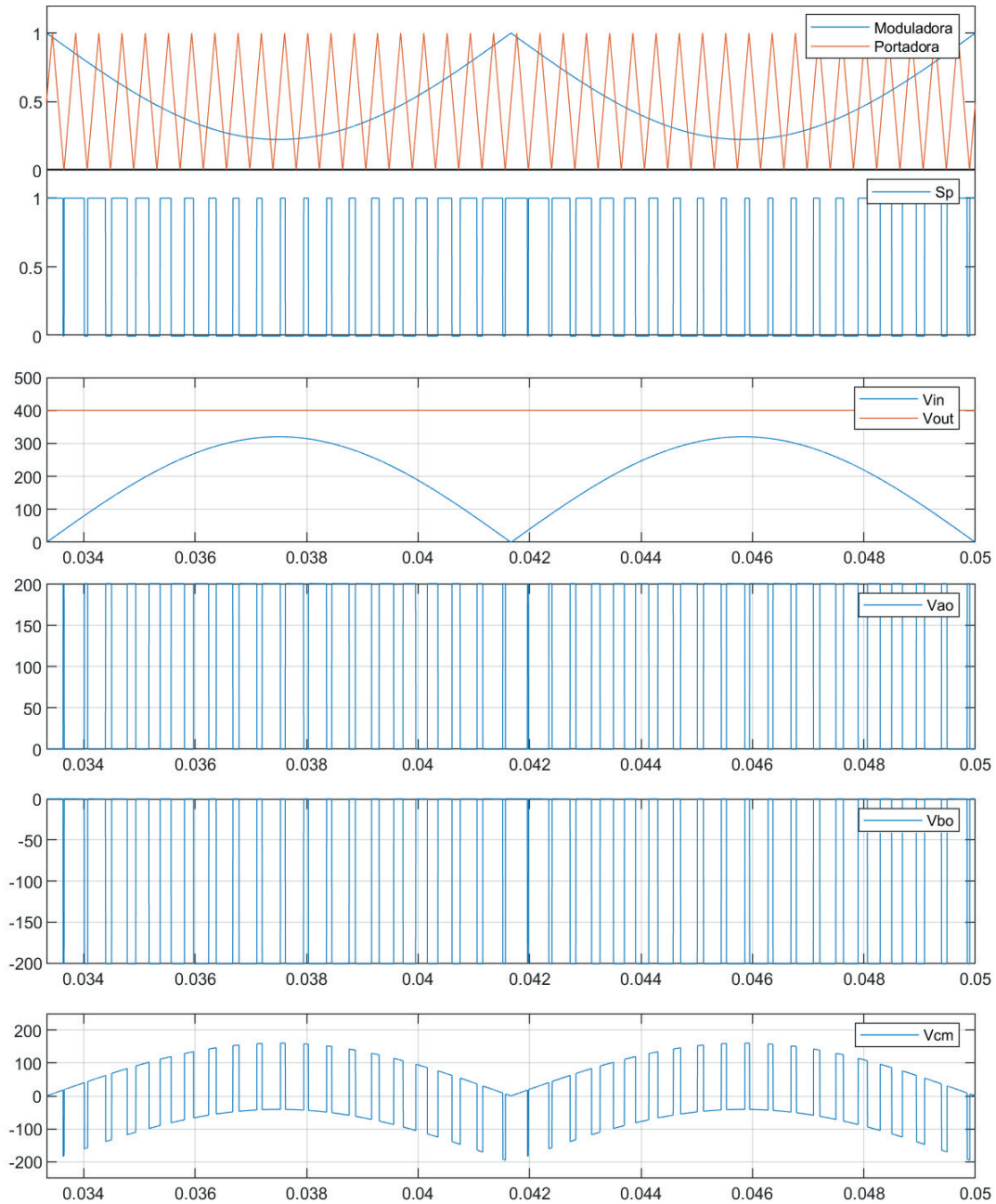


Figura 23 – Resultado tensão de modo comum para o conversor considerando $L2$ igual a zero.

Tabela 4 – Tensão de modo comum para diferentes condições de $L1$ e $L2$

Condição	Baixa Frequência	Alta Frequência
$L1 = L2 \neq 0$	0	0
$L1 \neq L2 \neq 0$	$v_{in} \cdot \frac{L1 \parallel L2}{2(L1-L2)}$	$(\frac{V_o}{2L1} - \frac{V_o}{2L2}) \frac{L1L2}{L1+L2}$
$L1 \neq 0, L2 = 0$	$\frac{v_{in}}{2}$	$-\frac{V_o}{2}$

4 PROJETO DO CONVERSOR

Neste capítulo, serão apresentados os principais aspectos relacionados ao projeto dos elementos do conversor, incluindo cálculos de perdas nos elementos mais passivos e ativos. O projeto de um conversor é geralmente pautado em um conjunto de parâmetros que determinam seu desempenho e rendimento. A Tabela 5 lista os principais parâmetros para o projeto de conversor, juntamente com seus respectivos valores.

Tabela 5 – Parâmetros de entrada para o projeto do conversor

Parâmetro	Valor
Tensão de entrada	220 V
Tensão de saída	375 V
Frequência da rede	60 Hz
Potência	3300 W
Ondulação de corrente máxima	30 %
Ondulação de tensão máxima	10 %

Os valores mencionados na tabela foram usados como entrada para o projeto descrito a seguir nesse capítulo.

4.1 PROJETO DO INDUTOR ACOPLADO

O projeto dos indutores segue uma metodologia que inclui a definição do núcleo e do enrolamento, a avaliação das perdas no núcleo e nos enrolamentos, e a estimativa da elevação de temperatura. Essas etapas são realizadas iterativamente para cada núcleo. A metodologia adotada é apresentada no livro (HURLEY; WOLFLE, 2013) e pode ser visualizada no fluxograma da Figura 24. Depois de definir o núcleo adequado, as perdas no núcleo e no cobre são calculadas e a elevação de temperatura são calculadas.

4.1.1 Definindo o núcleo

A escolha do núcleo do indutor é um processo importante no projeto de conversores. O núcleo é responsável por armazenar a energia magnética do indutor e é crucial para a eficiência e desempenho do conversor.

Existem muitos tipos de núcleos disponíveis no mercado, cada um com diferentes propriedades magnéticas e elétricas. Para escolher o núcleo adequado, é necessário levar em consideração a frequência de operação, a corrente do indutor, o tamanho físico do núcleo e a perda de energia.

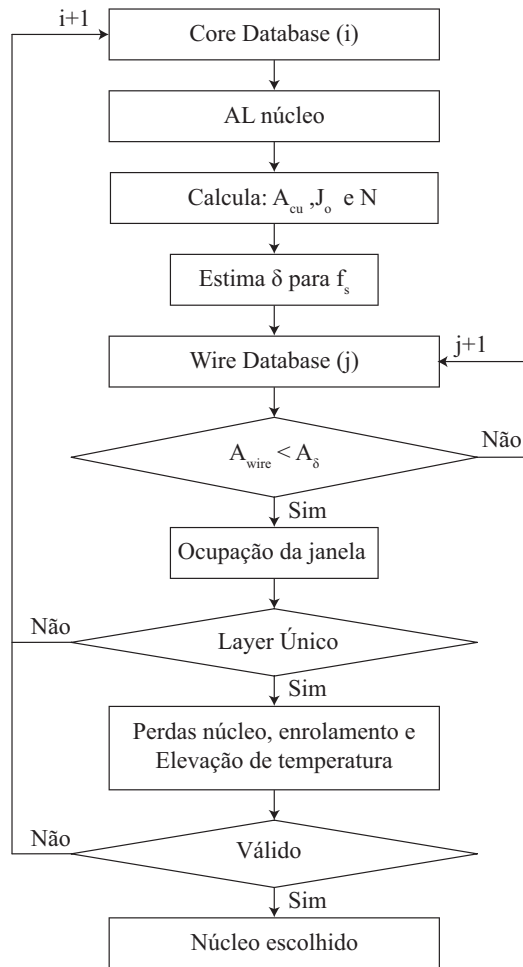


Figura 24 – Metodologia para projeto do indutor do conversor.

Um fator importante na escolha do núcleo é o seu material. Os materiais comuns utilizados em núcleos incluem ferrite, níquel e pó de ferro. Cada material tem sua própria curva de magnetização e resistência elétrica, o que pode afetar o desempenho do indutor.

O núcleo escolhido Advanced Powder Core APH47P60, ele foi escolhido dentro de algumas opções disponíveis no laboratório, o uso de núcleos toroidais em conversores PFC proporciona benefícios técnicos, como alta permeabilidade magnética, baixas perdas magnéticas, redução de emissões eletromagnéticas (baixa dispersão do fluxo magnético), facilidade de enrolamento e alta densidade de energia.

As perdas de energia no núcleo também devem ser consideradas. Perdas excessivas no núcleo podem levar a uma eficiência reduzida do conversor e a um aumento na temperatura do indutor. Por isso, é importante escolher um núcleo com perdas de energia adequadas para a aplicação específica do conversor.

Algumas características do núcleo escolhido podem ser observadas na Figura 25.

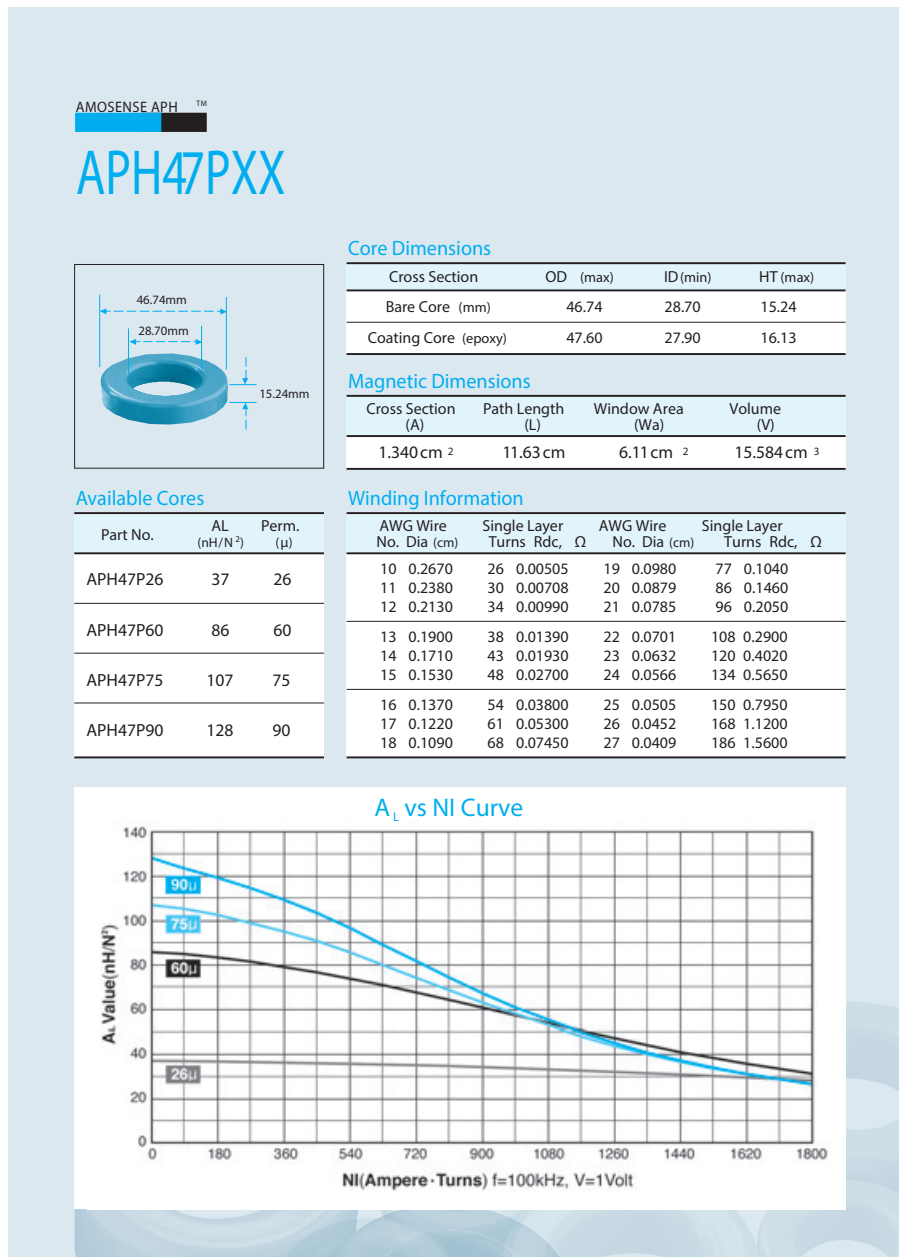


Figura 25 – Informações do núcleo utilizado fornecido pelo fabricante do núcleo Amosense.

4.1.2 Dimensionando os condutores do indutor

Primeiramente, é definida uma densidade de corrente J_o no enrolamento em A/cm^2 . Em seguida, é calculada a área de cobre necessária A_{cobre} para a densidade de corrente desejada a partir da corrente eficaz I_{rms} e da densidade de corrente J_o :

$$A_{cobre} = \frac{I_{rms}}{J_0} \quad (86)$$

A penetração para o cobre δ e a seção máxima do condutor $A_{cu_{max}}$ são calculadas considerando uma frequência de comutação f_s , pode ser aproximado para o material de cobre por:

$$\delta \approx \frac{0.075}{\sqrt{f_s}} \quad (87)$$

$$A_{cu_{max}} = \pi\delta^2 \quad (88)$$

É realizada uma rotina para buscar o fio AWG que tenha próximo de 100 % de ocupação, a fim de encontrar o diâmetro adequado do fio. Em seguida, o número de condutores em paralelo para a densidade de corrente definida é calculado usando a área de cobre calculada anteriormente:

$$N_{paralelo} = \left\lceil \frac{A_{cobre}}{A_{cup}} \right\rceil \quad (89)$$

Onde $\lceil \cdot \rceil$ denota o arredondamento para cima.

O código também realiza cálculos relacionados ao uso de Litz, onde é determinada a equivalência entre AWG33x100 e AWG14, e é especificada a quantidade de voltas (N) para os núcleos. A máxima indutância para cada modelo considerando o enrolamento single layer, sendo que a indutância é dada pelos enrolamentos $L_1 + L_2$ ou $L_3 + L_4$, assim divide-se por 2 o valor de voltas:

4.1.3 Perdas no núcleo dos indutores

Durante o trabalho, foram utilizados dois artigos científicos como base para a estimativa das perdas no núcleo do indutor. Esses artigos apresentam soluções analíticas para o cálculo das perdas no núcleo em retificadores boost PFC monofásico (JACOBOSKI; BASTIANI LANGE; HELDWEIN, Marcelo Lobo, 2019) e bridgeless PFC monofásico (JACOBOSKI; BASTIANI LANGE; HELDWEIN, Marcelo Lobo, 2018).

Os artigos apresentam uma solução analítica fechada para calcular as perdas no núcleo em retificadores PFC monofásicos, utilizando o método iGSE. O método proposto permite calcular as perdas no núcleo de forma rápida (computacionalmente), como mostras o resultado de (JACOBOSKI; BASTIANI LANGE; HELDWEIN, Marcelo Lobo, 2018).

4.1.3.1 Perda de comutação

Os valores utilizados para traçar as perdas de comutação dos transistores GaN foram obtidos de (JONES *et al.*, 2015) e (JONES *et al.*, 2016), onde o teste de duplo pulso (DPT) foi realizado para determinar os parâmetros que descrevem a energia dissipada na comutação. As equações resultantes de energia para entrada em condução do transistor é:

$$E_{on}(I_L) = k_1 I_L^2 + k_2 I_L + k_3, \quad (90)$$

e para para saída de condução:

$$E_{off}(I_L) = k_4 I_L^2 + k_5 I_L + k_6, \quad (91)$$

Os parâmetros para as equações (90) e (91) são apresentados na Tabelas 6 e 7.

Tabela 6 – Valores de k_1 , k_2 e k_3 em diferentes condições de operação

Condições de Operação	k_1	k_2	k_3
400 V 0-37 A	0.0247	0.9302	16.1406
300 V 0-27 A	0.0176	0.5667	10.0385
200 V 0-17 A	0.0119	0.2464	5.6710
100 V 0-7 A	-0.0228	0.2655	1.1071

Tabela 7 – Valores de k_4 , k_5 e k_6 em diferentes condições de operação.

Condições de operação	k_4	k_5	k_6
400 V 0-7 A	-0.0455	-0.8996	8.4322
400 V 7-37 A	0.0143	-0.1622	13.1320
300 V 0-7 A	-0.0626	0.7972	5.1050
300 V 7-27 A	-0.0024	0.2482	6.1443
200 V 0-17 A	-0.0157	0.3099	2.9879
200 V 7-17 A	-0.0155	0.4343	1.9345
100 V 0-7 A	-0.0095	0.1297	1.0273

A partir das equações que representam a energia para entrada em condução e para saída de condução, é possível calcular as perdas de comutação do conversor integrando-as em um período de grade:

$$P_{on} = \frac{1}{T_r} \int_0^{T_r} f_s \cdot E_{on}(i_L) dt, \quad (92)$$

$$P_{off} = \frac{1}{T_r} \int_0^{T_r} f_s \cdot E_{off}(i_L) dt, \tag{93}$$

sendo,

- T_r - Período de tensão da rede elétrica;
- i_L - Corrente que sobre o canal do transistor;
- $E_{On}(i_L)$ - Energia necessária para entrada em condução;
- $E_{Off}(i_L)$ - Energia necessária para saída de condução.

4.1.4 Perdas nos diodos

A tensão de condução do diodo é uma das principais características elétricas dos diodos semicondutores, e se refere à tensão mínima necessária para que o diodo comece a conduzir corrente. É um parâmetro importante na escolha de um diodo para uma aplicação específica, pois determina a queda de tensão que ocorrerá no diodo quando ele estiver conduzindo.

O diodo escolhido IDL12G65C5 da Infineon é um diodo de comutação rápida, projetado para operar em aplicações de alta potência e alta frequência. Ele é construído a partir de tecnologia de *silicon carbide* (SiC).

Além disso, o diodo IDL12G65C5 tem uma capacidade de recuperação reversa rápida (Qrr), o que significa que ele pode comutar rapidamente do estado de condução para a de bloqueio. Isso o torna adequado para aplicações de comutação de alta frequência.

As características de tensão de condução deste modelo de diodo são apresentadas na Figura 26, e foram usadas como base de cálculo para estimar a tensão de condução e as perdas de condução, esses dados calculados podem ser visualizados na Figura 27

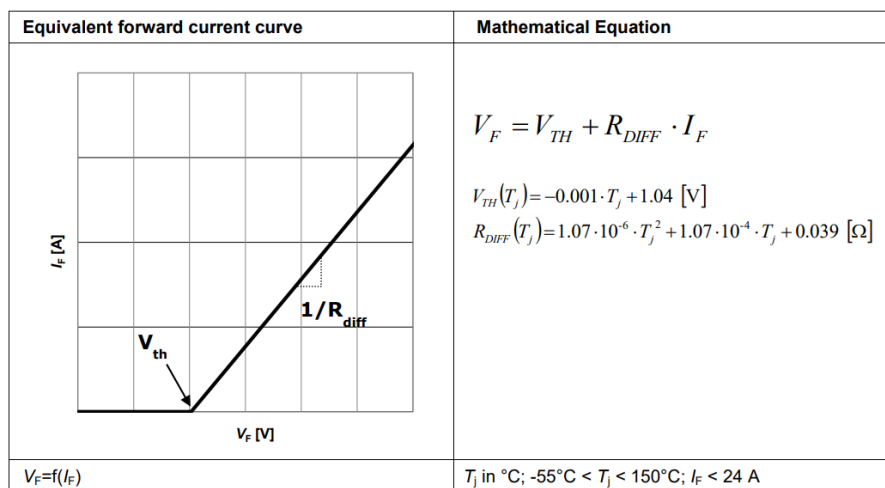


Figura 26 – Modelo simplificado de tensão de condução do diodo utilizado. Imagem obtida em (AG, 2019).

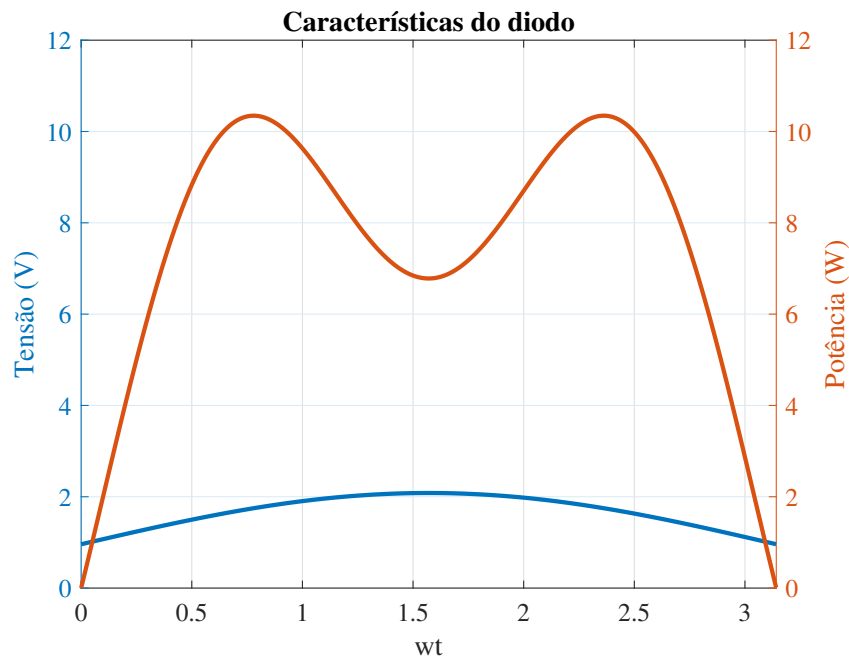


Figura 27 – Tensão de condução dos diodos ao longo do ciclo da rede, e perdas instantâneas.

4.1.5 Rendimento do conversor

O rendimento de um conversor é uma medida de eficiência que indica a quantidade de energia de entrada que é convertida em energia de saída. Ou seja, o rendimento é a razão entre a potência de saída e a potência de entrada, multiplicada por 100 para expressar o resultado em porcentagem.

A partir das perdas calculadas nessa seção é possível calcular as perdas totais do conversor através da soma de diversas parcelas que representam as perdas em diferentes componentes do circuito. As perdas incluem a dissipação de calor no cobre do indutor, nas perdas no núcleo do indutor, nas perdas de comutação e condução nos dispositivos semicondutores (GaN), bem como nas perdas nos diodos retificadores e nos diodos em série ao interruptor. E o resultado é apresentado na Figura 28.

O rendimento do conversor obtido em carga nominal foi de 97.67 % e a distribuição das perdas é apresentada na Figura 28. É importante lembrar que o rendimento do conversor depende de diversos fatores, como a topologia do circuito, os componentes utilizados, as condições de operação (por exemplo, a carga conectada) e o método de controle utilizado. Portanto, o rendimento apresentado na Figura 28, representa o rendimento do conversor em carga nominal.

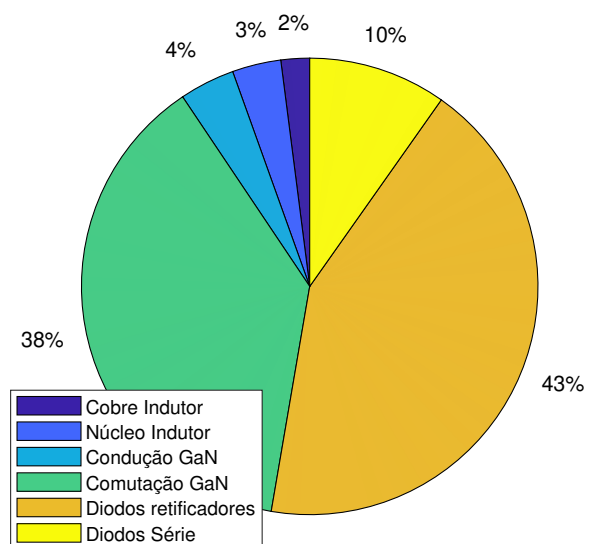


Figura 28 – Distribuição de perdas nos elementos do conversor

5 CONTROLE

O controle em conversores PFC (correção do fator de potência) ativo é fundamental para assegurar o correto funcionamento do conversor, uma vez que este não pode operar sem em modo de condução contínua. O objetivo do controle é manter a corrente de entrada do conversor em fase com a tensão da rede elétrica, resultando em um alto fator de potência. Além disso, é necessário uma malha externa de tensão para manter a tensão de saída regulada. A estratégia de controle utilizada é apresentada na Figura 29 e sua implementação é resumida neste capítulo. Para um aprofundamento maior na implementação, é possível consultar o Apêndice A.

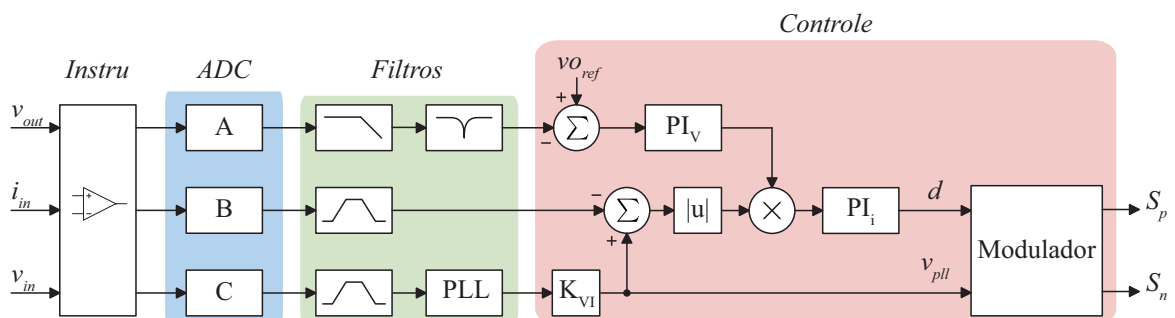


Figura 29 – Diagrama de blocos da estratégia de controle utilizada no conversor PFC.

5.1 ALGORITMO PLL

O SOGI-PLL (Second-Order Generalized Integrator Phase-Locked Loop) é um tipo de circuito utilizado em sistemas de controle onde é necessário sincronismo de fase, comumente utilizado em aplicações de eletrônica de potência conectados a rede elétrica, tais inversores e retificadores. A representação do SOGI-PLL pode ser visto na Figura 30.

O SOGI-PLL é comumente usado em sistemas monofásicos, devido a sua característica de gerar sinais de fase e quadratura a partir de um sinal de tensão de apenas uma fase. Para compreender o funcionamento divide-se ele em duas partes, SOGI-QSG e SRF-PLL. O SOGI-QSG é capaz de gerar sinais de fase e quadratura em um sistema elétrico monofásico, a partir da medição da tensão de entrada.

A equação da função de transferência do SOGI-QSG pode ser representada por duas equações, a função de transferência do integrador SOGI ($G_1(s)$) e a função de transferência do gerador de sinal em quadratura ($G_2(s)$). A função de transferência $G_1(s)$ representa a parte em fase do SOGI-QSG, enquanto a função de transferência $G_2(s)$ representa a parte em quadratura. As equações das funções de transferência são:

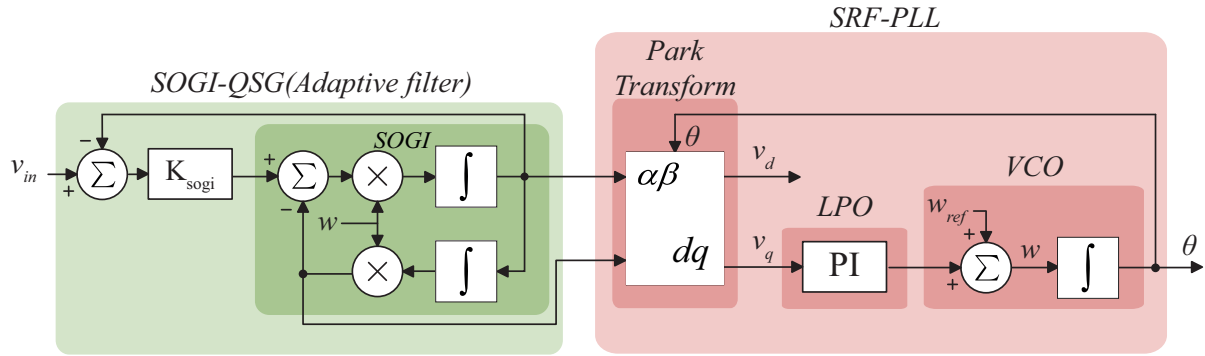


Figura 30 – Diagrama de blocos do SOGI-PLL, que consiste na união da estrutura SOGI-QSG e SRF-PLL

$$G_1(s) = \frac{V_\alpha(s)}{V_g(s)} = \frac{k\omega s}{s^2 + k\omega s + \omega^2} \quad (94)$$

$$G_2(s) = \frac{V_\beta(s)}{V_g(s)} = \frac{k\omega^2}{s^2 + k\omega s + \omega^2} \quad (95)$$

onde $V_\alpha(s)$ e $V_\beta(s)$ são as componentes de fase do sinal de entrada $V_g(s)$, k é o ganho do SOGI-QSG e ω é a frequência de ressonância do filtro.

A segunda parte do SOGI PLL consiste no (SRF-PLL), que é um estratégia de PLL utilizado para sincronização em sistemas elétricos. Ela é capaz de obter a informação de fase, frequência e magnitude a partir do sinais de fase e quadratura em sua entrada, permitindo a obtenção de uma referência de fase e magnitude para o controle dos dispositivos de potência conectados a rede elétrica.

5.2 MODULAÇÃO DO CONVERSOR

A estratégia de modulação de largura de pulso digital é o PWM clássico que consiste em um comparador do sinal modulador (d) e do sinal modulante (Portadora), como apresentado na Figura 31.

Para simplificação foi usado apenas um modulador PWM, obtido a partir da comparação com o valor absoluto da modulador, assim os sinais PWM, S_p e S_n , são o mesmo, a não ser pelo fato de que eles são habilitado somente em seus respectivos ciclo de atuação. A seleção do ciclo da rede é feita com a referencia de tensão gerada pelo SOGI-PLL, garantindo a habilitação dos sinais sincronizado com a rede.

Na Figura 32 é apresentado o resultado experimental da validação da estrategia apresentada na Figura 31. Na figura é possível a tensão de entrada do conversor e também

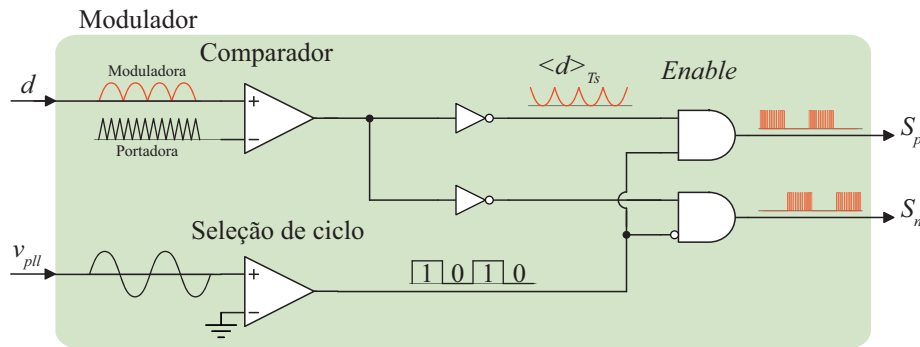


Figura 31 – Estratégia de modulação utilizada do conversor proposto. Consiste no modulador PWM tradicional, com um circuito de habilitação dos interruptores referente a cada ciclo da rede.

o sinal de seleção de ciclo, e por fim os sinais PWM, S_p e S_n atuando em seus respectivos ciclo da rede elétrica.

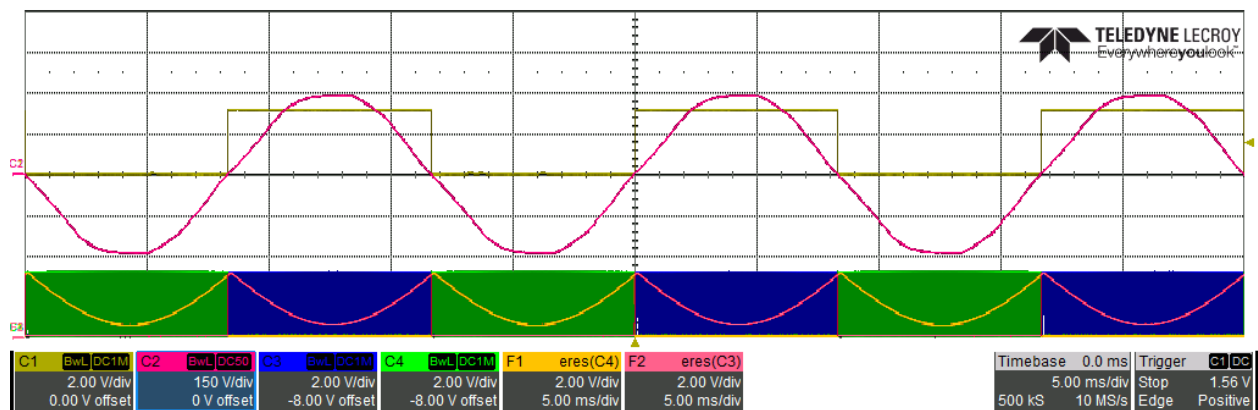


Figura 32 – Resultados práticos da modulação implementada.

5.3 ESTRATEGIA DE AMOSTRAGEM

A estratégia de amostragem do conversor implementada no DSP é apresentada na Figura 33. Ela foi estruturada em duas interrupções, a primeira é interrupção do PWM (isrPWM) e a segunda a interrupção do ADC (isrADC).

A interrupção do PWM foi configurada para ocorrer a cada evento com um pre-scaler definido. Na configuração são considerados eventos:

- Portadora igual a zero;
- Portadora igual ao valor do período;

ou seja, são gerado eventos no zero e no pico da portadora. Já o pre-scaler define quantos desses eventos precisão ocorre para de fato gerar uma interrupção. O pre-scaler foi definidos

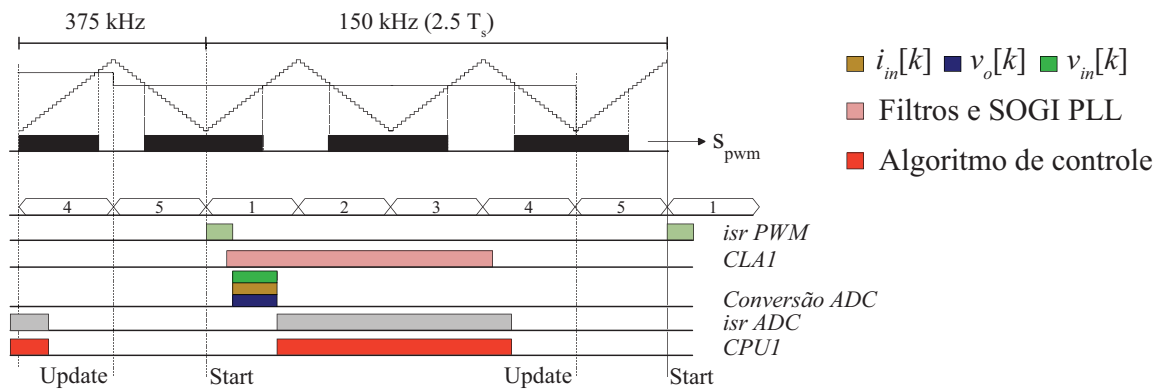


Figura 33 – A estratégia de amostragem do conversor implementada no DSP. A isr PWM é disparada a cada 5 eventos do modulador pwm, nela é iniciada a conversão do ADC e disparo da tarefa do CLA. O fim da conversão do ADC gera uma interrupção que roda a rotina de controle e atualiza os valores de modulação, finalizando as rotinas de controle.

de forma experimental sendo o requisito de tempo total de execução dos algoritmos serem menor que o período de atualiza, para que assim seja possível realizar todos os cálculos antes da próxima amostragem.

Dentro da isrPWM são executadas as seguintes instruções:

- Armazenamento das variáveis de controle processadas no CLA;
- Passagem dos variáveis amostradas para os filtros CLA;
- Disparo da tarefa CLA;
- Disparo das amostragem dos periféricos ADC;

Enquanto o CLA processa os filtros, o CPU aguarda os resultados do ADC para iniciar a rotina de controle, que é executada dentro da isrADC. A mesma que é disparada no final da conversão do periférico.

5.4 RESULTADOS DE SIMULAÇÃO

O resultado do controle apresentado nessa sessão é uma simulação das malhas de controle do conversor PFC, onde são habilitados os controles de corrente e tensão, além de ser testado um degrau de carga, apresentado na Figura 35.

O resultado da simulação é apresentado em seis figuras, onde são plotados os dados relevantes para o controle do conversor. Na primeira, são apresentadas as formas de onda da tensão de entrada, tanto a não filtrada quanto a filtrada, e o sinal de eixo direto da saída da transformada dq0 do SOGI-PLL. Já na segunda, são apresentadas as formas de onda da tensão de saída, tanto a não filtrada quanto a filtrada, e a referência de tensão da

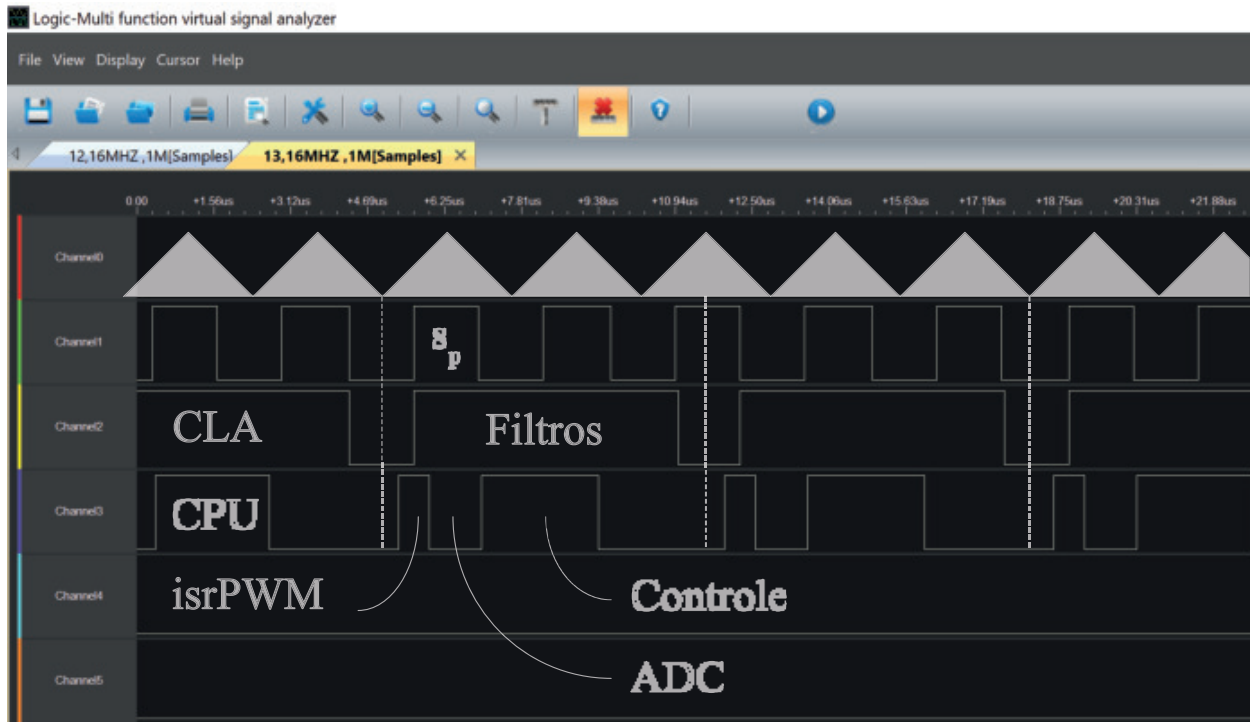


Figura 34 – A estratégia de amostragem do conversor implementada no DSP. Resultados práticos das temporizações das rotinas implementadas apresentados na Figura 33.

saída. Na terceira, são apresentadas as formas de onda da corrente de entrada, tanto a não filtrada quanto a filtrada. A quarta figura mostra as formas de onda dos sinais s_p , s_n , e seus respectivos valores médios no período de comutação t_s : $\langle s_p \rangle_{t_s}$ e $\langle s_n \rangle_{t_s}$, que representam os sinais de comando do conversor. A quinta apresenta as formas de onda do sinal de referência da fase da rede elétrica e da tensão da rede elétrica em quadratura. E, por fim, a sexta figura apresenta as formas de onda da corrente de saída, bem como os sinais de habilitação do controle de corrente e tensão. Esses dados são importantes para avaliar o desempenho do controle do conversor PFC e fazer ajustes necessários para otimizar as malhas de controle.

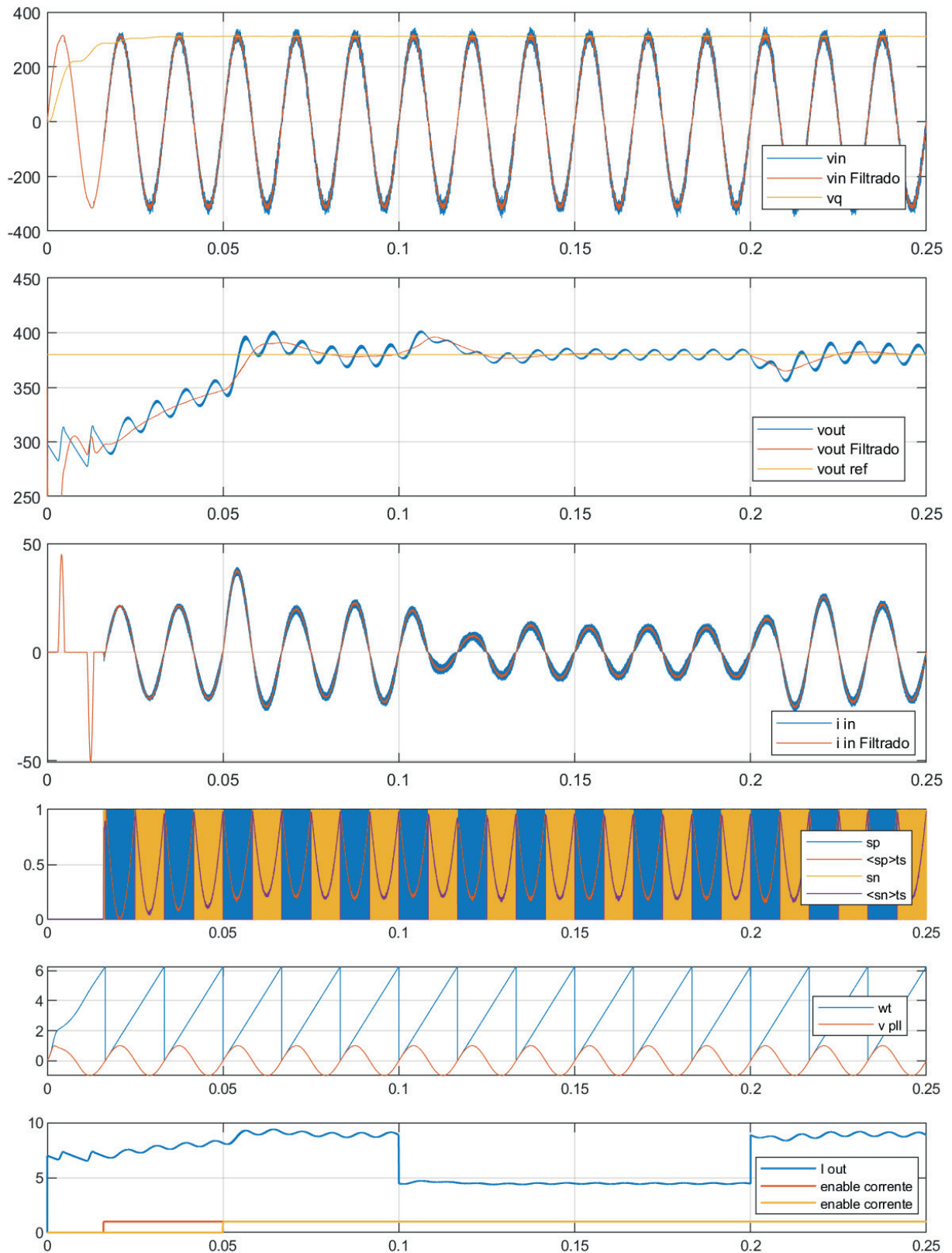


Figura 35 – A figura mostra os resultados da simulação do controle PFC apresentado. A figura consiste em seis partes que mostram as seguintes variáveis: (1) V_{in} (não filtrado), V_{in} (filtrado) e V_q (Eixo quadratura do SOGI-PLL); (2) V_{out} (não filtrado), V_{out} (filtrado) e V_{out} ref; (3) I_{in} (não filtrado) e I_{in} (filtrado); (4) S_p , $\langle S_p \rangle_{ts}$, S_n e $\langle S_n \rangle_{ts}$; (5) ωt e V_{pll} ; e (6) I_{out} .

6 RESULTADOS

Os resultado do conversor PFC foi dividido em duas partes para avaliação:

- Hardware-in-Loop (HIL): é uma técnica de teste usada para validar o desempenho do hardware do conversor PFC. Nesse método, o conversor é testado em um ambiente virtual (DSP) em tempo real, simulando as condições reais em que ele será utilizado. Com essa técnica, é possível testar e otimizar o desempenho do conversor antes de sua implantação real.
- Resultados reais do conversor montado: é a avaliação do desempenho do conversor PFC em condições reais de operação. Isso envolve a instalação do conversor em um ambiente de teste e a medição de seu desempenho.

Em resumo, os resultados práticos do conversor PFC podem ser avaliados por meio de testes em hardware-in-loop e em condições reais de operação. Esses testes permitem que o desempenho do conversor seja avaliado e otimizado antes de sua implantação real, o que pode levar a melhorias no controle, sem os riscos envolvidos em testes reais.

6.1 RESULTADOS HIL

O Hardware-in-Loop (HIL) é um sistema que une elementos de hardware e software para simular e testar o desempenho de sistemas eletrônicos em um ambiente controlado. Neste contexto, o processador digital de sinal (DSP) TMS320F28379D da Texas Instruments foi utilizado para a execução de tarefas de controle e emulação das equações diferenciais do conversor PFC.

O controle do conversor PFC foi realizado por meio do CPU1 e do acelerador de processamento de sinais em tempo real CLA1. O CPU1 foi dedicado para para controle do conversor PFC e troca de dados com CPU2, enquanto o CLA1 é um processador dedicado a executar de filtros.

Ademais, as equações diferenciais do conversor foram emuladas no CPU2 e no CLA2. O CPU2 realiza os cálculos das equações diferenciais discretizado do conversor enquanto o CLA2 emula os sinais de entrada do conversor. Esse estratégia pode ser visualizada na Figura 36.

A alocação de tarefas em diferentes núcleos de processamento do DSP permitiu a execução de tarefas em paralelo, possibilitando uma simulação do desempenho do conversor PFC em tempo real e a validação do controle do sistema.

Para realizar a comunicação entre os diferentes núcleos de processamento do DSP TMS320F28379D, foi utilizada a memória RAMGS e RAMLs do DSP.

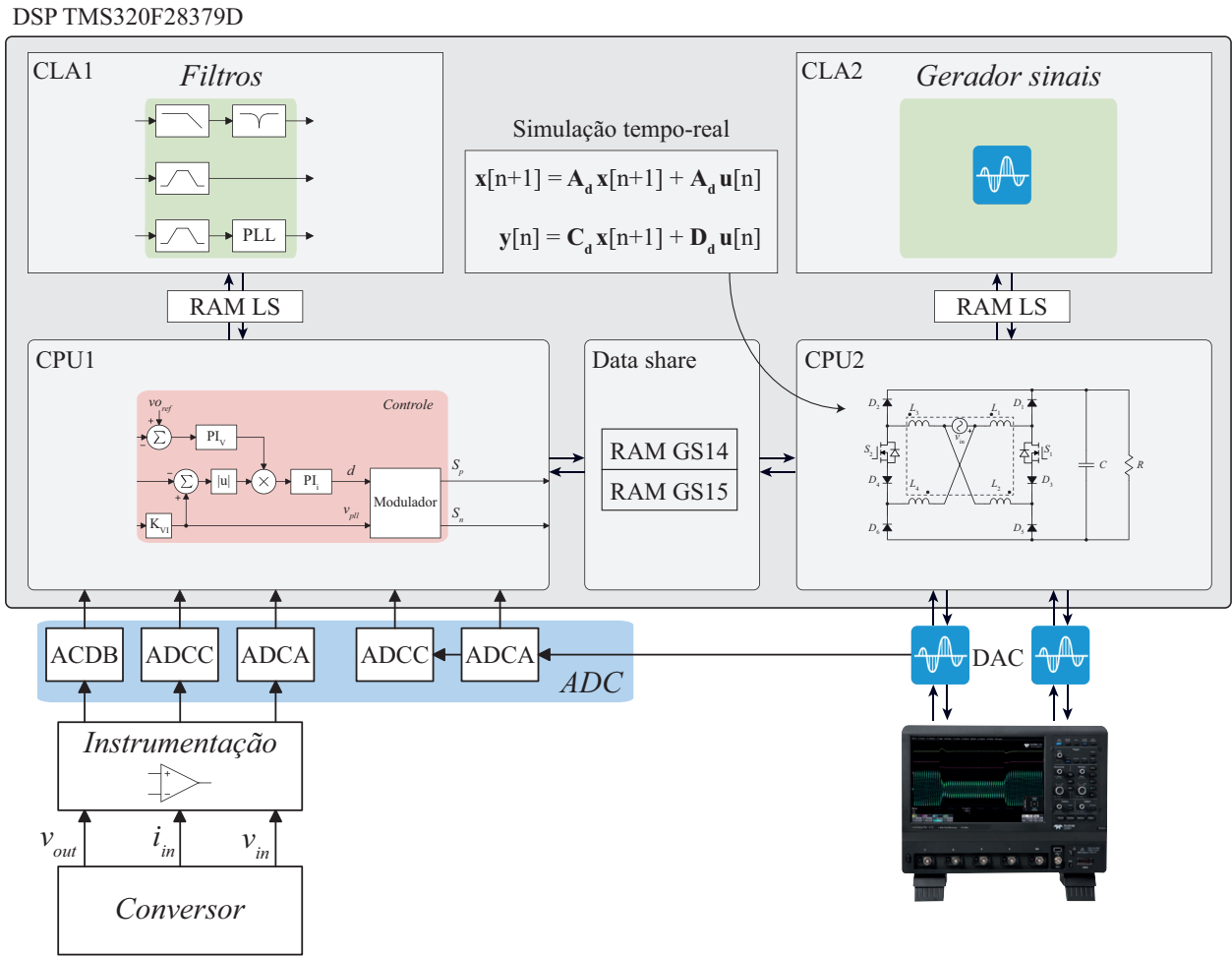


Figura 36 – Esquema ilustrativo da implementação do Hardware-in-the-Loop (HIL) nos dois CPU e CLA (Control Law Accelerator) do DSP (Digital Signal Processor) da Texas Instruments.

A RAMGS é a memória que pode ser compartilhada entre o CPU1 e o CPU2, permitindo a transferência de dados entre os núcleos de processamento. Já a RAMLS é uma memória compartilhada entre o CLA e o CPU.

Desta forma, o uso do DSP TMS320F28379D como HIL, possibilitou a realização de simulações do desempenho do conversor PFC e otimização do seu controle. A utilização de diferentes núcleos de processamento permitiu a execução dessas tarefas em paralelo.

Resultados da simulação de tempo real pode ser vista nas Figura 37 e 38. A figura 37 representa o momento onde é habilitado o controle do conversor, onde o retificado passa do modo de retificador passivo para ativo, é possível visualizar ainda um segundo onde a malha de tensão passa a atuar, isso ocorre quando a tensão do barramento é igual a 375 V. A segunda A figura 38, mostra um degrau carga de 50 % para 100 % e de 100 % para 50 % comprovando o funcionamento da malha de tensão e estabilidade nesta região.

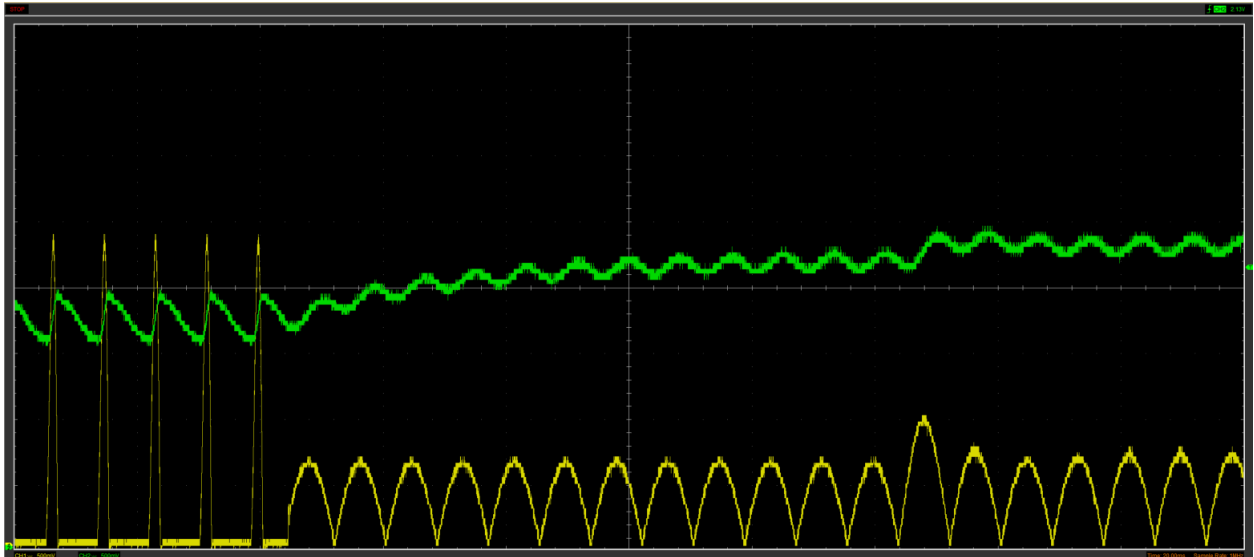


Figura 37 – Resultado de entrada de operação do conversor PFC, na parte inferior a corrente de entrada e na parte superior a tensão saída. A figura mostra o momento em que o controle é acionado, saindo do modo de retificador passivo para PFC ativo.

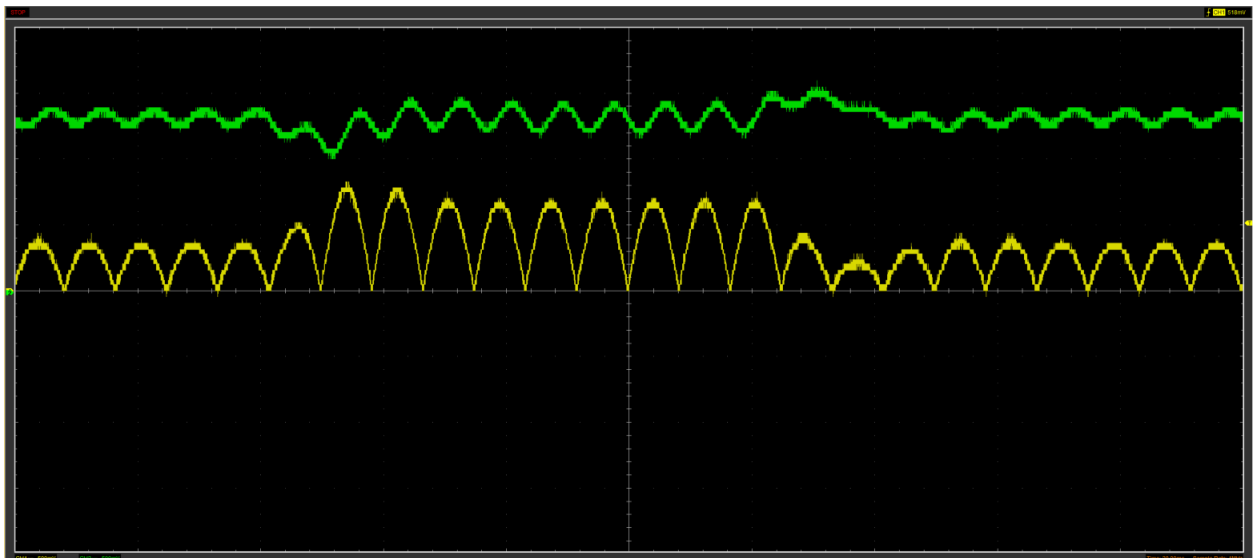


Figura 38 – A figura mostra um degrau carga de 50 % para 100 % e de 100 % para 50 %, na parte inferior a corrente de entrada e na parte superior a tensão saída.

6.2 RESULTADOS EXPERIMENTAIS DO CONVERSOR PROPOSTO

Para validar o projeto do conversor PFC e a estratégia de controle, a Figura 39 mostra o resultado do conversor operando em carga parcial, mostrando a tensão de entrada e saída e corrente no indutor. O canal C4 (I_{in}) mostra a corrente de entrada, o canal C2 (V_{in}) mostra a tensão de entrada e o canal C1 (V_{out}) mostram a tensão de saída.

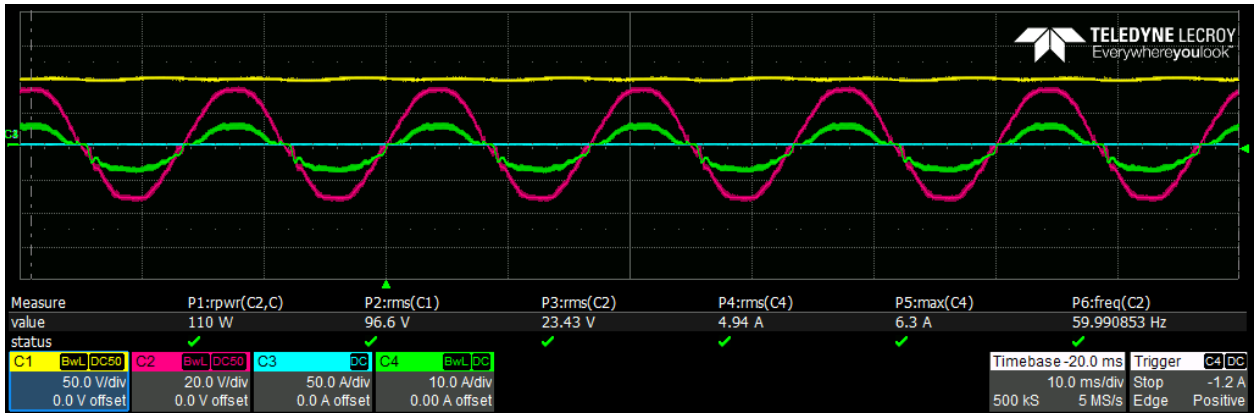


Figura 39 – Na figura é apresentado o conversor em regime permanente senoidal. No canal 1 é apresentado a tensão de saída do conversor, Canal 2 a entrada e por fim canal 3 a tensão de entrada do conversor.

A tensão de saída, mostrada na Figura 40, apresenta a ondulação em 120 Hz do barramento de tensão. A Figura 41 mostra a ondulação da corrente na entrada, o período da ondulação na Figura é de 2,66 us, equivalente a uma frequência de 375 kHz. A Figura 42 mostra um degrau carga de 50 % para 100 % e de 100 % para 50 % comprovando o funcionamento da malha de tensão e estabilidade nesta região. A Figura 43 semelhante a Figura 42, mostra as correntes no indutor L_1 e L_3 , respectivamente do semi ciclo positivo e negativo.

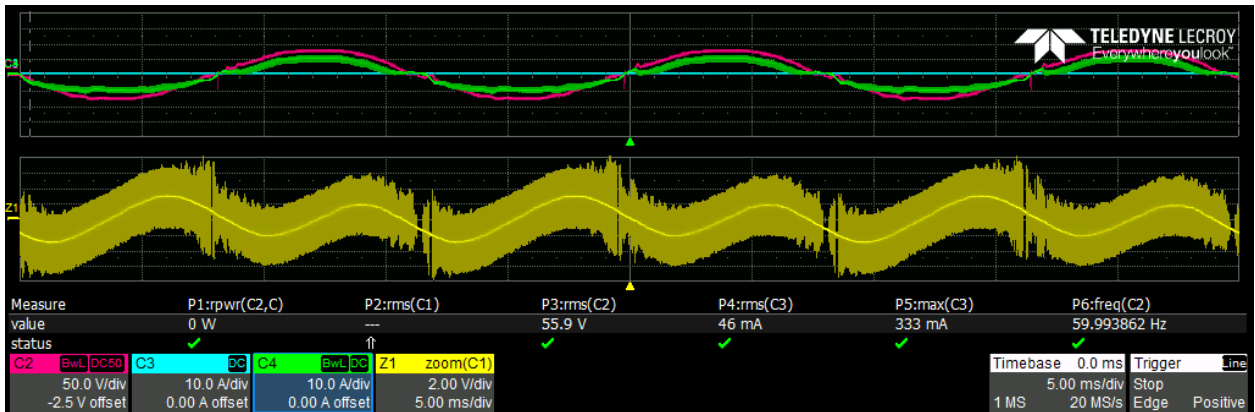


Figura 40 – Destaque na ondulação de tensão da saída V_{out} . É possível observar a componente em 120 Hz da ondulação do barramento, junto com a componente em alta frequência de comutação, 375 kHz

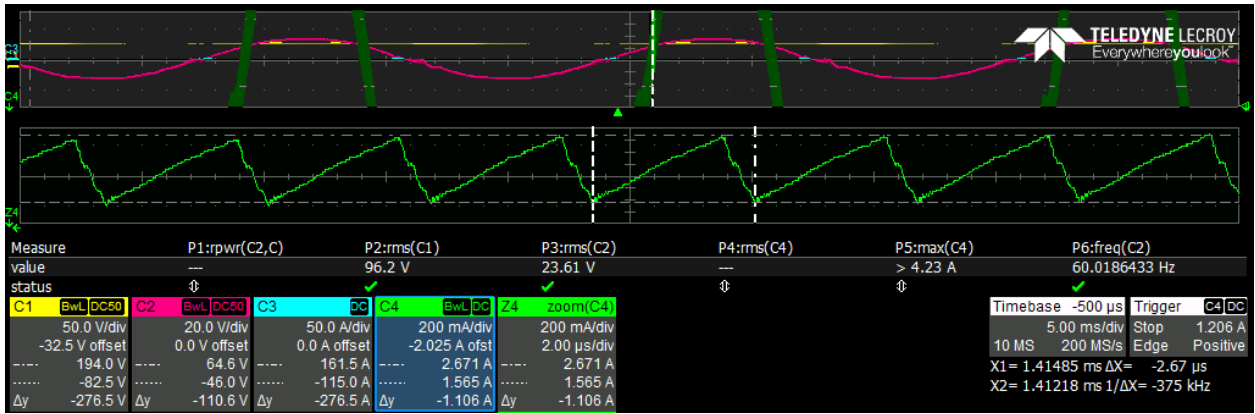


Figura 41 – Destaque na ondulação de corrente da entrada I_{in} . É possível observar a frequência da ondulação do indutor é de 375 kHz, que representa a frequência de comutação.

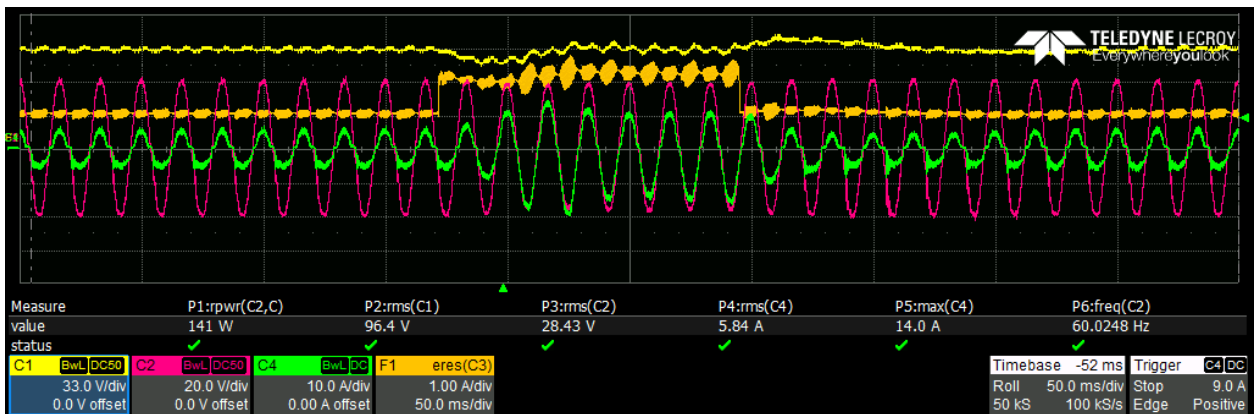


Figura 42 – O resultado para duas sequencias de degrau de carga. O primeiro degrau de 500 % para 100 % e segundo degrau de 100 % para 50 %, onde 100 % representa a carga nominal. É apresentado a corrente de saída, tensão de saída, tensão de entrada e corrente de entrada.

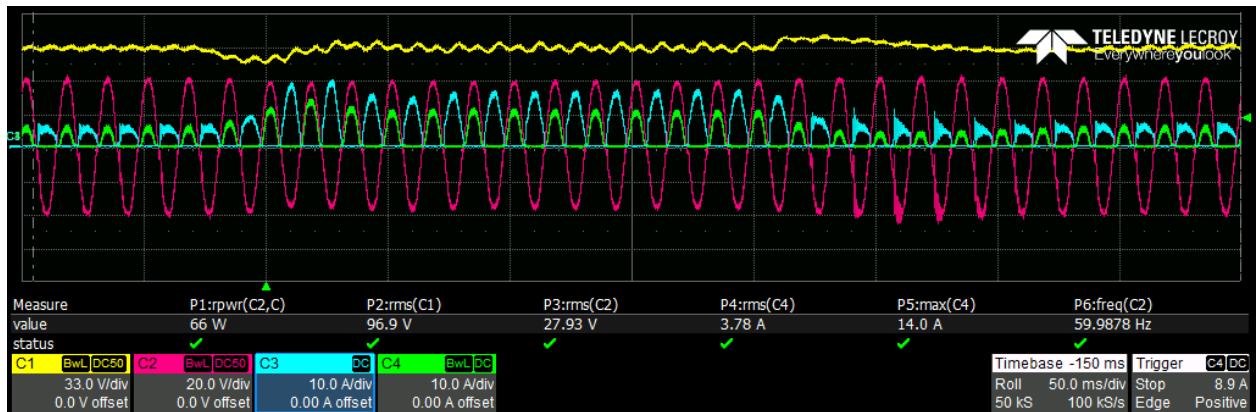


Figura 43 – O resultado para duas sequencias de degrau de carga. O primeiro degrau de 50 % para 100 % e segundo degrau de 100 % para 50 %, onde 100 % representa a carga nominal. É apresentado a tensão de saída, tensão de entrada e corrente no indutor L_1 e L_3 , respectivamente do semi ciclo positivo e negativo.

7 CONCLUSÃO

Em resumo, este trabalho abordou a modelagem, modulação e controle de uma nova topologia de conversor PFC. Embora tenha havido problemas técnicos na instrumentação que impediram a obtenção da potência e tensão nominal no conversor, os resultados obtidos em carga e tensão parcial demonstraram o funcionamento do conversor e das malhas de controle.

Durante o desenvolvimento do trabalho, foram exploradas tecnologias avançadas de semicondutores, como os transistores eGaN e os diodos SiC, permitindo a operação em alta frequência de comutação. A utilização de um microcontrolador DSP, como o TMS320F28379D, viabilizou a implementação de estratégias de controle e a criação de um ambiente de HIL para a validação das malhas de controle.

Além dos resultados e considerações mencionadas anteriormente, é importante ressaltar a vantagem da topologia proposta em relação à topologia bridgeless clássica em caso de falhas. Em situações em que ocorrem falhas nos interruptores do conversor, a topologia proposta pode continuar operando como uma retificação passiva, garantindo um fornecimento de energia contínuo e evitando interrupções indesejadas.

Essa capacidade de operação em falhas é de grande importância em aplicações críticas, onde a confiabilidade e a continuidade do fornecimento de energia são essenciais. Dessa forma, a nova topologia de conversor PFC apresenta uma solução mais robusta e confiável em comparação com a topologia bridgeless clássica. Outro modo de falha é situações em que ocorre a falha de apenas um dos interruptores do conversor, a topologia pode continuar operando em modo degradado, utilizando apenas o outro braço do conversor. Isso ocorre porque cada interruptor é responsável por meio ciclo da rede elétrica, garantindo que, mesmo com a falha de um dos interruptores, o conversor seja capaz de manter seu funcionamento com uma capacidade reduzida.

É importante destacar que o conversor desenvolvido apresenta uma redução significativa do modo comum. Embora não tenha sido explorado de forma prática e teórica devido a restrições de tempo, o modo comum reduzido foi investigado por meio de simulações. Essa característica é especialmente relevante, especialmente utilizando apenas um núcleo em comparação com algumas topologias convencionais apresentadas, que requerem um ou mais núcleos. Essa característica resulta em uma redução de núcleo, além de otimizar o espaço ocupado pelo conversor.

Dessa forma, podemos concluir que o trabalho alcançou seu objetivo principal, que foi a proposição de uma nova topologia de conversor PFC monofásico operando em alta frequência de comutação, além de abordar sua modelagem, modulação e controle. A validação do funcionamento do conversor e das malhas de controle por meio de resultados experimentais

demonstra a viabilidade da implementação da topologia proposta.

7.1 TRABALHOS FUTUROS

- Explorar a parte teórica e prática da tensão de modo comum no conversor.
- Explorar os modos de falhas validando as vantagens da topologia
- Obter resultados nominais do conversor.
- Testar outras variações da topologia apresentada, fazendo comparativo nos esforços nos componentes, tensão de modo comum.
- Caracterização dos indutores acoplados, modelagem dos parasitas do núcleo e a influência no conversor.
- Resultados de emissão eletromagnética conduzida no conversor, comparando com as 4 topologias apresentadas.
- Finalizar a modelagem das perdas para outros pontos de operação.

REFERÊNCIAS

- AG, Infineon Technologies. **Data Sheet: IDL12G65C5 SIC Diode**. [*S.l.: s.n.*], 2019. https://www.mouser.com/datasheet/2/196/Infineon-IDL12G65C5-DS-v02_00-en-1131071.pdf.
- ARBUGERI, José. **MasterThesisJoseArbugeri**. [*S.l.*]: GitHub, 2023. <https://github.com/Jarbugeri/MasterThesisJoseArbugeri>.
- BORGONOVO, D.; REMOR, J.P.; PERIN, A.J.; BARBI, I. A Self-Controlled Power Factor Correction Single-Phase Boost Pre-Regulator. *In*: 2005 IEEE 36th Power Electronics Specialists Conference. [*S.l.: s.n.*], 2005. P. 2351–2357.
- BORGONOVO, Deivis; HELDWEIN, Marcelo L.; MUSSA, Samir A. Application of the NIOS II processor-FPGA on the digital control of a single-phase PFC rectifier. *In*: 2008 11th Workshop on Control and Modeling for Power Electronics. [*S.l.: s.n.*], 2008. P. 1–7.
- BORGONOVO, Deivis; MUSSA, Samir Ahmad. Single-phase boost PFC voltage-doubler self-controlled using FPGA. *In*: 2008 IEEE Power Electronics Specialists Conference. [*S.l.: s.n.*], 2008. P. 4457–4463.
- CHEN, Hung-Chi. Duty Phase Control for Single-Phase Boost-Type SMR. **IEEE Transactions on Power Electronics**, v. 23, n. 4, p. 1927–1934, 2008.
- CHEN, Zhengge; DAVARI, Pooya; WANG, Huai. Single-Phase Bridgeless PFC Topology Derivation and Performance Benchmarking. **IEEE Transactions on Power Electronics**, v. 35, n. 9, p. 9238–9250, 2020.
- FIGUERES, Emilio; BENAVENT, Jos Manuel; GARCERA, Gabriel; PASCUAL, Marcos. A Control Circuit With Load-Current Injection for Single-Phase Power-Factor-Correction Rectifiers. **IEEE Transactions on Industrial Electronics**, v. 54, n. 3, p. 1272–1281, 2007.
- GAN SYSTEMS. **Gate Driver Design with GaN E-HEMTs**. [*S.l.: s.n.*], 2022. https://gansystems.com/wp-content/uploads/2022/03/GN012_Gate-Driver-Design-with-GaN-E-HEMTs_220308.pdf. Acessado em 23 de maio de 2023.

GHODKE, Dharmraj V.; SREERAJ, E.S.; CHATTERJEE, Kishore; FERNANDES, B. G. One-Cycle-Controlled Bidirectional AC-to-DC Converter With Constant Power Factor. **IEEE Transactions on Industrial Electronics**, v. 56, n. 5, p. 1499–1510, 2009.

GU, Bin; DOMINIC, Jason; CHEN, BaiFeng; LAI, Jih-Sheng. A high-efficiency single-phase bidirectional AC-DC converter with minimized common mode voltages for battery energy storage systems. *In*: 2013 IEEE Energy Conversion Congress and Exposition. [*S.l.*: *s.n.*], 2013. P. 5145–5149.

HULIEHEL, F.A.; LEE, F.C.; CHO, B.H. Small-signal modeling of the single-phase boost high power factor converter with constant frequency control. *In*: PESC '92 Record. 23rd Annual IEEE Power Electronics Specialists Conference. [*S.l.*: *s.n.*], 1992. 475–482 vol.1.

HURLEY, W.G.; WOLFLE, W.H. **Transformers and Inductors for Power Electronics**. [*S.l.*]: John Wiley & Sons, Ltd, 2013.

JACOBOSKI, Marcos José; BASTIANI LANGE, André de; HELDWEIN, Marcelo Lobo. Closed-Form Solution for Core Loss Calculation in Single-Phase Bridgeless PFC Rectifiers Based on the iGSE Method. **IEEE Transactions on Power Electronics**, v. 33, n. 6, p. 4599–4604, 2018.

JACOBOSKI, Marcos José; BASTIANI LANGE, André de; HELDWEIN, Marcelo Lobo. Closed-Form Solutions for Core and Winding Losses Calculation in Single-Phase Boost PFC Rectifiers. *In*: 2019 IEEE 15th Brazilian Power Electronics Conference and 5th IEEE Southern Power Electronics Conference (COBEP/SPEC). [*S.l.*: *s.n.*], 2019. P. 1–6.

JAPPE, Tiago Kommers; MUSSA, Samir Ahmad. Current technique applied in single phase PFC boost converter based on discrete-time One Cycle Control. *In*: 2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC). [*S.l.*: *s.n.*], 2011. P. 1–5.

JAPPE, Tiago Kommers; MUSSA, Samir Ahmad; COUTINHO, Daniel. Lyapunov based stability analysis of current self-control approach for single-phase PFC Boost converters subject to load variations. *In*: IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society. [*S.l.*]: IEEE, nov. 2011.

- JONES, E. A. *et al.* Characterization of an enhancement-mode 650-V GaN HFET. *In: IEEE. 2015 IEEE Energy Conversion Congress and Exposition (ECCE)*. [S.l.: s.n.], 2015. P. 400–407.
- JONES, E. A.; WANG, F.; COSTINETT, D.; ZHANG, Z.; GUO, B. Temperature-dependent turn-on loss analysis for GaN HFETs. *In: IEEE. 2016 IEEE Applied Power Electronics Conference and Exposition (APEC)*. [S.l.: s.n.], 2016. P. 1010–1017.
- LAI, Zheren; SMEDLEY, K.M. A family of continuous-conduction-mode power-factor-correction controllers based on the general pulse-width modulator. **IEEE Transactions on Power Electronics**, v. 13, n. 3, p. 501–510, 1998.
- LANGE, André De Bastiani; HELDWEIN, Marcelo Lobo. Light load stability improvement for single-phase boost PFC rectifier using input current self-control technique. *In: XI Brazilian Power Electronics Conference*. [S.l.: s.n.], 2011. P. 61–67.
- LANGE, André De Bastiani; SOEIRO, Thiago Batista; ORTMANN, Márcio Silveira; HELDWEIN, Marcelo Lobo. Three-Level Single-Phase Bridgeless PFC Rectifiers. **IEEE Transactions on Power Electronics**, v. 30, n. 6, p. 2935–2949, 2015.
- LIU, Zhengyang; LEE, Fred C.; LI, Qiang; YANG, Yuchen. Design of GaN-Based MHz Totem-Pole PFC Rectifier. **IEEE Journal of Emerging and Selected Topics in Power Electronics**, v. 4, n. 3, p. 799–807, 2016.
- MUNARETTO, Lucas; HELDWEIN, Marcelo L. Single-phase transformerless dual buck-based grid-connected inverter. *In: 2018 IEEE Applied Power Electronics Conference and Exposition (APEC)*. [S.l.: s.n.], 2018. P. 161–166.
- RAO, Vishnu Murahari; JAIN, Amit Kumar; REDDY, Kishore K.; BEHAL, Aman. Experimental Comparison of Digital Implementations of Single-Phase PFC Controllers. **IEEE Transactions on Industrial Electronics**, v. 55, n. 1, p. 67–78, 2008.
- RAO, Vishnu Murahari; JAIN, Amit Kumar; REDDY, Kishore K.; BEHAL, Aman. Experimental Comparison of Digital Implementations of Single-Phase PFC Controllers. **IEEE Transactions on Industrial Electronics**, v. 55, n. 1, p. 67–78, 2008.

ROSCHILD, Jonatas M.; COUTINHO, Daniel; SOUZA, Carlos E. de. Robust tuning of current self-controlled single-phase PFC boost converters. *In: 2015 IEEE Conference on Control Applications (CCA)*. [S.l.: s.n.], 2015. P. 478–483.

SCHWARZ, Francisc C. An Improved Method of Resonant Current Pulse Modulation for Power Converters. **IEEE Transactions on Industrial Electronics and Control Instrumentation**, IECI-23, n. 2, p. 133–141, 1976.

SENSITEC. **Sensitec CFS1000**. [S.l.: s.n.], s.d.

<https://www.sensitec.com/products-solutions/current-measurement/cfs1000>.

Acessado em 23 de maio de 2023.

SINGH, Bhim; SINGH, Sanjeev; CHANDRA, Ambrish; AL-HADDAD, Kamal. Comprehensive Study of Single-Phase AC-DC Power Factor Corrected Converters With High-Frequency Isolation. **IEEE Transactions on Industrial Informatics**, v. 7, n. 4, p. 540–556, 2011.

SIU, Ken K.M.; HO, Carl N.M. A critical review of Bridgeless PFC boost rectifiers with common-mode voltage mitigation. *In: IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. [S.l.: s.n.], 2016. P. 3654–3659.

SMEDLEY, K.M.; CUK, S. One-cycle control of switching converters. **IEEE Transactions on Power Electronics**, v. 10, n. 6, p. 625–633, 1995.

GLOSSÁRIO

IDL12G65C5 Diodo retificador SiC fabricado pela Infineon Technologies

TMS320F28379D TMS320F28379D é um microcontrolador de sinal digital (DSP) de 32 bits fabricado pela Texas Instruments

APÊNDICE A – APÊNDICE A

A.1 FILTROS SEGUNDA ORDEM DIGITAIS

Os filtros são dispositivos usados para permitir a passagem de certas frequências e atenuar outras. Eles são amplamente utilizados em sistemas eletrônicos para filtrar sinais indesejados ou para enfatizar determinados componentes de um sinal. Existem diferentes tipos de filtros, como filtros passa-baixa, passa-alta, passa-banda e rejeita-banda.

A resposta de um filtro é caracterizada por sua frequência central e sua largura de banda, que determinam as frequências nas quais o filtro tem a máxima atenuação e as frequências que ele permite passar. O ganho do filtro em sua frequência central é determinado pelo fator de qualidade do filtro, que é uma medida da "seletividade" do filtro.

Nesse apêndice, serão apresentados os procedimentos de discretização dos filtros mencionados anteriormente, ou seja, o filtro passa-baixa (LPF), filtro passa-banda (BPF), passa-alta (HBP) e rejeita-banda (BRF) utilizando a transformação bilinear de Tustin. Isso permitirá que esses filtros sejam implementados em sistemas digitais, mantendo suas características de resposta em frequência semelhantes às dos filtros analógicos correspondentes.

Tipo de Filtro	Equação
Filtro passa-baixas	$\frac{\omega_0^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$
Filtro passa-altas	$\frac{s^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$
Filtro passa-banda	$\frac{s\omega_0/Q}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$
Filtro rejeita-banda	$\frac{s^2 + \omega_0^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$

Tabela 8 – Equações dos 4 tipos de filtros de segunda ordem

Essa tabela apresenta os quatro tipos de filtros referentes ao Q Factor: passa-baixas, passa-altas, passa-banda e rejeita-banda. Cada filtro possui uma equação específica para sua implementação, sendo que todas as equações apresentam o parâmetro Q, que representa a qualidade do filtro e pode ser ajustado para alterar a sua largura de banda e sua seletividade.

A forma genérica para uma equação de segunda ordem é dada por:

$$\frac{Y(s)}{X(s)} = \frac{As^2 + Bs + C}{Ds^2 + Es + F}$$

Para discretizar esta equação, podemos usar a discretização de Tustin, também conhecida

como transformação bilinear. Esta transformação é dada por:

$$\frac{2(z-1)}{T(z+1)} \rightarrow S$$

Substituindo esta equação na equação de segunda ordem, obtemos:

$$Y(z) \left(F + \frac{2E(z-1)}{T(z+1)} + \frac{4D(z-1)^2}{T^2(z+1)^2} \right) = X(z) \left(C + \frac{2B(z-1)}{T(z+1)} + \frac{4A(z-1)^2}{T^2(z+1)^2} \right)$$

Realizando o mínimo comum, temos:

$$Y(z) \left(\frac{4D(z-1)^2 + 2ET(z-1)(z+1) + FT^2(z+1)^2}{T^2(z+1)^2} \right) = X(z) \left(\frac{4A(z-1)^2 + 2BT(z-1)(z+1) + CT^2(z+1)^2}{T^2(z+1)^2} \right)$$

Isolando os termos em $y(k)$, temos:

$$\begin{aligned} Y(z)(z^0(4D + 2ET + FT^2)) + z^{-1}(-8D + 2FT^2) \\ + z^{-2}(3D - 2ET + FT^2) \\ = X(z)(z^0(4A + 2BT + CT^2) + z^{-1}(-8A + 2CT^2) \\ + z^{-2}(3A - 2BT + CT^2)) \end{aligned}$$

Isolando $Y(z)$, temos:

$$\begin{aligned} Y(z)z^0 &= X(z)z^0 \left(\frac{4A + 2BT + CT^2}{AUX} \right) + \\ &X(z)z^{-1} \left(\frac{-8A + 2CT^2}{AUX} \right) + \\ &X(z)z^{-2} \left(\frac{3A - 2BT + CT^2}{AUX} \right) - \\ &Y(z)z^{-1} \left(\frac{-8D + 2FT^2}{AUX} \right) - \\ &Y(z)z^{-2} \left(\frac{3D - 2ET + FT^2}{AUX} \right) \end{aligned}$$

Onde:

$$AUX = (4D + 2ET + FT^2)$$

Aplicando a transformação do domínio Z para a equação das diferenças, temos:

$$Y[z] = b_0 X[z] + b_1 X[z-1] + b_2 X[z-2] - a_1 Y[z-1] - a_2 Y[z-2]$$

onde,

$$\begin{aligned}
 a_0 &= 1 \\
 a_1 &= \frac{-8D + 2FT^2}{AUX} \\
 a_2 &= \frac{3D - 2ET + FT^2}{AUX} \\
 b_0 &= \frac{4A + 2BT + CT^2}{AUX} \\
 b_1 &= \frac{-8A + 2CT^2}{AUX} \\
 b_2 &= \frac{3A - 2BT + CT^2}{AUX}
 \end{aligned}$$

A.1.1 Filtro passa baixa (LPF)

Para o filtro passa baixa temos a equação descrita como:

$$LPO(s) = \frac{\omega_0^2}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$$

Os termos genéricos da forma apresentada anteriormente são:

$$\begin{aligned}
 A &= 0 \\
 B &= 0 \\
 C &= w^2 \\
 D &= 1 \\
 E &= \frac{w}{Q} \\
 F &= w^2
 \end{aligned}$$

substituindo na forma genérica dos coeficientes discretizados obtemos:

$$\begin{aligned}
 a_0 &= 1 \\
 a_1 &= \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 a_2 &= \frac{T^2 w^2 - \frac{2 T w}{Q} + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_0 &= \frac{T^2 w^2}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_1 &= \frac{2 T^2 w^2}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_2 &= \frac{T^2 w^2}{T^2 w^2 + \frac{2 T w}{Q} + 4}
 \end{aligned}$$

$$\begin{aligned}
 LPO[z] = Y[z] &= + \frac{T^2 w^2}{T^2 w^2 + \frac{2 T w}{Q} + 4} X[z] \\
 &+ \frac{2 T^2 w^2}{T^2 w^2 + \frac{2 T w}{Q} + 4} X[z - 1] \\
 &+ \frac{T^2 w^2}{T^2 w^2 + \frac{2 T w}{Q} + 4} X[z - 2] \\
 &- \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} Y[z - 1] \\
 &- \frac{T^2 w^2 - \frac{2 T w}{Q} + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} Y[z - 2]
 \end{aligned}$$

A resposta ao degrau do filtro discretizado $LPO[z]$ comparado ao filtro contínuo $LPO[s]$ pode ser visualizado na Figura 44

A.1.2 Filtro passa banda (BPF)

Para o filtro passa banda temos a equação descrita como:

$$BPF(s) = \frac{s\omega_0/Q}{s^2 + s\frac{\omega_0}{Q} + \omega_0^2}$$

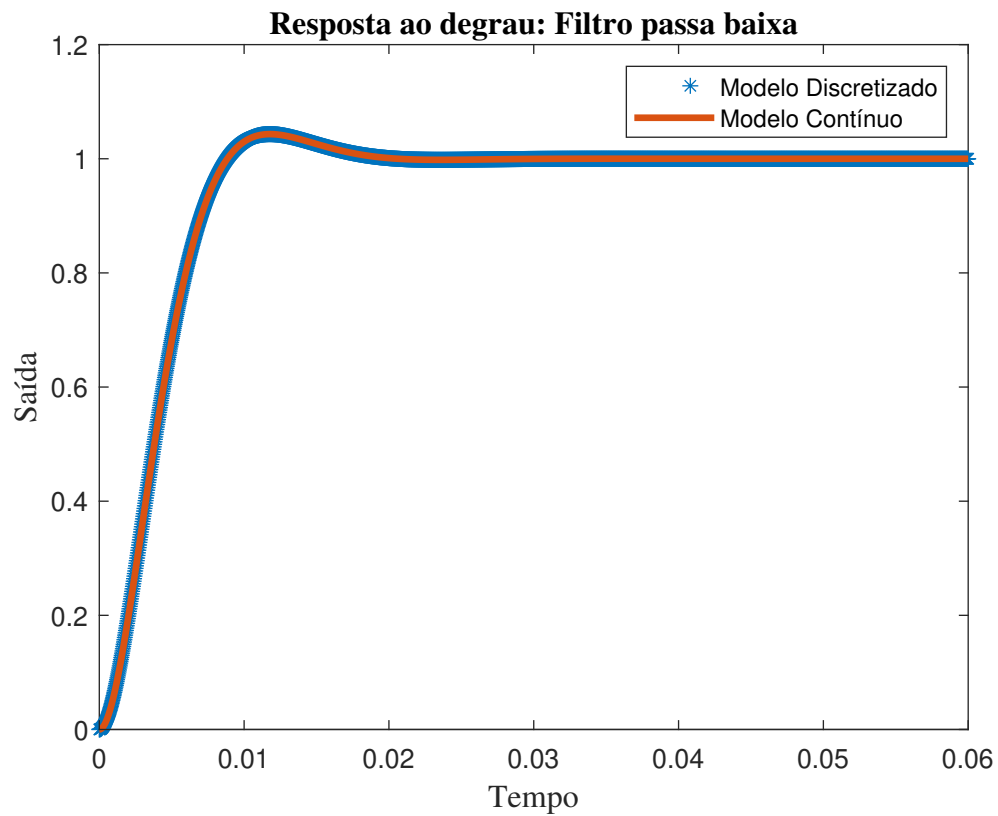


Figura 44 – Resposta ao degrau filtro passa baixa, do modelo Contínuo e modelo discretizado

Os termos genéricos da forma apresentada anteriormente são:

$$A = 0$$

$$B = \frac{w}{Q}$$

$$C = 0$$

$$D = 1$$

$$E = \frac{w}{Q}$$

$$F = w^2$$

substituindo na forma genérica dos coeficientes discretizados obtemos:

$$\begin{aligned}
 a_0 &= 1 \\
 a_1 &= \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 a_2 &= \frac{T^2 w^2 - \frac{2 T w}{Q} + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_0 &= \frac{2 T w}{Q \left(T^2 w^2 + \frac{2 T w}{Q} + 4 \right)} \\
 b_1 &= 0 \\
 b_2 &= -\frac{2 T w}{Q \left(T^2 w^2 + \frac{2 T w}{Q} + 4 \right)}
 \end{aligned}$$

$$\begin{aligned}
 BPF[z] = Y[z] &= + \frac{2 T w}{Q \left(T^2 w^2 + \frac{2 T w}{Q} + 4 \right)} X[z] \\
 &- \frac{2 T w}{Q \left(T^2 w^2 + \frac{2 T w}{Q} + 4 \right)} X[z - 2] \\
 &- \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} Y[z - 1] \\
 &- \frac{T^2 w^2 - \frac{2 T w}{Q} + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} Y[z - 2]
 \end{aligned}$$

A resposta ao degrau do filtro discretizado $BPF[z]$ comparado ao filtro contínuo $BPF[s]$ pode ser visualizado na Figura 45

A.1.3 Filtro rejeita faixa (BRF)

Para o filtro rejeita faixa (Notch) temos a equação descrita como:

$$BRF(s) = \frac{s^2 + \omega_0^2}{s^2 + s \frac{\omega_0}{Q} + \omega_0^2}$$

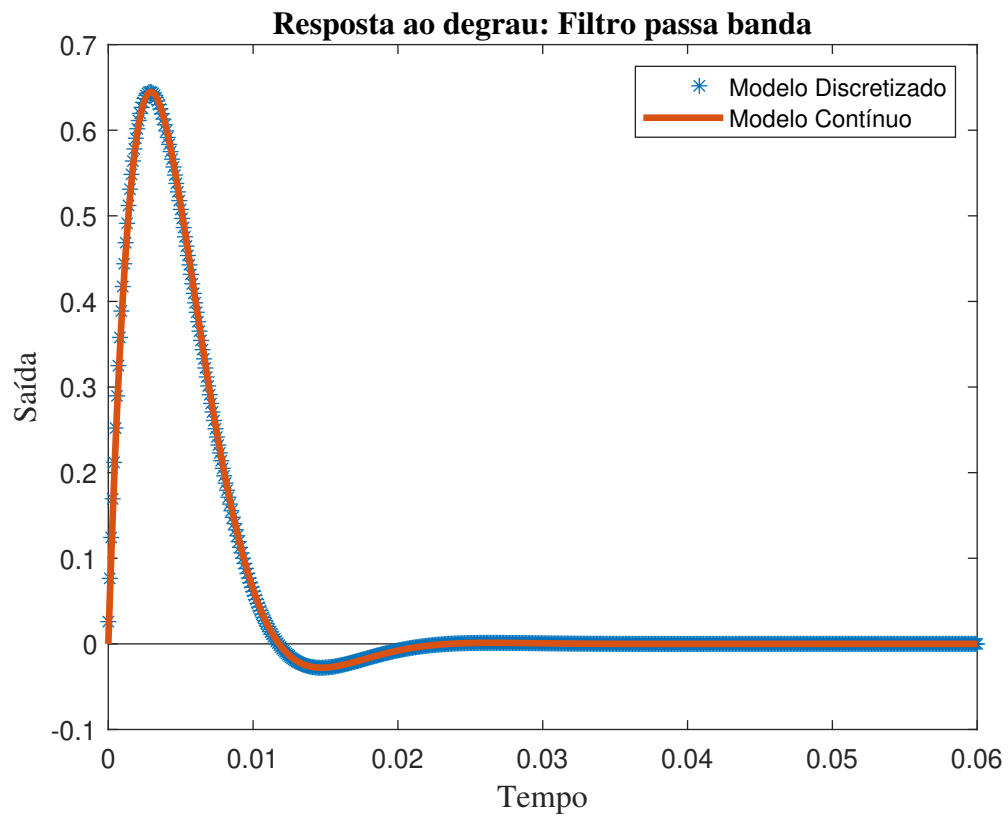


Figura 45 – Resposta ao degrau filtro passa banda, do modelo Contínuo e modelo discretizado

Os termos genéricos da forma apresentada anteriormente são:

$$A = 1$$

$$B = 0$$

$$C = w^2$$

$$D = 1$$

$$E = \frac{w}{Q}$$

$$F = w^2$$

substituindo na forma genérica dos coeficientes discretizados obtemos:

$$\begin{aligned}
 a_0 &= 1 \\
 a_1 &= \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 a_2 &= \frac{T^2 w^2 - \frac{2 T w}{Q} + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_0 &= \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_1 &= \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} \\
 b_2 &= \frac{T^2 w^2 + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4}
 \end{aligned}$$

$$\begin{aligned}
 BPF[z] = Y[z] &= + \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} X[z] \\
 &+ \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} X[z - 1] \\
 &+ \frac{T^2 w^2 + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} X[z - 2] \\
 &- \frac{2 T^2 w^2 - 8}{T^2 w^2 + \frac{2 T w}{Q} + 4} Y[z - 1] \\
 &- \frac{T^2 w^2 - \frac{2 T w}{Q} + 4}{T^2 w^2 + \frac{2 T w}{Q} + 4} Y[z - 2]
 \end{aligned}$$

A resposta ao degrau do filtro discretizado $BRF[z]$ comparado ao filtro contínuo $BRF[s]$ pode ser visualizado na Figura 46

A.1.4 Filtro rejeita faixa (HPF)

Para o filtro passa alta temos a equação descrita como:

$$BRF(s) = \frac{s^2}{s^2 + s \frac{\omega_0}{Q} + \omega_0^2}$$

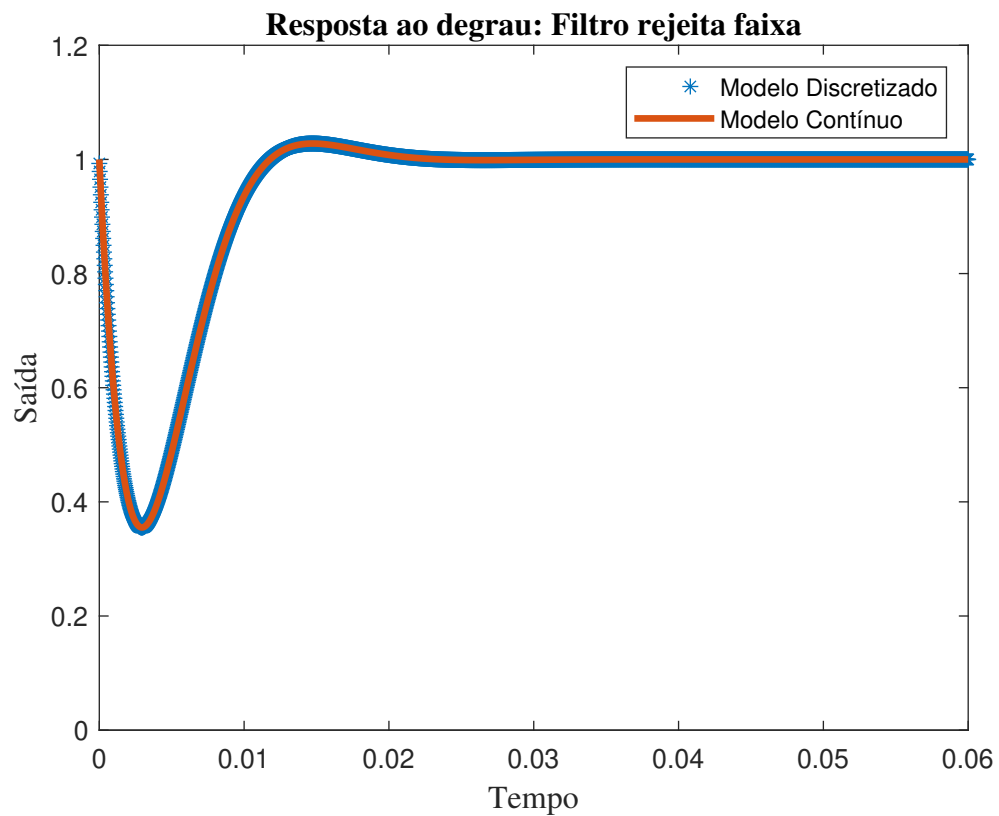


Figura 46 – Resposta ao degrau filtro rejeita faixa, do modelo Contínuo e modelo discretizado

Os termos genéricos da forma apresentada anteriormente são:

$$A = 1$$

$$B = 0$$

$$C = 0$$

$$D = 1$$

$$E = \frac{w}{Q}$$

$$F = w^2$$

substituindo na forma genérica dos coeficientes discretizados obtemos:

$$\begin{aligned}
 a_0 &= 1 \\
 a_1 &= \frac{2T^2 w^2 - 8}{T^2 w^2 + \frac{2Tw}{Q} + 4} \\
 a_2 &= \frac{T^2 w^2 - \frac{2Tw}{Q} + 4}{T^2 w^2 + \frac{2Tw}{Q} + 4} \\
 b_0 &= \frac{4}{T^2 w^2 + \frac{2Tw}{Q} + 4} \\
 b_1 &= -\frac{8}{T^2 w^2 + \frac{2Tw}{Q} + 4} \\
 b_2 &= \frac{4}{T^2 w^2 + \frac{2Tw}{Q} + 4}
 \end{aligned}$$

$$\begin{aligned}
 BPF[z] = Y[z] &= + \frac{4}{T^2 w^2 + \frac{2Tw}{Q} + 4} X[z] \\
 &- \frac{8}{T^2 w^2 + \frac{2Tw}{Q} + 4} X[z-1] \\
 &+ \frac{4}{T^2 w^2 + \frac{2Tw}{Q} + 4} X[z-2] \\
 &- \frac{2T^2 w^2 - 8}{T^2 w^2 + \frac{2Tw}{Q} + 4} Y[z-1] \\
 &- \frac{T^2 w^2 - \frac{2Tw}{Q} + 4}{T^2 w^2 + \frac{2Tw}{Q} + 4} Y[z-2]
 \end{aligned}$$

A resposta ao degrau do filtro discretizado $HPF[z]$ comparado ao filtro contínuo $HPF[s]$ pode ser visualizado na Figura 47

A.1.5 Biblioteca em C dos filtros discretizado

O código apresentado implementa uma estrutura para um filtro de segunda ordem com os tipos de filtros: passa-baixa, passa-banda, rejeita-banda e passa-alta. A estrutura do filtro contém as informações sobre os coeficientes do filtro e os estados de entrada e saída do filtro. Além disso, são definidas funções para atualizar os coeficientes dos filtros passa-baixa e passa-banda e para realizar a pré-distorção da frequência de corte do filtro. A pré-distorção é feita para ajustar a frequência de corte do filtro de acordo com a frequência de amostragem. O código foi escrito em linguagem C.

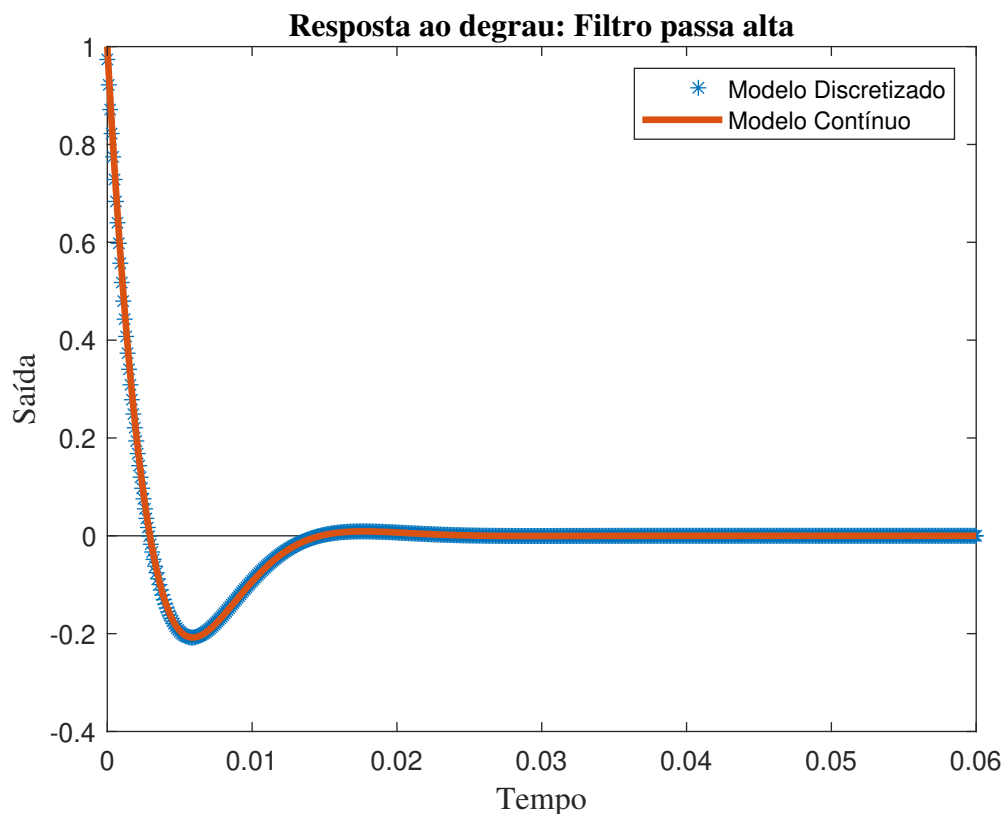


Figura 47 – Resposta ao degrau filtro passa alta, do modelo Contínuo e modelo discretizado

O arquivo header "so_filter_t" é responsável por definir as funções e estruturas de dados necessárias para implementar filtros digitais de segunda ordem. Ele inclui o arquivo "F28x_Project.h", que é um arquivo de cabeçalho utilizado para desenvolver projetos para a família de microcontroladores da Texas Instruments F28x.

O arquivo header define quatro tipos de filtros digitais de segunda ordem, que são lowpass, bandpass, notch e highpass, e permite ao usuário escolher se deseja aplicar a distorção de frequência de corte prévia ou não.

Ele também define três estruturas de dados: "coeffs_t", que contém os coeficientes do filtro; "states_t", que contém os estados de entrada e saída; e "so_filter_t", que contém todos os parâmetros de configuração e estados dos filtros digitais de segunda ordem.

Além disso, o arquivo header fornece protótipos de funções específicas para cada tipo de filtro e funções genéricas para qualquer filtro, incluindo inicialização, pré-distorção de frequência de corte, reset e processamento do filtro. As funções específicas para cada tipo de filtro são implementadas separadamente para o compilador da CPU e o compilador da CLA (Control Law Accelerator) da Texas Instruments.

```
2  * so_filter.h
3  *
4  * Created on: 18 de jan de 2023
5  * Author: Jose
6  */
7
8  #ifndef SO_FILTER_H_
9  #define SO_FILTER_H_
10
11 /*****
12  * Includes *
13  *****/
14
15 #include "F28x_Project.h"
16
17 /*****
18  * Defines *
19  *****/
20
21 #define SO_FILTER_LOWPASS          0
22 #define SO_FILTER_BANDPASS        1
23 #define SO_FILTER_NOTCH           2
24 #define SO_FILTER_HIGHPASS        3
25
26 #define PRE_WARP_ON                1
27 #define PRE_WARP_OFF               0
28
29 /*****
30  * Structs *
31  *****/
32
33 typedef struct _coeffs_t_
34 {
35     ///< Coeficientes do filtro
36     ///< a0 deve ser 1 para esse metodo
37     float a1;
```

```
38     float a2;
39     float b0;
40     float b1;
41     float b2;
42 }coeffs_t;
43
44 typedef struct _states_t_
45 {
46     ///< Estados de entrada e saida
47     float u0;    ///< Estado entrada u[n]
48     float u1;    ///< Estado entrada u[n-1]
49     float u2;    ///< Estado entrada u[n-2]
50     float y0;    ///< Estado entrada y[n]
51     float y1;    ///< Estado entrada y[n-1]
52     float y2;    ///< Estado entrada y[n-2]
53 }states_t;
54
55 typedef struct _so_filter_t_
56 {
57     ///< Parametros de configuracao do SOGI
58     Uint32  type;          ///< Tipo do filtro: 0 lowpass, 1 bandpass, 2
↪ notch e 3 highpass
59     Uint32  is_prewarped; ///< 1 Frequencia de corte pre distorcida, 0
↪ não distorcida
60     float Q;              ///< Fator de qualidade do filtro
61     float fs;            ///< Frequencia de amostragem do filtro
62     float ts;            ///< Período de amostragem do do filtro,
↪ calculado na inicializacao
63     float fc;            ///< Frequencia de corte em Hertz
64     float wc;            ///< Frequencia de corte em radianos/s,
↪ calculado na inicializacao a partir de fr
65
66     ///< Coeficientes e estados dos filtro sogi
67     coeffs_t coeffs;     ///< Coeficientes do filtro de segunda ordem
68     states_t states;     ///< Estados de entrada e saida do filtro
69 }so_filter_t;
```

```

70
71 /*****
72  * Functions prototypes *
73  *****/
74
75 #if !defined(__TMS320C28XX_CLA__)    //!< CPU Compiler
76     /* Funcoes especificas de cada tipo de filtro */
77     void so_filter_update_coeff_lowpass(so_filter_t * so_filter);
78     void so_filter_update_coeff_bandpass(so_filter_t * so_filter);
79     void so_filter_update_coeff_notch(so_filter_t * so_filter);
80     void so_filter_update_coeff_highpass(so_filter_t * so_filter);
81     /* Funcoes generica para qualquer filtro */
82     void so_filter_initialize(so_filter_t * so_filter, float fs, float fc,
↪ float Q, Uint32 type, Uint32 is_prewarped);
83     void so_filter_prewarp_frequency(so_filter_t * so_filter, float fc);
84     void so_filter_reset(so_filter_t * so_filter, float value);
85     void so_filter_run(so_filter_t * so_filter, float input);
86 #elif defined(__TMS320C28XX_CLA__)    //!< CLA Compiler
87     /* Funcoes especificas de cada tipo de filtro */
88     void so_filter_update_coeff_lowpass_cla(so_filter_t * so_filter);
89     void so_filter_update_coeff_bandpass_cla(so_filter_t * so_filter);
90     void so_filter_update_coeff_notch_cla(so_filter_t * so_filter);
91     void so_filter_update_coeff_highpass_cla(so_filter_t * so_filter);
92     /* Funcoes generica para qualquer filtro */
93     void so_filter_reset_cla(so_filter_t * so_filter, float value);
94     void so_filter_run_cla(so_filter_t * so_filter, float input);
95
96 #endif
97
98
99 #endif /* SO_FILTER_H_ */

```

O arquivo .c fornecido é um código fonte escrito em linguagem C, que implementa funções relacionadas ao processamento de filtros digitais. O arquivo começa com a inclusão de dois arquivos de cabeçalho, "so_filter.h" e "math.h".

Em seguida, são definidas três funções principais que realizam o cálculo dos coeficientes

dos filtros digitais para diferentes tipos de filtro: passa-baixa, passa-faixa e notch. Cada função recebe um ponteiro para uma estrutura "so_filter_t" que contém as informações necessárias para o cálculo dos coeficientes. Essas informações incluem a frequência de amostragem do sinal (fs), o tempo de amostragem (ts), a frequência de corte do filtro (fc) e o fator de qualidade (Q).

A primeira função, "so_filter_prewarp_frequency", realiza a distorção prévia da frequência de corte do filtro para compensar as diferenças entre as frequências analógicas e digitais. Essa função utiliza a fórmula matemática $w_p = \frac{2}{T} \tan\left(\frac{w_c T}{2}\right)$, onde w_p é a frequência de corte pré-distorcida em radianos, T é o tempo de amostragem, w_c é a frequência de corte em radianos.

As funções "so_filter_update_coeff_lowpass", "so_filter_update_coeff_bandpass" e "so_filter_update_coeff_notch" realizam o cálculo dos coeficientes do filtro para os tipos de filtro passa-baixa, passa-faixa e notch, respectivamente. Essas funções utilizam as fórmulas matemáticas para os coeficientes dos filtros digitais de segunda ordem, que dependem dos parâmetros w_c , Q e ts .

```

1
2 /*
3  * sogi_pll.c
4  *
5  * Created on: 18 de jan de 2023
6  * Author: Jose
7  */
8
9 /*****
10  * Includes *
11  *****/
12
13 #include "so_filter.h"
14 #include "math.h"
15
16 /*****
17  * Functions *
18  *****/
19
20 /**
21  * @brief Pre distorcao para a frequencia de corte do filtro

```

```

22  *
23  *       $w_p = 2/T * \tan(w_c T/2)$ 
24  *
25  * @param  fc          Frequencia de atualizacao do so_filter em Hz
26  * @return wp          Retorna a frequencia de corte pre distorcida
↪ em radianos
27  */
28 void so_filter_prewarp_frequency(so_filter_t * so_filter, float fc)
29 {
30     so_filter->fc = fc;
31     so_filter->wc = 2.0 * so_filter->fs * tan(
↪ 3.1415926535897932384626433832795 * fc * so_filter->ts );
32 }
33
34 /**
35  * @brief Recalcula os coeficientes dos filtros lowpass
36  *
37  * @param so_filter    Ponteiro para a estrutura do so_filter
38  * @param fs          Frequencia de atualizacao do so_filter em Hz
39  * @param Q           Fator de qualidade do filtro
40  * @param fc          Frequencia de corte em Hertz
41  */
42 void so_filter_update_coeff_lowpass(so_filter_t * so_filter)
43 {
44     ///< Variaveis auxiliares para minimizar processamento
45     float a = so_filter->ts * so_filter->ts * so_filter->wc *
↪ so_filter->wc;    ///< (ts*wc)^2
46     float b = (2.0 * so_filter->ts * so_filter->wc) / so_filter->Q;
↪ ///< 2*ts*wc/Q
47     float aux_div = 1.0 / ( 4 + b + a );
↪ ///< 1 / (4 + 2*ts*wc/Q + (ts*wc)^2)
48
49     ///< Coeficientes do filtro de eixo direto
50     so_filter->coeffs.a1 = ( (2.0 * a) - 8) * aux_div;
51     so_filter->coeffs.a2 = (a - b + 4) * aux_div;
52     so_filter->coeffs.b0 = a * aux_div;

```

```

53     so_filter->coeffs.b1 = 2.0 * so_filter->coeffs.b0;
54     so_filter->coeffs.b2 = so_filter->coeffs.b0;
55 }
56
57 /**
58  * @brief Recalcula os coeficientes dos filtros lowpass
59  *
60  * @param so_filter    Ponteiro para a estrutura do so_filter
61  * @param fs          Frequencia de atualizacao do so_filter em Hz
62  * @param Q           Fator de qualidade do filtro
63  * @param fc          Frequencia de corte em Hertz
64  */
65 void so_filter_update_coeff_bandpass(so_filter_t * so_filter)
66 {
67     ///< Variaveis auxiliares para minimizar processamento
68     float a = so_filter->ts * so_filter->ts * so_filter->wc *
↪ so_filter->wc;    ///< (ts*wc)^2
69     float b = 2.0 * so_filter->ts * so_filter->wc / so_filter->Q;
↪ ///< 2*ts*wc/Q
70     float aux_div = 1.0 / ( 4 + b + a );
↪ ///< 1 / (4 + 2*ts*wc/Q + (ts*wc)^2)
71
72     ///< Coeficientes do filtro de eixo direto
73     so_filter->coeffs.a1 = (2.0 * a - 8) * aux_div;
74     so_filter->coeffs.a2 = (a - b + 4) * aux_div;
75     so_filter->coeffs.b0 = b * aux_div;
76     so_filter->coeffs.b1 = 0.0;
77     so_filter->coeffs.b2 = - so_filter->coeffs.b0;
78 }
79
80 /**
81  * @brief Recalcula os coeficientes dos filtros lowpass
82  *
83  * @param so_filter    Ponteiro para a estrutura do so_filter
84  * @param fs          Frequencia de atualizacao do so_filter em Hz
85  * @param Q           Fator de qualidade do filtro

```

```

86  * @param fc          Frequencia de corte em Hertz
87  */
88  void so_filter_update_coeff_notch(so_filter_t * so_filter)
89  {
90      ///< Variaveis auxiliares para minimizar processamento
91      float a = so_filter->ts * so_filter->ts * so_filter->wc *
↪ so_filter->wc;    ///< (ts*wc)^2
92      float b = 2.0 * so_filter->ts * so_filter->wc / so_filter->Q;
↪ ///< 2*ts*wc/Q
93      float c = a + 4;
↪ ///< (ts*wc)^2 + 4
94      float aux_div = 1.0 / ( b + c );
↪ ///< 1 / (4 + 2*ts*wc/Q + (ts*wc)^2)
95
96      ///< Coeficientes do filtro de eixo direto
97      so_filter->coeffs.a1 = (2.0 * a - 8) * aux_div;
98      so_filter->coeffs.a2 = (c - b) * aux_div;
99      so_filter->coeffs.b0 = c * aux_div;
100     so_filter->coeffs.b1 = so_filter->coeffs.a1;
101     so_filter->coeffs.b2 = so_filter->coeffs.b0;
102 }
103
104 /**
105  * @brief Recalcula os coeficientes dos filtros lowpass
106  *
107  * @param so_filter    Ponteiro para a estrutura do so_filter
108  * @param fs          Frequencia de atualizacao do so_filter em Hz
109  * @param Q           Fator de qualidade do filtro
110  * @param fc          Frequencia de corte em Hertz
111  */
112  void so_filter_update_coeff_highpass(so_filter_t * so_filter)
113  {
114      float a = so_filter->ts * so_filter->ts * so_filter->wc *
↪ so_filter->wc;    ///< (ts*wc)^2
115      float b = 2.0 * so_filter->ts * so_filter->wc / so_filter->Q;
↪ ///< 2*ts*wc/Q

```

```

116     float c = a + 4;
    ↪    /// $(ts*wc)^2 + 4$ 
117     float aux_div = 1.0 / ( b + c );
    ↪    /// $1 / (4 + 2*ts*wc/Q + (ts*wc)^2)$ 
118     /// $\text{Coeficientes do filtro de eixo direto}$ 
119     so_filter->coeffs.a1 = (2.0 * a - 8)      * aux_div;
120     so_filter->coeffs.a2 = (c - b)           * aux_div;
121     so_filter->coeffs.b0 = 4.0               * aux_div;
122     so_filter->coeffs.b1 = - 2.0 * so_filter->coeffs.b0;
123     so_filter->coeffs.b2 = so_filter->coeffs.b0;
124 }
125
126 /**
127  * @brief Reseta e inicializa filtro com valor desejado
128  *
129  * @param so_filter Ponteiro para a estrutura do filtro
130  * @param value     Valore inicial de reset do filtro
131  */
132 void so_filter_reset(so_filter_t * so_filter, float value){
133     so_filter->states.u2 = value;
134     so_filter->states.u1 = value;
135     so_filter->states.u0 = value;
136     so_filter->states.y2 = value;
137     so_filter->states.y1 = value;
138     so_filter->states.y0 = value;
139 }
140
141 /**
142  * @brief Roda a estrutura do filtro SO
143  *
144  * @param so_filter Ponteiro para a estrutura do filtro
145  * @param input     Entrada do filtro
146  */
147 void so_filter_run(so_filter_t * so_filter, float input){
148     so_filter->states.u2 = so_filter->states.u1;
149     so_filter->states.u1 = so_filter->states.u0;

```

```

150     so_filter->states.u0 = input;
151     so_filter->states.y2 = so_filter->states.y1;
152     so_filter->states.y1 = so_filter->states.y0;
153     so_filter->states.y0 =  so_filter->states.u0 * so_filter->coeffs.b0 +
154                             so_filter->states.u1 * so_filter->coeffs.b1 +
155                             so_filter->states.u2 * so_filter->coeffs.b2 -
156                             so_filter->states.y2 * so_filter->coeffs.a2 -
157                             so_filter->states.y1 * so_filter->coeffs.a1;
158 }
159
160 /**
161  * @brief Inicia filtro de segunda ordem
162  *
163  * @param so_filter    Ponteiro para a estrutura do so_filter
164  * @param fs           Frequencia de atualizacao do so_filter em Hz
165  * @param Q            Fator de qualidade do filtro
166  * @param fc           Frequencia de corte em Hertz
167  * @param Type         Tipo do filtro: 0 lowpass, 1 bandpass, 2 notch e 3
168  *                    ↪ highpass
169  */
170 void so_filter_initialize(so_filter_t * so_filter, float fs, float fc,
171 ↪ float Q, Uint32 type, Uint32 is_prewarped)
172 {
173     ///< Atualiza os parametros usado no filtro
174     so_filter->type = type;
175     so_filter->Q = Q;
176     so_filter->fs = fs;
177     so_filter->ts = 1.0 / fs;
178     so_filter->fc = fc;
179     so_filter->is_prewarped = is_prewarped;
180
181     if (is_prewarped){
182         so_filter_prewarp_frequency(so_filter, fc);
183     }else{
184         so_filter->wc = 6.28318530717958647692528676655900 * fc;
185     }
186 }

```

```
184
185  ///< Atualiza Filtro baseado no tipo
186  switch (type)
187  {
188      case SO_FILTER_LOWPASS:
189          so_filter_update_coeff_lowpass(so_filter);
190          break;
191      case SO_FILTER_BANDPASS:
192          so_filter_update_coeff_bandpass(so_filter);
193          break;
194      case SO_FILTER_NOTCH:
195          so_filter_update_coeff_notch(so_filter);
196          break;
197      case SO_FILTER_HIGHPASS:
198          so_filter_update_coeff_highpass(so_filter);
199          break;
200      default:
201          so_filter_update_coeff_lowpass(so_filter);
202          break;
203  }
204 }
205
206
```

A.2 IMPLEMENTAÇÃO SOGI PLL

Como apresentado durante a seção do controle, a equação da função de transferência do SOGI-QSG pode ser representada por duas equações, a função de transferência do integrador SOGI ($G_1(s)$) e a função de transferência do gerador de sinal em quadratura ($G_2(s)$). A função de transferência $G_1(s)$ representa a parte em fase do SOGI-QSG, enquanto a função de transferência $G_2(s)$ representa a parte em quadratura.

A.2.1 Filtro $G_1(s)$, gerador em fase

Aplicando mesmo método que foi usado para as os filtros de segunda ordem,

Para o filtro em fase temos a equação descrita como:

$$G_1(s) = \frac{V_\alpha(s)}{V_g(s)} = \frac{k\omega s}{s^2 + k\omega s + \omega^2}$$

Os termos genéricos da forma apresentada anteriormente são:

$$A = 0$$

$$B = k\omega$$

$$C = 0$$

$$D = 1$$

$$E = k\omega$$

$$F = \omega^2$$

substituindo na forma genérica dos coeficientes discretizados obtemos:

$$a_0 = 1$$

$$a_1 = \frac{2T^2\omega^2 - 8}{T^2\omega^2 + 2kT\omega + 4}$$

$$a_2 = \frac{T^2\omega^2 - 2kT\omega + 4}{T^2\omega^2 + 2kT\omega + 4}$$

$$b_0 = \frac{2Tk\omega}{T^2\omega^2 + 2kT\omega + 4}$$

$$b_1 = 0$$

$$b_2 = -\frac{2Tk\omega}{T^2\omega^2 + 2kT\omega + 4}$$

$$\begin{aligned} G_1[z] = Y[z] = & + \frac{2Tk\omega}{T^2\omega^2 + 2kT\omega + 4} X[z] \\ & - \frac{2Tk\omega}{T^2\omega^2 + 2kT\omega + 4} X[z-2] \\ & - \frac{2T^2\omega^2 - 8}{T^2\omega^2 + 2kT\omega + 4} Y[z-1] \\ & - \frac{T^2\omega^2 - 2kT\omega + 4}{T^2\omega^2 + 2kT\omega + 4} Y[z-2] \end{aligned}$$

A resposta ao degrau do filtro discretizado $G_1[z]$ comparado ao filtro contínuo $G_1[s]$ pode ser visualizado na Figura 48

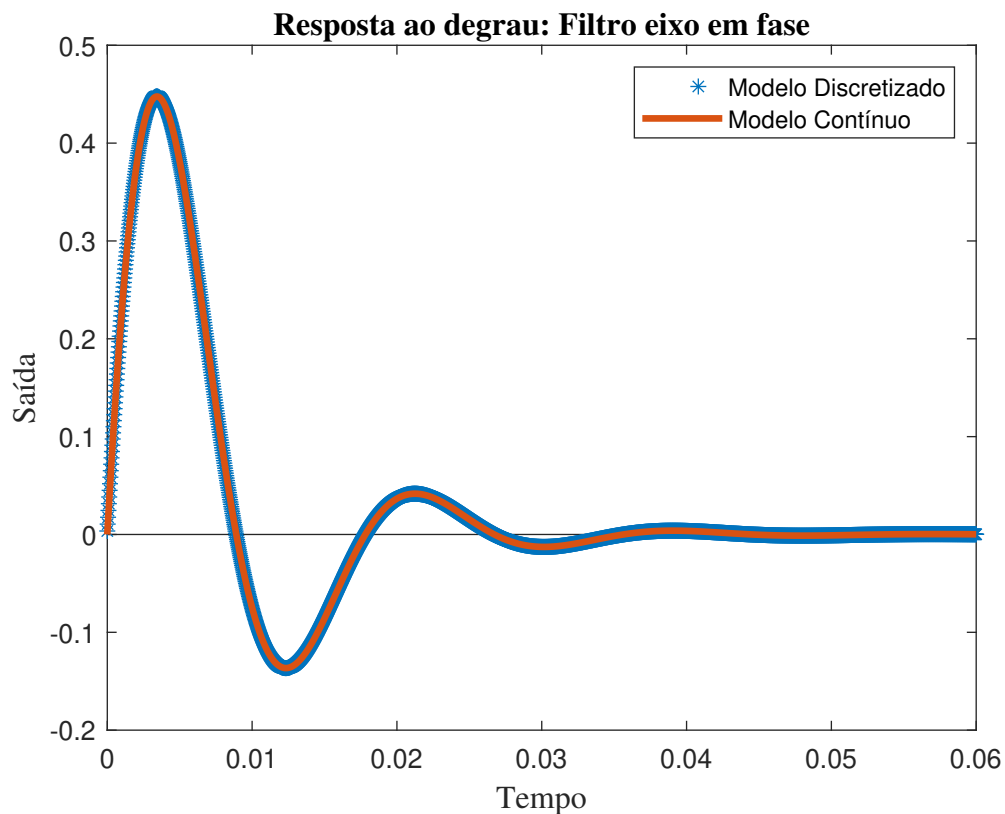


Figura 48 – Resposta ao degrau filtro G_1 , do modelo contínuo e modelo discretizado.

A.2.2 Filtro $G_2(s)$, gerador em quadratura

Aplicando mesmo método que foi usado para os filtros de segunda ordem, para o filtro em quadratura temos a equação descrita como:

$$G_2(s) = \frac{V_\beta(s)}{V_g(s)} = \frac{k\omega^2}{s^2 + k\omega s + \omega^2}$$

Os termos genéricos da forma apresentada anteriormente são:

$$A = 0$$

$$B = 0$$

$$C = k\omega^2$$

$$D = 1$$

$$E = k\omega$$

$$F = \omega^2$$

substituindo na forma genérica dos coeficientes discretizados obtemos:

$$\begin{aligned}
 a_0 &= 1 \\
 a_1 &= \frac{2 T^2 w^2 - 8}{T^2 w^2 + 2 k T w + 4} \\
 a_2 &= \frac{T^2 w^2 - 2 k T w + 4}{T^2 w^2 + 2 k T w + 4} \\
 b_0 &= \frac{T^2 k w^2}{T^2 w^2 + 2 k T w + 4} \\
 b_1 &= \frac{2 T^2 k w^2}{T^2 w^2 + 2 k T w + 4} \\
 b_2 &= \frac{T^2 k w^2}{T^2 w^2 + 2 k T w + 4}
 \end{aligned}$$

$$\begin{aligned}
 G_1[z] = Y[z] &= + \frac{T^2 k w^2}{T^2 w^2 + 2 k T w + 4} X[z] \\
 &+ \frac{2 T^2 k w^2}{T^2 w^2 + 2 k T w + 4} X[z - 2] \\
 &+ \frac{T^2 k w^2}{T^2 w^2 + 2 k T w + 4} X[z - 2] \\
 &- \frac{2 T^2 w^2 - 8}{T^2 w^2 + 2 k T w + 4} Y[z - 1] \\
 &- \frac{T^2 w^2 - 2 k T w + 4}{T^2 w^2 + 2 k T w + 4} Y[z - 2]
 \end{aligned}$$

A resposta ao degrau do filtro discretizado $G_2[z]$ comparado ao filtro contínuo $G_2[s]$ pode ser visualizado na Figura 49

A.2.3 Biblioteca em C: SOGI PLL

Esse é um arquivo de cabeçalho em C para implementação de um algoritmo de controle PLL (Phase Locked Loop) utilizando o método SOGI (Second Order Generalized Integrator).

O arquivo contém a definição de estruturas de dados para implementação do SOGI PLL, incluindo filtros, integradores e um controlador PI. Também inclui as definições de constantes e funções protótipo para inicialização, atualização e execução do algoritmo de controle SOGI PLL.

A estrutura de dados `sogi_pll_t` contém variáveis de entrada, coeficientes, estados e variáveis de saída para o controle PLL implementado.

As funções `sogi_pll_initialize()`, `sogi_pll_update_coeff_sogi()` e `sogi_pll_run()` são responsáveis pela inicialização, atualização e execução do controle SOGI PLL, respectiva-

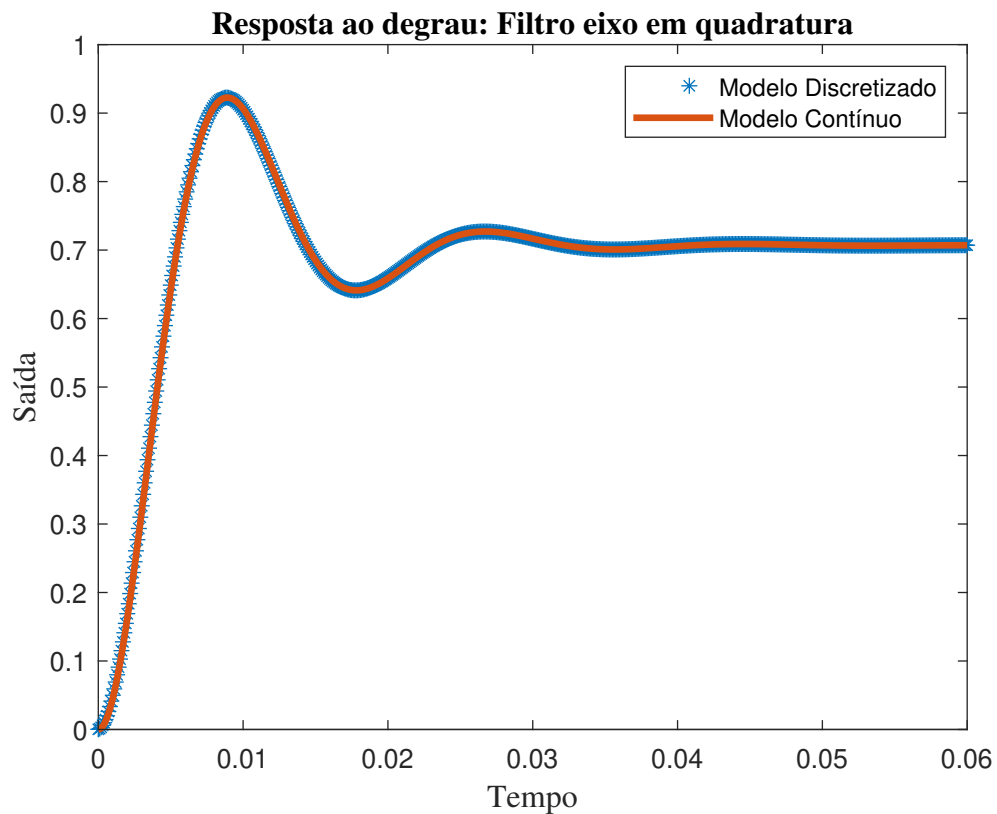


Figura 49 – Resposta ao degrau filtro G_2 , do modelo contínuo e modelo discretizado.

mente. As funções estão definidas para compilação na CPU ou no CLA (Control Law Accelerator) do processador Texas Instruments TMS320F28379D.

```

1  /*
2  * sogi_pll.h
3  *
4  * Created on: 23 de abr de 2022
5  * Author: Jose
6  */
7
8  #ifndef SOGI_PLL_H_
9  #define SOGI_PLL_H_
10
11 /*****
12 * Includes *
13 *****/
14

```

```
15 #include "F28x_Project.h"
16 #if !defined(__TMS320C28XX_CLA__)    //!< CPU Compiler
17 #include "math.h"
18 #elif defined(__TMS320C28XX_CLA__)  //!< CLA Compiler
19 #include "CLAmath.h"
20 #endif
21
22 /*****
23  * Defines *
24  *****/
25
26 #define TWO_PI          6.28318530717958647692528676655900
27 #define RAD2HZ          0.15915494309189533576888376337251
28
29 /*****
30  * Structs *
31  *****/
32
33 typedef struct _sogi_so_filter_t_
34 {
35     //!< Estados de entrada e saida
36     float u0;    //!< Estado entrada u[n]
37     float u1;    //!< Estado entrada u[n-1]
38     float u2;    //!< Estado entrada u[n-2]
39     float y0;    //!< Estado saida y[n]
40     float y1;    //!< Estado saida y[n-1]
41     float y2;    //!< Estado saida y[n-2]
42     //!< Coeficientes do filtro
43     // float a0;    //!< Coeff a0: deve ser 1 para esse metodo
44     float a1;
45     float a2;
46     float b0;
47     float b1;
48     float b2;
49 }sogi_so_filter_t;
50
```

```
51 typedef struct _sogi_integrador_t_
52 {
53     ///< Estados de entrada e saida
54     float u0;    ///< Estado entrada u[n]
55     float u1;    ///< Estado entrada u[n-1]
56     float y0;    ///< Estado saida y[n]
57     float y1;    ///< Estado saida y[n-1]
58     ///< Integrator gain
59     float Ki;    ///< Ganho Ki do integrador
60 }sogi_integrador_t;
61
62 typedef struct _sogi_pi_t_
63 {
64     ///< Estados de entrada e saida do integrador
65     float u0;    ///< Estado entrada u[n]
66     float u1;    ///< Estado entrada u[n-1]
67     float y0;    ///< Estado saida y[n]
68     float y1;    ///< Estado saida y[n-1]
69     ///< Ganho integrator e proporcional
70     float Ki;    ///< Ganho integral Ki do PI
71     float Kp;    ///< Ganho proporcional Kp do PI
72     ///< Saida do pi
73     float output; ///< Saida da estrutura do PI
74 }sogi_pi_t;
75
76 typedef struct _sogi_pll_t_
77 {
78     ///< Parametros de configuracao do SOGI
79     float k;      ///< Ganho k dos filtros sogi
80     float fs;     ///< Frequencia de amostragem do sogi pll
81     float ts;     ///< Periodo de amostragem do sogi pll, calculado
82     ↪ na inicializacao
83     float fr;     ///< Frequencia da rede em Hz
84     float wc;     ///< Frequencia da rede em radianos/s, calculado
85     ↪ na inicializacao a partir de fr
```

```

85     ///  
Entrada do pll_sogi
86     float input;           ///  
Estado de entrada do sogi pll
87
88     ///  
Coeficientes e estados dos filtros sogi
89     sogi_so_filter_t v_alpha;      ///  
Tensao eixo alpha
90     sogi_so_filter_t v_beta;      ///  
Tensao eixo beta
91
92     ///  
Variaveis em dq0
93     float d;                 ///  
Tensao eixo direto
94     float q;                 ///  
Tensao eixo quadratura
95
96     ///  
PI
97     sogi_pi_t pi;           ///  
Estrutura do PI do sogi pll
98
99     float w_ref;           ///  
Referencia de frequencia em
↪     radianos
100
101     ///  
Integrator
102     sogi_integrador_t integrador; ///  
Estrutura do I do sogi pll
103
104     ///  
Variaveis de saida do sogi
105     Uint32 zero_crossing;
106     float wt;              ///  
Angulo da tensao de entrada em
↪     radianos por segundo
107     float omega;          ///  
frequencia da tensao de entrada
↪     em radianos
108     float freq;          ///  
frequencia da tensao de entrada
↪     em Hertz
109     float sine;          ///  
Sinal seno de saida do pll
110     float cosine;        ///  
Sinal coseno de saida do pll
111 }sogi_pll_t;
112
113 /*  
*****
114 * Functions prototypes *
115 *****  
*/
116

```

```

117 #if !defined(__TMS320C28XX_CLA__)    //!< CPU Compiler
118     void sozi_pll_initialize(sozi_pll_t * sozi, float fs, float k, float
↪ fr, float Kp, float Ki);
119     void sozi_pll_update_coeff_sozi(sozi_pll_t * sozi);
120     void sozi_pll_run(sozi_pll_t * sozi, float input);
121 #elif defined(__TMS320C28XX_CLA__)    //!< CLA Compiler
122     void sozi_pll_update_coeff_sozi_cla(sozi_pll_t * sozi);
123     void sozi_pll_run_cla(sozi_pll_t * sozi, float input);
124 #endif
125
126
127 #endif /* SOGI_PLL_H_ */

```

E a implementação é descrita no arquivo `sozi_pll.c` abaixo, Esse arquivo `.c` é referente a uma implementação SOGI-PLL. Ele contém funções que atualizam os coeficientes dos filtros do SOGI e computam a estrutura do SOGI-PLL.

A função `sozi_pll_update_coeff_sozi` atualiza os coeficientes dos filtros de eixo alpha e beta do SOGI. A função recebe como parâmetros um ponteiro para a estrutura do SOGI-PLL, a frequência de atualização do SOGI-PLL em Hz, o ganho `k` das estruturas dos filtros PLL e a frequência da rede do SOGI-PLL em Hz.

```

1  /*
2  * sozi_pll.c
3  *
4  * Created on: 3 de mai de 2022
5  * Author: Jose
6  */
7
8  /*****
9  * Includes *
10 *****/
11
12 #include "sozi_pll.h"
13
14 /*****
15 * Functions *

```

```

16  *****/
17
18  /**
19   * @brief Recalcula os coeficientes dos filtros do sogi
20   *
21   * @param sogi  Ponteiro para a estrutura do sogi_pll
22   * @param fs    Frequencia de atualizacao do sogi_pll em Hz
23   * @param k     Ganho k das estruturas dos filtros pll
24   * @param fr    Frequencia da rede do sogi_pll em Hz
25   */
26  void sogi_pll_update_coeff_sogi(sogi_pll_t * sogi)
27  {
28      ///< Variaveis auxiliares para minimizar processamento
29      float a = sogi->ts * sogi->ts * sogi->wc * sogi->wc;      ///<
    ↪ (ts*wc)^2
30      float b = 2.0 * sogi->k * sogi->ts * sogi->wc;          ///<
    ↪ 2*k*ts*wc
31      float c = a + 4;                                         ///<
    ↪ (ts*wc)^2 + 4
32      float d = a * sogi->k;                                    ///<
    ↪ k*(ts*wc)^2
33      float aux_div = 1.0 / ( c + b );                          ///< 1 /
    ↪ (4 + 2*k*ts*wc + (ts*wc)^2)
34
35      ///< Coeficientes do filtro de eixo alpha
36      sogi->v_alpha.a1 = (2.0 * a - 8) * aux_div;
37      sogi->v_alpha.a2 = (c - b) * aux_div;
38      sogi->v_alpha.b0 = b * aux_div;
39      sogi->v_alpha.b1 = 0.0;
40      sogi->v_alpha.b2 = -sogi->v_alpha.b0;
41
42      ///< Coeficientes do filtro de eixo em beta
43      sogi->v_beta.a1 = sogi->v_alpha.a1;
44      sogi->v_beta.a2 = sogi->v_alpha.a2;
45      sogi->v_beta.b0 = d * aux_div;
46      sogi->v_beta.b1 = 2.0 * sogi->v_beta.b0;

```



```

47     sogi->v_beta.b2 = sogi->v_beta.b0;
48 }
49
50 void sogi_pll_so_filter_run(sogi_so_filter_t * so_filter, float input)
51 {
52     so_filter->u2 = so_filter->u1;
53     so_filter->u1 = so_filter->u0;
54     so_filter->u0 = input;
55     so_filter->y2 = so_filter->y1;
56     so_filter->y1 = so_filter->y0;
57     so_filter->y0 = so_filter->u0 * so_filter->b0 +
58                   so_filter->u1 * so_filter->b1 +
59                   so_filter->u2 * so_filter->b2 -
60                   so_filter->y2 * so_filter->a2 -
61                   so_filter->y1 * so_filter->a1;
62 }
63
64 /**
65  * @brief Roda o integrador da estrutura do sogi pll
66  *
67  * For a given step  $n > 0$  with simulation time  $t(n)$ , updates output  $y(n)$ 
68  * ↪ as follows:
69  *
70  * Forward Euler method:
71  *  $y(n) = y(n-1) + K_i * [t(n) - t(n-1)] * u(n-1)$ 
72  *
73  * Backward Euler method:
74  *  $y(n) = y(n-1) + K_i * [t(n) - t(n-1)] * u(n)$ 
75  *
76  * Trapezoidal method:
77  *  $y(n) = y(n-1) + K_i * [t(n) - t(n-1)] * [u(n) + u(n-1)] / 2$ 
78  *
79  * @param input Entrada do integrador
80 */
81 void sogi_pll_integrator_run(sogi_pll_t * sogi, float input)
82 {

```

```

82     ///< Update memory
83     sogi->integrador.u1 = sogi->integrador.u0;
84     sogi->integrador.u0 = input;
85     sogi->integrador.y1 = sogi->integrador.y0;
86
87     ///< Run integrator method
88     // sogi->integrador.y0 = sogi->integrador.y1 + sogi->integrador.Ki *
89     ↳ sogi->ts * sogi->integrador.u1; //!<
90     ↳ Forward Euler method
91     sogi->integrador.y0 = sogi->integrador.y1 + sogi->integrador.Ki *
92     ↳ sogi->ts * sogi->integrador.u0; //!<
93     ↳ Backward Euler method
94     // sogi->integrador.y0 = sogi->integrador.y1 + sogi->integrador.Ki *
95     ↳ sogi->ts * (sogi->integrador.u1 + sogi->integrador.u0) * 0.5; //!<
96     ↳ Trapezoidal method
97 }
98
99 /**
100  * @brief Roda o PI da estrutura do sogi pll
101  *
102  * @param input Entrada do integrador
103  */
104 void sogi_pll_pi_run(sogi_pll_t * sogi, float input)
105 {
106     ///< Update memory
107     sogi->pi.u1 = sogi->pi.u0;
108     sogi->pi.u0 = input;
109     sogi->pi.y1 = sogi->pi.y0;
110
111     ///< Run integrator method
112     // sogi->pi.y0 = sogi->pi.y1 + sogi->pi.Ki * sogi->ts * sogi->pi.u1;
113     ↳ //!< Forward Euler method
114     sogi->pi.y0 = sogi->pi.y1 + sogi->pi.Ki * sogi->ts * sogi->pi.u0;
115     ↳ //!< Backward Euler method
116     // sogi->pi.y0 = sogi->pi.y1 + sogi->pi.Ki * sogi->ts * (sogi->pi.u1 +
117     ↳ sogi->pi.u0) * 0.5; //!< Trapezoidal method

```

```
109
110     ///< Output integrator + proportional
111     sogi->pi.output = sogi->pi.y0 + (sogi->pi.Kp * sogi->pi.u0);
112 }
113
114 /**
115  * @brief Recalcula os coeficientes dos filtros do sogi
116  *
117  * @param sogi Ponteiro para a estrutura do sogi_pll
118  * @param fs Freqüencia de atualizaco do sogi_pll
119  * @param k Ganho k das estruturas dos filtros pll
120  * @param fr Freqüencia da rede do sogi_pll
121  * @param Kp Ganho proporcional do pi sogi
122  * @param Ki Ganho integral do pi sogi
123  */
124 void sogi_pll_initialize(sogi_pll_t * sogi, float fs, float k, float fr,
125 ↪ float Kp, float Ki){
126
127     ///< Atualiza os parametros usado no filtro PLL
128     sogi->k = k;
129     sogi->fs = fs;
130     sogi->ts = 1.0 / fs;
131     sogi->fr = fr;
132     sogi->wc = TWO_PI * fr;
133
134     ///< Update sogi filter
135     sogi_pll_update_coeff_sogi(sogi);
136
137     ///< Update pi filter
138     sogi->pi.Kp = Kp;
139     sogi->pi.Ki = Ki;
140
141     ///< Update integrator filter
142     sogi->integrador.Ki = 1.0; ///< Always 1 integrator gain.
143
144     ///< Referencia para integrador
```

```

144     sogi->w_ref = TWO_PI * fr;
145
146 }
147
148 /**
149  * @brief Subtraia o estado de saida do integrador pelo valor sub
150  *
151  * @param sogi Ponteiro para a estrutura do sogi_pll
152  * @param sub valor de decremento dos estados de saida
153  */
154 void sogi_pll_reset_integrador(sogi_pll_t * sogi, float sub){
155     sogi->integrador.y0 -= sub;
156     sogi->integrador.y1 -= sub;
157 }
158
159 /**
160  * @brief Roda a estrutura do pll sogi
161  *
162  * @param sogi Ponteiro para a estrutura do sogi_pll
163  * @param input Tensao de entrada da rede
164  */
165 void sogi_pll_run(sogi_pll_t * sogi, float input){
166
167     sogi->input = input;
168
169     /* SOGI Filter*/
170     sogi_pll_so_filter_run(&sogi->v_alpha, sogi->input);
171     sogi_pll_so_filter_run(&sogi->v_beta, sogi->input);
172
173     // alpha beta to DQ0
174     //!< Auxiliar sine and cosine
175     sogi->cosine = cos(sogi->wt);
176     sogi->sine   = sin(sogi->wt);
177
178     sogi->d = sogi->v_alpha.y0 * sogi->sine - sogi->v_beta.y0 *
↵     sogi->cosine;

```

```
179     sogi->q = sogi->v_alpha.y0 * sogi->cosine + sogi->v_beta.y0 *
↪     sogi->sine;
180
181     // PI(z)
182     sogi_pll_pi_run(sogi, sogi->q);
183
184     sogi->omega = sogi->w_ref + sogi->pi.output;
185
186     // I(z)
187     sogi_pll_integrator_run(sogi, sogi->omega);
188     if (sogi->integrador.y0 >= TWO_PI){
189         sogi_pll_reset_integrador(sogi, TWO_PI);
190         sogi->zero_crossing = 1;
191     }else{
192         sogi->zero_crossing = 0;
193     }
194
195     sogi->wt = sogi->integrador.y0;
196
197     // rad to Hz
198     sogi->freq = sogi->omega * RAD2HZ;
199
200 }
201
```

A.3 ROTINA ISRPWM

Este código é uma rotina de interrupção (ISR) para o sinal de interrupção do módulo EPWM4. A função é responsável por executar diversas tarefas, como a configuração de pinos de saída, reinicialização do contador do módulo EPwm1, chamada de uma tarefa do CLA, inicialização de novas conversões dos módulos ADC A, B e C, além de limpar o flag de interrupção e indicar ao controlador de interrupção do microcontrolador que a interrupção foi atendida.

A linha "#pragma CODE_SECTION(isr_epwm4, ".TI.ramfunc");" define a seção de código onde a rotina ISR será armazenada na memória RAM.

O flag de interrupção do módulo EPwm4 é limpo na linha "EPwm4Regs.ETCLR.bit.INT

= 1;".

Copia os valores filtrados dos sinais de entrada e saída do filtro digital gerados no CLA para a estrutura pfc.

A tarefa 1 do CLA é forçada a executar na linha "Cla1ForceTask1();".

São inicializadas novas conversões nos módulos ADC A, B e C nas linhas "AdccRegs.ADCSOCFRC1.bit.SOC0 = 1;", "AdcbRegs.ADCSOCFRC1.bit.SOC0 = 1;" e "AdcaRegs.ADCSOCFRC1.bit.SOC0 = 1;", respectivamente.

Por fim, o controlador de interrupção do microcontrolador é notificado de que a interrupção foi atendida na linha "PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;".

```

1  #pragma CODE_SECTION(isr_epwm4, ".TI.ramfunc");
2  __interrupt void isr_epwm4(void){
3
4      EPwm4Regs.ETCLR.bit.INT = 1;    ///< Reset interrupt flag
5
6      pfc.current_in.filtered = so_filter_lowpass_current_input.states.y0;
7      pfc.voltage_in.filtered = so_filter_lowpass_voltage_input.states.y0;
8      pfc.voltage_out.filtered = so_filter_notch_voltage_output.states.y0;
9
10     Cla1ForceTask1();
11
12     AdccRegs.ADCSOCFRC1.bit.SOC0 = 1;    ///< Init new conversion ADC C
13     AdcbRegs.ADCSOCFRC1.bit.SOC0 = 1;    ///< Init new conversion ADC B
14     AdcaRegs.ADCSOCFRC1.bit.SOC0 = 1;    ///< Init new conversion ADC A
15
16     PieCtrlRegs.PIEACK.all = PIEACK_GROUP3;
17 }

```

A.4 ROTINA DE INTERRUPTÃO (ISR ADC)

Aqui é apresentado a rotina de interrupção que é executado quando uma conversão de ADC do módulo ADC_A é concluída. A função da interrupção é coletar os resultados da conversão do ADC e executar uma rotina de controle do conversor PFC (Power Factor Correction), que utiliza o valor de entrada de corrente, tensão de entrada e tensão de saída do conversor.

O código também inclui uma rotina de proteção para desligar o PFC em caso de

corrente ou tensão de saída excessivamente alta e uma atualização dos valores de saída do DAC para um escopo de monitoramento. Por fim, a interrupção é encerrada e o controlador de interrupção do microcontrolador é tratada.

```

1  #pragma CODE_SECTION(isr_Adca, ".TI.ramfunc");
2  __interrupt void isr_Adca(void){
3
4      #if HIL_ON
5          pfc.current_in.raw = dataCPU2.current_input;  ///< Get result
↪   from ADC
6          pfc.voltage_in.raw = dataCPU2.voltage_input;  ///< Get result
↪   from ADC
7          pfc.voltage_out.raw = dataCPU2.voltage_output;  ///< Get result
↪   from ADC
8
9          limiter_saturation(pfc.current_in.raw, -40.00, 40.00);
10         limiter_saturation(pfc.voltage_in.raw, -500.0, 500.0);
11         limiter_saturation(pfc.voltage_out.raw, -500.0, 500.0);
12     #else
13         pfc.current_in.adc.value = (float) AdcaResultRegs.ADCRESULT0;
↪   ///< Get result from ADC
14         pfc.voltage_in.adc.value = (float) AdcbResultRegs.ADCRESULT0;
↪   ///< Get result from ADC
15         pfc.voltage_out.adc.value = (float) AdccResultRegs.ADCRESULT0;
↪   ///< Get result from ADC
16     #endif
17
18     AdcaRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
19     AdcbRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
20     AdccRegs.ADCINTFLGCLR.bit.ADCINT1 = 1;
21
22     /* PFC protection */
23     if (pfc.control.bit.enable_current_loop){
24         if (pfc.current_in.filtered >= 30.00) pfc.control.bit.enable_pfc
↪   = 0;
25         if (pfc.voltage_out.filtered >= 450.0) pfc.control.bit.enable_pfc
↪   = 0;

```

```
26     }
27
28     if(pfc.control.bit.enable_pfc){
29         pfc_control_run();
30     }else{
31         dataCPU1.duty_p_pu = 0.0;
32         pi_reset(&pi_current);
33         pi_reset(&pi_voltage);
34         pfc.control.bit.enable_current_loop = 0;
35         pfc.control.bit.enable_voltage_loop = 0;
36         pi_current.output = 0;
37         pi_voltage.output = 0;
38     }
39
40     /* Scope update */
41     if(++index_scopeA > (Uint16) SCOPE_SIZE) index_scopeA = 0;
42     scopeA[index_scopeA] = (float) * ptr_scopeA;
43     if(++index_scopeB > (Uint16) SCOPE_SIZE) index_scopeB = 0;
44     scopeB[index_scopeB] = (float) * ptr_scopeB;
45     if(++index_scopeC > (Uint16) SCOPE_SIZE) index_scopeC = 0;
46     scopeC[index_scopeC] = (float) * ptr_scopeC;
47
48     /* Update DAC output */
49     #if DAC_CPU1
50         DacaRegs.DACVALS.bit.DACVALS = (Uint16) ((( (float) * ptr_DACA) *
↪ DACA_gain) + DACA_offset);
51         DacbRegs.DACVALS.bit.DACVALS = (Uint16) ((( (float) * ptr_DACB) *
↪ DACB_gain) + DACB_offset);
52         DaccRegs.DACVALS.bit.DACVALS = (Uint16) ((( (float) * ptr_DACC) *
↪ DACC_gain) + DACC_offset);
53     #endif
54
55     PieCtrlRegs.PIEACK.all = PIEACK_GROUP1;
56 }
```


A.5 ROTINA DE CONTROLE

O código apresentado contém uma função que executa o controle do PFC, ele é executado dentro da `isrADC`, após o fim das conversões das variáveis de controle. O controle é realizado por meio de um controlador proporcional integral (PI) para a tensão de saída e outro PI para a corrente de entrada. A função utiliza variáveis de estrutura (`struct`) para armazenar as informações do estado do sistema, como tensão e corrente de entrada e saída, referências, erros e ganhos dos controladores.

O código começa verificando se a detecção de cruzamento de zero da rede elétrica (variável `"sogi_pll.zero_crossing"`) foi ativada e, se for o caso, ativa o loop de controle de corrente. Em seguida, verifica se a tensão de saída está acima de um valor de referência e se o loop de corrente já foi ativado, ativando o loop de controle de tensão.

Caso o loop de controle de tensão seja ativado, é calculado o erro entre a tensão de saída e a referência e a função `"pi_run()"` é usada para executar o controle PI da tensão de saída. O resultado é limitado para garantir que o valor da saída fique dentro de uma faixa aceitável. Caso o loop de controle de tensão não seja ativado, a saída do controlador de tensão é definida como zero.

Em seguida, a referência de corrente de entrada é calculada como uma função da tensão de referência do PLL, multiplicada pelo ganho do controlador de tensão. A função `"pi_run()"` é usada novamente para executar o controle PI da corrente de entrada, com o erro sendo calculado como a diferença entre a corrente de referência e a corrente filtrada de entrada. O resultado é limitado dentro de uma faixa aceitável.

Os valores de saída do controlador de corrente são então usados para calcular os valores de duty cycle e atualizar os registradores de comparação (CMPA e CMPB) do módulo PWM (Pulse Width Modulation). Esses valores também são armazenados em uma variável de estrutura para o HIL no CPU2.

```
1  #pragma CODE_SECTION(pfc_control_run, ".TI.ramfunc");
2  void pfc_control_run(void){
3
4      if(sogi_pll.zero_crossing) pfc.control.bit.enable_current_loop = 1;
5      if(pfc.voltage_out.filtered > 380.0 && sogi_pll.zero_crossing)
↪ pfc.control.bit.enable_voltage_loop = 1;
6
7      if (pfc.control.bit.enable_voltage_loop){
8          pfc.voltage_out.error = pfc.voltage_out.reference -
↪ pfc.voltage_out.filtered;
```

```
9     pi_run(&pi_voltage, pfc.voltage_out.error);
10     limiter_saturation(pi_current.output, -50.00, 50.00);
11 }else{
12     pi_voltage.output = 0.0;
13 }
14
15 pfc.current_in.reference = CURRENT_REFERENCE_GAIN * sozi_pll.sine *
↪ (1.0 + pi_voltage.output);
16
17 if (pfc.control.bit.enable_current_loop){
18     pfc.current_in.error = fabsf(pfc.current_in.reference) -
↪ fabsf(pfc.current_in.filtered);
19     pi_run(&pi_current, pfc.current_in.error);
20     limiter_saturation(pi_current.output, 0.00, 1.00);
21 }
22
23 pfc.duty_p.value_pu = pi_current.output;
24 pfc.duty_n.value_pu = pi_current.output;
25
26 EPwm4Regs.CMPA.bit.CMPA = pfc.duty_p.value_pu * pfc.duty_p.gain;
27 EPwm4Regs.CMPB.bit.CMPB = pfc.duty_n.value_pu * pfc.duty_n.gain;
28
29 dataCPU1.duty_p_pu = pfc.duty_p.value_pu;
30 dataCPU1.duty_n_pu = pfc.duty_n.value_pu;
31
32
33 }
```

A.6 ROTINA DE FILTROS CLA

Este é um código que define a rotina de interrupção Cla1Task1 para o processamento de filtros digitais e a estimação da fase da rede elétrica através do algoritmo SOGI-PLL.

A função INIT_COUNTER é chamada para inicializar um contador. Em seguida, três filtros digitais são executados usando a função so_filter_run_cla e seus valores de saída são armazenados nas estruturas de dados pfc.current_in.filtered, pfc.voltage_in.filtered e pfc.voltage_out.filtered, respectivamente.

Em seguida, a função `sogi_pll_run_cla` é chamada para executar o algoritmo SOGI-PLL e estimar a frequência da rede elétrica. Finalmente, a função `END_COUNTER` é chamada para encerrar a contagem do tempo de execução.

É importante notar que esta função é executada dentro do contexto de uma interrupção do CLA.

```
1  /**
2  * @brief Rotina de interrupção executada pela tarefa 1 do CLA.
3  *
4  * Essa rotina executa o processamento de filtros digitais e a estimação
5  * da fase da rede elétrica através do algoritmo SOGI-PLL.
6  *
7  * @note Essa função é executada dentro do contexto de uma interrupção do
8  * ↪ CLA.
9  */
10 __interrupt void Cla1Task1 ( void )
11 {
12     INIT_COUNTER(counter_cla);
13
14     so_filter_run_cla(&so_filter_lowpass_current_input,
15     ↪ pfc.current_in.raw);
16
17     so_filter_run_cla(&so_filter_lowpass_voltage_input,
18     ↪ pfc.voltage_in.raw);
19
20     so_filter_run_cla(&so_filter_lowpass_voltage_output,
21     ↪ pfc.voltage_out.raw);
22     so_filter_run_cla(&so_filter_notch_voltage_output,
23     ↪ so_filter_lowpass_voltage_output.states.y0);
24
25     sogi_pll_run_cla(&sogi_pll,
26     ↪ so_filter_lowpass_voltage_input.states.y0);
27
28     END_COUNTER(counter_cla);
29 }
```

APÊNDICE B – APÊNDICE B

B.1 PROTÓTIPO CONVERSOR

Na figura 50 é apresentado uma imagem do protótipo do conversor. O protótipo contém toda a parte de potência, instrumentação, gate drives, sensores e controle. Na figura 51 é apresentado os dois layers do protótipo do conversor no software de layout e ao lado a implementação física. Na figura 52 é apresentados o protótipo do gate drive montado. Na figura 53 é apresentado o kit de controle utilizando FPGA, é importante ressaltar que no conversor foi previsto a utilização de kit's DSP da Texas Instruments e ele foi desenvolvido de forma a ser pino compatível. Na figura 54 e 55 é apresentando o sensor de corrente montado e seu software de projeto, fornecido pelo fabricante do sensor.

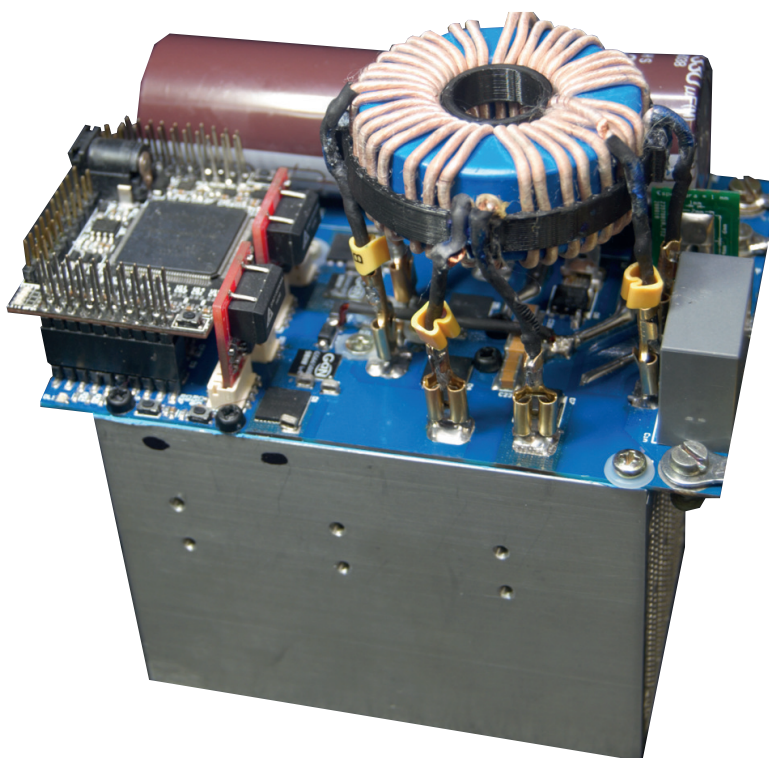


Figura 50 – Protótipo conversor PFC, visualização 3D real e alitum

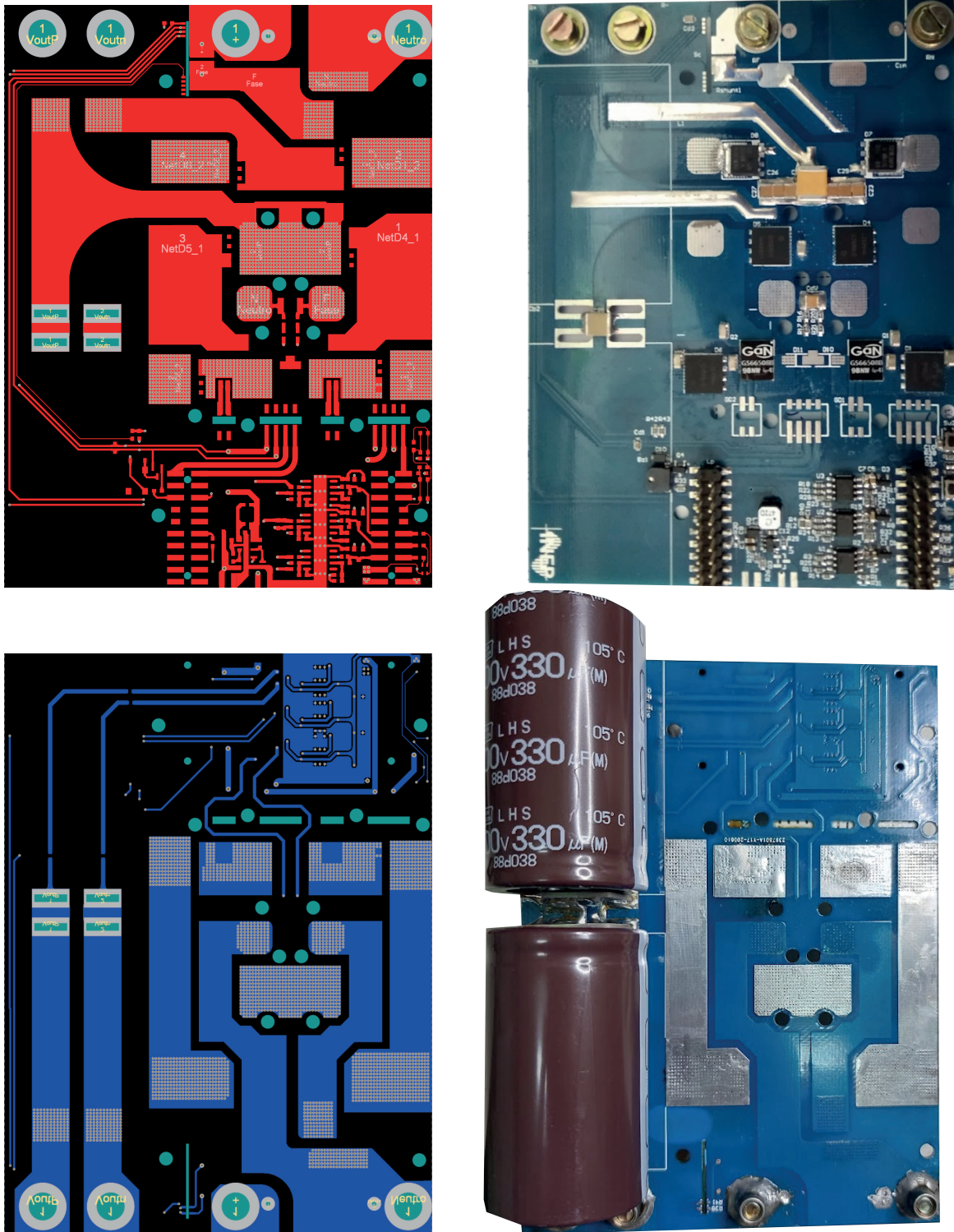


Figura 51 – Protótipo do Conversor PFC: Um protótipo desenvolvido em dois layers, que integra as partes de potência, instrumentação e controle em seu layout. A imagem exhibe o protótipo completo do conversor PFC, destacando sua estrutura em camadas e a distribuição dos componentes de potência, instrumentação e controle pelos layers.

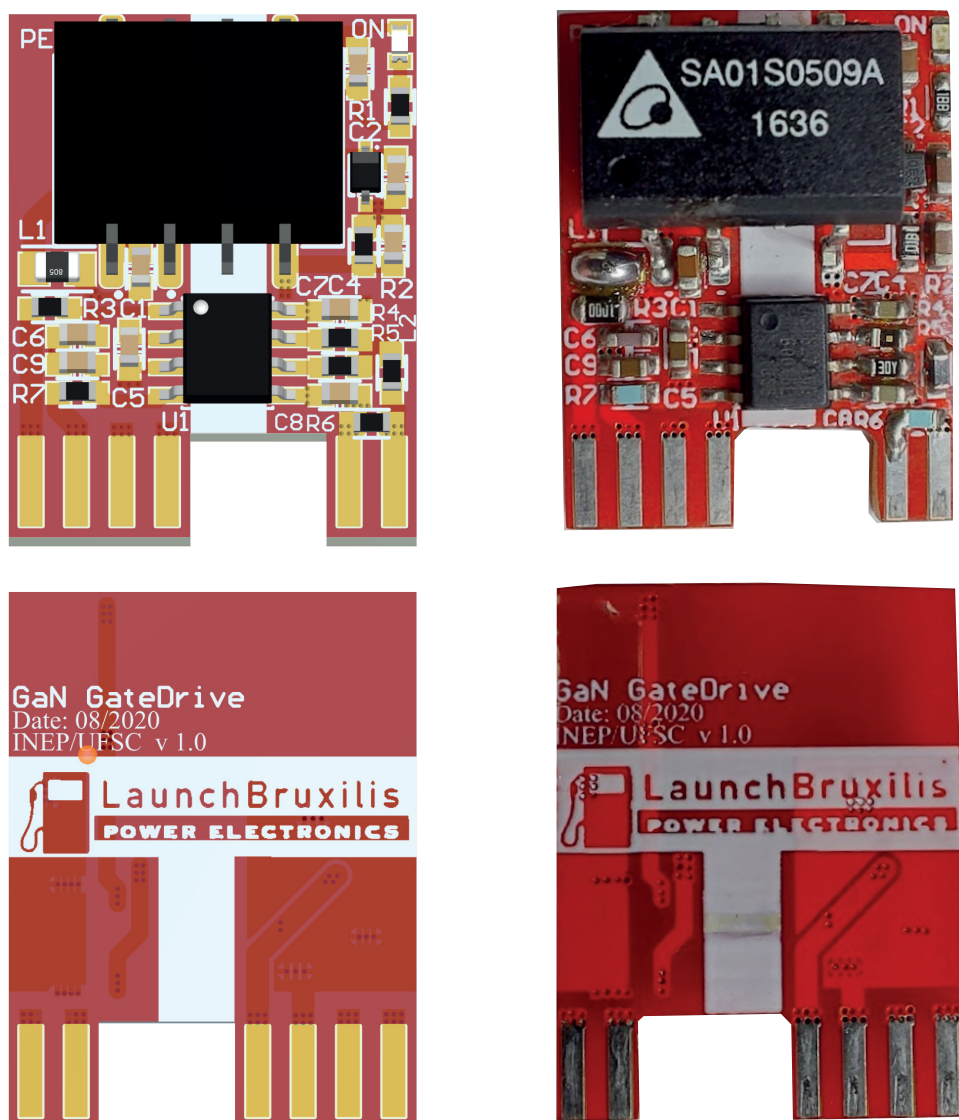


Figura 52 – Protótipo de Gate Drive: Um protótipo desenvolvido com base nas recomendações do fabricante Gan Systems, conforme descrito no documento (GAN SYSTEMS, 2022). Este protótipo de gate drive incorpora uma fonte isolada e pulsos isolados. A imagem ilustra o resultado do projeto do protótipo gate drive.

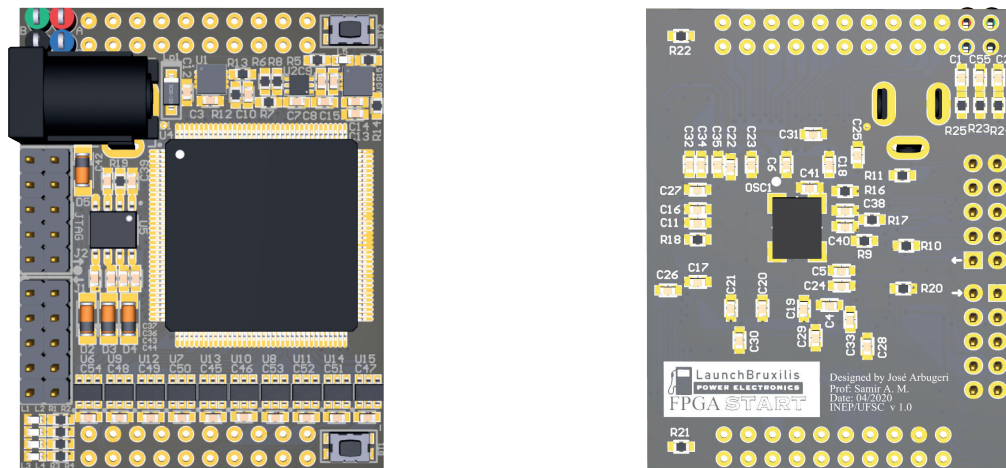


Figura 53 – Protótipo do Módulo de Controle FPGA: Um protótipo desenvolvido pelo autor utilizando o FPGA Cyclone IV e conectores compatíveis com o LaunchPad da Texas Instruments. Este módulo de controle FPGA oferece recursos para processamento e controle de dados, como: ADC, PWM, SPI entre outros recursos que podem ser implementado via descrição de hardware (VHDL).

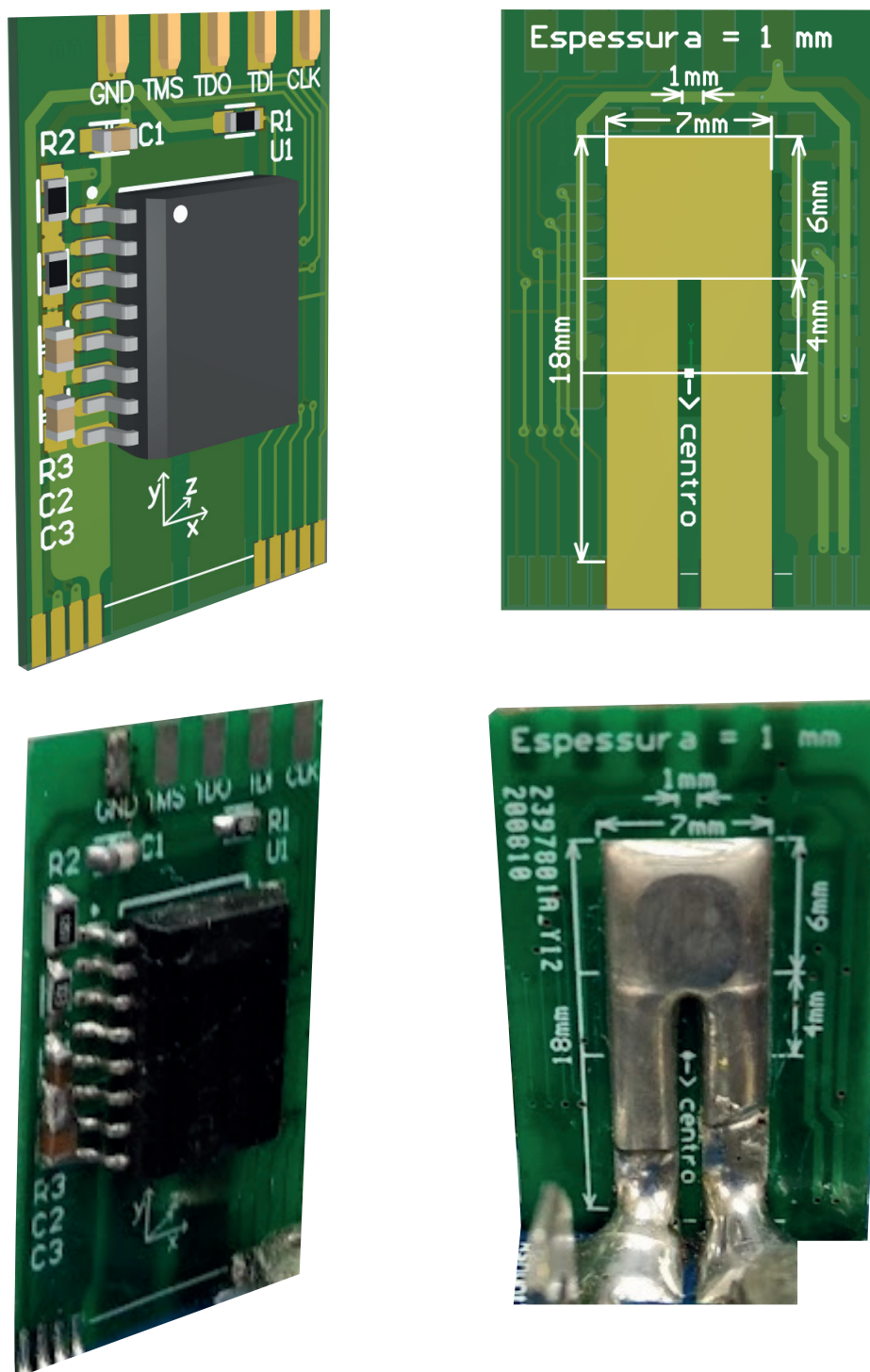


Figura 54 – Protótipo do sensor de corrente CFS1000: Um protótipo desenvolvido pelo autor utilizando o sensor CFS1000 da Sensitec. É evidenciado as dimensões da PCB na imagem.

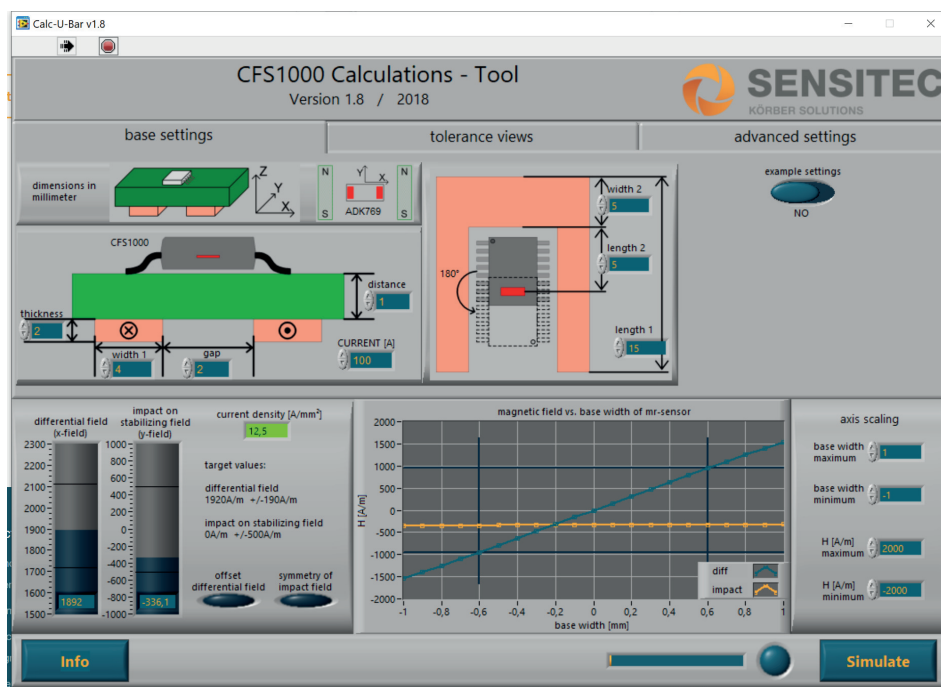


Figura 55 – Software fornecido pelo fabricante para projeto da PCB do sensor de corrente CFS1000. Fonte: (SENSITEC, s.d.)

APÊNDICE C – APÊNDICE C

C.1 REFLOW-OVEN (FORNINHO)

Para realizar a solda dos componentes foi criado um forno de refusão caseiro. Um forno de refusão, ou reflow oven, é um equipamento utilizado para fundir e soldar componentes eletrônicos em placas de circuito impresso (PCBs). Os fornos de refusão comerciais são caros e podem não estar acessíveis para todos os entusiastas de eletrônica. No entanto, é possível construir um forno de refusão caseiro com peças facilmente encontradas e a um custo acessível.



Figura 56 – Forno construído para solda das PCBs

Antes de construir um forno de refusão caseiro, é importante entender os princípios

básicos de como ele funciona. Os componentes eletrônicos são soldados nas PCBs usando pasta de solda, que contém ligas metálicas que se fundem a temperaturas específicas. O forno de refusão é aquecido a uma temperatura específica e mantido nessa temperatura por um período de tempo, permitindo que a pasta de solda derreta e as soldas sejam formadas.

Para construir um forno de refusão caseiro, foi utilizado um forno convencional de cozinha com controle de temperatura, um termopar, um controlador de temperatura, um relé de estado sólido. O forno de cozinha será modificado para se tornar um forno de refusão, com o termopar conectado ao controlador de temperatura para medir a temperatura dentro do forno.

O relé de estado sólido é utilizado para controlar a energia que é fornecida ao forno, e o ventilador é usado para garantir que o ar quente circule uniformemente dentro do forno, evitando áreas de temperatura mais elevada ou mais baixa. O forno construído pode ser visto na Figura 56. Toda a parte de código implementada e PCB está disponível para uso no repositório (ARBUGERI, 2023), dentro da pasta 05 Forninho.

C.2 TÉCNICA DE CONTROLE

O controle utilizado para para controle da temperatura do forno foi o controle usando preditor de smith. O controle preditor de Smith, também conhecido como controle preditivo de Smith ou simplesmente controle preditor, é uma técnica de controle avançado que é usada em processos industriais para melhorar o desempenho do sistema de controle.

O controle preditor de Smith foi desenvolvido por Smith em 1957, e é baseado em um modelo matemático do processo controlado que é utilizado para prever o comportamento futuro do sistema. Ele utiliza um modelo de predição para calcular a saída futura do processo, com base nas entradas futuras. Essas previsões são então usadas para calcular a ação de controle necessária para alcançar a saída desejada.

O controle preditor de Smith é particularmente útil em sistemas com atrasos, que são comuns em processos industriais. Os atrasos podem ser causados por uma variedade de fatores, como a inércia do sistema, o tempo de resposta dos sensores e atuadores, ou o tempo de transferência dos materiais dentro do processo. Esses atrasos podem dificultar a estabilidade e a precisão do sistema de controle.

O controle preditor de Smith funciona ajustando a resposta do sistema para compensar os atrasos, de modo que a saída do processo corresponda à saída desejada. Ele também é capaz de lidar com perturbações no processo, como mudanças na carga ou na temperatura.

O controle preditor de Smith é geralmente implementado em sistemas de controle de malha fechada, onde a saída do processo é comparada com a saída desejada e, em seguida,

a ação de controle é ajustada de acordo. Ele é frequentemente utilizado em aplicações de controle de processos, como na indústria química, na produção de energia e em processos de manufatura.

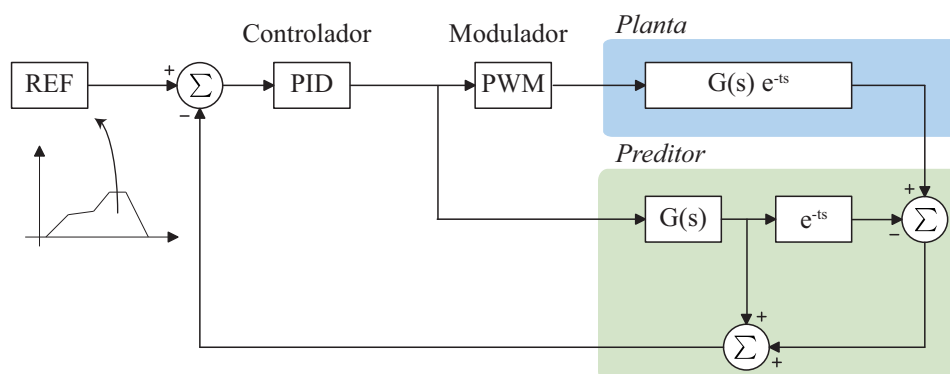


Figura 57 – Controle de temperatura usando preditor de smith

C.3 SOFTWARE DE CONTROLE

O controle do forno de refusão pode ser implementado em um Arduino usando um código simples e eficaz. O código deve incluir funções para ler a temperatura do forno usando um termopar e um controlador de temperatura usando máquina de estados, ajustar a potência fornecida ao forno usando um relé de estado sólido. O código desenvolvido está disponível no repositório (ARBUGERI, 2023).

C.4 APLICATIVOS USANDO PYTHON

Um aplicativo de monitoramento e plotagem foi criado para acompanhar o desempenho do forno de refusão controlado pelo Arduino. O aplicativo é usado para exibir informações sobre a temperatura do forno e outros parâmetros importantes em tempo real, permitindo que o operador monitore o processo de soldagem e faça ajustes conforme necessário. O código desenvolvido está disponível no repositório (ARBUGERI, 2023). Nas figuras apresentadas em 58, 59 e 60.

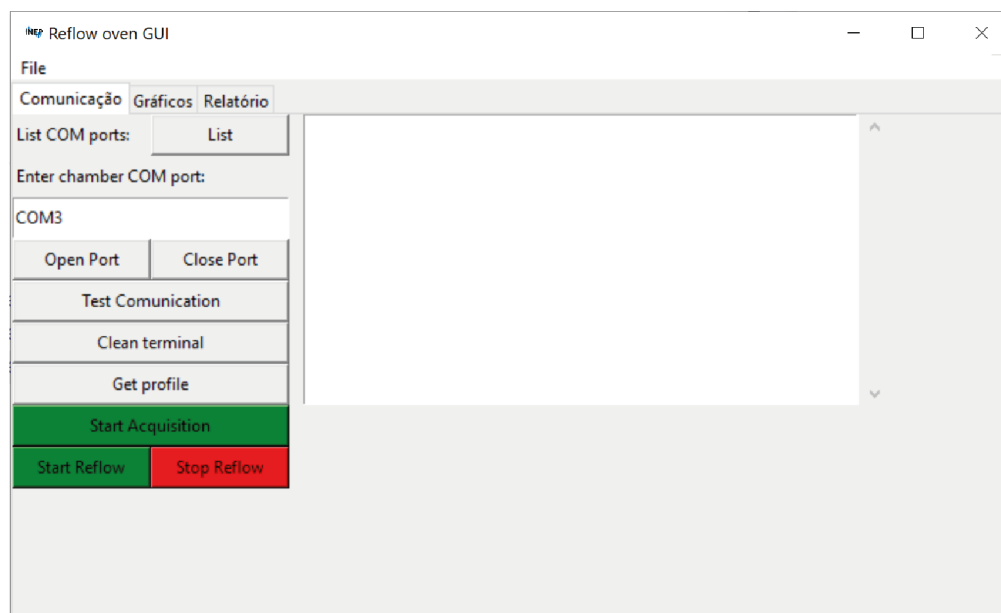


Figura 58 – Página 1 do aplicativo de monitoração implementado em python.

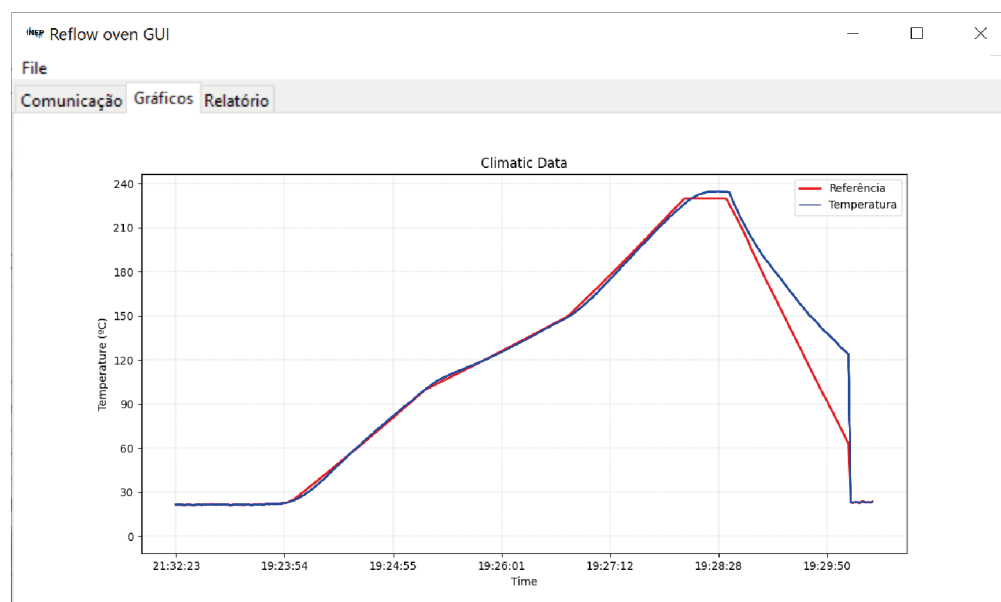


Figura 59 – Página 2 do aplicativo de monitoração implementado em python.

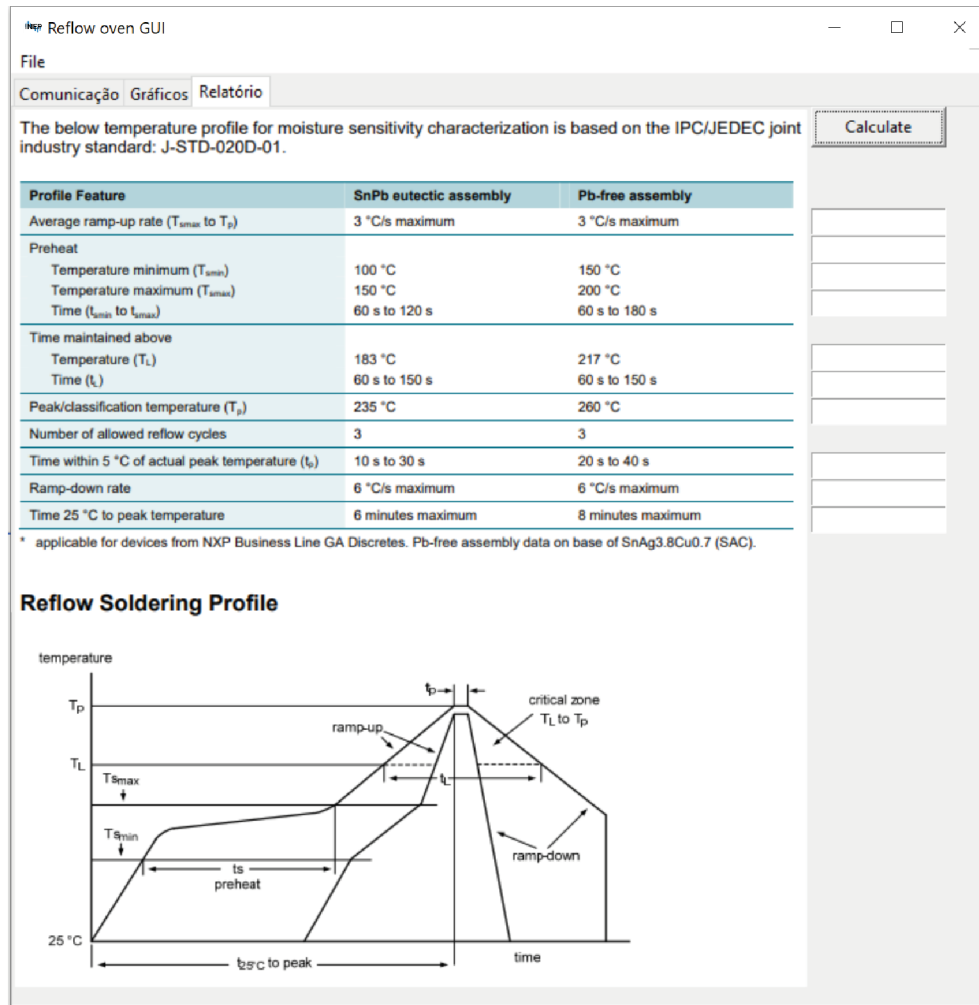


Figura 60 – Página 3 do aplicativo de monitoração implementado em python.