

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
FACULDADE DE ENGENHARIAS, ARQUITETURA E URBANISMO E GEOGRAFIA
CURSO DE ENGENHARIA ELÉTRICA

RODOLFO BIFFI POMPEU DE BARROS

IMPLEMENTAÇÃO EM FPGA DO ALGORITMO DE MODULAÇÃO
POR VETORES ESPACIAIS (SVPWM)

Campo Grande - MS

Julho, 2021

RODOLFO BIFFI POMPEU DE BARROS

**IMPLEMENTAÇÃO EM FPGA DO ALGORITMO DE MODULAÇÃO
POR VETORES ESPACIAIS (SVPWM)**

Trabalho de Conclusão de Curso apresentado
como exigência para obtenção do grau de
Bacharelado em Engenharia Elétrica da Uni-
versidade Federal de Mato Grosso do Sul –
UFMS.

Orientador: Raymundo Cordero Garcia

Campo Grande - MS

Julho, 2021



Serviço Público Federal
Ministério da Educação
Fundação Universidade Federal de Mato Grosso do Sul



ATA DE DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO

No dia 14 de julho de 2021, o acadêmico Rodolfo Biffi Pompeu de Barros, RGA nº 2016.2103.070-9, defendeu em sessão pública o Trabalho de Conclusão de Curso intitulado: "**Implementação em FPGA do Algoritmo de Modulação por Vetores Espaciais (SVPWM)**".

A seguinte Comissão Avaliadora se reuniu para a avaliação do trabalho:

Prof. Dr. Raymundo Cordero Garcia (orientador)
(Universidade Federal de Mato Grosso do Sul)

Prof. Dr. Luigi Galotto Junior
(Universidade Federal de Mato Grosso do Sul)

Prof. Dr. Marcio Luiz Magri Kimpara
(Universidade Federal de Mato Grosso do Sul)

Tendo sido considerado:

☒ **Aprovado** ☐ **Aprovado com restrições** ☐ **Reprovado**

Observações da banca (em caso de restrições ou de reprovação):

Coordenador do Curso de Graduação em Engenharia Elétrica (FAENG/UFMS):
Ciente,

Prof.ª Dr.ª Luciana Cambraia Leite
Coordenadora de curso, em exercício



Documento assinado eletronicamente por **Raymundo Cordero Garcia, Professor do Magisterio Superior**, em 14/07/2021, às 09:34, conforme horário oficial de Mato Grosso do Sul, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Luigi Galotto Junior, Professor do Magisterio Superior**, em 14/07/2021, às 09:35, conforme horário oficial de Mato Grosso do Sul, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Marcio Luiz Magri Kimpara, Professor do Magisterio Superior**, em 14/07/2021, às 09:35, conforme horário oficial de Mato Grosso do Sul, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufms.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2672397** e o código CRC **584F3FC7**.

FACULDADE DE ENGENHARIAS, ARQUITETURA E URBANISMO E GEOGRAFIA

Av Costa e Silva, s/nº - Cidade Universitária
Fone:
CEP 79070-900 - Campo Grande - MS



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



DECLARAÇÃO DE AUTORIA, RESPONSABILIDADE E AUTORIZAÇÃO DE PUBLICAÇÃO

Rodolfo Biffi Pompeu de Barros, residente e domiciliado na cidade de Campo Grande, Estado de Mato Grosso do Sul, portador do RG de nº 1.415.239 (SEJUSP/MS) e CPF nº 038.363.421-03, declaro que o Trabalho de Conclusão de Curso (TCC) apresentado com o título "**Implementação em FPGA do Algoritmo de Modulação por Vetores Espaciais (SVPWM)**" é de minha autoria e assumo a total responsabilidade pelo seu conteúdo e pela originalidade do texto.

Declaro que identifiquei e referenciei todas as fontes e informações gerais que foram utilizadas para construção do presente texto.

Declaro também que este texto não foi publicado, em parte, na íntegra ou conteúdo similar em outros meios de comunicação, tendo sido enviado com exclusividade para a conclusão do curso de graduação em Engenharia Elétrica da Universidade Federal de Mato Grosso do Sul (UFMS). Autorizo desta forma a publicação do trabalho no site do curso para divulgação.

Campo Grande, 14 de julho de 2021.

Rodolfo Biffi Pompeu de Barros

RGA 2016.2103.070-9

Curso de Engenharia Elétrica - UFMS



Documento assinado eletronicamente por **RODOLFO BIFFI POMPEU DE BARROS, Usuário Externo**, em 10/08/2021, às 16:11, conforme horário oficial de Mato Grosso do Sul, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufms.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2672452** e o código CRC **7D88DE71**.

COLEGIADO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA - BACHARELADO

RESUMO

Resumo: O objetivo deste trabalho foi o estudo da modulação por vetores espaciais, desde sua fundamentação teórica e implementação do algoritmo em FPGA (Field Programmable Gate Array) até testagem experimental. Esta técnica pode ser utilizada na modulação do inversor trifásico de dois níveis, a qual é muito utilizado para o acionamento de máquinas elétricas trifásicas. Com o avanço das tecnologias de veículos elétricos, motivado pela crescente preocupação das sociedades humanas com a preservação do meio ambiente, encontrar meios mais eficientes de acionar e controlar motores elétricos é uma preocupação constante para a indústria e também para academia. Comparada com outras técnicas de modulação, como senoidal e por banda de histerese, a modulação por vetores espaciais apresenta melhor eficiência em relação a TDH e também melhor aproveitamento da tensão de barramento CC. Dessa forma, o avanço nas pesquisas envolvendo a otimização do algoritmo, garante um melhor aproveitamento dos recursos disponíveis. Neste trabalho, serão apresentadas os resultados de cada uma das etapas principais da modulação por vetores espaciais para que seja possível validar o funcionamento do algoritmo. Estas ondas são, a seleção de setores, os tempos e também os resultados de tensão e corrente.

ABSTRACT

Abstract: The aim of this work was the study of space vector modulation, from its theoretical foundation and implementation of the FPGA (Field Programmable Gate Array) algorithm to experimental testing. This technique can be used without modulation of the two-level three-phase inverter, a method widely used to drive three-phase electrical machines. With the advancement of electric vehicle technologies, motivated by the growing concern of human societies with environmental preservation, finding more efficient ways to control the drive-train of electric motors is a constant concern for industry and also for academy. Compared with other modulation techniques, such as sinusoidal and hysteresis band, modulation by spatial vectors presents better efficiency in relation to THD and also better use of DC voltage bus. Thus, the advance in research involving an optimization of the algorithm, guaranteed a better use of available resources. In this work, the results of each of the main steps of the modulation by space vectors will be accepted so that it is possible to validate the functioning of the algorithm. These waves are, a selection of sectors, times and also voltage and current results.

LISTA DE ILUSTRAÇÕES

Figura 1 – <i>Drivetrain</i>	8
Figura 2 – Estrutura de Controle de um VE utilizando um FPGA	10
Figura 3 – Inversor monofásico de meia ponte	15
Figura 4 – Saída do inversor	16
Figura 5 – Inversor onda completa	17
Figura 6 – Topologia inversor trifásico de dois níveis	18
Figura 7 – Sinais de comando das chaves elétricas para um inversor trifásico . . .	19
Figura 8 – Configuração de condução, 5-6-1 ou (1,0,1)	20
Figura 9 – Configuração de condução, 6-2-1 - (1,0,0)	20
Figura 10 – Configuração de condução, 1-2-3 - (1,1,0)	20
Figura 11 – Configuração de condução, 2-3-4 - (0,1,0)	21
Figura 12 – Configuração de condução, 3-4-5 - (0,1,1)	21
Figura 13 – Configuração de condução, 4-5-6 - (0,0,1)	21
Figura 14 – Gráfico das tensões de fase.	23
Figura 15 – Gráfico das tensões de fase.	24
Figura 16 – Gráfico das Tensões de fase.	25
Figura 17 – Gráfico das tensões de fase deslocado.	27
Figura 18 – Esquema de controle e modulação.	30
Figura 19 – Hexagono Regular.	30
Figura 20 – Combinação de vetores.	31
Figura 21 – Ordem de chaveamento.	33
Figura 22 – Cálculo do setor.	36
Figura 23 – Cálculo do setor.	37
Figura 24 – Cálculo dos Tempos.	39
Figura 25 – Cálculo dos Tempos.	40
Figura 26 – Fluxograma do código apresentado.	41
Figura 27 – Bancada Experimental.	42
Figura 28 – Overview	43
Figura 29 – Setor de trabalho.	43
Figura 30 – tempo t_a	44
Figura 31 – tempo t_b	44
Figura 32 – tempo t_x	45
Figura 33 – tempo t_y	45
Figura 34 – Tensões de linha simulação.	46
Figura 35 – Tensões de linha na saída do FPGA.	46
Figura 36 – Tensões de linha na saída do inversor.	47
Figura 37 – Corrente de fase.	48

LISTA DE TABELAS

Tabela 1	– Tensões de fase e linha para cada esquema de condução	22
Tabela 2	– Tensões de saída do inversor em α e β	29
Tabela 3	– Tabela de cálculo dos tempos t_a e t_b	32
Tabela 4	– Seleção do setor de trabalho.	34
Tabela 5	– Otimização do algoritmo.	38

SUMÁRIO

1	INTRODUÇÃO	7
1.1	Veículos Elétricos (VE)	7
1.1.1	Controle Em FPGA para Veículos Elétricos	9
1.1.2	Modulação Por Vetores Espaciais	11
1.2	Objetivos	12
1.2.1	Objetivo Geral	12
1.2.2	Objetivos Específicos	12
1.3	Organização do Trabalho	12
2	FUNDAMENTAÇÃO TEÓRICA.	14
2.1	Inversor Trifásico de Dois Níveis	14
2.1.1	Topologias Básicas.	14
2.1.2	Inversores Trifásicos.	18
2.1.3	Análise dos esquemas de condução.	20
2.2	Representação por vetores espaciais.	25
2.2.1	Transformada de Clarke	28
2.3	Modulação Por Vetores Espaciais.	29
2.4	Representação em ponto fixo.	34
3	METODOLOGIA	36
3.1	Geração das tensões de referência em VHDL.	36
3.2	Cálculo do setor em VHDL.	36
3.3	Cálculo dos tempos.	37
3.4	Geração do sinal <i>PWM</i>	39
4	RESULTADOS	42
4.0.1	Resultados da simulação.	42
4.0.2	Resultados experimentais.	46
5	CONCLUSÃO	49
	REFERÊNCIAS	50
	APÊNDICE A – CÓDIGO COMPLETO	51

1 INTRODUÇÃO

A crescente preocupação da sociedade moderna com as consequências ambientais da ação humana motivou esforços para que fossem desenvolvidos meios alternativos de transporte com menor impacto ambiental. Os veículos elétricos híbridos (VEH) e veículos elétricos (VE) figuram como uma das principais alternativas aos meios de transporte movidos a combustíveis fósseis. Nos últimos anos, os centros de pesquisa e a indústria têm se esforçado para elevar a eficiência destes meios de transporte. Um dos principais fatores que contribuem para o aumento da eficiência são os algoritmos que controlam as máquinas elétricas por meio de seus atuadores. Um controle mais eficiente significa menores perdas e também um ganho considerável em segurança, uma vez que a precisão destas máquinas está relacionada a precisão do controlador (1).

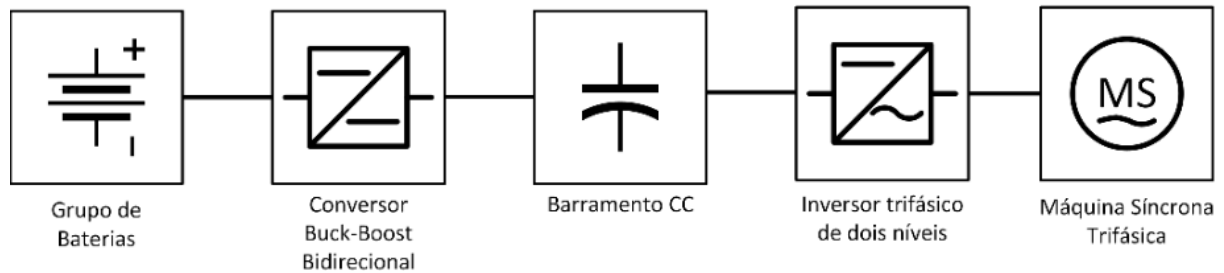
1.1 Veículos Elétricos (VE)

O número de veículos a combustão, tanto para fins comerciais quanto para uso pessoal, atingiu quase 1.3 bilhão de unidades em 2015 (2). A produção destes veículos vem aumentando a cada ano, em 2019 foram produzidos mais de 96 milhões de automóveis, segundo a ACEA (3). Meios de transporte movidos por meio de motor a combustão representam 15,9% do total de emissões de CO_2 provenientes da atividade humana na atmosfera, segundo a OICA (4). Desta forma, meios de transporte alimentados por combustíveis fósseis são um dos principais agentes das mudanças climáticas causadas pela ação humana. A dependência humana de fontes de energia não renováveis é outro aspecto negativo da massiva utilização destes tipos de veículos. Todos estes fatores contribuíram para pesquisas na área dos transportes com baixa ou nenhuma emissão, focados não só em transporte pessoal, mas em diversos tipos de veículos com aplicação comercial e industrial. Veículos elétricos não emitem CO_2 na atmosfera e veículos híbridos produzem um volume substancialmente menor em relação aos veículos a combustão (1).

Os veículos elétricos apresentam ganhos que vão além do escopo ambiental e de sustentabilidade, estes veículos oferecem uma série de vantagens para o usuário. Os gastos financeiros com energia para alimentar um veículo elétrico, comparado aos custos dos combustíveis convencionais, chega a ser 50% menor. Mesmo em casos onde este veículo é utilizado para longas viagens, onde sua eficiência é inferior em função da menor recorrência do estado regenerativo. O estado regenerativo é caracterizado pela inversão do sentido da corrente do *Drivetrain*, ocasionado por uma característica inerente às máquinas elétricas em processo de frenagem. A frenagem regenerativa aumenta o estado de carga da bateria sempre que acionada, em um ambiente urbano, este acionamento é mais frequente de modo a aumentar o rendimento do VE em relação a um cenário onde viaja longas distâncias (1). A indústria e a academia buscam desenvolver métodos para melhorar a utilização dos recursos empregados nestes veículos. Os estudos relacionados ao aumento da eficiência

nestes carros, geralmente está focado em um dos componentes do *Drivetrain*: assim é chamado o conjunto de equipamentos que tem a função de fornecer potência ao motor elétrico de maneira a atender suas demandas de torque.

Figura 1 – *Drivetrain*



Fonte:(1)

A Figura 1, apresenta um diagrama simplificado do *Drivetrain* real, apenas com os principais componentes do sistema. A bateria tem a função de fornecer potência ao sistema, mas em geral, seus níveis de tensão estão abaixo do que seria necessário para alimentar o motor. O conversor Buck-Boost Bidirecional tem a função de elevar os níveis de tensão fornecidos pela bateria ao barramento CC e ao passar para o estado regenerativo, o conversor adequa os níveis de tensão fornecidas pelo barramento à bateria. O barramento CC funciona como um ponto de conexão dos componentes, geralmente contém um capacitor de filtro com a função de manter os níveis de tensão adequados ao projeto. O inversor trifásico de dois níveis converte a tensão contínua em tensão alternada que alimenta a máquina síncrona trifásica (5).

O controlador do inversor é o responsável pelos níveis de tensão e corrente enviados a máquina elétrica. Otimizar os algoritmos do controlador destes atuadores é um método efetivo para melhorar a eficiência dos carros, uma vez que não demanda adição de componentes e nem modificações relevantes na planta. Existe um grande esforço científico para criar técnicas de modulação e controle com maior confiabilidade, precisão e eficiência. A modulação tem como função definir o estado das chaves elétricas do atuador de modo a sintetizar os níveis de tensão demandados pelo controlador (6). O aprimoramento das técnicas de controle e modulação, possibilitaram a substituição dos motores de corrente contínua (CC) pelo conjunto inversor e máquina de corrente alternada (CA). Motores de indução, quando comparados a motores de corrente contínua, tem um menor custo de produção e menor frequência de manutenção. Em conjunto com um inversor controlado, é capaz de atingir níveis de precisão no controle de velocidade e posição superiores ao motor de corrente contínua. Segundo (7), o motor síncrono de ímã permanente (MSIP, ou Permanent Magnet Synchronous Motor – PMSM em inglês) vem se destacando no mercado de veículos elétricos em relação ao motor de corrente contínua e o motor de indução. Esta máquina, apresenta maior densidade de potência e maior eficiência devido ao ímã

permanente que substitui os enrolamentos de campo da máquina. Além de apresentar uma maior capacidade de gerar torque, o seu modelo matemático poder ser obtido de maneira mais simplificada em relação aos outros.

1.1.1 Controle Em FPGA para Veículos Elétricos

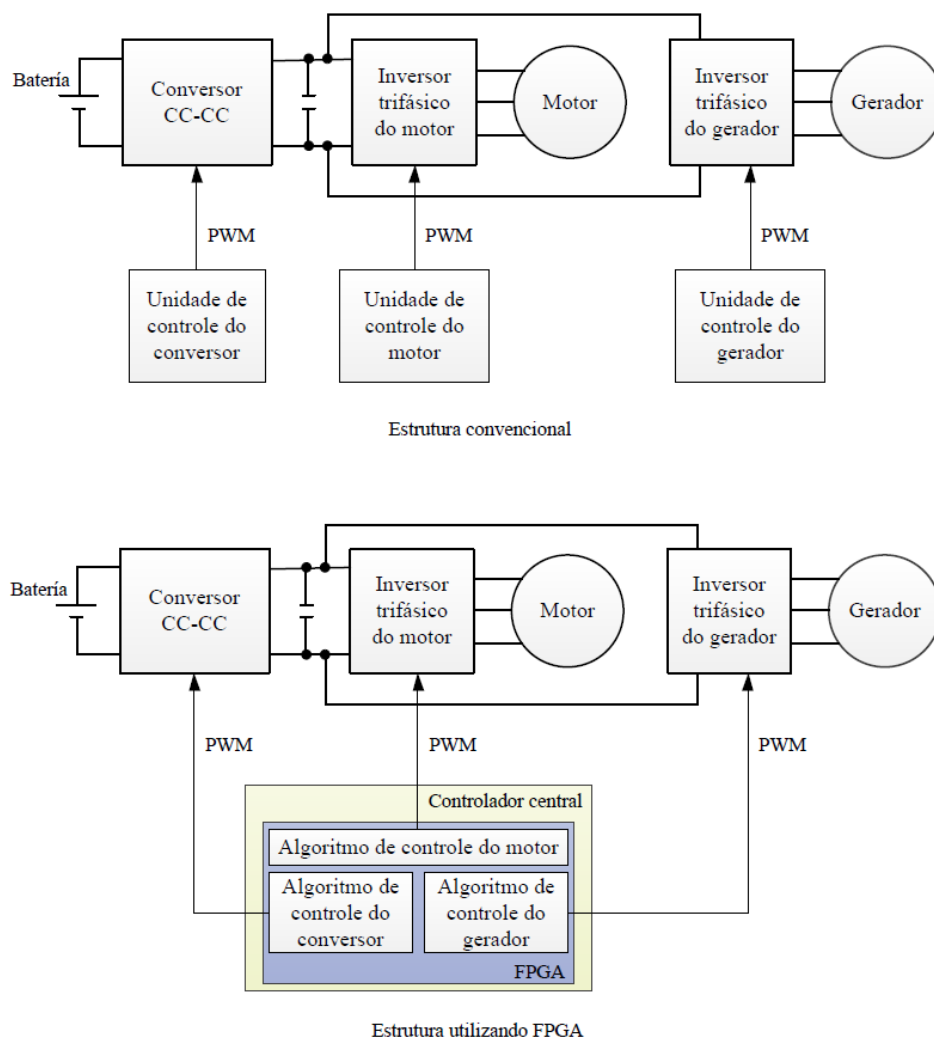
Os métodos de controle digital em tempo real estão ganhando espaço em função de suas muitas vantagens em relação ao controle realizado de forma analógica. Utilizando microprocessadores, é possível desenvolver um algoritmo efetivo de controle de maneira versátil e prática, além de substituir uma quantidade considerável de componentes que seriam necessários para implementar o controle analógico (8). Algoritmos de modulação e controle de máquinas trifásicas são implementados em diversas arquiteturas de hardware, como micro controladores, DSPs e DSPICs. Estes dispositivos apresentam altas velocidades de processamento e dispõe de memória suficiente para armazenar as instruções necessárias para realizar o controle. Outras características que justificam a utilização destas arquiteturas são o baixo custo financeiro, amplo e fácil acesso a recursos técnicos e facilidade de implementação. No entanto, o fato de executarem instruções de maneira sequencial, acaba sendo uma desvantagem em casos onde são necessárias simultâneas ações de controle e monitoramento de estados. Em plantas com esta característica, seriam necessários mais de um destes dispositivos para garantir o tempo de resposta e estabilidade do controle, elevando o custo deste tipo de projeto (7).

Neste contexto, existe uma demanda por dispositivos capazes de executar instruções paralelas, possibilitando o monitoramento e controle simultâneo. O FPGA (*Field Programmable Gate Array em Inglês*) utiliza blocos lógico programáveis permitindo codificar modelos matemáticos em linguagem de Hardware, isso significa que a cada vez que um código é compilado o dispositivo modifica suas conexões internas. A linguagem de hardware mais utilizado no desenho de sistemas digitais em FPGAs é o VHDL (*VHSIC Hardware Description Language, em Inglês*). O FPGA pode operar com lógica sequencial ou paralela, permitindo que vários algoritmos sejam executados simultaneamente em um mesmo dispositivo. Esta característica permite que o FPGA seja capaz de executar ações de controle e monitoramento de estados em diversos componentes da planta ao mesmo tempo e com hierarquia bem definida. Além disso, com o auxílio de softwares é possível fazer simulações de plantas controladas por algoritmos embarcados no FPGA, o que reduz significativamente o custo de materiais e em tempo para realizar testes com algoritmos em desenvolvimento (7), (9).

Os veículos Elétricos (VE) e Veículos Elétricos Híbridos (VEH) requerem controle simultâneo dos seus componentes. Estes veículos podem possuir mais de um motor, geralmente um para tração e o outro para arranque, ambos necessitam ser monitorados e sofrer ações de controle simultaneamente. Ademais, seus componentes, como conversor

CC-CC, inversor, baterias, freio dinâmico, entre outros, também necessitam destas ações e monitoramento simultâneo. Para que estes dispositivos se comportem de maneira adequada, são necessárias uma grande quantidade de cálculos executadas em um curto espaço de tempo e muitas vezes de forma paralela. Assim o FPGA tem ganhado espaço entre os projetos de controladores para veículos elétricos, em função das suas capacidades citadas anteriormente (7), (9). A figura 2, mostra uma comparação entre um sistema de controle de um VE convencional controlado por unidades independentes e outro controlado por um único FPGA. Ilustrando a capacidade do FPGA de performar diversas ações de controle em um único dispositivo.

Figura 2 – Estrutura de Controle de um VE utilizando um FPGA



Fonte:(7)

Além dos dispositivos relacionados ao drivetrain, existe uma crescente preocupação com sistemas auxiliares relacionados a segurança do motorista e também de pedestres como câmeras de segurança para processamento de imagens, freios automáticos, efeitos sonoros para alertar pedestres, entre outros. O FPGA se mostra uma boa opção para

realizar o processamento das informações destes sistemas, o que contribui para considerar este tipo de processador em um projeto de controlador para VE e VEH (9).

1.1.2 Modulação Por Vetores Espaciais

Em diversas aplicações de inversores, é necessário que se faça um controle adequado dos níveis de tensão de saída, este controle é feito alterando o ganho do inversor. Um dos métodos mais utilizados para controle de ganho do inversor é a modulação por largura de pulso (*Pulse Width Modulation – PWM*, do inglês) (10). Este método de modulação geralmente é implementado realizando a comparação entre os sinais de referência e um sinal do tipo portadora de alta frequência. Por meio de comparações de módulo feitas dentro de um controlador digital ou até mesmo por um controlador analógico, é possível definir os estados das chaves elétricas de um inversor para cada instante de tempo, de modo a sintetizar uma onda de tensão muito próxima às referências aplicadas no controlador. Existem muitos métodos de PWM, como a modulação por largura de pulsos múltiplos. Utilizando o princípio de que ao aumentar a frequência da portadora, também aumentam o número de ciclos de comparação o que resulta em um menor conteúdo harmônico. A tensão de saída pode ser modificada através do índice de modulação, este coeficiente varia de 0 até 1 e é definido pela equação (1), (9) - (10).

$$m = \frac{|V_r|}{\frac{2V_{cc}}{\pi}} \quad (1)$$

- V_r Pico da onda senoidal de referência.
- V_{cc} Tensão do barramento cc.

Outro método amplamente utilizado é o da modulação por banda de histerese, ou modulação Delta. Nesta técnica uma referência de corrente varia entre dois limitantes, chamados de banda de histerese. Esta técnica demanda elevada velocidade de processamento quando se deseja alto nível de precisão, em decorrência da banda de histerese ser pequena o controlador deve realizar a comutação em curtos períodos de tempo. O método de modulação mais utilizado em aplicações na indústria é a modulação senoidal. Neste método um sinal senoidal com frequência fundamental é comparado a um sinal triangular, de modo a sintetizar um sinal PWM cuja a fundamental possui características senoidais. Este método possui algumas variações, como utilizar mais de um seno para facilitar o projeto do filtro, ou até adicionar harmônicas ao sinal fundamental para obter um melhor aproveitamento da tensão de barramento. No entanto, este último apresenta alto grau de complexidade para ser embarcado em um processador digital. A modulação senoidal no geral é bastante efetiva e não requer alta capacidade de cálculo por parte dos dispositivos de processamento e por isso acaba tendo um baixo custo de implementação (6), (10).

A técnica abordada neste trabalho é a modulação por vetores espaciais, (Space Vector Pulse Width Modulation – SVPWM). Esta técnica sintetiza um vetor espacial a partir dos sinais de referência que alimentam o sistema, mostrando um melhor aproveitamento da tensão de barramento do inversor, sem necessidade da adição de harmônicos à fundamental. Além de apresentar uma menor taxa de distorção harmônica e maior rendimento em relação as técnicas citadas (modulação senoidal e histerese). A modulação por vetores espaciais possui um alto custo computacional associado (requer muitas operações matemáticas). Isso motiva pesquisas para otimizar o citado algoritmo (6), (10).

1.2 Objetivos

1.2.1 Objetivo Geral

Implementar o algoritmo de modulação por vetores espaciais em VHDL, utilizando ferramentas como representação em ponto fixo e *look-up table*. Montar uma bancada para testar o algoritmo em um motor real.

1.2.2 Objetivos Específicos

Implementação em FPGA do algoritmo *Space Vector* e montagem de uma bancada experimental com atuador, planta e sensoramento para realização de ensaios e obtenção de resultados experimentais. Para facilitar a compreensão e a divisão do tempo para realizar estas atividades, foram criados tópicos com as etapas de elaboração, segue:

- Simulação da técnica de modulação por vetores espaciais.
- Codificação e verificação do algoritmo.
- Implementação do algoritmo em FPGA.
- Montagem da bancada com sensoramento.
- Ensaios experimentais.

1.3 Organização do Trabalho

- No Capítulo 1, é apresentado uma revisão bibliográfica básica no intuito de situar o trabalho no contexto geral, e a motivação para o desenvolvimento do trabalho.
- No Capítulo 2, é apresentada a fundamentação teórica do trabalho. Descrevendo os principais aspectos da teoria e funcionamento do inversor trifásico de dois níveis, da

teoria de campos girantes, da modulação por vetores espaciais, FPGA e também a teoria de ponto fixo.

- No Capítulo 3, são apresentadas as metodologias de implementação do algoritmo.
- No Capítulo 4, são apresentados resultados simulados e experimentais sobre a implementação de vetores espaciais em FPGA.
- No Capítulo 5, são apresentadas as conclusões finais e propostas de continuidade do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA.

2.1 Inversor Trifásico de Dois Níveis

Os conversores estáticos que fazem a conversão de corrente contínua (CC) em corrente alternada (CA), são conhecidos como inversores. Estes dispositivos são capazes de controlar amplitude e frequência de tensão alternada de saída a partir de uma fonte de tensão contínua. A tensão fornecida pode variar de acordo com as demandas de projeto quando este dispositivo está sendo controlado em malha fechada. O controlador regula o ganho do inversor, de modo a adequar a amplitude da tensão de saída mesmo a partir de uma tensão de entrada fixa ou não controlada. Este ganho é ajustado a partir do controle do período em que os dispositivos semicondutores permanecem ativos, fornecendo potência à carga. Como mencionado anteriormente, um método amplamente utilizado para realizar o controle do ganho é através da modulação por largura de pulso. Idealmente, as formas de onda de saída são senoidais, no entanto estes sinais contêm harmônicos indesejados. Estes harmônicos são inerentes ao processo de chaveamento, mas podem ser significativamente reduzidos ao aplicar uma técnica de modulação adequada em seus dispositivos semicondutores. Inversores são largamente utilizados na indústria, é mais comum serem observados em acionamento e controle de máquinas de corrente alternada e para conversão da energia gerada em painéis solares (CC) em eletricidade própria para consumo na maioria dos equipamentos do mercado (CA). Mas também são essenciais para a maioria dos dispositivos embarcados que possuem máquinas de corrente alternada, já que estes dispositivos geralmente são alimentados por meio de baterias que fornecem tensão contínua. Nos carros elétricos desempenha papel fundamental, já que através deste conversor foi possível viabilizar a utilização de motores menores, mais leves, com maior capacidade torque e rendimento mais elevado, tornando o veículo elétrico um produto competitivo em muitos aspectos. Estes inversores podem ser monofásicos ou trifásicos e as chaves elétricas podem variar entre BJTs, MOSFETs, IGBTs, GTOs, entre outros. Estes dispositivos são geralmente acionados por sinais PWM para gerar a tensão de saída alternada. A escolha do tipo de chave depende das especificações de cada projeto, como frequência de chaveamento, tolerância de variação de tensão e corrente e também o custo financeiro.

2.1.1 Topologias Básicas.

A figura 3 ajuda na compreensão do princípio de operação do inversor monofásico de meia ponte. Quando $Q1$ está conduzindo, a tensão sobre a carga é de $\frac{V_s}{2}$, isso ocorre por um período de $\frac{T_0}{2}$. Quando $Q1$ deixa de conduzir e $Q2$ inicia a condução e a tensão sobre a carga passa a ser $-\frac{V_s}{2}$, por um período de $\frac{T_0}{2}$. Para evitar um curto circuito, a

modulação deve ser feita de modo a prevenir que ambas as chaves conduzam ao mesmo tempo, isso é feito ao adicionar um "tempo morto" no acionamento de uma das chaves. Este tempo funciona como um atraso para acionamento, isso é necessário porque em aplicações práticas a condução de uma chave elétrica custa um curto período de tempo para cessar. Por este fato, as chaves de um mesmo braço de um inversor são chamadas complementares, isso significa que apenas uma delas irá conduzir corrente para cada intervalo. O processo de chaveamento, faz surgir na saída do inversor uma onda quadrada de tensão com amplitude $\frac{V_s}{2}$, mas que possui valor negativo em metade do tempo de condução. Esta onda está expressa na figura 4. Para obter a tensão eficaz de saída, pode-se utilizar (2), esta equação funciona para essa e também outras topologias de inversores, da mesma forma, (3) permite o cálculo da tensão de saída para cada uma de suas harmônicas.

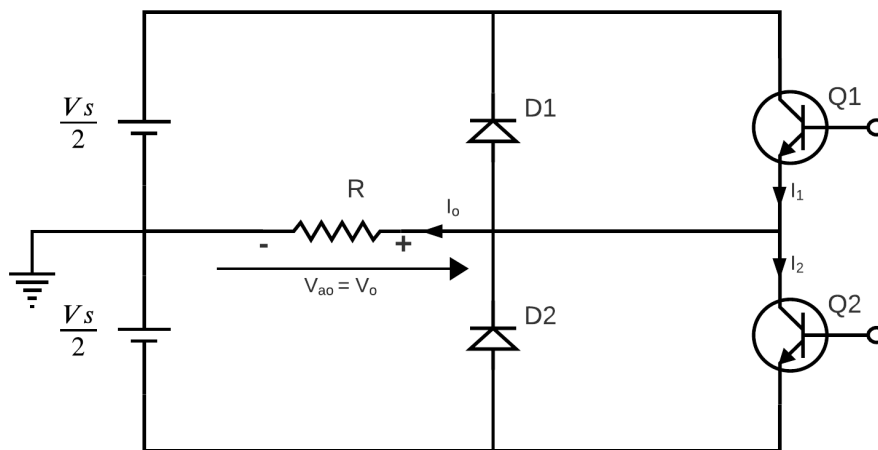
$$V_{RMS} = \sqrt{\frac{2}{T_0} \cdot \int_0^{\frac{T_0}{2}} V^2 dt} \quad (2)$$

A tensão instantânea de saída pode ser expressa na seguinte série de Fourier.

$$v_0 = \sum_{n=1,3,5,\dots}^{\infty} \frac{2V_s}{n\pi} \cdot \sin(n\omega t) \quad (3)$$

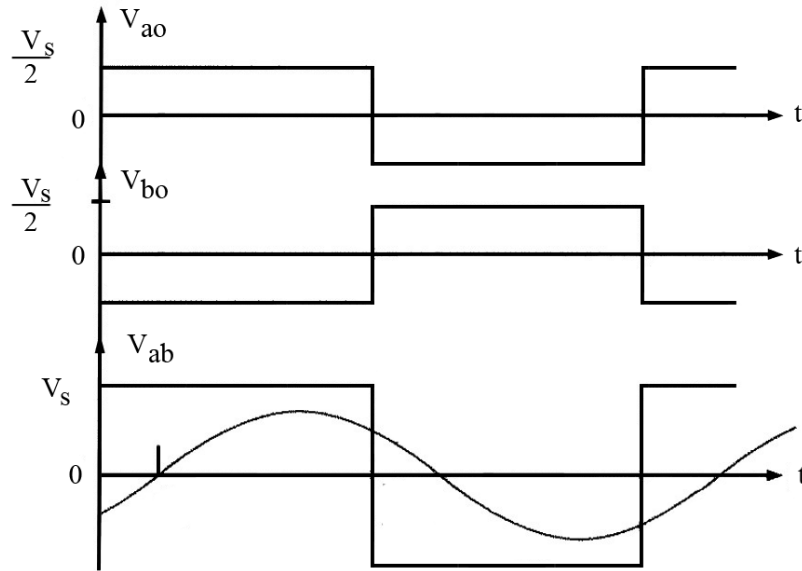
Onde $\omega = 2\pi f$ é a frequência da tensão de saída em rad/s .

Figura 3 – Inversor monofásico de meia ponte



Fonte:(10)

Figura 4 – Saída do inversor



Fonte:(10)

Para a topologia do inversor de meia ponte, a tensão eficaz de saída é expressa em (4).

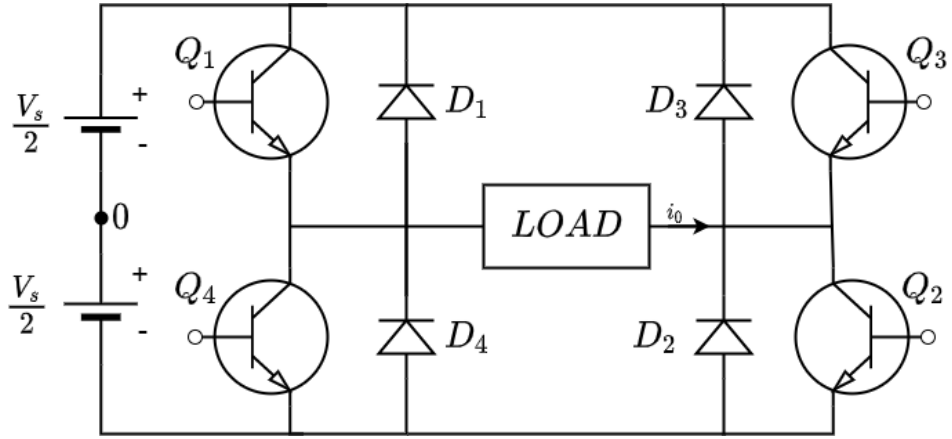
$$V_0 = \frac{V_s}{2} \quad (4)$$

Em (5) é mostrado a tensão instantânea da componente fundamental, $n = 1$.

$$V_1 = \frac{2V_s}{\sqrt{2\pi}} = 0.45V_s \quad (5)$$

A figura 5 mostra a topologia do inversor de ponte completa, esse conversor possui 4 chaves elétricas, aqui representadas por 4 transistores e 4 diodos. Em relação ao inversor de meia ponte, esta topologia funciona de forma parecida, as chaves dos braços funcionam de maneira complementar e da mesma forma, existe uma inversão na polaridade da tensão de saída de um estágio para o outro. No entanto, as chaves Q_1 e Q_2 conduzem durante o primeiro estágio e durante o segundo Q_3 e Q_4 conduzem simultaneamente. Isso faz com que a carga seja exposta à tensão V_s e $-V_s$ ao invés de apenas metade da tensão. O que ocorre é que a carga não está mais conectada ao ponto de neutro, então para cada ciclo de condução a carga é submetida a toda a tensão de alimentação. As equações (2) e (3) podem ser aplicadas para obter as tensões de saída, assim segue:

Figura 5 – Inversor onda completa



Fonte:(10)

A tensão eficaz de saída está expressa em (6).

$$V_{RMS} = \sqrt{\frac{2}{T_0} \cdot \int_0^{\frac{T_0}{2}} V_s^2 dt} = V_s \quad (6)$$

A tensão instantânea de saída está expressa em (7).

$$v_0 = \sum_{n=1,3,5,\dots}^{\infty} \frac{4V_s}{n\pi} \cdot \sin(n\omega t) \quad (7)$$

A tensão instantânea de saída da fundamental, $n = 1$.

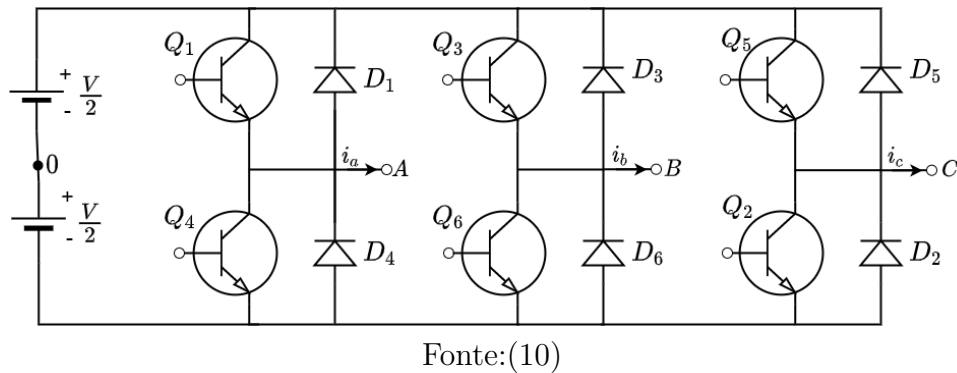
$$V_1 = \frac{4V_s}{\sqrt{2\pi}} = 0.90V_s \quad (8)$$

As topologias apresentadas, de meia ponte e ponte completa são a base do estudo destes tipos de conversores. O princípio de operação e as equações de análise de performance, podem ser aplicadas em diversas topologias deste conversor. Nas análises de tensão de saída, pôde-se constatar que as tensões de saída do inversor de ponte completa são muito superiores, no entanto, estes conversores são mais caros em função do maior número de chaves elétricas e por este mesmo motivo apresentam mais perdas por condução e chaveamento em relação ao de meia ponte. Para aplicações utilizando maior potência, são utilizados inversores trifásicos, neste trabalho a aplicação de inversores foi feita utilizando esta topologia, no tópico a seguir serão feitas análises acerca do funcionamento desta topologia de conversores.

2.1.2 Inversores Trifásicos.

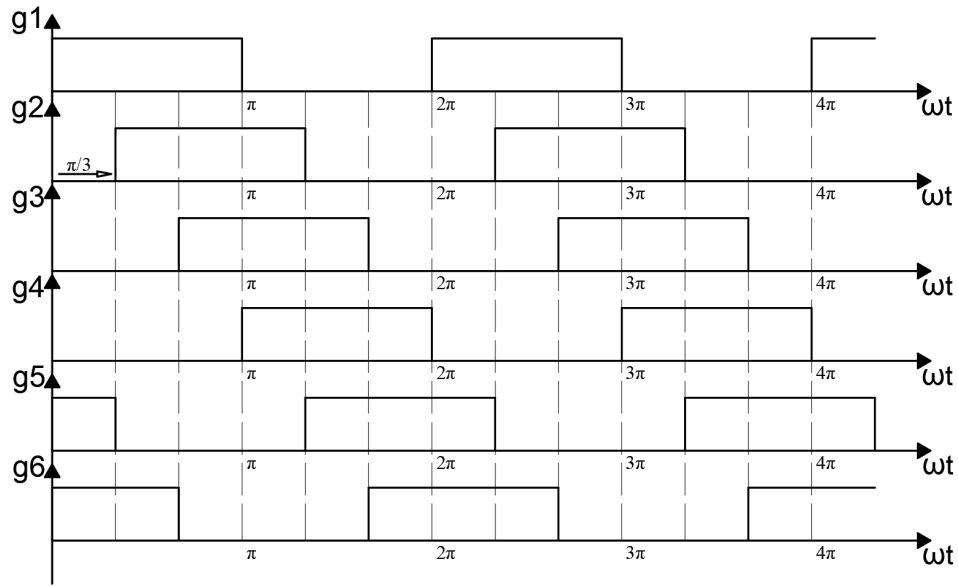
Os inversores trifásicos de dois níveis, geralmente são montados como 3 inversores de meia ponte conectados em paralelo como mostrado na figura 6. Esta topologia é composta de 6 chaves elétricas distribuídas em pares pelos 3 braços do inversor. As chaves Q_1 , Q_3 e Q_5 , são chamadas *High Side Switches* ou chaves superiores e as chaves Q_4 , Q_6 e Q_2 são chamadas *Low Side Switches* ou chaves inferiores. As chaves de cada um dos braços do inversor operam de forma complementar, ou seja, se uma chave está ligada a outra deve estar desligada afim de evitar um curto circuito. Cada um dos braços deve receber um sinal de controle defasado de 120° de modo a criar na saída uma tensão trifásica balanceada.

Figura 6 – Topologia inversor trifásico de dois níveis



Estes conversores possuem dois tipos de condução, condução por 180° e condução por 120° , neste trabalho será abordado apenas o método de condução por 180° . Neste tipo de condução, cada transistor conduz por um período de 180° e três transistores irão conduzir em cada instante de tempo. Na figura 7 estão ilustrados os sinais de comando dos transistores, estes sinais geram uma sequência de comutação, esta sequência geralmente é mostrada como um número de três dígitos, onde a ordem destes números representa a sequência de condução dos transistores. Por exemplo, a sequência "123", significa que o transistor 1 está conduzindo e depois de 60° o transistor 2 também entra em condução e nos 60° seguintes, entra o transistor 3, ou seja a cada 60° o esquema de condução é modificado. Cada transistor conduz 180° por período, o que implica em um *Duty Cycle* de 50%.

Figura 7 – Sinais de comando das chaves elétricas para um inversor trifásico



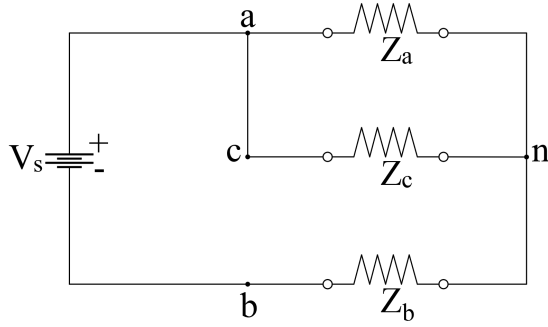
Fonte:(10)

Ao analisar os sinais da figura 7, é possível perceber que a cada 60° a sequência de comutação é modificada. A sequência de comutação para um período de 360° , 5-6-1, 6-1-2, 1-2-3, 2-3-4, 3-4-5, 4-5-6. Uma notação alternativa para representar o estado das chaves é no formato de um vetor binário de 3 posições, onde "1" significa que a chave está conduzindo e "0" significa que a chave não está conduzindo. Estes vetores costumam tomar como referência apenas o conjunto dos *High Side Switches* ou chaves superiores. Desta forma, pela natureza complementar da condução, é possível saber qual é o estado de cada uma das 6 chaves apenas olhando as três chaves superiores ou inferiores. As fases são identificadas observando as posições do vetor, sendo que a primeira representa o estado da chave da fase *a*, a segunda da fase *b* e a terceira da fase *c*. Assim, nesta notação, a configuração representada pelos números 561 é escrita como (1,0,1). A seguir, serão feitas as análises das tensões de saída para cada uma destas configurações, considerando uma carga trifásica balanceada conectada em estrela na saída do conversor.

2.1.3 Análise dos esquemas de condução.

Tomando $Z_a = Z_b = Z_c = Z$, em função de se tratar de uma carga trifásica equilibrada, seguem as análises das configurações de condução para cada estado de chaveamento.

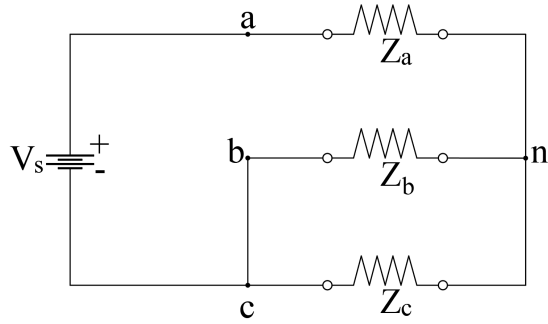
Figura 8 – Configuração de condução, 5-6-1 ou (1,0,1)



Fonte:(10)

$$\begin{aligned} V_{an} &= \frac{Z/2}{Z + Z/2} \cdot V_s = \frac{1}{3} V_s \\ V_{bn} &= \frac{Z}{Z + Z/2} \cdot -V_s = -\frac{2}{3} V_s \\ V_{cn} &= \frac{Z/2}{Z + Z/2} \cdot V_s = \frac{1}{3} V_s \\ V_{ab} &= V_s \quad V_{bc} = -V_s \quad V_{ca} = 0 \end{aligned}$$

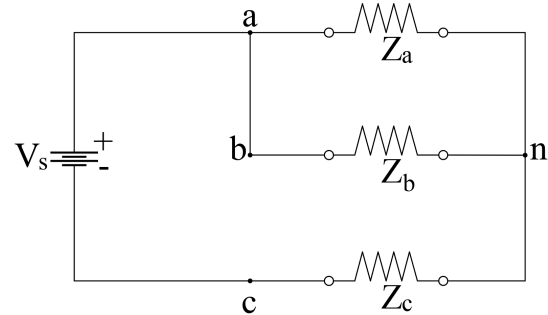
Figura 9 – Configuração de condução, 6-2-1 - (1,0,0)



Fonte:(10)

$$\begin{aligned} V_{an} &= \frac{Z}{Z + Z/2} \cdot V_s = \frac{2}{3} V_s \\ V_{bn} &= \frac{Z/2}{Z + Z/2} \cdot -V_s = -\frac{1}{3} V_s \\ V_{cn} &= \frac{Z/2}{Z + Z/2} \cdot -V_s = -\frac{1}{3} V_s \\ V_{ab} &= V_s \quad V_{bc} = 0 \quad V_{ca} = -V_s \end{aligned}$$

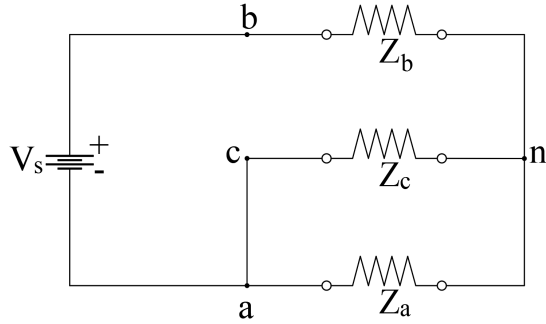
Figura 10 – Configuração de condução, 1-2-3 - (1,1,0)



Fonte:(10)

$$\begin{aligned} V_{an} &= \frac{Z/2}{Z + Z/2} \cdot V_s = \frac{1}{3} V_s \\ V_{bn} &= \frac{Z/2}{Z + Z/2} \cdot V_s = \frac{1}{3} V_s \\ V_{cn} &= \frac{Z}{Z + Z/2} \cdot -V_s = -\frac{2}{3} V_s \\ V_{ab} &= 0 \quad V_{bc} = V_s \quad V_{ca} = -V_s \end{aligned}$$

Figura 11 – Configuração de condução,
2-3-4 - (0,1,0)



Fonte:(10)

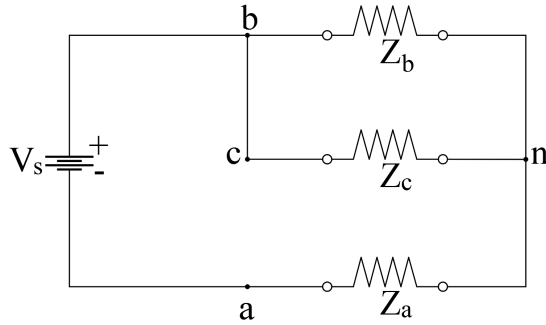
$$V_{an} = \frac{Z/2}{Z + Z/2} \cdot -V_s = -\frac{1}{3}V_s$$

$$V_{bn} = \frac{Z}{Z + Z/2} \cdot V_s = \frac{2}{3}V_s$$

$$V_{cn} = \frac{Z/2}{Z + Z/2} \cdot -V_s = -\frac{1}{3}V_s$$

$$V_{ab} = -V_s \quad V_{bc} = V_s \quad V_{ca} = 0$$

Figura 12 – Configuração de condução,
3-4-5 - (0,1,1)



Fonte:(10)

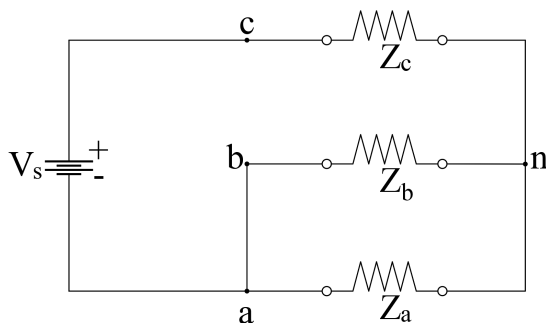
$$V_{an} = \frac{Z}{Z + Z/2} \cdot -V_s = -\frac{2}{3}V_s$$

$$V_{bn} = \frac{Z/2}{Z + Z/2} \cdot V_s = \frac{1}{3}V_s$$

$$V_{cn} = \frac{Z/2}{Z + Z/2} \cdot V_s = \frac{1}{3}V_s$$

$$V_{ab} = -V_s \quad V_{bc} = 0 \quad V_{ca} = V_s$$

Figura 13 – Configuração de condução,
4-5-6 - (0,0,1)



Fonte:(10)

$$V_{an} = \frac{Z/2}{Z + Z/2} \cdot -V_s = -\frac{1}{3}V_s$$

$$V_{bn} = \frac{Z/2}{Z + Z/2} \cdot V_s = -\frac{1}{3}V_s$$

$$V_{cn} = \frac{Z}{Z + Z/2} \cdot -V_s = -\frac{2}{3}V_s$$

$$V_{ab} = 0 \quad V_{bc} = -V_s \quad V_{ca} = V_s$$

Com as análises de cada um dos estados de chaveamento feitas acima, é possível montar uma tabela com as tensões de fase e linha de saída do inversor trifásico de dois níveis.

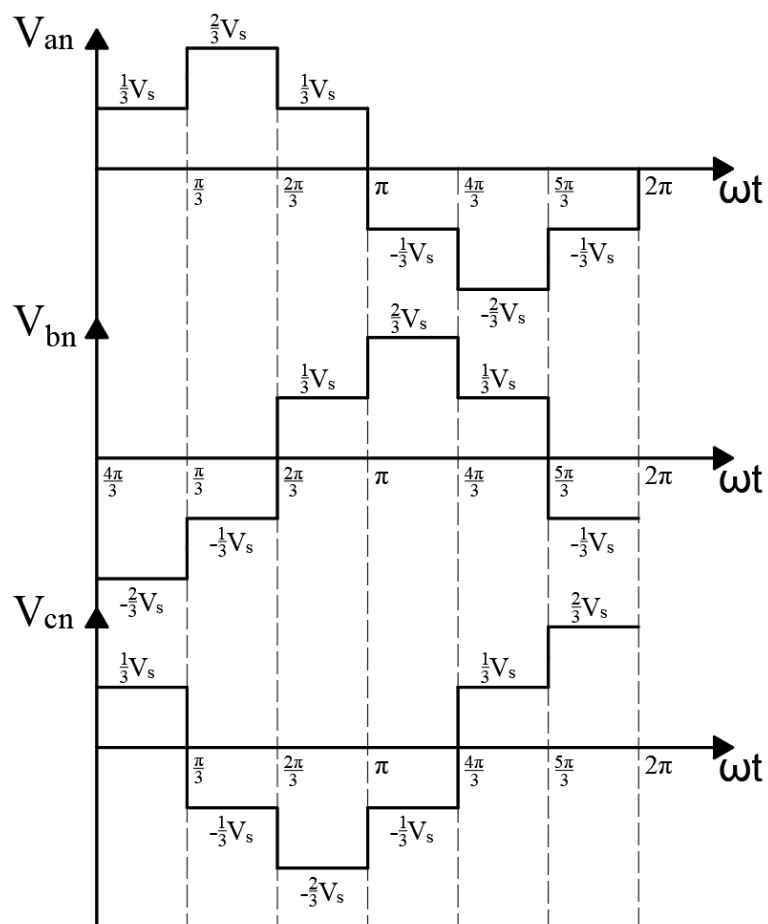
Tabela 1 – Tensões de fase e linha para cada esquema de condução

Estados de Chaveamento	Código Chaves Superiores	V_{an}	V_{bn}	V_{cn}	V_{ab}	V_{bc}	V_{ca}
5-6-1	(1,0,1)	$\frac{1}{3}V_s$	$-\frac{2}{3}V_s$	$\frac{1}{3}V_s$	$+V_s$	$-V_s$	0
6-1-2	(1,0,0)	$\frac{2}{3}V_s$	$-\frac{1}{3}V_s$	$-\frac{1}{3}V_s$	$+V_s$	0	$-V_s$
1-2-3	(1,1,0)	$\frac{1}{3}V_s$	$\frac{1}{3}V_s$	$-\frac{2}{3}V_s$	0	$+V_s$	$-V_s$
2-3-4	(0,1,0)	$-\frac{1}{3}V_s$	$\frac{2}{3}V_s$	$-\frac{1}{3}V_s$	$-V_s$	$+V_s$	0
3-4-5	(0,1,1)	$-\frac{2}{3}V_s$	$\frac{1}{3}V_s$	$\frac{1}{3}V_s$	$-V_s$	0	$+V_s$
4-5-6	(0,0,1)	$-\frac{1}{3}V_s$	$-\frac{1}{3}V_s$	$\frac{2}{3}V_s$	0	$-V_s$	$+V_s$
4-6-2	(0,0,0)	0	0	0	0	0	0
1-3-5	(1,1,1)	0	0	0	0	0	0

Fonte:(10)

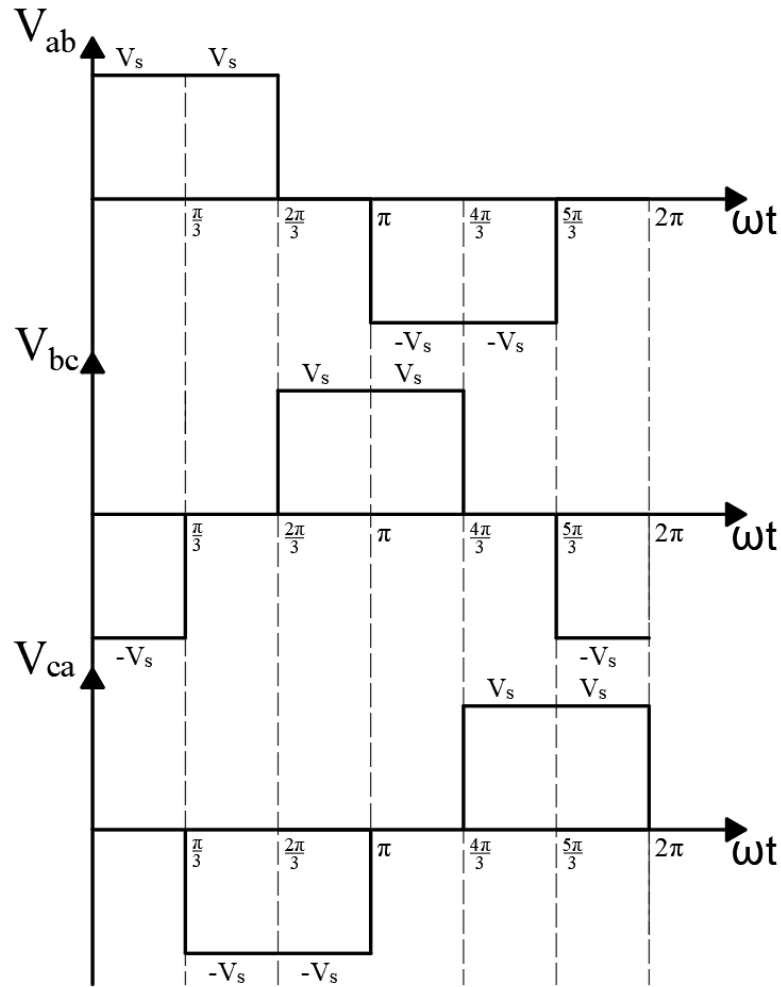
A partir dos resultados da tabela 1, é possível construir o gráfico das tensões de saída, de fase e de linha do inversor trifásico de dois níveis.

Figura 14 – Gráfico das tensões de fase.



Fonte:(10)

Figura 15 – Gráfico das tensões de fase.



Fonte:(10)

A partir das curvas acima, é possível calcular o valor RMS das tensões de fase e de linha.

$$V_F = \sqrt{\frac{2}{2\pi} \cdot \left(\int_0^{\frac{\pi}{3}} \frac{V_s^2}{3} d\omega t + \int_{\frac{\pi}{3}}^{\frac{2\pi}{3}} \frac{2V_s^2}{3} d\omega t + \int_{\frac{2\pi}{3}}^{\pi} \frac{V_s^2}{3} d\omega t \right)} = \frac{\sqrt{2}}{3} V_s \quad (9)$$

$$V_L = \sqrt{\frac{2}{2\pi} \int_0^{2\pi} \frac{2\pi}{3} V_s^2 d\omega t} = \sqrt{\frac{2}{3}} V_s \quad (10)$$

Para mostrar que o resultado trata de um sistema trifásico equilibrado, basta fazer o quociente entre a tensão de linha e a tensão de fase obtidos.

$$\frac{V_L}{V_F} = \frac{\sqrt{\frac{2}{3}}V_s}{\frac{\sqrt{2}}{3}V_s} = \sqrt{3} \implies V_L = \sqrt{3}V_F \quad (11)$$

O resultado obtido em (11) implica em uma carga trifásica balanceada conectada em estrela, que é a justamente a condição de carga indicada no início das análises.

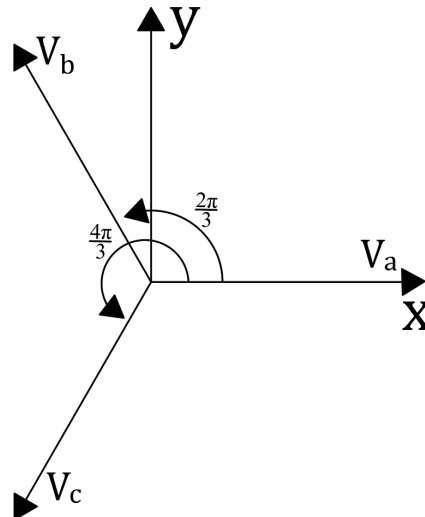
2.2 Representação por vetores espaciais.

A representação por vetores espaciais tem como objetivo representar o sistema trifásico em um plano como função de duas coordenadas. Este processo é feito por meio de projeções. Cada componente de cada uma das fases a , b e c são projetadas sobre x e y . As componentes de um sistema trifásico estão representadas pelas equações abaixo.

$$\begin{aligned} V_a &= V_m \text{sen}(\omega_e t) \\ V_b &= V_m \text{sen}(\omega_e t - \frac{2\pi}{3}) \\ V_c &= V_m \text{sen}(\omega_e t - \frac{4\pi}{3}) \end{aligned} \quad (12)$$

Onde o termo V_m é a máxima amplitude das componentes de cada uma das fases. Assim, para realizar a projeção de cada uma das fases nos eixos x e y será necessário uma representação gráfica, como segue na figura.

Figura 16 – Gráfico das Tensões de fase.



Fonte:(10)

O objetivo é escrever V_a , V_b e V_c em termos de x e y , desta forma segue:

$$\begin{aligned}x &= V_a \cos(0) + V_b \cos\left(\frac{2\pi}{3}\right) + V_c \cos\left(\frac{4\pi}{3}\right) \\y &= V_a \sin(0) + V_b \sin\left(\frac{2\pi}{3}\right) + V_c \sin\left(\frac{4\pi}{3}\right)\end{aligned}\quad (13)$$

Para a representação por vetores espaciais de sinais trifásicos é mais indicado que se utilize a base dos números complexos. Isto se deve ao fato de que um número complexo carrega a informação de módulo e posição angular de forma natural. A próxima equação mostra a aplicação desta base numérica ao conjunto de equações (13).

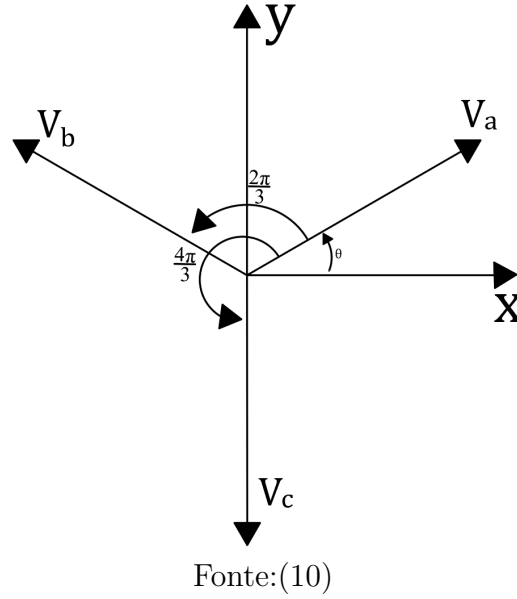
$$\begin{aligned}z &= x + jy \\z &= V_a (\cos(0) + j\sin(0)) + V_b \left(\cos\left(\frac{2\pi}{3}\right) + j\sin\left(\frac{2\pi}{3}\right)\right) + \\&+ V_c \left(\cos\left(\frac{4\pi}{3}\right) + j\sin\left(\frac{4\pi}{3}\right)\right)\end{aligned}\quad (14)$$

Aplicando a equação de Euler $e^{j\theta} = \cos(\theta) + j\sin(\theta)$ pode-se escrever a equação 14 na sua forma trigonométrica, como segue:

$$z = V_a + V_b e^{j\frac{2\pi}{3}} + V_c e^{j\frac{4\pi}{3}} \quad (15)$$

O conjunto de equações (12) mostrou o termo $\omega_e t$ no argumento das funções senoidais, isso significa que estas grandezas variam no tempo, como tensões, correntes e também fluxos. Pode ocorrer que o sistema trifásico esteja defasado no espaço, nestes casos o defasamento no tempo dos sinais continuaria o mesmo, sendo modificados apenas por um defasamento angular θ somado ao defasamento no tempo. Segue um exemplo gráfico apresentado na figura 17 e também as equações resultantes.

Figura 17 – Gráfico das tensões de fase deslocado.



No caso de um sistema com um defasamento espacial, utiliza-se a substituição de variáveis para modelar as equações que o definem.

$$z = V_a e^{j(\theta)} + V_b e^{j(\theta + \frac{2\pi}{3})} + V_c e^{j(\theta + \frac{4\pi}{3})} \quad (16)$$

Assim, pode-se definir um sistema em vetores espaciais com componentes variantes no tempo, da seguinte forma:

$$\vec{V}_r = e^{j\theta} \left[V_a(t) + V_b(t)e^{j\frac{2\pi}{3}} + V_c(t)e^{j\frac{4\pi}{3}} \right] \quad (17)$$

Fica explicitado no argumento a dependência temporal das variáveis que compõe o vetor espacial, desta forma para cada período haverá um vetor espacial. É comum na literatura que o termo $e^{j\frac{2\pi}{3}}$ seja representado por $\vec{a}$, assim pode-se escrever a equação 17 no seguinte formato.

$$\vec{V}_r = e^{j\theta} \left[V_a(t) + V_b(t)\vec{a} + V_c(t)\vec{a}^2 \right] \quad (18)$$

Neste trabalho o desenvolvimento do vetor espacial será feito para o caso particular da figura (16), onde $\theta = 0$. Para adequar o módulo do vetor espacial ao módulo das componentes, é multiplicado uma constante de invariância de amplitude, como mostrado na equação que segue(10).

$$\vec{V}_r = \frac{2}{3} \left[V_a(t) + V_b(t)\vec{a} + V_c(t)\vec{a}^2 \right] \quad (19)$$

2.2.1 Transformada de Clarke

Mudanças de base são muito comuns na engenharia. Permitem representar vetores em outros sistemas de referência de forma a simplificar as análises destas grandezas. A transformada de Clarke ou $\alpha\beta$ permite representar o sistema trifásico em apenas duas componentes, uma real α e a outra imagiária β . Desta forma, o vetor espacial definido na equação (19) que dependia dos valores das três componentes trifásicas, pode ser representado neste novo sistema de coordenadas dependendo apenas das componentes α e β . A transformada de Clarke é feita por meio de uma matriz que está expressa abaixo (11).

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \frac{2}{3} \cdot \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \cdot \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \quad (20)$$

A transformada permite representar tensões, correntes e fluxos trifásicos em termos de α e β . Na tabela 2 serão mostrados os valores de tensão de saída do inversor trifásico de dois níveis da tabela 1, em termos de α e β . Estes valores serão mostrados em uma tabela onde cada configuração de condução do inversor será chamado de um vetor que vai de V_0 até V_7 . Na tabela 1 foram mostrados apenas 6 vetores dos 8 que o inversor é capaz de produzir, isto se deve ao fato de que os vetores $V_0(000)$ e $V_7(111)$ são vetores nulos, ou seja, a tensão de saída no inversor para essa configuração de chaves é zero. Tem-se como notação padrão o código dos *High Side Switches*, aqui chamados de vetores básicos. Desta forma, segue:

Tabela 2 – Tensões de saída do inversor em α e β

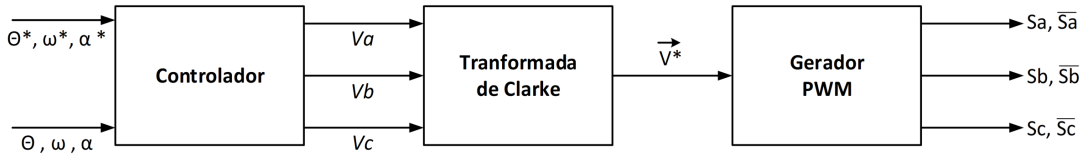
V	V_α	V_β	$Ae^{j\theta}$	S_a	S_b	S_c
V_0	0	0	0	0	0	0
V_1	$\frac{2}{3}V_s$	0	$\frac{2}{3}V_se^{j0^\circ}$	1	0	0
V_2	$\frac{1}{3}V_s$	$\frac{1}{\sqrt{3}}V_s$	$\frac{2}{3}V_se^{j60^\circ}$	1	1	0
V_3	$-\frac{1}{3}V_s$	$\frac{1}{\sqrt{3}}V_s$	$\frac{2}{3}V_se^{j120^\circ}$	0	1	0
V_4	$-\frac{2}{3}V_s$	0	$\frac{2}{3}V_se^{j180^\circ}$	0	1	1
V_5	$-\frac{1}{3}V_s$	$-\frac{1}{\sqrt{3}}V_s$	$\frac{2}{3}V_se^{j240^\circ}$	0	0	1
V_6	$\frac{1}{3}V_s$	$-\frac{1}{\sqrt{3}}V_s$	$\frac{2}{3}V_se^{j300^\circ}$	1	0	1
V_7	0	0	0	1	1	1

Fonte:(11)

2.3 Modulação Por Vetores Espaciais.

A modulação tem o objetivo de determinar o estado das chaves elétricas de um inversor para que este seja capaz de sintetizar em seus terminais a tensão demandada por um controlador. A figura 18 ilustra o funcionamento deste sistema, são inseridos valores de referência de velocidade e posição que são comparados com os valores medidos, a comparação é feita dentro do controlador que para compensar este erro realiza uma ação de controle que se manifesta como uma demanda de tensão. A tensão demandada pelo controlador passa pela transformada de Clarke e o vetor de referência é aplicado no algoritmo de modulação, que por sua vez determina qual a configuração e por quanto tempo o inversor deve mantê-la para obter em seus terminais a tensão solicitada pelo controle.

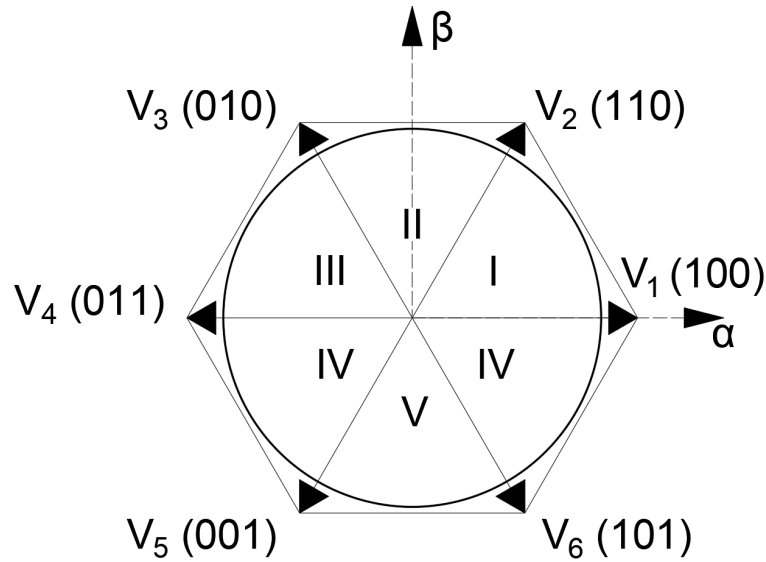
Figura 18 – Esquema de controle e modulação.



Fonte:(11)

Os vetores básicos mostrados na tabela 2, podem ser arranjados no plano $\alpha\beta$ de modo a formar o hexágono regular (10).

Figura 19 – Hexágono Regular.



Fonte:(11)

Um vetor espacial de referência V_r que representa as tensões trifásicas do inversor, pode ser escrito como uma combinação dos vetores de referência. O inversor não é capaz de sintetizar vetores além dos 8 vetores básicos, mas ele é capaz de controlar o tempo em que eles ficam ativos, desta forma pode-se combina-los em uma soma de múltiplos dos vetores básicos por períodos em que ficam ativos. O objetivo do SVPWM é utilizar este conceito para obter o vetor de referência em qualquer posição solicitada pelo controlador. As seguintes equações mostram como este processo é executado (12).

$$V_r * t_{pwm} = V_a t_a + V_b t_b \quad (21)$$

O valores de t_a e t_b são os tempos em que as configurações de condução que geram os vetores V_a e V_b ficam ativos. O t_{pwm} é o periodo total que esses padrões de comutação

ficam ativos, no entanto os valores de t_a e t_b não são iguais ao período de chaveamento em todos as configurações.

$$t_{pwm} \geq t_a + t_b \quad (22)$$

Para diminuir o conteúdo harmônico o vetores nulos são ativos por um tempo chamado de t_0 para que o período seja sempre constante, assim segue que:

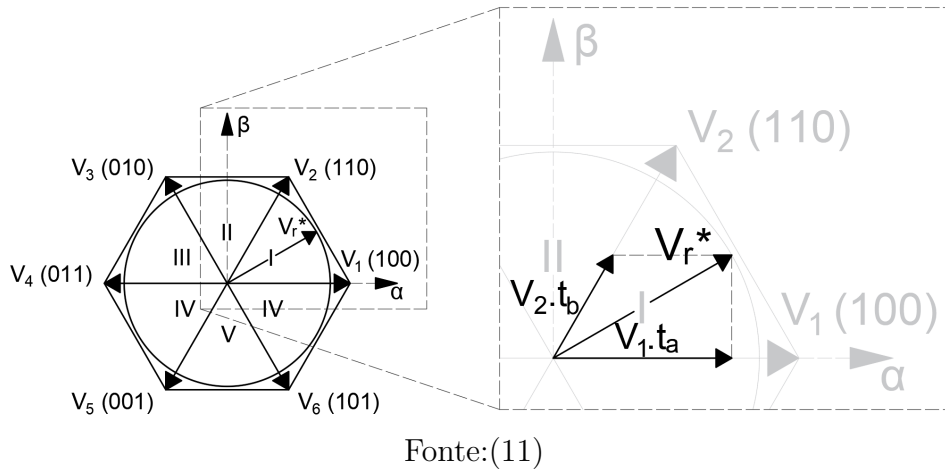
$$t_{pwm} = t_a + t_b + t_0 \quad (23)$$

A partir da equação (23) pode-se reescrever a equação (21) utilizando os vetores nulos e o tempo de vetor nulo, t_0 . Sendo t_0 o tempo em que os vetores nulos V_0 e/ou V_7 são utilizados.

$$V_r * t_{pwm} = V_a t_a + V_b t_b + V_{nulo} t_0 \quad (24)$$

A figura 20 mostra um representação gráfica da combinação de vetores que foi tratada acima.

Figura 20 – Combinação de vetores.



Para encontrar os valores dos tempos, basta utilizar a seguinte matriz (11).

$$\begin{bmatrix} t_a \\ t_b \end{bmatrix} = t_{pwm} \cdot \begin{bmatrix} V_{\alpha 1} & V_{\beta 1} \\ V_{\alpha 2} & V_{\beta 2} \end{bmatrix}^{-1} \cdot \begin{bmatrix} V_{\alpha} \\ V_{\beta} \end{bmatrix} \quad (25)$$

Onde $V_{\alpha 1}$, V_{β} , $V_{\alpha 2}$ e V_{β} , são as componentes α e β dos vetores básicos que delimitam o setor no qual o vetor de referência será sintetizado. Fazendo a substituição dos valores da tabela 2 para cada vetor básico, obten-se os resultados da tabela 3.

Tabela 3 – Tabela de cálculo dos tempos t_a e t_b .

Setor	t_a	t_b
I	$t_{pwm} \frac{3V_\alpha - \sqrt{3}V_\beta}{2V_s}$	$t_{pwm} \frac{\sqrt{3}V_\beta}{V_s}$
II	$t_{pwm} \frac{3V_\alpha + \sqrt{3}V_\beta}{2V_s}$	$-t_{pwm} \frac{3V_\alpha - \sqrt{3}V_\beta}{2V_s}$
III	$t_{pwm} \frac{\sqrt{3}V_\beta}{V_s}$	$-t_{pwm} \frac{3V_\alpha + \sqrt{3}V_\beta}{2V_s}$
IV	$-t_{pwm} \frac{3V_\alpha - \sqrt{3}V_\beta}{2V_s}$	$-t_{pwm} \frac{\sqrt{3}V_\beta}{V_s}$
V	$-t_{pwm} \frac{3V_\alpha + \sqrt{3}V_\beta}{2V_s}$	$t_{pwm} \frac{3V_\alpha - \sqrt{3}V_\beta}{2V_s}$
VI	$-t_{pwm} \frac{\sqrt{3}V_\beta}{V_s}$	$t_{pwm} \frac{3V_\alpha + \sqrt{3}V_\beta}{2V_s}$

Fonte:(11)

Para que as ondas tenham simetria de modo a reduzir a distorção harmônica e reduzir o número de chaveamentos por período de modulação, t_a e t_b são escritas em termos de t_x e t_y , aplica-se a lógica do seguinte conjunto de equações (11).

$$\begin{cases} m = 1,3,5 \implies t_x = t_a; & t_y = t_b \\ m = 2,4,6 \implies t_x = t_b; & t_y = t_a \end{cases} \quad (26)$$

O mesmo é feito para os vetores básicos, são dispostos em termos de V_x e V_y de acordo com o seguinte conjunto de equações (11).

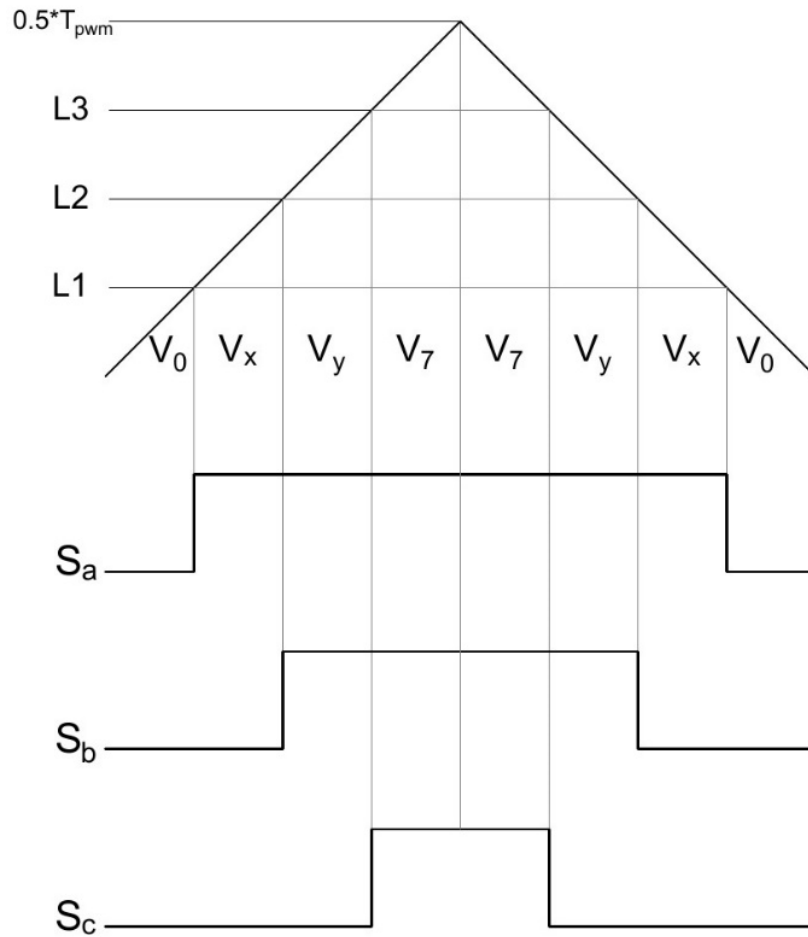
$$\begin{cases} m = 1,3,5 \implies V_x = V_a; & V_y = V_b \\ m = 2,4,6 \implies V_x = V_2; & V_y = V_1 \end{cases} \quad (27)$$

Com os vetores V_x e V_y definidos e com os tempos t_x e t_y especificados, é possível definir uma ordem de comutação que evite ao máximo as perdas por chaveamento, este seria o caso onde apenas um dos braços do inversor comuta para cada vetor básico ativado. Os vetores nulos V_0 e V_7 são utilizados neste ponto para garantir que as transições de vetores básicos aconteçam com apenas uma comutação. Para obter a ordem de comutação adequada as especificações, são criados três limites a partir dos tempos t_x e t_y , estes limites são definidos nas equações a seguir (11).

$$\begin{cases} L_1 = \frac{t_0}{4} \\ L_2 = L_1 + \frac{t_x}{2} \\ L_3 = L_2 + \frac{t_y}{2} \end{cases} \quad (28)$$

A partir dos resultados obtidos e dos limitantes definidos pelo conjunto de equações 28, pode-se definir a ordem de chaveamento a partir da comparação entre os limites e um portadora triangular, onde cada limite irá definir um vetor a ser ativo pelo inversor trifásico de dois níveis. Esta comparação está ilustrada na figura a seguir.

Figura 21 – Ordem de chaveamento.



Fonte:(11)

Para que seja possível performar os cálculos acima, é necessário conhecer o setor onde se localiza o vetor de referência para cada instante de tempo. Este procedimento é feito a partir de comparações entre múltiplos das componentes α e β . Como mostrado na tabela 4:

Tabela 4 – Seleção do setor de trabalho.

Setor	Angulo do vetor	Condicionais V_α e V_β
I	$[0^\circ, 60^\circ)$	$0 \leq V_\beta < \sqrt{3}V_\alpha$
II	$[60^\circ, 120^\circ)$	$0 \leq \sqrt{3}V_\alpha \leq V_\beta$ ou, $0 < \sqrt{3}V_\alpha < V_\beta$
III	$[120^\circ, 180^\circ)$	$0 < V_\beta \leq -\sqrt{3}V_\alpha$
IV	$[180^\circ, 240^\circ)$	$\sqrt{3}V_\alpha < V_\beta \leq 0$
V	$[240^\circ, 300^\circ)$	$V_\beta \leq \sqrt{3}V_\alpha \leq 0$ ou, $V_\beta < -\sqrt{3}V_\alpha < 0$
VI	$[300^\circ, 360^\circ)$	$-\sqrt{3}V_\alpha \leq V_\beta < 0$

Fonte:(13)

2.4 Representação em ponto fixo.

Em sistemas digitais, existem principalmente duas técnicas de representação de números decimais: representação em ponto flutuante e representação em ponto fixo. A representação em ponto flutuante permite representar números mais grandes ou mais pequenas em comparação à representação em ponto fixo. Porém, o custo computacional das operações aritméticas com a representação de ponto flutuante é maior. Por isso, neste trabalho utiliza-se a representação em ponto fixo, a qual é uma extensão simples da técnica de representação dos números inteiros. Na representação em ponto fixo, reserva-se N bits para a parte inteira e M bits para representar a parte fracionária. A quantidade total de bits utilizados para representar um número é $W = N + M$. Em muitas aplicações, $W = 16, 24, 32$ ou 64 bits. Sejam $b_{N-1}, b_{N-2}, \dots, b_1, b_0, b_{-1}, b_{-2}, \dots, b_{-M}$ os bits utilizados para representar o número (observar que são W bits), sendo b_{N-1} o bit mais significativo. O valor do número X representado através destes bits é:

$$X = \sum_{j=-M}^{N-1} b_j 2^j \quad (29)$$

A representação em ponto fixo pode ser sem sinal (representar números só positivos) ou com sinal (representar números positivos e negativos). Neste trabalho usou-se a representação com sinal. A técnica de complemento a 2 usa-se para obter o negativo de um

número. O valor de N depende do número inteiro mais grande que deve ser representado, enquanto que o valor de M depende do valor decimal mais pequeno que deve ser representado (resolução numérica, que é igual a 2^{-M}).

3 METODOLOGIA

3.1 Geração das tensões de referência em VHDL.

As referências senoidais foram embarcadas utilizando uma *look-up table* (LUT). Um LUT é uma memória que armazena valores específicos de uma função para serem utilizadas em algum cálculo. A memória ROM do FPGA foi utilizada para implementar LUTs que representam as referências senoidais. Tais referências foram geradas no MATLAB e aqueles dados foram colocados em um arquivo ".mif" (arquivo de configuração da memória ROM do FPGA). Este arquivo gerou LUTs através de uma ferramenta chamada *Mega Wizard* do software Quartus. As três senoides foram geradas com módulo unitário e defasadas de 120° . A partir das senoides geradas foram calculados os valores de V_α e V_β como mostrado no trecho de código mostrado na figura 22:

Figura 22 – Cálculo do setor.

```

230
231 if (cont = 0) then -- Definindo amplitude em 90% do valor total para evitar sobre-modulação
232
233     vas1      <= resize (90*vas,vas1);
234     vbs1      <= resize (90*vbs,vbs1);
235     vcs1      <= resize (90*vcs,vcs1);
236
237 elsif (cont = 1) then --calcular valfa vbeta
238
239     valphas <= resize(((vas1+vas1)-(vbs1+vcs1))*c_inv3,valphas);
240     vbetas  <= resize((vbs1 - vcs1)*c_inv_r3,vbetas);
241
242
243
244
245
246 elsif (cont = 2) then
247
248 -- variaveis auxiliares com valores das constantes
249 valpha1 <= resize((c3*valphas), valpha1);
250 valpha2 <= resize((c4*valphas), valpha2);
251
252
253

```

Fonte:[Autor]

O módulo das senoides foi ajustado para 90 de modo a obter um fator de modulação igual a 90% e evitar sobremodulação.

3.2 Cálculo do setor em VHDL.

Com as referência em α e β definidas e convertidas para ponto-fixa, basta implementar a tabela 3 para realizar o cálculo do setor. A variável $Valpha1 = \sqrt{3}V_\alpha$ e a variável $Valpha2 = -\sqrt{3}V_\alpha$. Desta forma, segue o código.

Figura 23 – Cálculo do setor.

```

255
256
257 elif (cont = 3) then
258
259     ----- 1 2 3
260
261     if (vbetas>=0 and valpha1>vbetas) then
262         setor <= "001";
263     end if;
264
265     if (valpha1>=0 and vbetas>=valpha1) then
266         setor <= "010";
267     end if;
268
269     if (valpha2>0 and vbetas>valpha2) then
270         setor <= "010";
271     end if;
272
273     if (vbetas>0 and valpha2>=vbetas) then
274         setor <= "011";
275     end if;
276
277     ----- 4 5 6
278
279
280
281     if (0>=vbetas and vbetas>valpha1) then
282         setor <= "100";
283     end if;
284
285     if (0>=valpha1 and valpha1>=vbetas) then
286         setor <= "101";
287     end if;
288
289     if (0>=valpha2 and valpha2>vbetas) then
290         setor <= "101";
291     end if;
292
293     if (0>vbetas and vbetas>=valpha2) then
294         setor <= "110";
295     end if;
296
297
298 elif (cont = 5) then
299     setor_s <= setor;
300
301

```

Fonte:[Autor]

A variável $setor_s$ armazena a informação do setor da referência e passa a ser utilizado como condicional para o cálculo dos tempos, como será visto a seguir.

3.3 Cálculo dos tempos.

Ao analisar a tabela 4, é possível perceber que t_a e t_b assumem os mesmos valores em diferentes setores, possibilitando que ambos sejam escritos em função de apenas um dos tempos. Assim, é possível tabular as equações dos tempos, tal como indicado na tabela 5.

Tabela 5 – Otimização do algoritmo.

Setor	t_a	t_b	t_x	t_y
I	t_{a1}	t_{a3}	t_{a1}	t_{a3}
II	t_{a2}	t_{a4}	t_{a4}	t_{a2}
III	t_{a3}	t_{a5}	t_{a3}	t_{a5}
IV	t_{a4}	t_{a6}	t_{a6}	t_{a4}
V	t_{a5}	t_{a1}	t_{a5}	t_{a1}
VI	t_{a6}	t_{a2}	t_{a2}	t_{a6}

Fonte:(11)

Basta apenas calcular os tempos indicados na tabela 5 para englobar todos os setores. Este processo reduz o custo computacional e também facilita revisão do código. Desta forma, segue a figura que mostra o algoritmo de cálculo dos tempos.

Figura 24 – Cálculo dos Tempos.

```

305  elsif (cont = 6) then
306
307  ----- cálculo dos tempos
308
309  ta1 <= resize ((c11*valphas+c12n*vbetas),ta1);
310  ta2 <= resize ((c11*valphas+c12*vbetas),ta2);
311  ta3 <= resize ((c13*vbetas),ta3);
312  ta4 <= resize ((c11n*valphas+c12*vbetas),ta4);
313  ta5 <= resize ((c11n*valphas+c12n*vbetas),ta5);
314  ta6 <= resize ((c13n*vbetas),ta6);
315
316
317  ----- txs, tys, vx e vy
318
319  elsif (cont = 7) then
320
321  if (setor = "001") then
322      txs <= ta1;
323      tys <= ta3;
324      vx <= V1;
325      vy <= V2;
326  end if;
327  if (setor = "010") then
328      txs <= ta4;
329      tys <= ta2;
330      vx <= V3;
331      vy <= V2;
332  end if;
333  if (setor = "011") then
334      txs <= ta3;
335      tys <= ta5;
336      vx <= V3;
337      vy <= V4;
338  end if;
339  if (setor = "100") then
340      txs <= ta6;
341      tys <= ta4;
342      vx <= V5;
343      vy <= V4;
344  end if;
345  if (setor = "101") then
346      txs <= ta5;
347      tys <= ta1;
348      vx <= V5;
349      vy <= V6;
350  end if;
351  if (setor = "110") then
352      txs <= ta2;
353      tys <= ta6;
354      vx <= V1;
355      vy <= V6;
356  end if;
357

```

Fonte:[Autor]

3.4 Geração do sinal *PWM*

Com as informações dos tempos t_x e t_y para cada setor, segundo a tabela 5, pode-se definir a sequência de vetores espaciais do inversor segundo a Figura 21. Foi criada uma onda triangular para definir a sequência de vetores espaciais do inversor (algoritmo indicado no apêndice A). Lembrando que devido a natureza complementar dos pares de chaves alocadas nos braços do inversor trifásico de dois níveis, ao definir as chaves superiores (*High Side Switches*) ou chaves inferiores (*Low Side Switches*) são os complementares. A figura 25 mostra o trecho de código onde foi gerado o sinal de *PWM*.

Figura 25 – Cálculo dos Tempos.

```

358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408

```

```

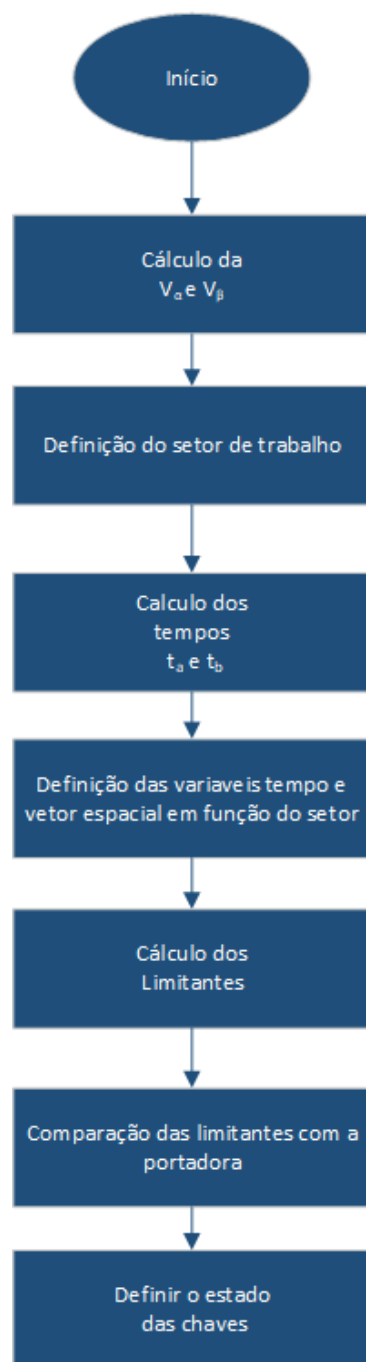
    elsif (cont=4999) then
        L0<=resize( (c_0p25*tpwm) + (c_m0p25*txs)+ (c_m0p25*ty), L0);
        L1<=resize( (c_0p25*tpwm) + (c_0p25*txs) + (c_m0p25*ty), L1);
        L2<=resize( (c_0p25*tpwm) + (c_0p25*txs) + (c_0p25*ty), L2);
    end if;
--
end if;
end process; -- contagem
--
process(cont, clk)
begin
    if (cont=0) then
        sa <='0';
        sb <='0';
        sc <='0';
    elsif (cont=4999) then
        sa <='0';
        sb <='0';
        sc <='0';
    else
        if (Tr < L0) then
            sa <='0';
            sb <='0';
            sc <='0';
        elsif (Tr < L1) then
            sa <=Vx(2);
            sb <=Vx(1);
            sc <=Vx(0);
        elsif (Tr < L2) then
            sa <=Vy(2);
            sb <=Vy(1);
            sc <=Vy(0);
        else
            sa <='1';
            sb <='1';
            sc <='1';
        end if;
    end if;
end process;
end architecture rtl;

```

Fonte:[Autor]

Em seguida, basta fazer o mapeamento das I/Os do FPGA para levar os sinais de cada uma das chaves para o inversor.

Figura 26 – Fluxograma do código apresentado.

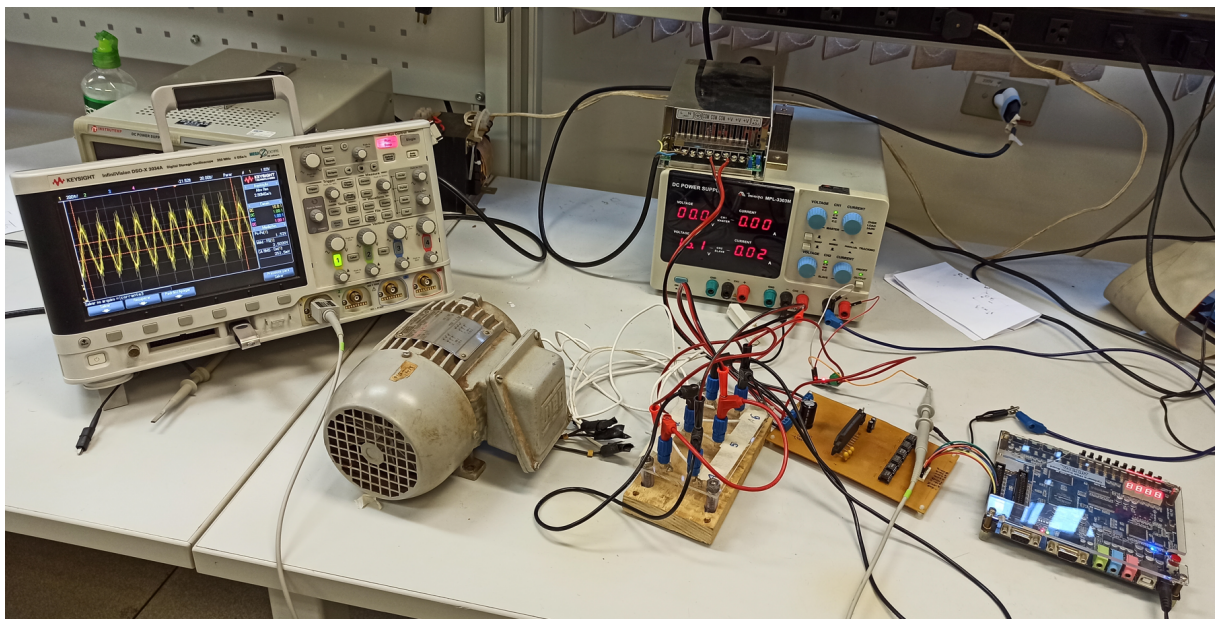


Fonte:[Autor]

4 RESULTADOS

Simulações foram efetuadas em MATLAB/SIMULINK. Adicionalmente, para obter os resultados experimentais, os sinais extraídos do FPGA foram alimentados no inversor IRAMX16UP60A e em seguida as saídas do inversor foram conectadas aos terminais de um motor de indução trifásico 0,25 CV; 60Hz; 1720 RPM; 220 V. A tensão do barramento CC é 100 V, e a frequência de chaveamento do inversor é 10 kHz. Uma das fases do inversor passou por um sensor de efeito Hall para que se pudesse obter a forma de onda da corrente no momento de acionamento da máquina. Abaixo a figura 27 mostra a bancada de testes.

Figura 27 – Bancada Experimental.

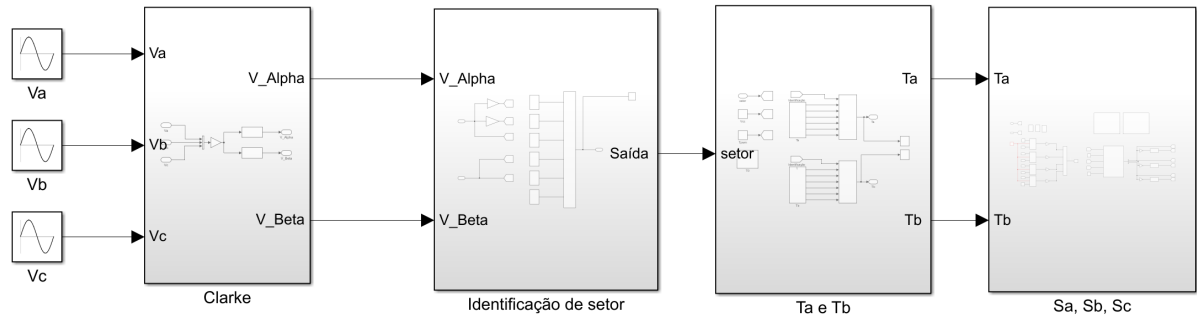


Fonte:[Autor]

4.0.1 Resultados da simulação.

Abaixo seguem os resultados obtidos na simulação via SIMULINK, mostrados nas figuras 28 até 33 juntamente com um *overview* da simulação mostrado na figura 28.

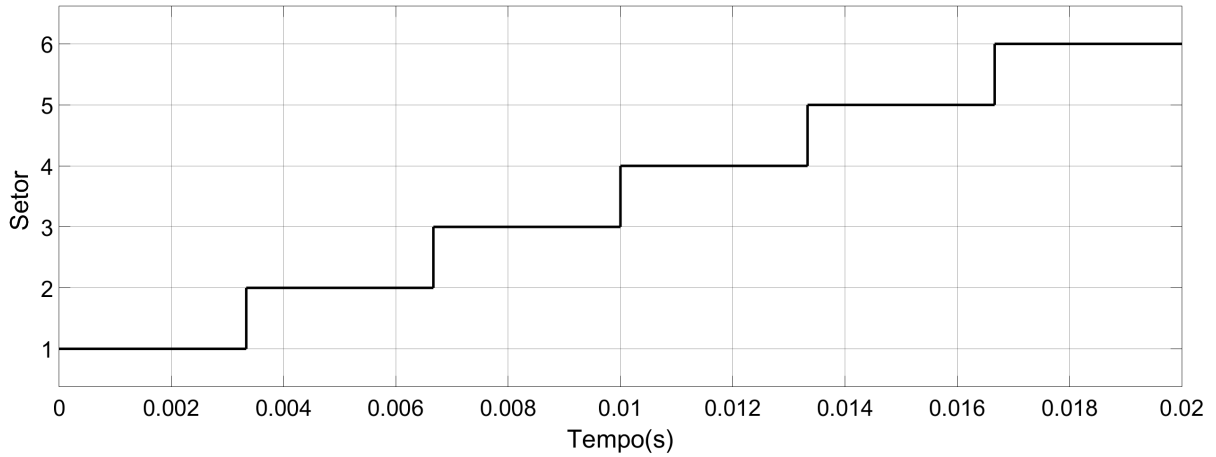
Figura 28 – Overview



Fonte:[Autor]

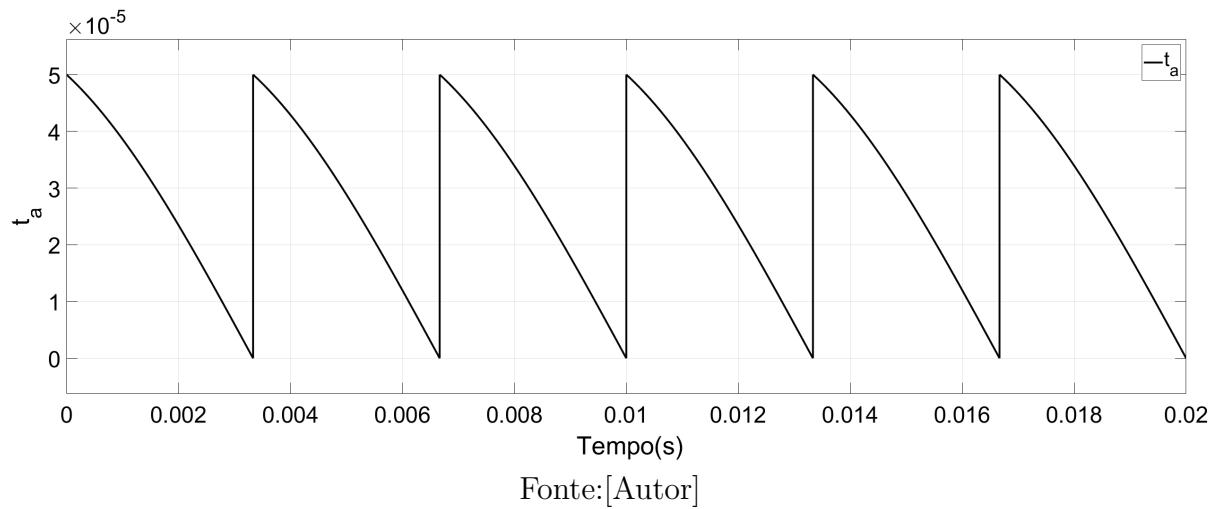
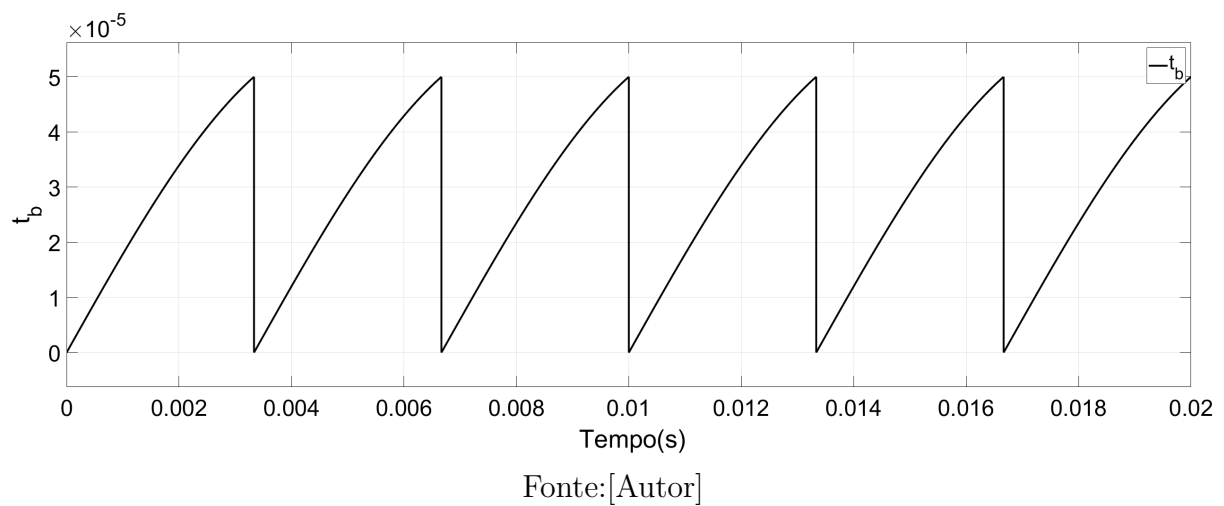
Todos os passos para obtenção nomeiam os sub-sistemas mostrados, no bloco chamado "Clarke" são geradas as referências $\alpha\beta$ a partir da referência trifásica. No bloco de identificação do setor, são feitas as comparações mostradas na tabela 4 para obter o setor de trabalho. No bloco chamado " t_a e t_b " são implementados os resultados da tabela 3 para obter os tempos de ativação dos vetores básicos. E no bloco " S_a , S_b e S_c " a ordem de chaveamento é definida.

Figura 29 – Setor de trabalho.

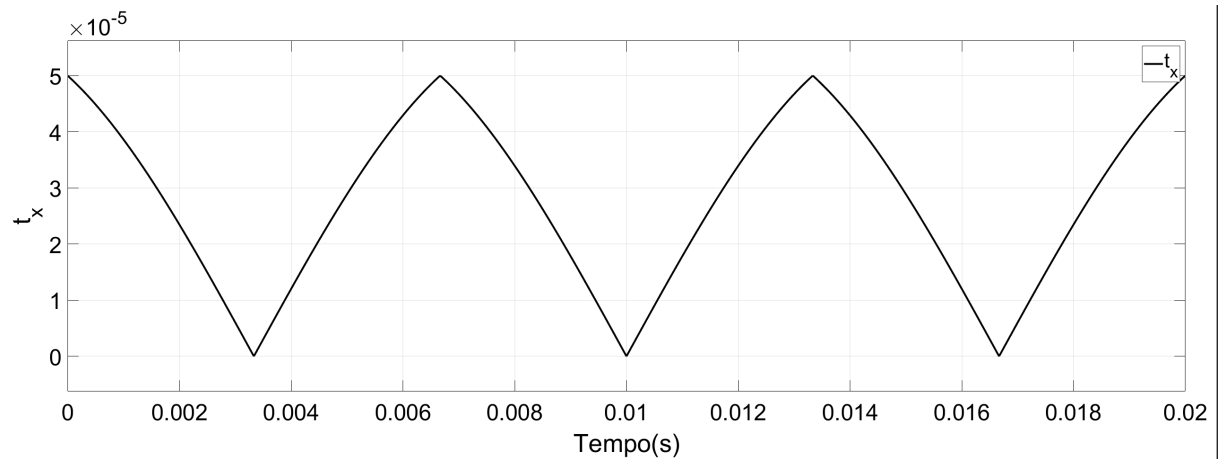


Fonte:[Autor]

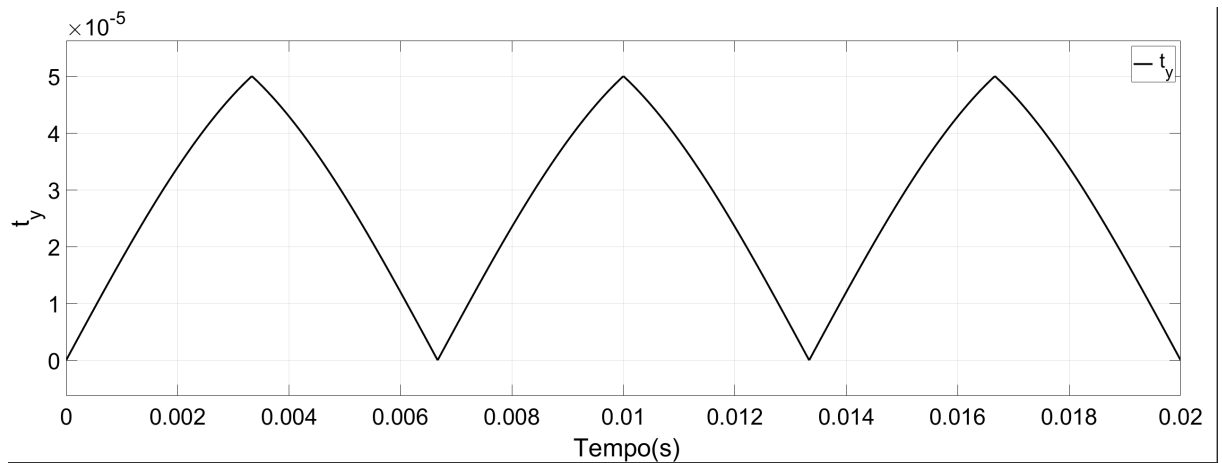
O setor varia de 1 a 6, indicando os seis setores do hexágono regular, este é um sinal periódico indicando que o vetor de referência gira em torno do eixo do hexágono regular. As figuras 30 e 31 apresentam os gráficos dos tempos " t_a e t_b ".

Figura 30 – tempo t_a .Figura 31 – tempo t_b .

Como indicado anteriormente os tempos t_a e t_b tem uma curva assimétrica, necessitando da aplicação do conjunto de equações (26) para obtenção de uma curva simétrica e por tanto com menor conteúdo harmônico. As figuras 32 e 33 apresentam os tempos t_x e t_y .

Figura 32 – tempo t_x .

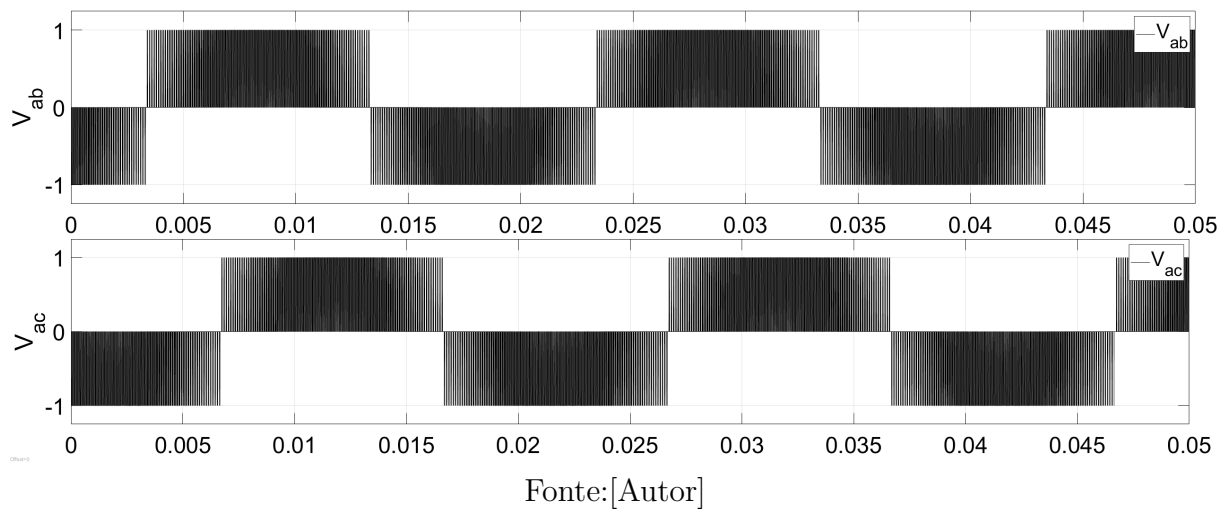
Fonte:[Autor]

Figura 33 – tempo t_y .

Fonte:[Autor]

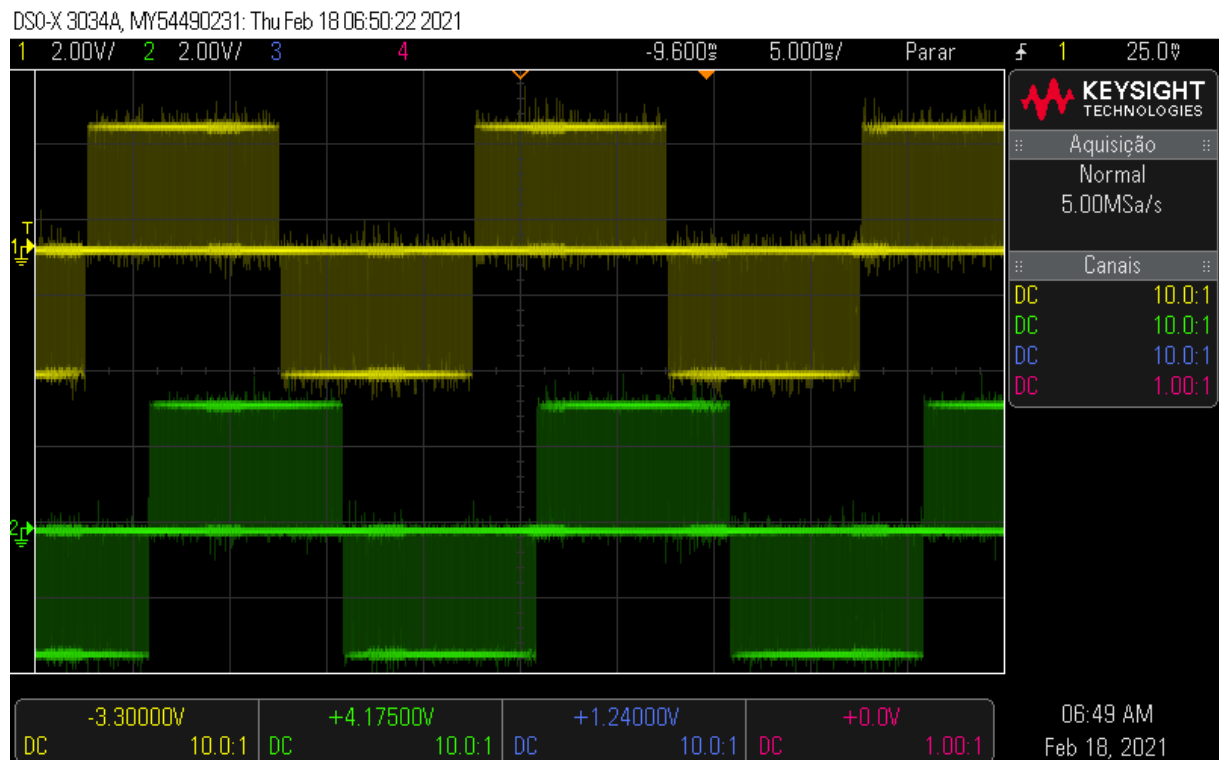
Acima estão indicados as curvas t_x e t_y . Em seguida, o resultado da tensão de linha V_{ab} e V_{ac} da simulação. A figura 34 mostra as tensões de linha da simulação.

Figura 34 – Tensões de linha simulação.



4.0.2 Resultados experimentais.

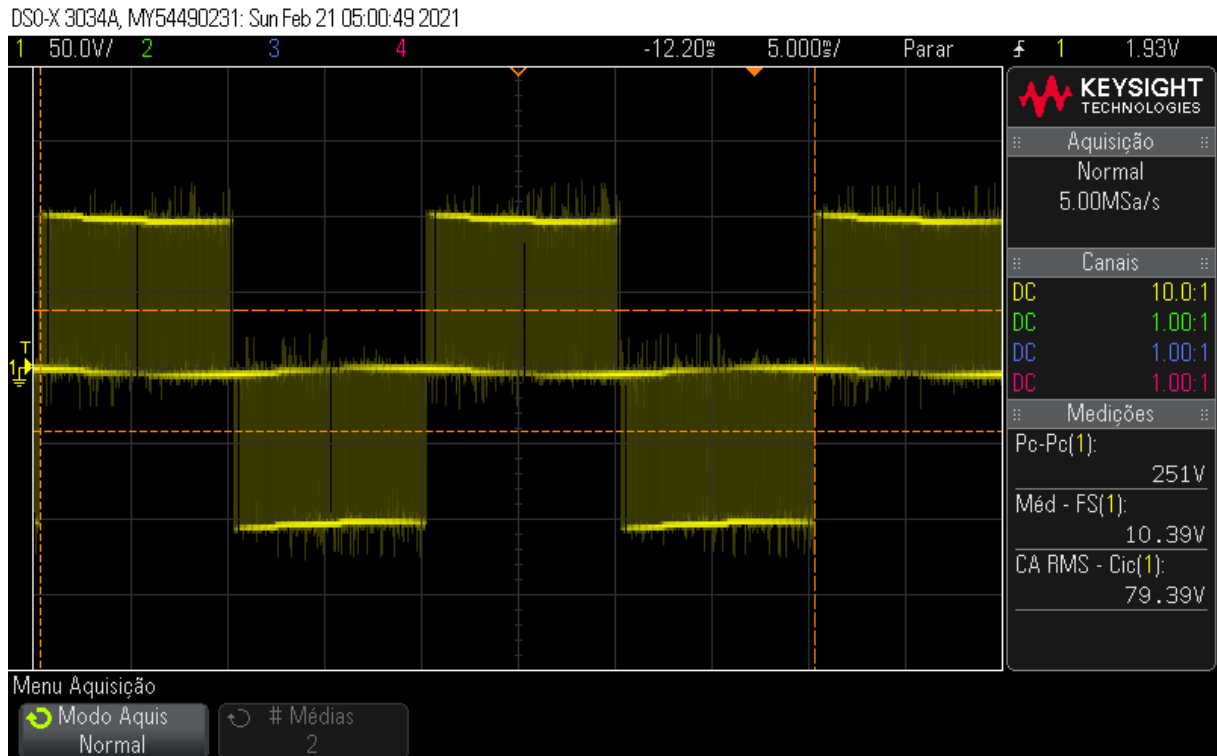
Na figura 35 é mostrado o perfil do sinal de linha V_{ab} e V_{ac} obtida na saída do *FPGA*.

Figura 35 – Tensões de linha na saída do *FPGA*.

A tensão apresenta perfil muito semelhante a observada figura 33, que é um

resultado de simulação, indicando resultado satisfatório. Na figura 36 é mostrada a tensão de linha V_{ab} medida na saída do inversor trifásico de dois níveis.

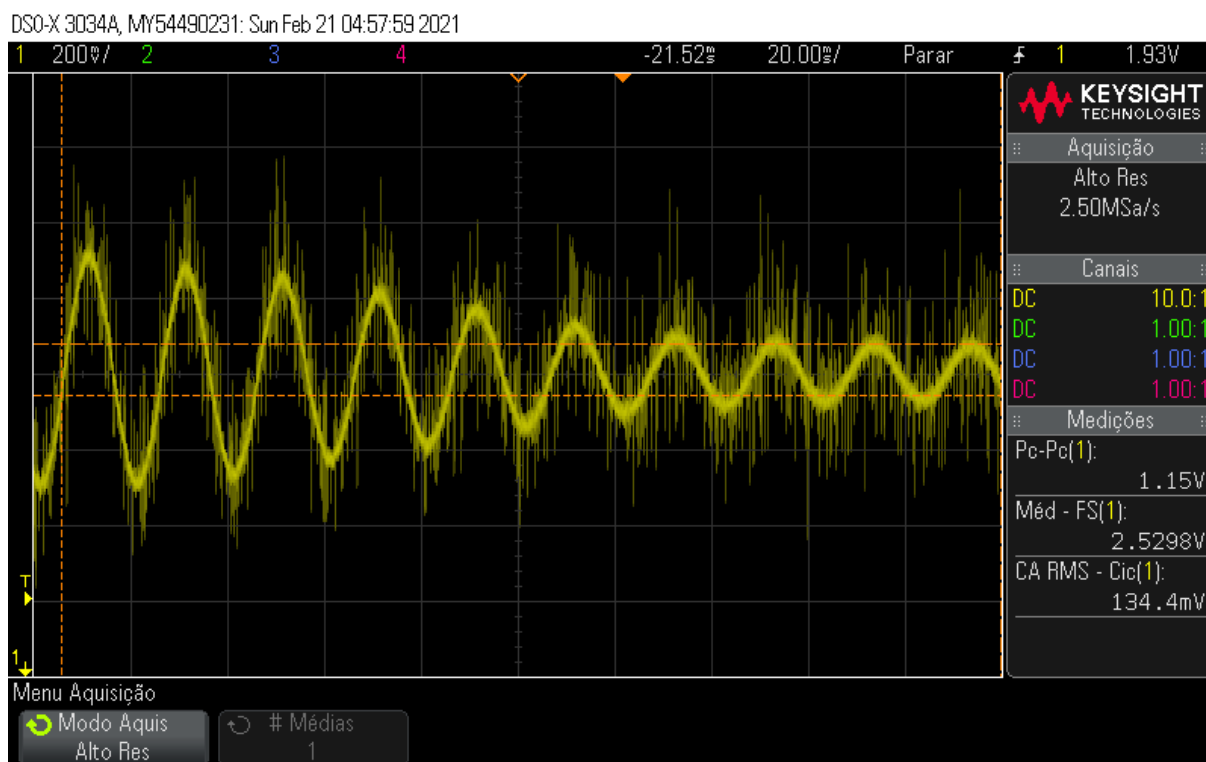
Figura 36 – Tensões de linha na saída do inversor.



Fonte:[Autor]

Pode ser observada grande semelhança na tensão de linha, tanto entre a simulação e a onda de saída do *FPGA*, quanto a tensão medida na saída do inversor. Na figura 37 é mostrado o perfil do sinal de corrente medido com auxílio do sensor de efeito Hall.

Figura 37 – Corrente de fase.



Fonte:[Autor]

A mudança de amplitude observada é decorrente do período transitório da máquina, mas o perfil da onda é claramente senoidal, mostrando que a modulação funciona da forma aceitável.

5 CONCLUSÃO

O trabalho de conclusão de curso apresentou a implementação experimental do algoritmo SVPWM em um FPGA. As equações que compõem o algoritmo SVPWM foram implementadas em linguagem VHDL com representação numérica em ponto fixo com sucesso. As saída de tensão no inversor são coerentes com os valores esperados teoricamente (ondas PWM com defasagem de 120°). Os códigos VHDL produzidos podem ser utilizados como um subsistema que faz parte do sistema de acionamento de motores em malha aberta ou fechada, baseados em FPGA. Neste trabalho, as tensões de referência foram geradas em um *look-up table*. Porém, em um sistema de controle em malha fechada, tais referências são produzidas pelo algoritmo de controle. Portanto, a codificação em VHDL da modulação SVPWM é simples de utilizar em sistemas de controle em malha fechada. Como trabalho futuro, sugere-se a utilização do código desenvolvido no controle de velocidade em malha fechada baseado em FGPA.

REFERÊNCIAS

- 1 HUSAIN, I. *Electric and Hybrid Vehicles: Design Fundamentals*. [S.l.]: CRC Press, 2003. ISBN 9780429956355.
- 2 NUMBER of passenger cars and commercial vehicles in use worldwide from 2006 to 2015 in (1,000 units). 2015. <<https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide>>. Acesso em 05/01/2021.
- 3 WORLD Motor Vehicle Production from 2004 to 2019. 2019. <EuropeanAutomobileManufacturersAssociation>. Acesso em 06/01/2021.
- 4 MANUFACTURERS, I. O. of M. V. *Man-Made CO2 Emissions*. <<https://www.oica.net/category/climate-change-and-co2/>>. Acesso em 06/01/2021.
- 5 MAN-MADE CO2 Emissions. <<https://www.oica.net/category/climate-change-and-co2/>>. Acesso em 06/01/2021.
- 6 KRISHNAN, R. *Electric Motor Drives: Modeling, Analysis, and Control*. Upper Saddle River, Nova Jersey, Estados Unidos: Prentice Hall, 2001.
- 7 GARCÍA, R. C. *Controle de Velocidade de Motor Síncrono de Ímã Permanente Utilizando Redes Neurais Artificiais e Multiplexação em Frequência*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Junho 2015.
- 8 MONZANI, R. C. *Controladores Analógicos e Digitais: uma análise comparativa*. Monografia (Trabalho de conclusão de curso) — Faculdade de Odontologia, Universidade Camilo Castelo Branco, Londrina, Paraná, 2010.
- 9 RIBEIRO, A. D. M. *Implementação em FPGA de Algoritmos para o Controle de Motores Trifásicos Baseados em Redes Neurais Artificiais com Função de Ativação Projetada Usando Algoritmos Genéticos*. Dissertação (Mestrado) — Universidade Federal de Mato Grosso do Sul (UFMS), Mato Grosso do Sul, 2020.
- 10 RASHID, M. H. *Electric and Hybrid Vehicles: Design Fundamentals*. [S.l.]: Elsevier, 2001. ISBN 978-0128114070.
- 11 GARCÍA, R. C. *Controle preciso de posição do motor síncrono de ímã permanente usando redes neurais artificiais para aplicação em robótica*. 2009.
- 12 SHARMA, D.; BHAT, A. H.; AHMAD, A. Ann based svpwm for three-phase improved power quality converter under disturbed ac mains. In: *2017 6th International Conference on Computer Applications In Electrical Engineering-Recent Advances (CERA)*. [S.l.: s.n.], 2017. p. 533–538.
- 13 SHU, Z. et al. An efficient svpwm algorithm with low computational overhead for three-phase inverters. *IEEE Transactions on Power Electronics*, v. 22, n. 5, p. 1797–1805, 2007.

APÊNDICE A – CÓDIGO COMPLETO

```

1  // Estrutura principal modulacao vetorial SV-PWM
2  // Rodolfo Biffi Pompeu de Barros
3  // UFMS, Campo Grande, 27/07/2020
4
5  library ieee;
6  use ieee.std_logic_1164.all;
7  use ieee.std_logic_1164.all;
8  use ieee.std_logic_arith.all;
9  use ieee.std_logic_unsigned.all;
10 use ieee.numeric_std.all;
11 use ieee.fixed_pkg.all;
12 use ieee.fixed_float_types.all;
13
14
15 entity principal2 is
16
17     port(
18         clk, reset          : in          std_logic    ;
19         setor_s              : out         std_logic_vector(2 downto 0 );
20         sa, sb, sc, tri      : out         std_logic
21     );
22
23 end entity;
24
25 architecture rtl of principal2 is
26
27     -- LUTs
28     // -----
29     -- Sin(0)
30
31
32     component s1 IS
33     PORT
34     (
35         address          : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
36         clock             : IN STD_LOGIC      := '1';
37         q                 : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
38     );
39     end component s1;
40
41     -- Sin(120)
42
43     component s2 IS PORT
44     (
45         address          : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
46         clock             : IN STD_LOGIC      := '1';
47         q                 : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
48     );
49     end component s2;
50
51     -- Sin(240)
52
53     component s3 IS PORT (
54         address          : IN STD_LOGIC_VECTOR (7 DOWNTO 0);
55         clock             : IN STD_LOGIC      := '1';
56         q                 : OUT STD_LOGIC_VECTOR (31 DOWNTO 0)
57     );

```

```

57         end component s3;
58 //-----
59
60
61
62 // Variaveis de contagem
63
64 signal cont, cont2          :std_logic_vector(15 downto 0);
65 signal endereco             :std_logic_vector(7 downto 0);
66
67
68 // Declaracao das variaveis em ponto fixo
69
70
71 signal Vas, Vbs, Vcs,Vas1,Vbs1,Vcs1      : sfixed (15 downto -16);
72 signal Valphas,Vbetas, Valpha1,Valpha2    : sfixed (15 downto -16);
73 signal c1, c2, c3, c4 , c_inv3, c_inv_r3   : sfixed (7 downto -16);
74 signal c5, c6, c7, c8, c9, c10             : sfixed (1 downto -30);
75 signal c11, c12, c13, c11n, c12n, c13n     : sfixed (1 downto -30);
76 signal ta1,ta2,ta3,ta4,ta5,ta6            : sfixed (1 downto -30);
77 signal txs, tys, tpwm                    : sfixed (1 downto -30);
78 signal L0, L1, L2, Tr , passo             : sfixed (1 downto -30);
79 signal c_0p5,c_0p25,c_m0p25              : sfixed (1 downto -3);
80
81 // Declaracao de variaveis em std_logic
82
83 signal va_std,vb_std,vc_std               : std_logic_vector(31 downto 0);
84 signal setor                             : std_logic_vector(2 downto 0);
85 signal V0, V1, V2, V3, V4, V5, V6, V7    : std_logic_vector(2 downto 0);
86 signal vx, vy                            : std_logic_vector(2 downto 0);
87 signal S                                  : std_logic;
88
89
90 // declarar alpha1 e alpha2
91 begin
92
93 //-----
94         -- Port sin(0)
95
96         SIN1: s1 port map(
97
98             address => endereco,
99             clock => clk,
100            q        => va_std
101
102            );
103
104         -- Port sin(120)
105
106         SIN2: s2 port map(
107
108             address => endereco,
109             clock => clk,
110            q        => vb_std
111
112            );
113

```

```

114
115         -- Port sin(240)
116
117     SIN3: s3 port map(
118
119         address => endereco,
120         clock   => clk,
121         q       => vc_std
122
123         );
124 //-----
125 V0 <= ("000");
126 V1 <= ("100");
127 V2 <= ("110");
128 V3 <= ("010");
129 V4 <= ("011");
130 V5 <= ("001");
131 V6 <= ("101");
132 V7 <= ("111");
133
134
135
136 contador : process(clk, reset)
137 begin
138
139     if (reset = '1') then
140
141
142         -- declaracao dos valores das constantes
143 //-----
144
145         passo                <= to_sfixed (0.00000002,passo);
146         c_0p5                <= to_sfixed (0.5, c_0p5);
147         c_0p25               <= to_sfixed (0.25, c_0p25);
148         c_m0p25              <= to_sfixed (-0.25, c_m0p25);
149         tpwm                  <= to_sfixed (0.0001, tpwm);
150         c_inv3                <= to_sfixed (0.3333333333333333,c_inv3);
151         c_inv_r3              <= to_sfixed (0.577350269189626,c_inv_r3);
152         c2                    <= to_sfixed (3.0,c2);
153         c3                    <= to_sfixed (1.73205080757,c3);
154         c4                    <= to_sfixed (-1.73205080757,c4);
155
156         c11                   <= to_sfixed (0.0000005,c11);
157         c12                   <= to_sfixed (0.0000002886751346,c12);
158         c13                   <= to_sfixed (0.0000005773502692,c13);
159         c11n                  <= to_sfixed (-0.0000005,c11n);
160         c12n                  <= to_sfixed (-0.0000002886751346,c12n);
161         c13n                  <= to_sfixed (-0.0000005773502692,c13n);
162
163
164
165
166 //-----
167
168 // zerando os valores das variaveis
169
170 //-----

```



```

171
172     Vas                <= (others => '0');
173     Vbs                <= (others => '0');
174     Vcs                <= (others => '0');
175     Valphas            <= (others => '0');
176     Valpha1            <= (others => '0');
177     Valpha2            <= (others => '0');
178     Vbetas             <= (others => '0');
179     setor              <= (others => '0');
180     ta1                <= (others => '0');
181     ta2                <= (others => '0');
182     ta3                <= (others => '0');
183     ta4                <= (others => '0');
184     ta5                <= (others => '0');
185     ta6                <= (others => '0');
186     L0                 <= (others => '0');
187     L1                 <= (others => '0');
188     L2                 <= (others => '0');
189
190 //-----
191
192
193     elsif (rising_edge(clk)) then
194
195
196         // contador
197 //-----
198         if (cont = 4999) then
199             cont                <= (others => '0');
200             Vas                 <= to_sfixed (va_std,Vas);
201             Vbs                 <= to_sfixed (vb_std,Vbs);
202             Vcs                 <= to_sfixed (vc_std,Vcs);
203
204 //-----atualizacao de enderecamento -----
205             if (endereco < 199) then
206                 endereco <= endereco +1; // incremento o endereco
207             else // endereco e 199
208                 endereco <= (others => '0'); // zerar o endereco
209             end if;
210 //-----
211
212             else \\ cont vai de 0 ate 4998
213                 cont <= cont + 1; // incrementar o contador
214             end if;
215 //-----
216
217 //-----criacao da triangular-----
218             if (cont = 0) then
219                 Tr <= (others => '0');
220             elsif (cont <= 2500) then \\ incrementar triangular
221                 Tr <= resize(Tr+passo,Tr);
222             else \\ decrementar triangular
223                 Tr <= resize(Tr-passo,Tr);
224             end if;
225
226 //-----
227

```

```

228 //Inicio da etapa de calculo:
229
230 //-----
231
232
233
234 // Definindo aptitude em 90% do valor total para evitar sobre-modulacao
235     if (cont = 0) then
236
237         Vas1      <= resize (90*Vas,Vas1);
238         Vbs1      <= resize (90*Vbs,Vbs1);
239         Vcs1      <= resize (90*Vcs,Vcs1);
240
241     elseif (cont = 1) then \\ calcular valfa vbeta
242
243         Valphas <= resize(((Vas1+Vas1)-(Vbs1+Vcs1))*c_inv3,Valphas);
244         Vbetas  <= resize((Vbs1 - Vcs1)*c_inv_r3,Vbetas);
245
246
247
248
249
250     elseif (cont = 2) then
251
252 // Variaveis auxiliares com valores das constantes
253
254         Valpha1 <= resize((c3*Valphas), Valpha1);
255         Valpha2 <= resize((c4*Valphas), Valpha2);
256
257
258
259
260
261
262     elseif (cont = 3) then
263
264 //-----1 2 3
265
266         if (Vbetas>=0 and Valpha1>Vbetas) then
267             setor <= "001";
268         end if;
269
270         if (Valpha1>=0 and Vbetas>=Valpha1) then
271             setor <= "010";
272         end if;
273
274         if (Valpha2>0 and Vbetas>Valpha2) then
275             setor <= "010";
276         end if;
277
278         if (Vbetas>0 and Valpha2>=Vbetas) then
279             setor <= "011";
280         end if;
281
282
283 //-----4 5 6
284

```

```

285
286     if (0>=Vbetas and Vbetas>Valpha1)           then
287         setor <= "100";
288     end if;
289
290     if (0>=Valpha1 and Valpha1>=Vbetas)         then
291         setor <= "101";
292     end if;
293
294     if (0>Valpha2 and Valpha2>Vbetas)           then
295         setor <= "101";
296     end if;
297
298     if (0>Vbetas and Vbetas>=Valpha2)           then
299         setor <= "110";
300     end if;
301
302
303     elsif (cont = 5) then
304
305         setor_s <= setor;
306
307
308
309
310     elsif (cont = 6) then
311
312
313 //-----Calculo dos tempos
314
315         ta1 <= resize ((c11*Valphas+c12n*Vbetas),ta1);
316         ta2 <= resize ((c11*Valphas+c12*Vbetas),ta2);
317         ta3 <= resize ((c13*Vbetas),ta3);
318         ta4 <= resize ((c11n*Valphas+c12*Vbetas),ta4);
319         ta5 <= resize ((c11n*Valphas+c12n*Vbetas),ta5);
320         ta6 <= resize ((c13n*Vbetas),ta6);
321
322
323 //-----txs, tys, vx e vy
324     elsif (cont = 7)      then
325
326         if (setor = "001") then
327             txs <= ta1;
328             tys <= ta3;
329             vx  <= V1;
330             vy  <= V2;
331         end if;
332         if (setor = "010") then
333             txs <= ta4;
334             tys <= ta2;
335             vx  <= V3;
336             vy  <= V2;
337         end if;
338         if (setor = "011") then
339             txs <= ta3;
340             tys <= ta5;
341             vx  <= V3;

```

```

342         vy <= V4;
343     end if;
344     if (setor = "100") then
345         txs <= ta6;
346         tys <= ta4;
347         vx <= V5;
348         vy <= V4;
349     end if;
350     if (setor = "101") then
351         txs <= ta5;
352         tys <= ta1;
353         vx <= V5;
354         vy <= V6;
355     end if;
356     if (setor = "110") then
357         txs <= ta2;
358         tys <= ta6;
359         vx <= V1;
360         vy <= V6;
361     end if;
362
363
364     elsif (cont=4999) then
365
366         L0<=resize( (c_0p25*tpwm) + (c_m0p25*txs)+ (c_m0p25*tys), L0);
367         L1<=resize( (c_0p25*tpwm) + (c_0p25*txs) + (c_m0p25*tys), L1);
368         L2<=resize( (c_0p25*tpwm) + (c_0p25*txs) + (c_0p25*tys), L2);
369
370     end if;
371
372
373     end if;
374 end process; // contagem
375
376 process(cont, clk)
377 begin
378     if (cont=0) then
379         sa <='0';
380         sb <='0';
381         sc <='0';
382
383
384     elsif (cont=4999) then
385         sa <='0';
386         sb <='0';
387         sc <='0';
388     else
389         if (Tr < L0) then
390             sa <='0';
391             sb <='0';
392             sc <='0';
393
394
395         elsif (Tr < L1) then
396             sa <=Vx(2);
397             sb <=Vx(1);
398             sc <=Vx(0);

```

```
399
400         elsif (Tr < L2) then
401             sa <= Vy(2);
402             sb <= Vy(1);
403             sc <= Vy(0);
404
405         else
406             sa <= '1';
407             sb <= '1';
408             sc <= '1';
409         end if;
410     end if;
411
412 end process;
413 end architecture rtl;
```