

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
MESTRADO EM ENGENHARIA ELÉTRICA

THIAGO TONON

**CONTROLE NEURAL APLICADO A UM CONVERSOR *BOOST*
UTILIZADO EM AEROGERADORES DE BAIXA POTÊNCIA**

DISSERTAÇÃO

CORNÉLIO PROCÓPIO

2014

THIAGO TONON

**CONTROLE NEURAL APLICADO A UM CONVERSOR *BOOST*
UTILIZADO EM AEROGERADORES DE BAIXA POTÊNCIA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Mestre em Engenharia Elétrica”.

Orientador: Prof. Dr. Hélio Voltolini

Co-orientador: Prof. Dr. Alessandro Goedel.

CORNÉLIO PROCÓPIO
2014

Dados Internacionais de Catalogação na Publicação

- T616 Tonon, Thiago.
Controle neural aplicado a um conversor *boost* utilizado em aerogeradores de baixa potência / Thiago Tonon. – 2014.
71 p. : il. color. ; 30 cm.
- Orientador: Hélio Voltolini.
Coorientador: Alessandro Goedel
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Engenharia Elétrica. Cornélio Procopio, 2014.
Referências: p. 70-71.
1. Conversores de corrente elétrica. 2. Controladores elétricos. 3. Redes Neurais (Computação). 4. Energia eólica. 5. Engenharia elétrica – Dissertações. I. Voltolini, Hélio, orient. II. Goedel, Alessandro, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

CDD (22. ed.) 621.3



Título da Dissertação Nº 20 :

“Controle Neural Aplicado a um Conversor *Boost* Utilizado em Aerogeradores de Baixa Potência”.

por

Thiago Tonon

Orientador: Prof. Dr. Hélio Voltolini

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM ENGENHARIA ELÉTRICA – Área de Concentração: Sistemas Eletrônicos Industriais, pelo Programa de Pós-Graduação em Engenharia Elétrica – PPGE – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 14:00h do dia 27 de agosto de 2014. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

Prof. Dr. Hélio Voltolini
Presidente

Prof. Dr. Claudionor Francisco do Nascimento
Universidade Federal de São Carlos

Prof. Dr. Maurício dos Santos Kaster
Universidade Tecnológica Federal do Paraná

Visto da coordenação:

Paulo Rogério Scalassara

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Av. Alberto Carazzai, 1640 - 86.300-000- Cornélio Procópio – PR.

Tel. +55 (43) 3520-4007 / e-mail: ppgee-cp@utfpr.edu.br / www.utfpr.edu.br/cornelioprocopio/ppgee

Dedico este trabalho a toda a minha família, em especial aos meus pais Edinir Braz Tonon e Iraci Pelisson Tonon, e a minha esposa Rhawenna Zanella Santana e minha filha Maria Júlia Zanella Tonon.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelo dom da vida e perseverança.

Agradeço a minha esposa Rhawenna Zanella Santana e à minha filha Maria Julia Zanella Santana, pelos dias de ausência e paciência no decorrer deste trabalho.

Agradeço também meus pais Edinir Braz Tonon e Iraci Pelisson Tonon pelas orações e palavras de apoio.

Agradeço aos meus sogros Julio de Almeida Santana e Maria A. Zanella Santana pelo apoio.

Ao meu orientador, Prof. Dr. Hélio Voltolini que, mesmo diante da distância, sempre dedicou seu tempo e conhecimento nas orientações relativas ao trabalho.

Um agradecimento especial ao meu co-orientador Prof. Dr. Alessandro Goedtel por toda ajuda no decorrer destes anos.

A todos os professores do Programa de Pós-Graduação em Engenharia Elétrica da UTFPR - câmpus Cornélio Procópio, pelos conhecimentos passados no decorrer destes anos.

A todos os amigos que fizeram parte desta caminhada durante estes anos, por todos os momentos e conhecimentos compartilhados.

Um agradecimento ao suporte financeiro da Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Paraná (Processo N° 06/56093-3), do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Processo N° 474290/2008-5, 473576/2011-2, 552269/2011-5), e CAPES-DS.

“Todas as vitórias ocultam uma abdicação.”
Simone de Beauvoir

“E ainda que tivesse o dom da profecia, e conhecesse todos os mistérios e toda a ciência, e ainda que tivesse toda fé de maneira tal que transportasse os montes, e não tivesse amor, nada seria.”
(Cor 1, 13: 2).

RESUMO

TONON, Thiago. CONTROLE NEURAL APLICADO A UM CONVERSOR *BOOST* UTILIZADO EM AEROGERADORES DE BAIXA POTÊNCIA. 71 f. Dissertação – Mestrado em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

Este trabalho apresenta o estudo, projeto e simulação de um conversor CC *boost*, com controle da tensão de saída utilizando redes neurais. O conversor *boost* estudado neste trabalho está sendo utilizado em um aerogerador de baixa potência, com potência máxima de 3kW. Devido à operação em velocidade variável do aerogerador, a tensão de entrada no conversor também é variável em uma ampla faixa de operação, e a saída deve ser estabilizada para uma tensão fixa. Isso faz com que haja a necessidade do controle do conversor para que as diferenças na tensão de entrada sejam compensadas. Para a compensação dessa diferença, foi projetado um controlador utilizando o método de lugar das raízes. A modelagem do conversor *boost* também é apresentada. O controlador fornece o tempo de atuação da chave semicondutora de potência utilizada no conversor, controlando assim a tensão de saída. A variação na tensão de entrada, que pode variar de 50V a 350V, faz com que o controlador não atue de forma otimizada para todos os pontos de operação. Dessa forma, um controlador neural foi projetado para que trabalhe como controlador, compensando distúrbios de tensão de entrada do conversor. A validação dos controles foi efetuada através de simulação utilizando o software Matlab/Simulink, para confirmação do desenvolvimento teórico apresentado no trabalho.

Palavras-chave: Modelagem do Conversor *Boost*, Controle do Conversor *Boost*, Controle Neural, Gerador Eólico.

ABSTRACT

TONON, Thiago. NEURAL CONTROL APPLIED TO A *BOOST* CONVERTER USED IN LOW POWER WIND GENERATORS. 71 f. Dissertação – Mestrado em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

This work presents the study, design and simulation of a DC boost converter with output voltage control using neural networks. The boost converter studied in this work is being used in a wind turbine low power, with maximum power of 3kW. Due to the operation of the variable speed of the wind turbine, the input voltage of the converter is also variable over a wide operating range, and the output must be stabilized to a fixed voltage. This means that there is a need to control the converter so that differences in input voltage are compensated. To compensate this difference, a controller was designed using the root locus method. The modeling of the boost converter is also presented. The controller provides the time of operation of power semiconductor switch used in the converter, thereby controlling the output voltage. The variation in input voltage, which can vary from 50V to 350V, makes the driver does not act optimally for all operating points. Thus, a neural controller is designed to work as a controller, compensating disturbances of the converter input voltage. The validation of this controller was performed by simulation using Matlab / Simulink software, to confirm the theoretical development presented in the work.

Keywords: Modeling Boost Converter, Boost Control, Neural Control, Wind Power Generator.

LISTA DE ILUSTRAÇÕES - FIGURAS

Figura 1 - Diagrama do aerogerador.....	15
Figura 2 - Conversor <i>Boost</i>	20
Figura 3 - Conversor <i>Boost</i> com chave em condução.	20
Figura 4 - Conversor <i>Boost</i> com chave em corte.....	21
Figura 5 - Conversor <i>Boost</i> simulado no Matlab/ <i>Simulink</i>	36
Figura 6 - Tensão de saída pela tensão de entrada em malha aberta.	37
Figura 7 - Lugar das raízes do sistema.	39
Figura 8 - Diagrama de blocos do sistema.	40
Figura 9 - Resposta do sistema ao degrau unitário.....	40
Figura 10 - Conversor <i>Boost</i> controlado simulado no Matlab/ <i>Simulink</i>	41
Figura 11 - Controle aplicado ao conversor <i>Boost</i>	42
Figura 12 - Tensão de saída do conversor para uma tensão de entrada de 200V.....	42
Figura 13 - Tensão de saída do conversor para uma tensão de entrada de 350V.....	43
Figura 14 - Tensão de saída do conversor para uma tensão de entrada de 280V.....	43
Figura 15 - Tensão de saída do conversor para uma tensão de entrada de 210V.....	44
Figura 16 - Tensão de saída do conversor para uma tensão de entrada de 140V.....	44
Figura 17 - Tensão de saída do conversor para uma tensão de entrada de 70V.....	45
Figura 18 - Tensão de saída do conversor para uma tensão de entrada de 50V.....	45
Figura 19 - Neurônio Artificial.....	48
Figura 20 - Camadas de uma Rede Neural Artificial.	50
Figura 21 - Diagrama de treinamento da rede neural.	57
Figura 22 – Topologia da rede neural utilizada como controle do conversor.	59
Figura 23 - Resposta do sistema para uma tensão de entrada de 200V.....	60
Figura 24 - Resposta do sistema para uma tensão de entrada de 350V.....	61
Figura 25 - Resposta do sistema para uma tensão de entrada de 280V.....	62
Figura 26 - Resposta do sistema para uma tensão de entrada de 210V.....	63
Figura 27 - Resposta do sistema para uma tensão de entrada de 140V.....	63
Figura 28 - Resposta do sistema para uma tensão de entrada de 70V.....	64
Figura 29 - Resposta do sistema para uma tensão de entrada de 50V.....	64
Figura 30 - Resposta do sistema para uma tensão de entrada em rampa.....	66
Figura 31 - Resposta do sistema para uma tensão de entrada em rampa.....	67

LISTA DE TABELAS

Tabela 1 - Parâmetros do conversor <i>boost</i>	36
Tabela 2 – Funções de ativação do neurônio artificial.	49
Tabela 3 – Exemplo dos dados utilizados no treinamento.	56
Tabela 4 – Resumo das simulações para coleta de dados de operação do conversor.....	56
Tabela 5 – Amostras para rede neural.	58
Tabela 6 – Dados da rede neural.	59
Tabela 7 – Índice ITAE para tensão de entrada de 200V.....	61
Tabela 8 – Índice ITAE para tensão de entrada de 350V.....	62
Tabela 9 – Índice ITAE para tensão de entrada de 280V.....	62
Tabela 10 – Índice ITAE para tensão de entrada de 210V.....	63
Tabela 11 – Índice ITAE para tensão de entrada de 140V.....	64
Tabela 12 – Índice ITAE para tensão de entrada de 70V.....	65
Tabela 13 – Índice ITAE para tensão de entrada de 50V.....	65
Tabela 14 – Índice ITAE para tensão de entrada em rampa.....	66
Tabela 15 – Índice ITAE para tensão de entrada em rampa.....	67

LISTA DE ABREVIATURAS E SIGLAS

AG	Algoritmos Genéticos
GSIM	Gerador Síncrono de Imãs Permanentes
kHz	KiloHertz
kW	KiloWatts
LKC	Lei de Kirchhoff das Correntes
LKT	Lei de Kirchhoff das Tensões
MSIM	Máquina Síncrona de Imãs Permanentes
PI	Proporcional Integrativo
PMSM	<i>Permanent Magnetic Synchronous Machine</i>
PWM	<i>Pulse Width Modulation</i>
RBF	<i>Radial Basis Function</i>
RNA	Rede Neural Artificial
ITAE	<i>Integral of Time by Absolute Error</i>

LISTA DE SÍMBOBLOS

v_o	Tensão de saída do conversor
v_g	Tensão de entrada do conversor
D	Razão cíclica
i_L	Corrente no indutor
i_C	Corrente no capacitor
i	Corrente do conversor
L	Indutor do conversor
C	Capacitor do conversor
R	Carga de saída do conversor
R_c	Resistência interna do capacitor
$G(s)$	Função de Transferência do Conversor <i>boost</i>
$C(s)$	Função de Transferência do Controlador
H_c	Ganho do sensor de tensão

SUMÁRIO

1	INTRODUÇÃO	13
1.1	PROBLEMA A SER PESQUISADO	14
1.2	OBJETIVOS DA PESQUISA.....	16
1.2.1	Objetivo geral	16
1.2.2	Objetivos específicos	16
1.3	JUSTIFICATIVA.....	17
1.4	ORGANIZAÇÃO DO TRABALHO.....	18
2	MODELAGEM E SIMULAÇÃO DO CONVERSOR <i>BOOST</i>	19
2.1	INTRODUÇÃO	19
2.2	MODELAGEM DO CONVERSOR <i>BOOST</i>	20
2.2.1	Chave de potência em condução.....	22
2.2.2	Chave de potência em condução.....	24
2.3	<i>STATE-SPACE AVERAGING</i>	26
2.4	APLICANDO O <i>STATE-SPACE AVERAGING</i>	30
2.5	EXTRAINDO A FUNÇÃO DE TRANSFERÊNCIA	33
2.6	SIMULAÇÃO DO CONVERSOR <i>BOOST</i>	35
2.7	ÍNDICE DE DESEMPENHO	37
2.8	PROJETO DO CONTROLADOR.....	38
2.9	CONCLUSÃO	45
3	REDES NEURAIS ARTIFICIAS	47
3.1	INTRODUÇÃO	47
3.2	O NEURÔNIO ARTIFICIAL.....	47
3.3	ARQUITETURA DAS REDES NEURAIS ARTIFICIAIS	49
3.3.1	Redes <i>feedforward</i> multicamadas.....	51
3.4	TREINAMENTO E ASPECTO DE APRENDIZADO DAS REDES NEURAIS	51
3.5	REDE <i>PERCEPTRON</i> MULTICAMADAS	52
3.5.1	Algoritmo de aprendizado <i>backpropagation</i>	53
4	CONTROLE NEURAL APLICADO AO CONVERSOR <i>BOOST</i>	55
4.1	INTRODUÇÃO	55
4.2	CONTROLE NEURAL	55
4.3	SIMULAÇÃO DO CONVERSOR COM CONTROLE NEURAL	60
5	CONCLUSÃO	68
5.1	CONCLUSÃO	68
5.2	TRABALHOS FUTUROS	68
5.3	PUBLICAÇÃO ASSOCIADA AO TRABALHO.....	69
	REFERÊNCIAS	70

1 INTRODUÇÃO

O crescente aumento no consumo de energia elétrica aliado com a busca por sistemas geradores com o menor impacto ambiental vem fazendo com que cada dia mais pesquisadores busquem soluções que equilibrem essas necessidades. Das inúmeras formas de se gerar energia de maneira ecologicamente correta, a que mais se destaca é a geração eólica. Sua instalação é relativamente rápida e gera um menor impacto ambiental, quando comparado com outros tipos de geração de energia elétrica, como no caso de usinas hidráulicas.

Entretanto, a utilização de forças mecânicas intermitentes para conexão aos geradores, como por exemplo a força do vento, cria a necessidade da operação do sistema em velocidade variável. É preciso padronizar a tensão gerada para que o sistema possa operar em paralelo com outros geradores, ou ainda, poder utilizar essa potência gerada. Geralmente a necessidade da utilização de conversores eletrônicos, bem como o crescente desenvolvimento de componentes com mais eficiência energética, permite o controle e a regulação de tensões geradas por esses tipos de sistemas.

Em geral, para aerogeradores de baixa potência, o gerador elétrico utilizado gera tensões com amplitude e frequência variáveis que dependem da velocidade do vento. Assim, um sistema eletrônico converte a tensão gerada na forma CA com amplitude e frequência variável em uma tensão CA de amplitude e frequência fixa compatível com a rede elétrica. Dessa forma, o projeto e controle desses conversores são fundamentais para um efetivo funcionamento do sistema.

Os conversores CC-CC controlados são dispositivos capazes de fornecer uma tensão média de saída com amplitude fixa, mesmo que sua tensão de entrada ou sua carga possa variar. Seu princípio de funcionamento está baseado no conjunto de chave potência, indutor e capacitor. Juntos, os conversores são capazes de regular a tensão de saída para valores abaixo da tensão de entrada, como em conversores tipo *buck*, acima da tensão de entrada, como nos conversores tipo *boost*, ou sistemas que regulam a tensão de saída para valores acima ou abaixo da entrada, como em conversores *buck-boost*. Outras topologias podem ser encontradas na literatura, além dessas.

O conversor *boost* é um dispositivo capaz de fornecer uma tensão de saída com amplitude maior que o valor de tensão de entrada. O conjunto é formado fundamentalmente por um indutor, um capacitor, um diodo e uma chave de potência. A chave de potência deve ser controlada para fornecer tensão de saída com amplitude desejada. Efetuar o adequado controle do tempo de

chaveamento determina a tensão de saída que o conversor irá fornecer. Assim, o sinal de controle adequado à chave de potência é fundamental.

Técnicas convencionais de controle desses conversores consideram os componentes como ideais, sem perdas, e com funcionamento linear. Assim, o ponto de operação do sistema é estável, podendo ser modelado como um sistema de equações lineares. A partir dessas equações, um controlador pode ser desenvolvido, a fim de operar o sistema em malha fechada.

Entretanto, na prática, esses componentes não apresentam condições de funcionamento lineares. Além disso, a tensão de entrada, assim como a carga, podem variar, tornando o sistema instável. Por isso, uma solução é a utilização de sistemas de controles adaptativos. (MAHDAVI *et al.*, 2005)

Algumas soluções de implementação de redes neurais podem ser encontrados, como em (JAWHAR, 2006) e (KABOLI, 2013) que propõem o ajuste dos ganhos do controle utilizando sistemas neuro-fuzzy. (HASSAN, 2010) propõe o controle PI neural para controle da posição de servo motor atuando em sistemas pneumáticos. (ROSTAMI, 2010) propõe uma rede neural para previsão de ações futuras na planta para ação do controle PI. (UTOMO, 2011) aplica uma rede neural como controle do erro de um conversor *buck-boost*, obtendo a razão cíclica da chave a partir da diferença entre a tensão de saída e a tensão desejada e da diferença entre a tensão de saída atual com a tensão anterior e (DHIVYA, 2013) que propõe um controle neural para um conversor *boost* a partir da leitura de quatro variáveis do sistema para obtenção da razão cíclica do conversor.

1.1 PROBLEMA A SER PESQUISADO

O projeto do controle de um conversor *boost* é executado a partir da função de transferência que descreve matematicamente o comportamento do sistema. Essa função é obtida a partir das características do sistema, considerando os componentes como ideais e lineares. Entretanto, não há linearidade no sistema real. Dentre as possibilidades, pode-se projetar um controlador adaptativo.

O controle neural apresenta uma solução interessante para esse problema, uma vez que ele é capaz de fazer a aproximação de uma função. Podemos considerar, no tempo, que a saída do controle é uma função. A rede neural, trabalhando nessa aproximação, fará com que o sinal

fornecido ao sistema tenha uma aproximação de uma função linear, eliminando assim os períodos de funcionamento dinâmico do sistema.

Será considerado um sistema eólico de baixa potência que utiliza o Gerador Síncrono de Ímãs Permanentes. A partir da entrada de torque na máquina, gerado pela turbina de vento, a máquina gera uma tensão com amplitude e frequência que variam conforme a velocidade do vento. A saída desse gerador é conectada a um retificador não controlado. A saída desse conversor fornece tensão com amplitude variável para o conversor *boost* que deverá fornecer uma tensão com amplitude fixa na sua saída para, enfim, ser convertido em tensão CA pelo inversor de frequência. A faixa de operação desse conversor será com tensão de entrada entre 50V e 350V. Para valores de entrada maiores ou menores, o sistema não entrará em operação. A tensão de saída do conversor deve ser fixada em 400V. A potência fornecida pelo conversor será proporcional à tensão de entrada. Dessa forma, a potência fornecida pelo aerogerador será proporcional à velocidade do vento que, conseqüentemente, faz a variação da tensão de entrada do conversor. A Figura 1 apresenta o diagrama do sistema que faz parte desse projeto.

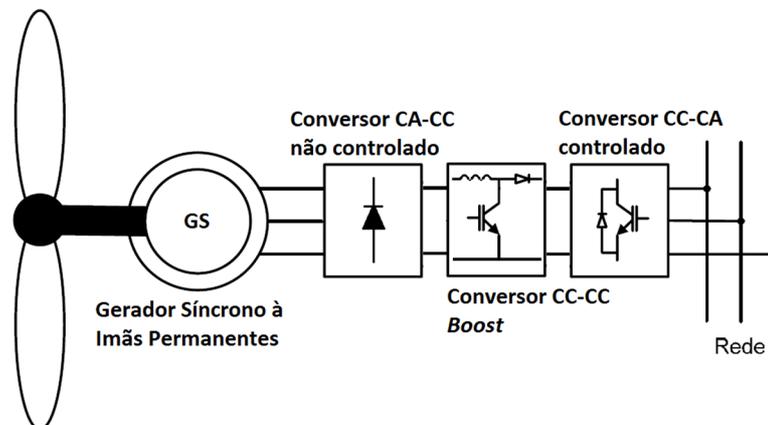


Figura 1 - Diagrama do aerogerador.

Fonte: Autoria própria.

Conforme foi apresentado, a utilização do controle convencional não fará com que o sistema opere de forma satisfatória, uma vez que no controle convencional o projeto será realizado a partir de um ponto de operação fixo. O conceito de controle convencional deste trabalho se aplica ao controle projetado a partir da função de transferência do sistema, utilizando técnicas de sintonia do controlador, como, por exemplo, lugar das raízes. As variações de tensão de entrada, bem como as variações de potência de saída farão com que o sistema sofra oscilações ou respostas transitórias indesejadas.

Enfim, a utilização de um controle neural fará com que o sistema possa operar de forma satisfatória, eliminando as oscilações em modo de operação constante, e seu funcionamento poderá ser melhorado para toda a faixa de tensões de entrada.

1.2 OBJETIVOS DA PESQUISA

Os objetivos que conduzem a pesquisa neste trabalho são divididos em objetivo geral e específicos.

1.2.1 Objetivo geral

Este trabalho visa o estudo e a simulação de um controle neural aplicado a um conversor estático tipo *boost*. Assim, uma análise comparativa entre a proposta de controle convencional e o controle neural poderá ser analisada.

1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são definidos como segue:

- Estudar os conceitos relacionados ao conversor *boost*;
- Estudar os conceitos relacionados ao controle do conversor *boost*;
- Estudar os conceitos relacionados com redes neurais artificiais;
- Simular o conversor *boost* no *Simulink/Matlab* em malha aberta e validar o funcionamento do sistema;
- Modelar o sistema e obter a função de transferência do conversor.
- Aplicar um controle convencional e simular o sistema com controle em malha fechada;

- Gerar dados para o treinamento e para validação da rede neural;
- Treinar uma rede neural artificial e validar a partir dos dados de treinamento e validação;
- Simular o conversor *boost* controlado pelo controlador neural;
- Validar a proposta, avaliando e comparando o desempenho dinâmico e estático do controle neural em relação ao controle convencional.

1.3 JUSTIFICATIVA

A operação do aerogerador de energia elétrica possui uma característica intrínseca: a operação em modo não contínuo. A potência do vento utilizada na conversão de energia não é contínua e sofre uma quantidade grande de variação. Por isso, a tensão de saída de um gerador conectado à turbina de vento sofre grande variação. Assim, a potência de saída do gerador será proporcional à velocidade do vento.

Com a variação na tensão de entrada, o controlador sintonizado a partir de um ponto de operação não traz resultados satisfatórios, pois o sistema sofre continuamente perturbações na tensão de entrada. Por isso, a utilização da rede neural para o controle deste conversor é importante, pois ela será capaz de controlar o conversor de forma mais efetiva, trazendo resultados otimizados para estas características de funcionamento.

O controle neural fará o controle utilizando apenas a tensão de entrada e de saída do conversor. Além disso, a rede neural será capaz de efetuar um melhor controle em relação aos modos de operação em regime transitório.

Dessa forma, o controle do conversor terá um ganho com relação à qualidade da tensão de saída, melhorando sua tensão de saída.

Além disso, o treinamento da rede neural para atuar como controlador irá utilizar dados de operação do sistema. Assim, um controle neural pode ser projetado a partir de um sistema que já esteja em operação, sem a necessidade de modelagem do mesmo.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho é dividido em cinco capítulos, conforme citados a seguir.

O primeiro capítulo apresenta uma abordagem introdutória, com justificativa e objetivos do trabalho.

O segundo capítulo apresenta a modelagem do conversor *boost*, objetivando a determinação da função de transferência do sistema. Na sequência, é apresentada uma abordagem do controle do conversor, bem como a simulação e validação do controle proposto.

O terceiro capítulo apresenta os conceitos básicos das redes neurais artificiais, além de uma abordagem da rede *Perceptron* multicamadas, utilizada na elaboração deste trabalho.

O quarto capítulo apresenta a simulação e o controle neural aplicado ao conversor, bem como resultados comparativos entre a proposta convencional e a proposta neural para controle.

No quinto capítulo são apresentadas as conclusões referentes à proposta deste trabalho.

2 MODELAGEM E SIMULAÇÃO DO CONVERSOR *BOOST*

2.1 INTRODUÇÃO

Um conversor CC-CC é um equipamento capaz de converter um nível de tensão CC de entrada para um nível de tensão CC desejado na saída, a fim de satisfazer os requisitos de cada carga.

Existe uma série de topologias capazes de fazer com que a tensão de saída possa ter um nível mais baixo ou mais alto. Um exemplo de um conversor capaz de fornecer uma tensão de saída com amplitude inferior à tensão de entrada é o conversor *Buck*. É possível também construir conversores capazes de fornecer na saída tensões com amplitudes superiores do que a entrada, como por exemplo, conversores tipo *boost*.

Devido às estas características, os conversores são utilizados, por exemplo, para o condicionamento de energia de fontes alternativas, tais como painéis fotovoltaicos, geradores eólicos e sistemas que não possuam tensão de saída com amplitude contínua. Assim, os conversores possuem um grande potencial de mercado futuro (KOTHAPALLI, 2008).

As fontes de alimentação CC possuem uma grande área de aplicabilidade, desde sistemas que necessitem de baixas potências, como por exemplo, em carregadores de celulares, até para grandes aplicações de potência, como em motores elétricos de locomotivas, por exemplo.

Cada aplicação requer uma tensão e uma corrente específica. Esse condicionamento de tensão é alvo de estudos e já possuem uma série de metodologias para obtenção de resultados.

O conversor utilizado neste trabalho é o conversor *boost*. Sua escolha foi feita, pois as condições de tensão de entrada são variáveis e há a necessidade de se obter uma tensão de saída com amplitude fixa, mais alta que a tensão de entrada.

Neste capítulo, são apresentadas a modelagem do conversor e sua simulação utilizando a ferramenta Matlab/*Simulink*.

2.2 MODELAGEM DO CONVERSOR *BOOST*

O conversor estático de potência utilizado no sistema e, dessa forma, utilizado para elaboração deste trabalho é o conversor elevador de tensão, tipo *boost*.

A topologia do conversor *boost* é apresentada na Figura 2.

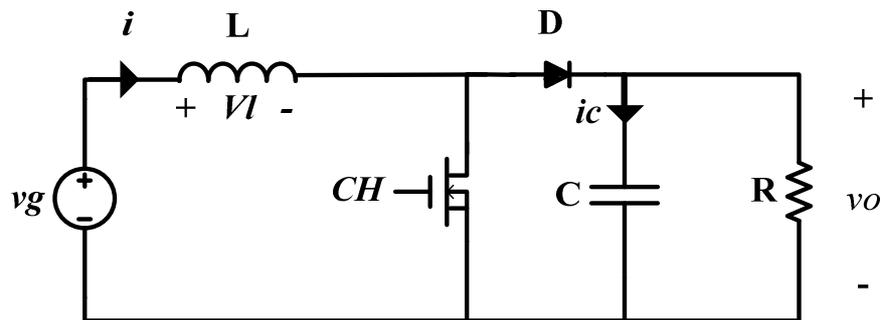


Figura 2 - Conversor *Boost*.

Fonte: Autoria própria.

O conversor *boost* possui dois pontos de operação distintos: quando a chave semicondutora de potência está em condução ou quando a chave semicondutora de potência está em corte.

Quando a chave de potência do conversor está em condução, sua topologia pode ser expressa conforme a Figura 3.

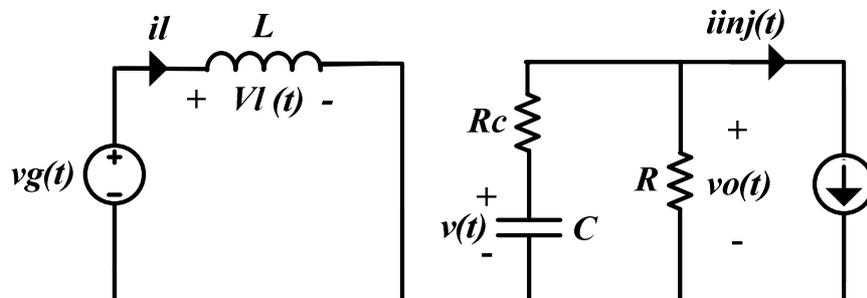


Figura 3 - Conversor *Boost* com chave em condução.

Fonte: Autoria própria.

Através das figuras, é possível observar que foi adicionada uma fonte de corrente. Ele injeta a corrente i_{inj} para os estágios do conversor de saída. Esta corrente é um sinal de entrada e é necessária para determinar a impedância de saída do conversor. A tensão de entrada do conversor é representado pela fonte de tensão v_g , e a tensão de saída representada por v_o . O indutor é representado por L , o diodo representado por D e o capacitor por C . A variável R representa a carga de saída o conversor e R_C a resistência interna do capacitor. Com a chave de potência em condução, a fonte fornece corrente para o indutor. Quando há a passagem de corrente sobre o indutor, um campo magnético é formado no núcleo do indutor. Quando uma tensão é aplicada a um indutor, as linhas de força desse campo se expandem, armazenando energia. Quando a tensão é desconectada do componente, essas linhas de campo se contraem, induzindo tensão nos terminais do indutor. Quando mais rápida for a contração desse campo, maior será a tensão induzida nos terminais do indutor. Dessa forma, quando maior for a variação de corrente, maior será a amplitude da tensão que aparecerá em seus terminais.

Assim, quando a chave de potência está em modo de condução, a fonte passa a carregar este indutor.

A tensão de saída, nesse modo de operação, é a tensão que está sobre o capacitor. Assim, nesse modo, o capacitor passa a ser a fonte de tensão para a carga. O diodo não conduz nesse modo, pois a tensão na saída do inversor é maior que a tensão que de entrada.

Quando a chave de potência do conversor está em corte, ou seja, não está conduzindo, sua topologia pode ser expressa conforme a Figura 4.

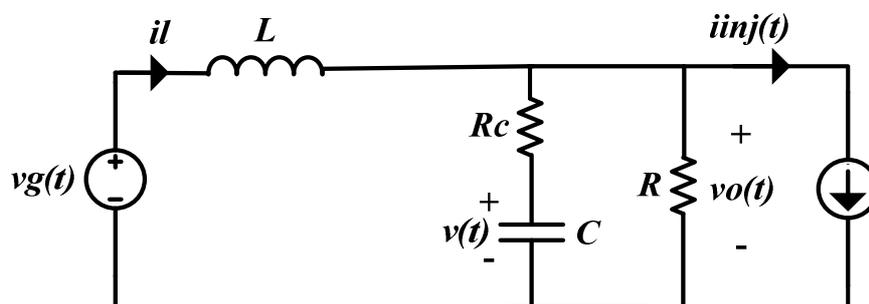


Figura 4 - Conversor *Boost* com chave em corte.

Fonte: Autoria própria.

Quando o conversor está operando nesta topologia, o indutor passa a descarregar sua energia armazenada e a fonte passa a fornecer tensão à saída do conversor. A tensão de saída é a somatória da tensão da fonte mais a tensão que o indutor está fornecendo no momento que está descarregando. Como essa tensão é superior à tensão de saída, o diodo passa a conduzir corrente. Uma parte dessa corrente é entregue diretamente à carga e outra parte passa a carregar o capacitor, que, no modo de operação anterior descrito, se tornará fonte fornecedora para a carga.

A modelagem matemática desse conversor foi desenvolvida de acordo com (JOHANSSON, 2003).

2.2.1 Chave de potência em condução

Para esse modo de operação, as seguintes equações podem ser encontradas:

$$\frac{di_L(t)}{dt} = \frac{1}{L} v_g(t) \quad (2.1)$$

$$\frac{dv(t)}{dt} = \frac{1}{C} \left(-\frac{v_o(t)}{R} - i_{inj}(t) \right) \quad (2.2)$$

$$v_o(t) = v(t) + R_C \left(-\frac{v_o(t)}{R} - i_{inj}(t) \right) \quad (2.3)$$

A equação (2.3) pode ser reorganizada como:

$$v_o(t) + \frac{R_C}{R} v_o(t) = v(t) - R_C i_{inj}(t) \quad (2.4)$$

$$v_o(t) = -\frac{v(t) - R_C i_{inj}(t)}{1 + R_C/R} i_{inj}(t) \quad (2.5)$$

$$v_o(t) = \frac{R}{R + R_C} v(t) - \frac{RR_C}{R + R_C} i_{inj}(t) \quad (2.6)$$

A equação (2.6) é usada para substituir (2.2).

$$\frac{dv(t)}{dt} = -\frac{1}{(R + R_c)C}v(t) + \frac{R_c}{(R + R_c)C}i_{inj}(t) - \frac{1}{C}i_{inj}(t) \quad (2.7)$$

A equação (2.7) é simplificada:

$$\frac{dv(t)}{dt} = -\frac{1}{(R + R_c)C}v(t) - \frac{R}{(R + R_c)C}i_{inj}(t) \quad (2.8)$$

Utilizando (2.1), (2.8) e (2.6), o seguinte sistema de espaço de estados é obtido:

$$\begin{cases} \frac{dx(t)}{dt} = A_1x(t) + B_1u(t) \\ y(t) = C_1x(t) + E_1u(t) \end{cases} \quad (2.9)$$

Onde:

$$x(t) = \begin{bmatrix} i_L(t) \\ v(t) \end{bmatrix} \quad (2.10)$$

$$u(t) = \begin{bmatrix} v_g(t) \\ i_{inj}(t) \end{bmatrix} \quad (2.11)$$

$$y(t) = v_o(t) \quad (2.12)$$

$$A_1 = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{(R + R_c)C} \end{bmatrix} \quad (2.13)$$

$$B_1 = \begin{bmatrix} \frac{1}{L} & 0 \\ 0 & -\frac{R}{(R + R_c)C} \end{bmatrix} \quad (2.14)$$

$$C_1 = \begin{bmatrix} 0 & \frac{R}{(R + R_c)} \end{bmatrix} \quad (2.15)$$

$$E_1 = \left[0 \quad -\frac{RR_C}{R + R_C} \right] \quad (2.16)$$

2.2.2 Chave de potência em condução

Operando nesta condição, as seguintes equações podem ser encontradas.

$$\frac{di_L(t)}{dt} = \frac{1}{L} (v_g(t) - v_o(t)) \quad (2.17)$$

$$\frac{dv(t)}{dt} = \frac{1}{C} \left(i_L(t) - \frac{v_o(t)}{R} - i_{inj}(t) \right) \quad (2.18)$$

$$v_o(t) = v(t) + R_C \left(i_L(t) - \frac{v_o(t)}{R} - i_{inj}(t) \right) \quad (2.19)$$

A equação (2.19) pode ser reorganizado como:

$$v_o(t) + \frac{R_C}{R} v_o(t) = v(t) + R_C (i_L(t) - i_{inj}(t)) \quad (2.20)$$

$$v_o(t) = -\frac{v(t) - R_C (i_L(t) - i_{inj}(t))}{1 + R_C/R} \quad (2.21)$$

$$v_o(t) = \frac{RR_C}{R + R_C} i_L(t) + \frac{R}{R + R_C} v(t) - \frac{RR_C}{R + R_C} i_{inj}(t) \quad (2.22)$$

A equação (2.22) é usado para substituir (2.18).

$$\frac{di_L(t)}{dt} = -\frac{RR_C}{(R + R_C)L} i_L(t) - \frac{R}{(R + R_C)L} v(t) + \frac{1}{L} v_g + \frac{RR_C}{R + R_C} i_{inj}(t) \quad (2.23)$$

$$\begin{aligned} \frac{dv(t)}{dt} = & \frac{1}{C}i_L(t) - \frac{R_C}{(R + R_C)C}i_L(t) - \frac{1}{(R + R_C)C}v(t) + \frac{R_C}{(R + R_C)C}i_{inj}(t) \\ & - \frac{1}{C}i_{inj}(t) \end{aligned} \quad (2.24)$$

A equação (2.24) é simplificada:

$$\frac{dv(t)}{dt} = \frac{R}{(R + R_C)C}i_L(t) - \frac{1}{(R + R_C)C}v(t) - \frac{R}{(R + R_C)C}i_{inj}(t) \quad (2.25)$$

Utilizando (2.23), (2.25) e (2.22), o seguinte sistema de espaço de estados é obtido:

$$\begin{cases} \frac{dx(t)}{dt} = A_2x(t) + B_2u(t) \\ y(t) = C_2x(t) + E_2u(t) \end{cases} \quad (2.26)$$

Onde:

$$x(t) = \begin{bmatrix} i_L(t) \\ v(t) \end{bmatrix} \quad (2.27)$$

$$u(t) = \begin{bmatrix} v_g(t) \\ i_{inj}(t) \end{bmatrix} \quad (2.28)$$

$$y(t) = v_o(t) \quad (2.29)$$

$$A_2 = \begin{bmatrix} -\frac{RR_C}{(R + R_C)L} & -\frac{R}{(R + R_C)L} \\ \frac{R}{(R + R_C)C} & -\frac{1}{(R + R_C)C} \end{bmatrix} \quad (2.30)$$

$$B_2 = \begin{bmatrix} \frac{1}{L} & \frac{RR_C}{(R + R_C)L} \\ 0 & -\frac{R}{(R + R_C)C} \end{bmatrix} \quad (2.31)$$

$$C_2 = \begin{bmatrix} \frac{RR_C}{R + R_C} & \frac{R}{(R + R_C)} \end{bmatrix} \quad (2.32)$$

$$E_2 = \begin{bmatrix} 0 & -\frac{RR_C}{R + R_C} \end{bmatrix} = E_1 \quad (2.33)$$

2.3 STATE-SPACE AVERAGING

O conversor trabalha com a variação de dois sistemas lineares invariantes no tempo, (2.9) e (2.26), durante o período de chaveamento, por isso parece com um sistema variante no tempo. Assim, o método de *State-Space Averaging* será utilizado para possibilitar a aproximação deste sistema em um sistema linear contínuo invariante no tempo (JOHANSSON, 2003).

O primeiro passo é calcular um sistema não linear invariante no tempo e a segunda etapa consiste na linearização do sistema.

$$\begin{cases} \frac{dx(t)}{dt} = (d(t)A_1 + (1 - d(t))A_2)x(t) + (d(t)B_1 + (1 - d(t))B_2)u(t) \\ y(t) = (d(t)C_1 + (1 - d(t))C_2)x(t) + (d(t)E_1 + (1 - d(t))E_2)u(t) \end{cases} \quad (2.34)$$

A equação (2.34) é uma aproximação do sistema variante no tempo e novas variáveis devem ser utilizadas. Para limitar o número de variáveis, isto não é feito. A razão cíclica, $d(t)$, é um sinal de entrada adicional em (2.34). Um novo vetor de entrada é, portanto, definido como:

$$u'(t) = \begin{bmatrix} u(t) \\ d(t) \end{bmatrix} \quad (2.35)$$

Isto não é feito em apresentações tradicionais de *State-Space Averaging*, onde o sinal de controle $d(t)$ é mantido separado dos sinais v_g e i_{inj} . No entanto, na teoria do sistema, todos os sinais de controle e perturbação são colocados em um vetor de entrada.

O sistema em (2.34) é um sistema não linear e invariante no tempo, pois existem produtos de dois sinais de entrada e é invariante no tempo já que todos os coeficientes não dependem do tempo. Um sistema não linear invariante no tempo pode ser descrito como:

$$\begin{cases} \frac{dx(t)}{dt} = f(x(t), u'(t)) \\ y(t) = g(x(t), u'(t)) \end{cases} \quad (2.36)$$

Uma linearização direta é aplicada, onde definimos os desvios a partir de um ponto de funcionamento como se segue:

$$x(t) = X + \hat{x}(t) \quad (2.37)$$

$$u'(t) = U' + \hat{u}(t) \quad (2.38)$$

$$y(t) = Y + \hat{y}(t) \quad (2.39)$$

As letras maiúsculas representam o ponto de operação (CC, em estado estacionário) e os valores com chapéu (^) denotam sinais de perturbação (AC). Suponha uma operação em um ponto de equilíbrio, ou seja:

$$\begin{aligned} f(x(t), u'(t)) &= 0 \\ x(t) &= X \\ u'(t) &= U' \end{aligned} \quad (2.40)$$

Os valores de saída são:

$$\begin{aligned} Y &= g(x(t), u'(t)) \\ x(t) &= X \\ u'(t) &= U' \end{aligned} \quad (2.41)$$

O sistema linearizado pode ser obtido através de:

$$\begin{cases} \frac{d\hat{x}(t)}{dt} = A'\hat{x}(t) + B'\hat{u}(t) \\ \hat{y}(t) = C'\hat{x}(t) + E'\hat{u}(t) \end{cases} \quad (2.42)$$

Onde:

$$A' = \left[\frac{\partial f}{\partial x} \right]$$

para $x(t) = X$
 $u'(t) = U'$

(2.43)

$$B' = \left[\frac{\partial f}{\partial u'} \right]$$

para $x(t) = X$
 $u'(t) = U'$

(2.44)

$$C' = \left[\frac{\partial f}{\partial x} \right]$$

para $x(t) = X$
 $u'(t) = U'$

(2.45)

$$E' = \left[\frac{\partial f}{\partial u'} \right]$$

para $x(t) = X$
 $u'(t) = U'$

(2.46)

A equação (2.42) é uma aproximação do sistema variante no tempo e novas variáveis devem ser utilizadas. Para limitar o número de variáveis, isto não é feito.

A equação (2.34) é um caso especial de (2.36). As equações (2.38) e (2.40) – (2.46) serão reescritas para este caso especial. As seguintes equações são obtidas se (2.38) for aplicada em (2.35):

$$u' = \begin{bmatrix} u(t) \\ d(t) \end{bmatrix} = \begin{bmatrix} U \\ D \end{bmatrix} + \begin{bmatrix} \hat{u}(t) \\ \hat{d}(t) \end{bmatrix}$$
(2.47)

As seguintes variáveis podem ser definidas:

$$d'(t) = 1 - d(t)$$
(2.48)

$$D' = 1 - D$$
(2.49)

A variável $d'(t)$ é equivalente à fração de tempo em que a chave de potência está desligada. D' é o ponto de operação de $d'(t)$. (2.40) e (2.41) serão reescritas utilizando (2.34):

$$\begin{cases} 0 = AX + BU \\ Y = CX + EU \end{cases} \quad (2.50)$$

Onde:

$$A = DA_1 + D'A_2 \quad (2.51)$$

$$B = DB_1 + D'B_2 \quad (2.52)$$

$$C = DC_1 + D'C_2 \quad (2.53)$$

$$E = DE_1 + D'E_2 \quad (2.54)$$

A equação (2.50) é reescrita como:

$$\begin{cases} X = -A^{-1}BU \\ Y = (-CA^{-1}B + E)U \end{cases} \quad (2.55)$$

A equações (2.43) – (2.46) são reformuladas utilizando (2.34):

$$A' = \left[\frac{\partial f}{\partial x} \right] = (d(t)A_1 + (1 - d(t))A_2) = A \quad (2.56)$$

para $x(t) = X$
 $u'(t) = U'$

$$B' = \left[\frac{\partial f}{\partial u'} \right] = \left[\frac{\partial f}{\partial u} \quad \frac{\partial f}{\partial d} \right] =$$

$$\left[d(t)B_1 + (1 - d(t))B_2 (A_1 - A_2)x(t) + (B_1 - B_2)u(t) \right] =$$

$$\left[B (A_1 - A_2)X + (B_1 - B_2)U \right] \quad (2.57)$$

para $x(t) = X$
 $u'(t) = U'$

$$C' = \left[\frac{\partial f}{\partial x} \right] = (d(t)C_1 + (1 - d(t))C_2) = C \quad (2.58)$$

para $x(t) = X$
 $u'(t) = U'$

$$\begin{aligned}
E' &= \left[\frac{\partial g}{\partial u'} \right] = \left[\frac{\partial g}{\partial u} \quad \frac{\partial g}{\partial d} \right] = \\
& \left[d(t)E_1 + (1 - d(t))E_2 (C_1 - C_2)x(t) + (E_1 - E_2)u(t) \right] = \\
& \left[E (C_1 - C_2)X + (E_1 - E_2)U \right] \\
& \text{para } x(t) = X \\
& u'(t) = U'
\end{aligned} \tag{2.59}$$

A equação (2.42) pode ser reescrita como:

$$\begin{cases} \frac{d\hat{x}(t)}{dt} = A\hat{x}(t) + B'\hat{u}'(t) \\ \hat{y}(t) = C\hat{x}(t) + E'\hat{u}'(t) \end{cases} \tag{2.60}$$

Os elementos B' e E' são definidos como:

$$B' = [B \quad B_d] \tag{2.61}$$

$$E' = [B \quad E_d] \tag{2.62}$$

Onde:

$$B_d = (A_1 - A_2)X + (B_1 - B_2)U \tag{2.63}$$

$$E_d = (C_1 - C_2)X + (E_1 - E_2)U \tag{2.64}$$

2.4 APLICANDO O STATE-SPACE AVERAGING

As seguintes equações são encontradas, utilizando as matrizes $A_1, A_2, B_1, B_2, C_1, C_2, E_1$ e E_2 , nas equações (2.51) – (2.54),

$$A = \begin{bmatrix} -D' \frac{RR_C}{(R + R_C)L} & -D' \frac{R}{(R + R_C)L} \\ D' \frac{R}{(R + R_C)C} & -\frac{1}{(R + R_C)C} \end{bmatrix} \tag{2.65}$$

$$B = \begin{bmatrix} \frac{1}{L} & D' \frac{RR_C}{(R + R_C)L} \\ 0 & -\frac{R}{(R + R_C)C} \end{bmatrix} \quad (2.66)$$

$$C = \left[D' \frac{RR_C}{R + R_C} \quad \frac{R}{(R + R_C)} \right] \quad (2.67)$$

$$E = \left[0 \quad -\frac{RR_C}{R + R_C} \right] = E_1 \quad (2.68)$$

As equações seguintes são obtidas se (2.50) for expandido:

$$0 = -\frac{RR_C D'}{(R + R_C)L} I_L - \frac{RD'}{(R + R_C)L} V + \frac{1}{L} V_g \quad (2.69)$$

$$0 = -\frac{RD'}{(R + R_C)C} I_L - \frac{1}{(R + R_C)C} V \quad (2.70)$$

$$V_o = \frac{RR_C D'}{(R + R_C)} I_L - \frac{R}{(R + R_C)} V \quad (2.71)$$

A equação (2.70) é simplificada para:

$$V = RD' I_L \quad (2.72)$$

A equação (2.71) é utilizada para reescrever (2.72).

$$V_o = \frac{R_C}{(R + R_C)} V + \frac{R}{(R + R_C)} V = V \quad (2.73)$$

A equação (2.71) é utilizada para reescrever (2.69).

$$0 = -\frac{R_C}{(R + R_C)} V - \frac{R}{(R + R_C)} D' V + V_g \quad (2.74)$$

$$\frac{V}{V_g} = \frac{R + R_C}{RD' + R_C} = \frac{R + R_C}{R - RD + R_C} = \frac{1}{1 - \frac{R}{R + R_C}D} \quad (2.75)$$

A tensão CC através do capacitor é nula e isso explica o resultado em (2.73). O valor médio da corrente de carga é igual ao valor médio da corrente do díodo. A corrente do díodo é igual à corrente do indutor durante a fração D' do tempo e igual a zero no caso contrário. O valor médio da corrente do díodo é, portanto, aproximadamente igual a $D'I_L$ isso explica o resultado em (2.72). A equação (2.75) indica a amplificação CC do conversor que é maior do que 1.

As equações (2.63) e (2.64) podem ser expandidas e simplificadas como:

$$B_d = \begin{bmatrix} \frac{RR_C}{(R + R_C)L} & \frac{R}{(R + R_C)L} \\ -\frac{R}{(R + R_C)C} & 0 \end{bmatrix} X + \begin{bmatrix} 0 & -\frac{RR_C}{(R + R_C)L} \\ 0 & 0 \end{bmatrix} U = \begin{bmatrix} \frac{RR_C}{(R + R_C)L} I_L + \frac{R}{(R + R_C)L} V \\ -\frac{R}{(R + R_C)C} \end{bmatrix} \quad (2.76)$$

$$E_d = \begin{bmatrix} -\frac{RR_C}{R + R_C} & 0 \end{bmatrix} X + 0U = \frac{RR_C}{R + R_C} I_L \quad (2.77)$$

Os elementos I_L e V das equações (2.76) e (2.77) são substituídas utilizando (2.72) e (2.75).

$$B_d = \begin{bmatrix} \frac{R_C}{(R + R_C)LD'} V + \frac{RD'}{(R + R_C)LD'} V \\ -\frac{1}{(R + R_C)CD'} V \end{bmatrix} = \begin{bmatrix} \frac{(RD' + R_C)V}{(R + R_C)LD'} \\ \frac{V}{(R + R_C)CD'} \end{bmatrix} = \begin{bmatrix} \frac{V_g}{LD'} \\ -\frac{V_g}{(RD' + R_C)CD'} \end{bmatrix} \quad (2.78)$$

$$E_d = \frac{R_c}{(R + R_c)D'} V = - \frac{R_c V_g}{(RD' + R_c)D'} \quad (2.79)$$

As equações (2.61) e (2.62) podem ser expandido como:

$$B' = \begin{bmatrix} \frac{1}{L} & \frac{RR_c D'}{(R + R_c)L} & \frac{V_g}{LD'} \\ 0 & -\frac{R}{(R + R_c)C} & -\frac{V_g}{(RD' + R_c)CD'} \end{bmatrix} \quad (2.80)$$

$$E' = \begin{bmatrix} 0 & -\frac{RR_c}{R + R_c} & -\frac{R_c V_g}{D'(RD' + R_c)} \end{bmatrix} \quad (2.81)$$

Todos os coeficientes do modelo CA (2.50) estão agora disponíveis.

2.5 EXTRAINDO A FUNÇÃO DE TRANSFERÊNCIA

A função de transferência do controle para a saída pode ser extraída a partir do sistema linearizado (2.60). Admitindo que o estado é zero inicialmente, a transformada de Laplace (2.60) é:

$$\begin{cases} s\hat{x}(s) = A\hat{x}(s) + B'\hat{u}'(s) \\ \hat{y}(s) = C\hat{x}(s) + E'\hat{u}'(s) \end{cases} \quad (2.82)$$

Para evitar novos nomes de variáveis, a transformada de Laplace de um sinal é denotado pelo mesmo nome que o sinal, por exemplo, $L\{v(t)\} = v(s)$, ainda que isso não é uma notação formalmente correta. A equação (2.82) é reescrita como:

$$\begin{cases} \hat{x}(s) = (sI - A)^{-1} B'\hat{u}'(s) \\ \hat{y}(s) = C\hat{x}(s) + E'\hat{u}'(s) \end{cases} \quad (2.83)$$

A primeira equação de (2.83) é escrito como:

$$\hat{x}(s) = \begin{bmatrix} \frac{RR_c D'}{(R+R_c)L} + s & \frac{RD'}{(R+R_c)L} \\ -\frac{RD'}{(R+R_c)C} & \frac{1}{(R+R_c)C} + s \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{L} & \frac{RR_c D'}{(R+R_c)L} & \frac{V_g}{LD'} \\ 0 & -\frac{R}{(R+R_c)C} & -\frac{V_g}{(RD'+R_c)CD'} \end{bmatrix} \hat{u}'(s) \quad (2.84)$$

A matriz inversa de (2.84) é calculada:

$$\hat{x}(s) = \frac{1}{\left(\frac{RR_c D'}{(R+R_c)L} + s\right)\left(\frac{1}{(R+R_c)C} + s\right) + \frac{R^2 D'^2}{(R+R_c)^2 LC}} \cdot \begin{bmatrix} \frac{1}{(R+R_c)C} + s & -\frac{RD'}{(R+R_c)L} \\ \frac{RD'}{(R+R_c)C} & \frac{RR_c D'}{(R+R_c)L} + s \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{L} & \frac{RR_c D'}{(R+R_c)L} & \frac{V_g}{LD'} \\ 0 & -\frac{R}{(R+R_c)C} & -\frac{V_g}{(RD'+R_c)CD'} \end{bmatrix} \hat{u}'(s) \quad (2.85)$$

A equação (2.85) é modificada para:

$$\hat{x}(s) = \frac{1}{\frac{RR_c D' + R^2 D'^2}{(R+R_c)^2 LC} + s \frac{L + RR_c CD'}{(R+R_c)^2 LC} + s^2} \cdot \begin{bmatrix} \frac{1}{(R+R_c)C} + s \frac{1}{L} & \frac{RR_c D'}{(R+R_c)^2 LC} + s \frac{RR_c D'}{(R+R_c)L} + \frac{R^2 D'}{(R+R_c)^2 LC} \\ \frac{RD'}{(R+R_c)LC} & \frac{R^2 R_c D'^2}{(R+R_c)^2 LC} - \frac{R^2 R_c D'}{(R+R_c)^2 LC} - s \frac{R}{(R+R_c)C} \\ \frac{V_g}{(R+R_c)LCD'} + s \frac{V_g}{LD'} + \frac{RD'V_g}{(R+R_c)(RD'+R_c)LCD'} \\ \frac{RV_g}{(R+R_c)LC} - \frac{RR_c D'V_g}{(R+R_c)(RD'+R_c)LCD'} - s \frac{V_g}{(RD'+R_c)CD'} \end{bmatrix} \hat{u}'(s) \quad (2.86)$$

$$\hat{x}(s) = \frac{1}{\frac{RD'(RD'+R_c)}{(R+R_c)} + s(L + RR_c CD') + s^2(R+R_c)LC} \cdot \begin{bmatrix} 1 + s(R+R_c)C & RD'(1 + sR_c C) & \frac{V_g}{D'} \left(1 + \frac{RD'}{RD'+R_c} + s(R+R_c)C\right) \\ RD' & -\frac{R^2 R_c DD'}{(R+R_c)} - sRL & \frac{V_g}{D'(RD'+R_c)} (R^2 D'^2 - s(R+R_c)L) \end{bmatrix} \hat{u}'(s) \quad (2.87)$$

Algumas funções de transferências podem ser deduzidas a partir de (2.87):

$$\frac{\hat{i}_L(s)}{\hat{d}(s)} = \frac{\frac{V_g}{D'} \left(1 + \frac{RD'}{RD' + R_c} + s(R + R_c)C \right)}{\frac{RD'(RD' + R_c)}{(R + R_c)} + s(L + RR_c CD') + s^2(R + R_c)LC} \quad (2.88)$$

$$\frac{\hat{v}(s)}{\hat{d}(s)} = \frac{\frac{V_g}{D'(RD' + R_c)} (R^2 D'^2 - s(R + R_c)L)}{\frac{RD'(RD' + R_c)}{(R + R_c)} + s(L + RR_c CD') + s^2(R + R_c)LC} \quad (2.89)$$

A segunda equação de (2.83) é expandido:

$$\hat{y}(s) = \left[\frac{RR_c}{R + R_c} D' \quad \frac{R}{R + R_c} \right] \hat{x}(s) + \left[0 \quad -\frac{RR_c}{R + R_c} \quad -\frac{R_c V_g}{(RD' + R_c)D'} \right] \hat{u}'(s) \quad (2.90)$$

A função de transferência para controle de saída é obtida combinando as equações (2.90) (2.88) (2.89):

$$\begin{aligned} \frac{\hat{v}_o(s)}{\hat{d}(s)} &= \frac{RR_c D'}{R + R_c} \frac{\hat{i}_L(s)}{\hat{d}(s)} + \frac{R}{R + R_c} \frac{\hat{v}(s)}{\hat{d}(s)} - \frac{R_c V_g}{(RD' + R_c)D'} = \\ &= \frac{\frac{V_g}{(RD' + R_c)D'} (R^2 D'^2 - s(R + R_c)L)(1 + sR_c C)}{\frac{RD'(RD' + R_c)}{(R + R_c)} + s(L + RR_c CD') + s^2(R + R_c)LC} \end{aligned} \quad (2.91)$$

Assim, a equação (2.91) é a função de transferência do sistema utilizado no projeto deste controlador.

2.6 SIMULAÇÃO DO CONVERSOR *BOOST*

Este trabalho visa a melhoria do controle do conversor *boost* que já foi projetado. Dessa forma, a Tabela 1 apresenta as características do conversor.

Tabela 1 - Parâmetros do conversor *boost*.

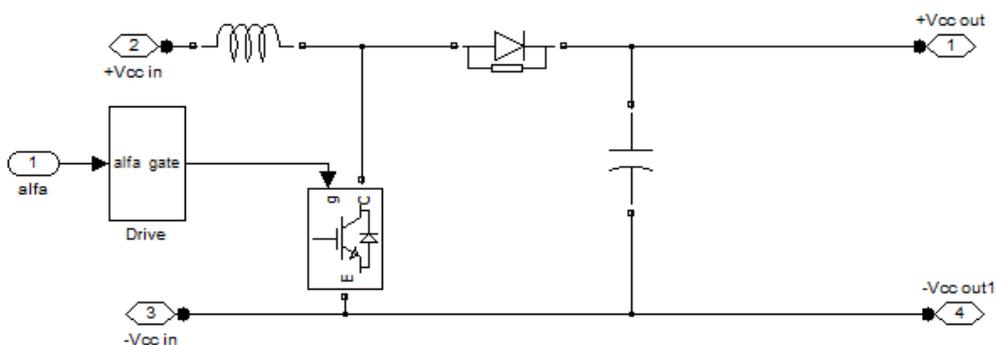
Tensão de entrada (VCC)	50 a 350 V
Tensão de saída (VCC)	400 V
Indutor	4mH
Resistência do capacitor	desprezível
Capacitor	940µF
Frequência de chaveamento	30kHz
Potência do aerogerador	100 a 3 kW

Fonte: Autoria própria.

A relação de tensão de saída pela razão cíclica do conversor *boost* pode ser descrita por:

$$D = 1 - \frac{V_i}{V_o} \quad (2.92)$$

Como forma de validar o modelo que foi simulado, foi escolhida uma tensão de entrada de 200V. Assim, utilizando a equação (2.92), o valor de encontrado é $D = 0.5$. A Figura 5 apresenta o modelo do conversor *boost* simulado.

**Figura 5 - Conversor *Boost* simulado no Matlab/Simulink.**

Fonte: Autoria própria.

Dessa forma, considerando uma tensão de entrada fixa de 200V e considerando o sistema operando em malha aberta, a resposta do sistema é apresentada conforme a Figura 6.

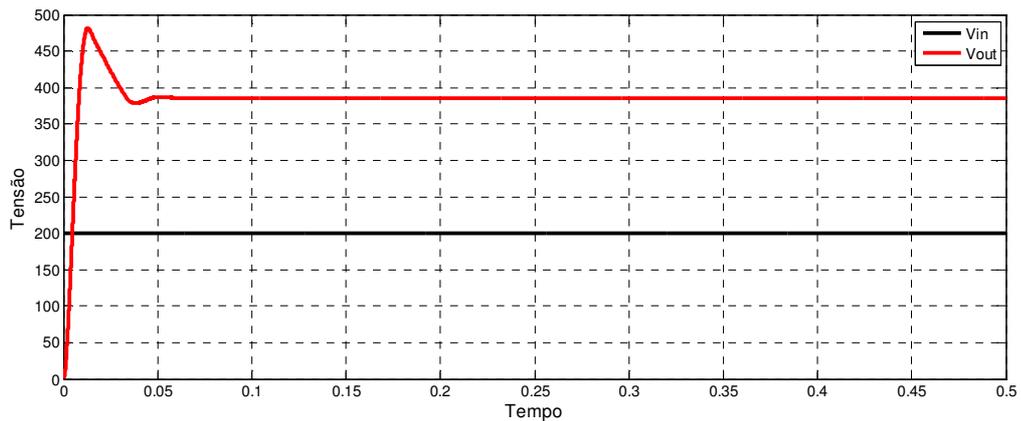


Figura 6 - Tensão de saída pela tensão de entrada em malha aberta.

Fonte: Autoria própria.

É possível perceber através da Figura 6 que o sistema está funcionando de forma correta, entregando na saída uma tensão proporcional ao valor de D calculado conforme equação (2.92). É possível perceber um erro em regime permanente, ocasionado pelas perdas internas no circuito, como resistência do indutor e resistência da chave. Assim, é possível afirmar que o sistema está simulado de forma a representar aproximadamente um sistema real.

Entretanto, para a representação aproximada da situação real, é necessário encontrar um controlador capaz de fazer com que o sistema opere em malha fechada, para compensar o erro em regime e para compensar a tensão de saída para tensões de entrada não fixas.

2.7 ÍNDICE DE DESEMPENHO

Um índice de desempenho pode ser calculado e utilizado para medir o desempenho de um sistema. A medida quantitativa do desempenho de um sistema é necessária para a operação de sistemas de controle adaptativos, para a otimização paramétrica de sistemas de controle e para o projeto ótimo de sistemas (DARF E BISHOP, 2001). Dessa forma, atribuir o desempenho do sistema através de um índice permite que a proposta do trabalho possa ser verificada de forma mais eficaz.

Uma técnica de medir o desempenho de um sistema é calculando o erro entre o valor desejado do sistema que o controle está atuando, com o sinal de saída do sistema. Um sistema pode ser considerado ótimo quando sua saída não apresenta nenhum erro em relação à saída desejada. O acúmulo do erro pode representar um sistema operando fora da otimização.

Entretanto, os sistemas apresentam um modo de operação transitório que sempre acarreta em um erro em relação ao valor de saída desejado. Dessa forma, o índice utilizado neste trabalho é o índice ITAE (*Integral of Time by Absolute Error*), que reduz a contribuição de grandes erros iniciais. Este índice é obtido pela integral do tempo multiplicado pelo valor absoluto do erro. A equação (2.93) apresenta o índice de desempenho ITAE.

$$ITAE = \int_0^T t|e(t)|dt \quad (2.93)$$

Todas as simulações comparativas entre os métodos de controle utilizam este índice para medir sua eficiência.

2.8 PROJETO DO CONTROLADOR

O controlador recebe um sinal de referência e compara com o sinal de saída do conversor. Assim, ele consegue encontrar valores para ação para atuar na chave de potência a fim de fazer com que o sistema responda da forma desejada.

Diversas formas de obtenção do controlador estão disponíveis. Para este trabalho, o projeto utiliza o método lugar das raízes para o projeto do controlador. O método do lugar das raízes consiste em traçar a resposta do sistema em malha fechada e conhecer os pólos e zeros de sua função de transferência. Com o auxílio computacional, é possível traçar esses pontos e obter os melhores pontos para compensação.

Para o estudo do lugar das raízes, é necessário conhecer a função de transferência do sistema.

Aplicando os dados da Tabela 1 na equação (2.91), é possível encontrar a função de transferência que representa o circuito do conversor *boost*. Utilizando o software Matlab, a função de transferência encontrada é apresentada em (2.94).

$$G(s) = \frac{-3,2s + 18670}{0,0003509s^2 + 0.004s + 0.25} \quad (2.94)$$

Utilizando a ferramenta *rltool* do Matlab, o lugar das raízes do sistema é apresentado na Figura 7.

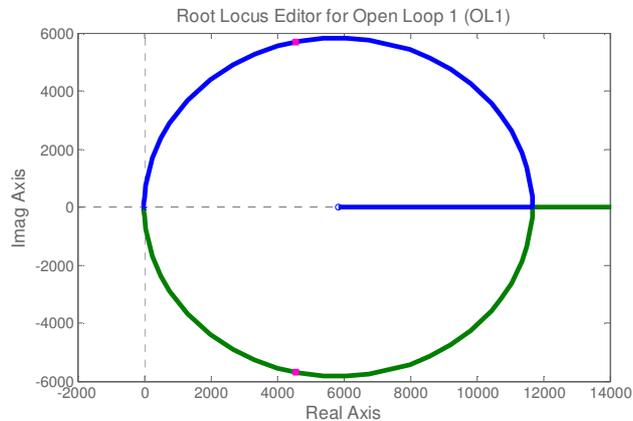


Figura 7 - Lugar das raízes do sistema.

Fonte: Autoria própria.

Através da figura, é possível perceber que o zero do sistema em malha fechada está no semi-plano direito, fazendo com que o sistema seja instável. O controlador deve compensar os polos e os zeros do sistema, de forma que ele consiga ser estável no ponto de operação.

Vale lembrar que este lugar das raízes foi encontrado a partir de uma entrada de 200V. Dessa forma, para entradas de tensão diferentes deste valor, o sistema controlador não estará projetado de forma correta, uma vez que a função de transferência muda e, dessa forma, os polos e zeros do sistema também mudarão.

Utilizando a ferramenta *rltool* do Matlab, o compensador foi encontrado. Para que o controlador seja capaz de compensar o sistema, é necessária a alocação de um polo na origem, que fará o papel de integrador, um zero conjugado para compensar os polos conjugados do sistema e mais um polo no semi-plano esquerdo, fazendo com que o sistema possa ter um ganho maior. Este procedimento permitiu que o sistema que antes estava instável se tornasse estável.

O compensador encontrado atendendo esses requisitos é expresso em (2.95).

$$C(s) = \frac{0,0001752s^2 + 0,00216s + 0,1401}{s^2 + 139,8s} \quad (2.95)$$

O projeto do controlador foi executado em tempo contínuo e toda a simulação foi executada em tempo discreto. Para isso, a discretização do sistema foi executado utilizando o método *Tustin*. O sistema pode ser expresso pelo diagrama de blocos, apresentado na Figura 8.

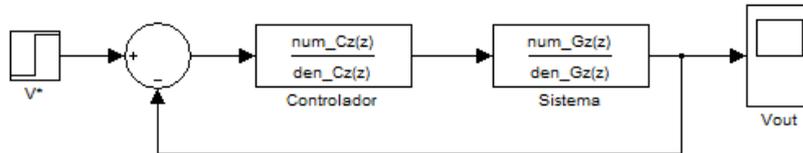


Figura 8 - Diagrama de blocos do sistema.

Fonte: Autoria própria.

A entrada V^* representa a tensão de referência do sistema, o bloco Controlador representa o controlador encontrado no projeto e o bloco Sistema representa o sistema através de sua função de transferência. A realimentação é unitária.

A Figura 9 apresenta a resposta do sistema no diagrama de blocos. A resposta em preto apresenta a tensão de entrada do sistema e, em vermelho, a tensão de saída do sistema com o controlador.

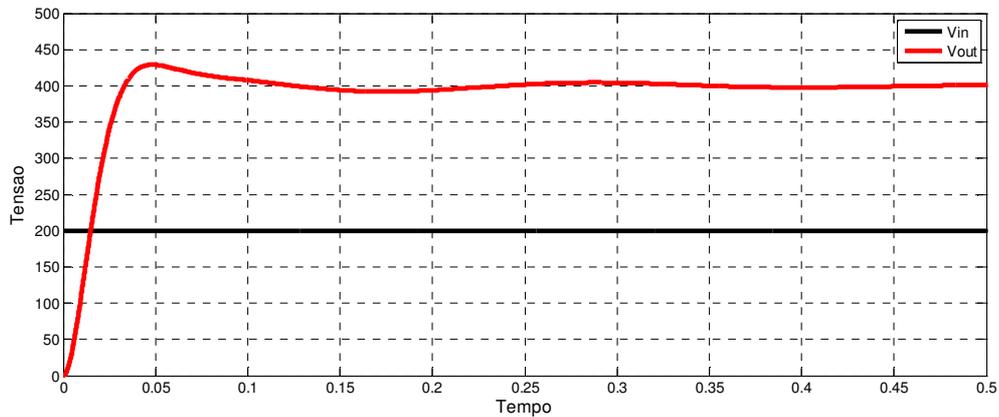


Figura 9 - Resposta do sistema ao degrau unitário.

Fonte: Autoria própria.

Com o sistema controlado pelo controlador encontrado no projeto, a resposta do sistema apresenta a correção do erro em regime permanente, apresentado na simulação com o sistema e malha aberta. Entretanto, na simulação em diagrama de blocos, a dinâmica do sistema não é corretamente representada, trazendo a necessidade da simulação com o circuito.

Para a simulação, alguns pontos devem ser levados em consideração. Primeiramente a realimentação do sistema deve ser feita utilizando um transdutor de tensão para a medição da tensão de saída e a representação de uma tensão de referência no controlador. Além disso, é necessário considerar o ganho do PWM.

O conversor está conectado a uma carga com valor que varia em razão da tensão de entrada. Isso porque o conversor deverá fornecer uma tensão de saída de 400V com corrente que será proporcional à tensão de entrada. Assim, a potência fornecida pelo conversor será proporcional à potência da entrada do mesmo. Quando a tensão de entrada no conversor for de 350V, o aerogerador deverá entregar a potência nominal de 3kW. Entretanto esta potência será reduzida quando houver variação na tensão de entrada do conversor. A saída do controlador é a razão cíclica D , que representa o tempo que a chave permanecerá em condução. A frequência de chaveamento é de 30kHz. Com essa proposta, a Figura 10 apresenta o sistema simulado no software *Simulink*. O bloco que representa a fonte de corrente variável que está conectada no lugar da carga emula a corrente de carga, em relação a tensão de entrada e tensão de saída do conversor. Assim, é possível simular uma potência proporcional que o sistema está fornecendo, em relação à potência de entrada.

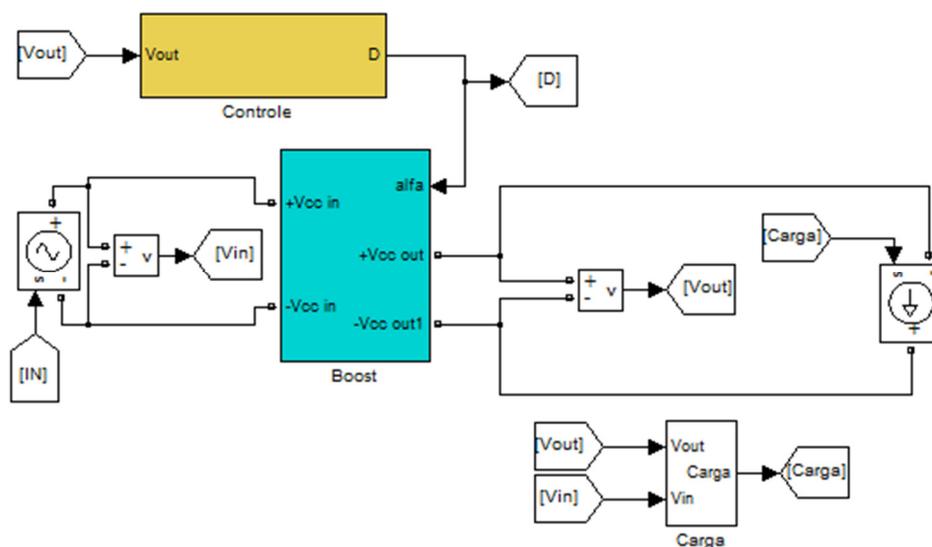


Figura 10 - Conversor *Boost* controlado simulado no Matlab/Simulink .

Fonte: Autoria própria.

A Figura 11 apresenta o bloco de Controle.

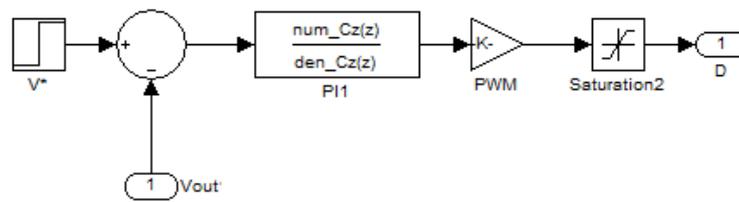


Figura 11 - Controle aplicado ao conversor *Boost* .

Fonte: Autoria própria.

A simulação apresentada na Figura 10 permitiu a validação do sistema através de uma série de possibilidades de tensões de entrada no conversor. Todas as simulações foram executadas variando a carga conectada na saída do conversor.

Dessa forma, a Figura 12 apresenta a resposta do sistema controlado para uma entrada de 200V. O projeto do controlador foi executado a partir da função de transferência obtida para esta tensão de entrada. A curva em preto representa a tensão de entrada do conversor e a curva em vermelho representa a tensão de saída.

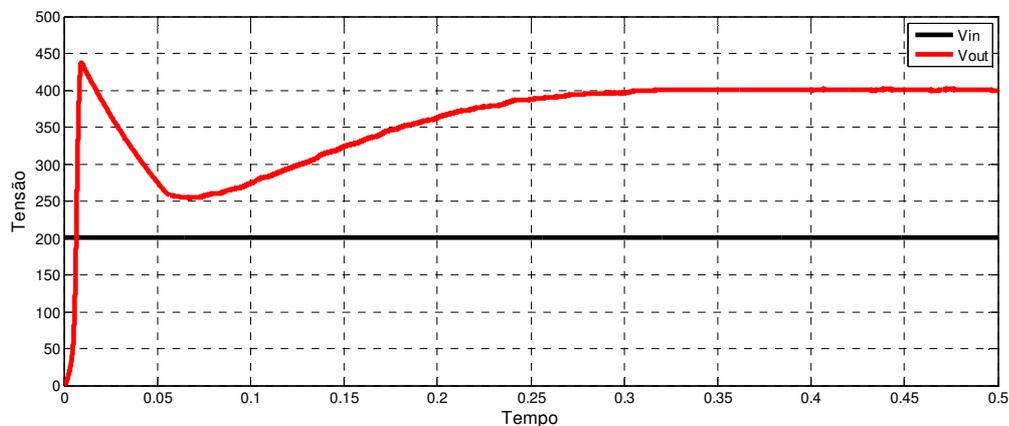


Figura 12 - Tensão de saída do conversor para uma tensão de entrada de 200V.

Fonte: Autoria própria.

É possível perceber que o conversor *boost* está sendo controlado, fornecendo a tensão de saídas desejada de 400V. Entretanto, devido a dinâmica do circuito e a resposta do controlador, é possível perceber o tempo de assentamento da tensão de saída em torno de 0.3s.

As Figuras 13 e 14 apresentam, respectivamente, a tensão de saída do conversor para uma tensão de entrada de 350V e 280V. É possível perceber que o tempo de assentamento é

praticamente o mesmo para a simulação anterior, porém o sobressinal é proporcional à tensão de entrada. Quanto maior a tensão de entrada, maior o sobressinal.

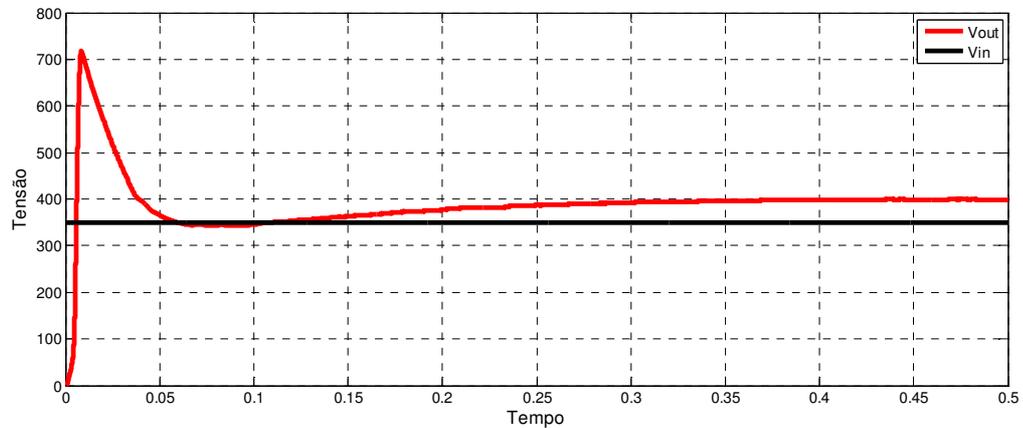


Figura 13 - Tensão de saída do conversor para uma tensão de entrada de 350V.

Fonte: Autoria própria.

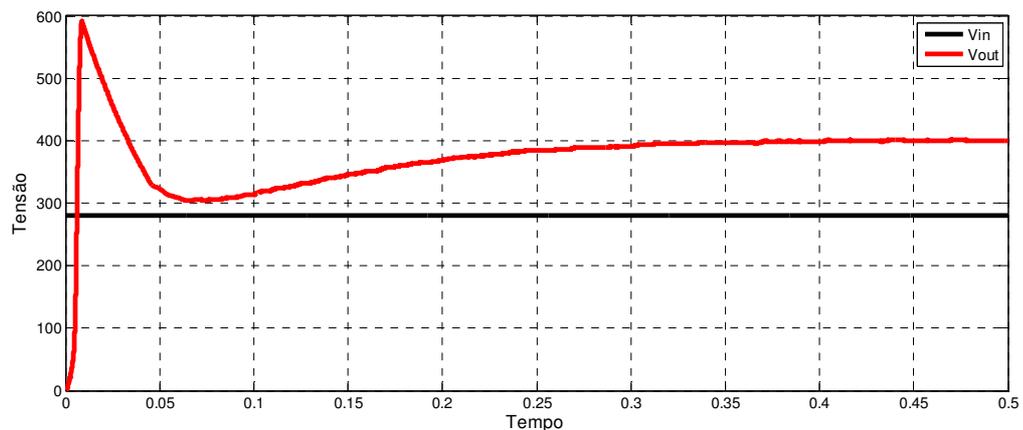


Figura 14 - Tensão de saída do conversor para uma tensão de entrada de 280V.

Fonte: Autoria própria.

A Figura 15 apresenta a tensão de saída do conversor para uma tensão de entrada de 210V. Novamente é possível perceber que o tempo de assentamento da tensão de saída, bem como o sobressinal que diminuiu em proporção da tensão de entrada.

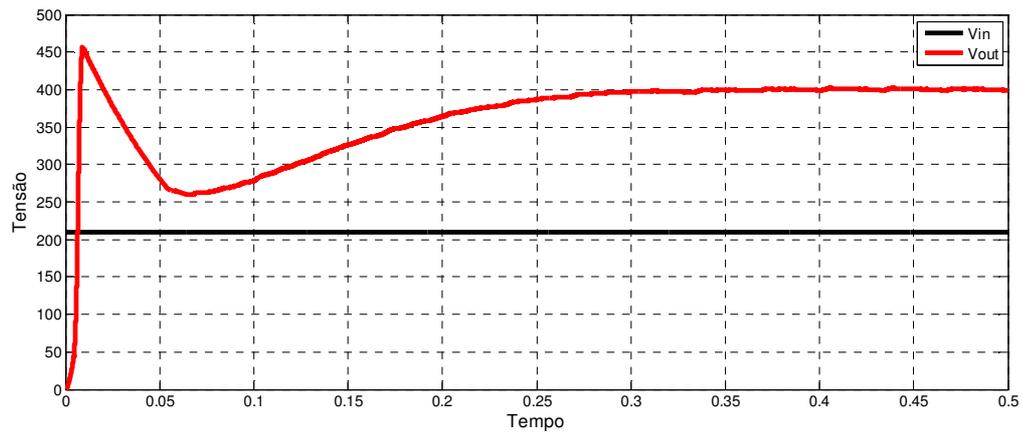


Figura 15 - Tensão de saída do conversor para uma tensão de entrada de 210V.

Fonte: Autoria própria.

As Figuras 16, 17 e 18 representam a saída do conversor para as tensões de entrada de 140V, 70V e 50V. O tempo de assentamento da tensão de saída para a tensão de entrada de 140V é o mesmo para as simulações anteriores. Entretanto, para as outras simulações, o tempo de assentamento da tensão de saída foi maior. Nestas condições, é possível perceber que não há o sobressinal, como houve nos casos anteriores.

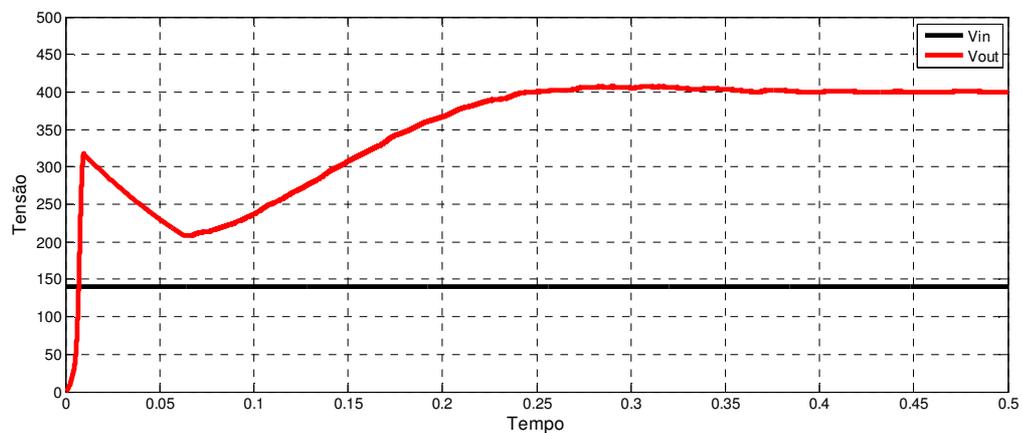


Figura 16 - Tensão de saída do conversor para uma tensão de entrada de 140V.

Fonte: Autoria própria.

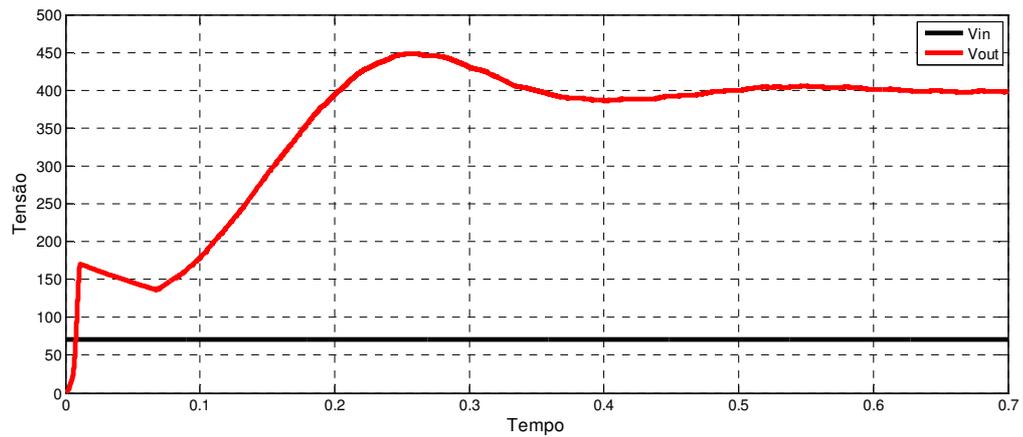


Figura 17 - Tensão de saída do conversor para uma tensão de entrada de 70V.

Fonte: Autoria própria.

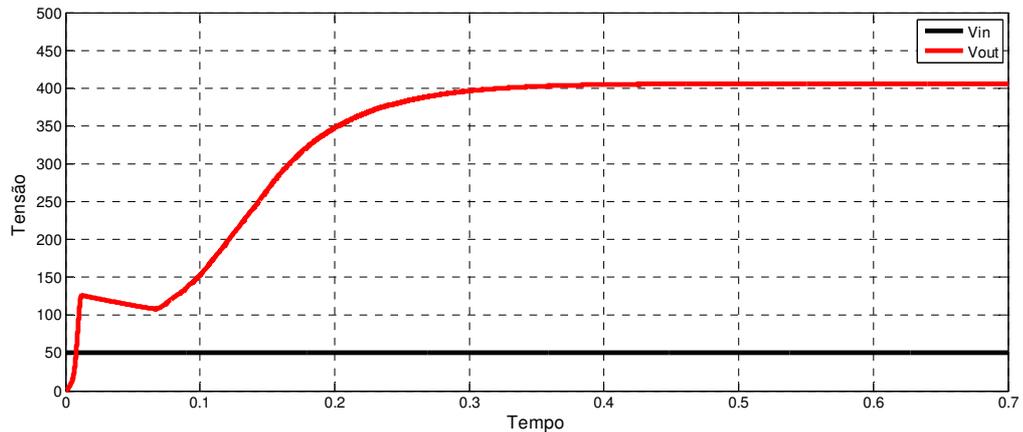


Figura 18 - Tensão de saída do conversor para uma tensão de entrada de 50V.

Fonte: Autoria própria.

2.9 CONCLUSÃO

Este capítulo apresentou a modelagem do conversor *boost* e o projeto do controlador utilizando o método do lugar das raízes.

Também foram apresentadas as simulações que permitiram a validação do modelo e do controle do sistema.

É possível observar que o comportamento da tensão de saída durante o regime transitório sofre alterações conforme variação na tensão de entrada, pois o projeto é baseado na função de transferência do sistema que, para a obtenção, é utilizado os valores de tensão de

entrada. Assim, para valores de entrada diferentes da tensão utilizada para a obtenção dos valores do controlador, o sistema continua controlando, mas apresenta diferença no modo de operação transitório.

A tentativa de projetar um controlador mais rápido, para compensar o tempo de assentamento para tensões mais baixas, acarreta em sobressinal maior para tensões de entrada elevadas. O mesmo ocorre quando se deseja projetar um controlador que proporcione um sobressinal menor para as tensões elevadas. A consequência é um tempo de assentamento maior quando a tensões de entrada são inferiores.

3 REDES NEURAIIS ARTIFICIAS

3.1 INTRODUÇÃO

A pesquisa de sistemas computacionais com capacidade de aprendizado e adaptação é fruto de trabalho de pesquisadores há mais de 50 anos. Entretanto, essa linha de pesquisa tornou-se mais forte a partir dos anos de 1990, devido à evolução computacional que permitiu simulações e implementações das propostas pesquisadas anteriormente (NUNES; SPATTI; FLAUZINO, 2010).

Dentre os sistemas de inteligência artificial, destacam-se as Redes Neurais Artificiais (RNA), Lógicas Nebulosas (Sistemas *Fuzzy*) e Algoritmos Genéticos (AG). O ponto em comum entre essas técnicas está na capacidade que o computador desenvolve, a partir do desenvolvimento de algoritmos, para o aprendizado do comportamento de sistemas.

As RNA possuem um princípio de funcionamento baseado nos neurônios dos seres vivos, onde a capacidade de aprendizado se dá através do processamento em paralelo de uma série de unidades de simples processamento.

As aplicações de RNA podem ser feitas para a predição do comportamento de um processo, classificações de padrões, aproximador universal de funções, entre outras.

Sua área de aplicabilidade está em crescente desenvolvimento, abrangendo áreas da engenharia, médica, farmacêutica, economia, genética, entre outras (NUNES; SPATTI; FLAUZINO, 2010).

Com o aumento da capacidade de processamento dos computadores, as áreas de aplicabilidade são ilimitadas, sobretudo onde a modelagem do sistema pode ser impossível ou extremamente trabalhosa, devido à não-linearidades ou desconhecimentos da planta.

3.2 O NEURÔNIO ARTIFICIAL

Uma rede neural artificial é composta por uma série de elementos simples, denominados neurônio artificial. Esses neurônios são desenvolvidos baseados na estrutura do neurônio de um ser vivo, onde sua saída é a somatória das entradas, ponderadas por um peso

sináptico. Dessa forma, o conjunto trabalhando em paralelo de uma série de neurônios é capaz de fazer com que uma saída da rede neural possa representar um sistema a partir de suas entradas.

A Figura 19 representa a estrutura de um neurônio artificial.

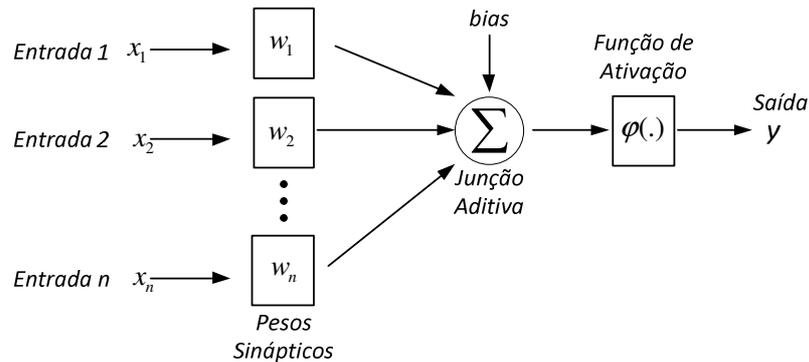


Figura 19 - Neurônio Artificial.

Fonte: Autoria própria.

A estrutura básica de um neurônio consiste no conjunto de valores de entrada $\{x_1, x_2, \dots, x_n\}$ que são ponderadas pelos pesos sinápticos $\{w_1, w_2, \dots, w_n\}$. Na sequência, é efetuada a somatória dos valores obtidos pelo processo anterior e por um elemento de ativação, denominado limiar de ativação. Esse valor de limiar de ativação define se a saída do neurônio será ou não ativada. Na sequência, uma função de ativação é utilizada para definir a saída do neurônio.

A equação (3.1) descreve matematicamente um neurônio artificial.

$$y = \varphi \left(\sum_{j=1}^n w_j \cdot x_j + b \right) \quad (3.1)$$

onde:

j é o número de entradas do neurônio;

w_j é o peso associado a j -ésima entrada;

b é o limiar de ativação associado ao neurônio;

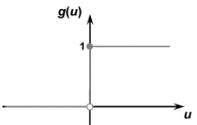
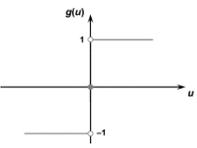
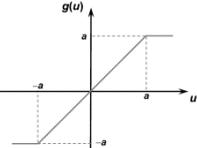
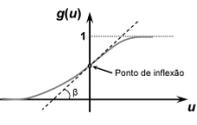
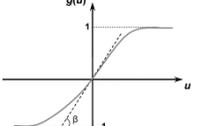
x_j é a j -ésima entrada do neurônio;

$\varphi(\cdot)$ é a função de ativação do neurônio;

y é a saída do neurônio.

A função de ativação tem como objetivo definir o tipo de saída que o neurônio irá fornecer baseado no sinal de ativação vindo do processo de soma anterior. A função de ativação está diretamente relacionada com o tipo de saída pretendido da rede neural. A Tabela 2 apresenta as principais funções de ativação.

Tabela 2 – Funções de ativação do neurônio artificial.

Tipo	Equação	Função
Degrau	$\varphi(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases}$	
Degrau Bipolar	$\varphi(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases}$	
Rampa Simétrica	$\varphi(u) = \begin{cases} 1, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < -a \end{cases}$	
Logística	$\varphi(u) = \frac{1}{1 + e^{-\beta \cdot u}}$	
Tangente Hiperbólica	$\varphi(u) = \frac{1 - e^{-\beta \cdot u}}{1 + e^{-\beta \cdot u}}$	

Fonte: Autoria própria.

3.3 ARQUITETURA DAS REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais possuem uma ilimitada quantidade de arranjos de seus neurônios. Os neurônios são arranjados de maneira a formar um grupo, por onde as informações podem ser processadas, gerando sinais de saídas esperadas no projeto. Entretanto, de uma maneira geral, as redes neurais artificiais podem ser divididas em três partes. Cada uma dessas partes é denominada de camadas, e não classificadas conforme a localidade onde se encontram.

A primeira camada de neurônios é denominada camada de entrada. Esta camada recebe as informações da entrada da rede, processa conforme seu padrão de funcionamento e sua saída são conectados à segunda camada.

A segunda camada é denominada camada intermediária ou camada oculta. Essa denominação se deve ao fato dessa camada conter uma série de neurônios que estão localizados entre a primeira camada e a última camada. O número de neurônios dessa camada pode variar conforme a necessidade da rede, constatado durante o processo de aprendizagem e validação da rede. Em geral, os dados de entrada dessa camada são os sinais vindos da camada de entrada e sua saída é a informação que será enviada para a camada de saída. Assim, os dados processados nessa camada são dados de processamento da rede, não representando informações que possam ser utilizadas diretamente. Por isso, sua denominação também de camada oculta. Praticamente, todo o processamento da rede neural é executado nessa camada.

A terceira e última camada é denominada Camada de saída. Nesta camada, o número de neurônios é o número de saída que a rede possui e sua função é coletar os dados da camada anterior, fazer suas ponderações e fornecer o sinal de saída da rede.

A Figura 20 apresenta essa topologia de rede.

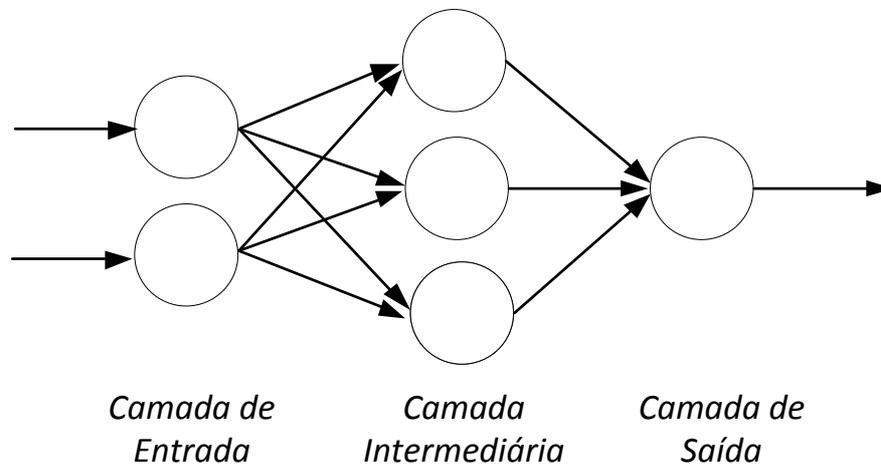


Figura 20 - Camadas de uma Rede Neural Artificial.

Fonte: Autoria própria.

Além dessa topologia, também é possível encontrar redes com mais de uma camada intermediária. A quantidade de camadas e a quantidade de neurônios que cada camada deverá possuir devem ser levantadas pelo responsável pelo projeto no momento do treinamento e validação da rede.

As redes neurais artificiais possuem também arquiteturas, que definem a forma como os neurônios estão interligados e a disposição de seus neurônios. Essas arquiteturas podem ser divididas em redes *feedforward* (alimentação à frente) de camada simples, redes *feedforward* de camadas múltiplas, redes recorrentes e redes reticuladas.

Todos os modos de operação da rede podem ser encontrados em (NUNES; SPATTI; FLAUZINO, 2010). Para elaboração do projeto proposto, a utilização da arquitetura *feedforward* multicamadas foi utilizado.

3.3.1 Redes *feedforward* multicamadas

A arquitetura das redes *feedforward* multicamadas são compostas pela camada de entrada, camada intermediária e camada de saída de neurônios. As principais aplicações para esse tipo de rede estão relacionadas com aproximação de funções, classificação de padrões, identificação de sistemas de otimização, robótica, controle de processos, etc. (NUNES; SPATTI; FLAUZINO, 2010).

A camada de entrada é composta pelas entradas da rede. A quantidade depende exclusivamente do número de entradas da rede. A camada intermediária possui números de neurônios diferentes do número de entradas. O número de neurônios está relacionado com o tipo de sinal que a rede neural irá trabalhar que depende do número de entradas, tipo de dados utilizado para o aprendizado, qualidade dos dados, entre outros. Por fim a camada de saída terá o mesmo número de neurônios correspondentes com o número de saídas que a rede possui.

Os principais tipos de redes pertencentes a essa arquitetura são as redes *Perceptron* Multicamadas e as redes de base radial (*radial basis function* - RBF).

3.4 TREINAMENTO E ASPECTO DE APRENDIZADO DAS REDES NEURAIAS

Uma rede neural é capaz de aprender o funcionamento de determinado sistema, sem que haja a necessidade de se modelar o problema proposto. Isso porque o seu processo de aprendizado se deve ao fato de apresentar à rede um padrão de funcionamento de determinado

sistema e ela é capaz de aprender seu funcionamento baseado nas informações que são dispostas para o aprendizado.

Uma grande vantagem na utilização de uma RNA, é que, a partir do aprendizado para determinados pontos de operação, a rede consegue generalizar soluções para pontos de operação que não foram apresentados durante o treinamento. Portanto, para que a rede possa aprender o funcionamento de determinado sistema é necessário apresentar os padrões de funcionamento em alguns pontos de operação.

Assim, em geral, a partir dos dados de funcionamento de um determinado sistema, aproximadamente 60% a 90% desses padrões de funcionamento são apresentados à rede para o aprendizado e de 10% a 40% dos dados de funcionamento da rede devem ser utilizados para a validação da rede. A validação consiste em apresentar padrões que não foram utilizados no processo de aprendizagem da rede e comparar os dados obtidos pela rede com os dados reais. O erro não deve ser superior a um valor estipulado em projeto. Se o erro for menor que este valor, a rede é considerada como treinada para a operação.

Uma técnica de aprendizagem utilizado em RNA é o aprendizado supervisionado. Essa técnica de aprendizagem consiste em fornecer para a rede o conjunto de informações de entrada e fornecer também a saída desejada para aquele padrão de entradas. A rede então passa a observar os valores de entradas e realiza o ajuste dos pesos dos neurônios para que a saída possa gerar um valor próximo ao desejado. O ajuste é realizado continuamente até que a saída obtida pela rede consiga obter a saída desejada. Essa saída desejada, em geral, é um valor próximo do valor desejado, porém dentro de uma margem de erro estipulada no projeto do sistema.

Portanto, para que a rede possa aprender o funcionamento de determinado sistema é necessário apresentar os padrões de funcionamento em alguns pontos de operação.

A rede utilizada neste trabalho é a Rede *Perceptron* Multicamadas. Essa possui um algoritmo de treinamento próprio e tem a capacidade de generalizar sua saída como um aproximador de funções, ou seja, sua saída será uma função próxima da saída desejada durante seu treinamento.

3.5 REDE *PERCEPTRON* MULTICAMADAS

As redes *Perceptron* multicamadas possuem uma arquitetura básica composta por uma camada de neurônios de entrada, pelo menos uma camada de neurônios escondidas e uma

camada de dados de saída. Cada neurônio que compõe essa rede é capaz de identificar padrões que sejam linearmente separáveis. Dessa forma, o conjunto de neurônios composto pela *Perceptron* é capaz de identificar todos os padrões de funcionamento de um sistema, tornando a rede uma das mais versáteis.

A rede *Perceptron* multicamadas pertence à arquitetura *feedforward*, onde os dados percorrem todos os neurônios da rede, fornecendo assim o valor de saída. Nesse caso, a saída dos neurônios da camada de entrada será a entrada para os neurônios da camada intermediária. Ainda, a saída dos neurônios da camada intermediária será a entrada dos neurônios da camada de saída. Por fim, a saída dos neurônios da última camada é o sinal de saída da rede. Como o sentido dos sinais percorrem os neurônios rumo à saída, sem retornar, o sistema é considerado *feedforward*. Seu treinamento é do tipo supervisionado, ou seja, durante o treinamento, as saídas desejadas para determinado padrão de entrada são fornecidas para que o ajuste dos pesos sinápticos possa ser efetuado, a fim de que a rede consiga obter a mesma saída desejada.

A camada intermediária desse tipo de rede é responsável por extrair a maior quantidade de informações da rede. O número de neurônios dessa camada está relacionado com a complexidade do sistema a ser aprendido pela rede.

O algoritmo utilizado para esta rede no processo de aprendizado é denominado *backpropagation* ou algoritmo de retropropagação do erro.

3.5.1 Algoritmo de aprendizado *backpropagation*

O procedimento adotado para o treinamento da rede *Perceptron* é o algoritmo *backpropagation*. Esse algoritmo possui duas fases de operação bastante distintas. A primeira fase de operação consiste no processo *forward* (propagação a diante). Nessa fase, os sinais de entrada são inseridos na rede e são ponderados por todos os neurônios, até que seja fornecido o sinal de saída. O sinal de saída será comparado com o sinal desejado e, havendo um erro entre eles, será calculado o erro entre a saída desejada e a saída encontrada pela rede.

A segunda fase de operação é composta pelo processo *backpropagation* (propagação reversa). Nessa fase, o erro calculado entre a saída desejada e a saída obtida pela rede será utilizada para a alteração dos pesos sinápticos de cada neurônio. O peso de cada neurônio será recalculado conforme o erro da saída. Esse processo é repetido até que todos os neurônios de todas as camadas sofram essa interação.

O repetido processo de verificação da saída desejada e do ajuste dos pesos é repetido até que a saída calculada esteja dentro de uma margem de erro aceitável com relação à saída desejada.

O processo de treinamento de uma rede *Perceptron* pode levar demasiado tempo e requerer um potencial computacional bastante elevado. Por isso, algumas técnicas para o aperfeiçoamento do treinamento são propostas.

Em 1963, os pesquisadores Levenberg e Marquardt propuseram uma técnica de melhoria no processo de aprendizado da rede. Eles propõem uma alteração que resulta em uma derivada do erro, otimizando o processo de aprendizado (HAGAN; MENHAJ, 1994).

O algoritmo de treinamento para a rede *Perceptron* é claramente exemplificado (NUNES; SPATTI; FLAUZINO, 2010), e será utilizado para o desenvolvimento do treinamento da rede proposta por este trabalho.

4 CONTROLE NEURAL APLICADO AO CONVERSOR *BOOST*

4.1 INTRODUÇÃO

Conforme visto no capítulo anterior, a rede neural tem capacidade de trabalhar como um aproximador universal de funções. A saída do controle, no tempo, pode ser descrito como uma função. Por isso, a utilização da rede neural como controlador pode ser empregada, uma vez que a rede fará a aproximação de seu sinal de saída conforme o sinal de entrada.

Se a rede neural for treinada de forma correta, a principal vantagem é que a rede terá condições de efetuar o controle do sistema com um sinal de controle com menor oscilação, isso porque sua saída será uma função próxima da função em que ela foi treinada.

Baseado nesse conceito, a rede neural irá substituir o controlador convencional do conversor *boost* com o intuito de que a ação de controle no conversor possa ser mais suave e não apresentar tantas oscilações, em especial para nas faixas de operação de tensão de entrada na qual o controlador não foi projetado, além de apresentar uma melhora no tempo de resposta do sistema.

4.2 CONTROLE NEURAL

Para o desenvolvimento de uma rede neural capaz de substituir o controle convencional desenvolvido no conversor, o primeiro procedimento é coletar informações de operação para que esses dados possam servir para o treinamento da rede neural.

O controle convencional utilizado nesse projeto está trabalhando com a informação de tensão de saída. O controle compara a tensão de saída desejada com a tensão de saída real do conversor. Então, gera um sinal de controle para atuar no chaveamento do conversor. A saída do controlador é um sinal entre 0 e 1 que representa a razão cíclica da chave de potência do controlador.

O controlador neural terá duas entradas, sendo tensão de entrada e tensão de saída do conversor e sua saída será a razão cíclica, para atuação da chave.

A Tabela 3 apresenta uma pequena parcela dos dados utilizados no treinamento, para melhor visualização dos dados de treinamento. Os dados, antes de serem apresentados para o treinamento da rede neural, foram normalizados. Assim, durante a operação, os valores de tensão de entrada e de saída também devem ser normalizados.

Tabela 3 – Exemplo dos dados utilizados no treinamento.

Tensão de entrada	Tensão de Saída	Saída Desejada (Razão cíclica)
70	400,0063	0,848517227172852
126	400,9564	0,706730461120606
189	400,1131	0,548554706573486
294	400,6766	0,276128864288330
350	400,6049	0,145763373374939

Fonte: Autoria própria.

Para o treinamento da rede neural, um conjunto de informações foi coletado, a partir da simulação do sistema com controle convencional. Como a faixa de operação do sistema será para tensões de entrada que variam entre 50V e 350V, para o treinamento e validação foram coletadas amostras conforme a Tabela 4.

Tabela 4 – Resumo das simulações para coleta de dados de operação do conversor.

Tensão de entrada	Tempo	Amostras
50V a 350V, com passos de 7V	0,7s (cada entrada)	700.000 (cada entrada)

Fonte: Autoria própria.

Dessa forma, foram executadas 44 simulações, com tensão de entrada inicial de 50V até 350V, com passos de 7V. Este passo representa uma variação na tensão de entrada de 2%. O tempo de execução de cada treinamento foi de 0,7s. Este tempo foi escolhido, pois para todas as entradas, era o suficiente para que o sistema operasse no modo dinâmico e operasse um período regime permanente. Isso permitiu que as amostras contivessem dados do sistema para o transitório e para o regime permanente. Essa condição é importante para que a rede tenha parâmetros suficientes para o correto aprendizado do funcionamento do sistema. A simulação deste projeto no software foi executada no modo discreto. A taxa de amostragem utilizada foi de 1 μ s. Assim, cada amostra coletada gerou uma matriz de três colunas por 700.000 linhas. Cada coluna representa, respectivamente, a tensão de entrada, a tensão de saída e a razão cíclica.

A forma como estes dados são apresentados para uma rede neural é primordial para que o sistema possa operar de forma satisfatória. Dessa forma, organizar estes dados é fundamental para que a resposta da rede atenda às necessidades do projeto. Além disso, a topologia da rede utilizada também altera a resposta do sistema.

De uma maneira geral, a rede neural vai aprender determinado padrão de funcionamento a partir dos dados de treinamento. Quanto mais informação contiver na matriz de treinamento, e quanto mais neurônios tiver a rede, os valores de saída da rede serão mais próximos aos valores dos dados desejado. Se os dados de treinamento não tiverem informações suficientes para que a rede possa aprender de forma correta, ela fará com que a saída seja uma aproximação daquilo que poderia ser a saída desejada.

Dessa forma, o processo de treinamento da rede neural consiste em processar os dados de operação, escolher uma topologia e treinar a rede. Na sequência, validar a rede e verificar se ela foi capaz de atender as expectativas do projeto. O processo de tratamento dos dados e treinamento da rede neural se repete até que a resposta da rede atenda às necessidades do projeto. O diagrama da Figura 21 apresenta este processo de treinamento.

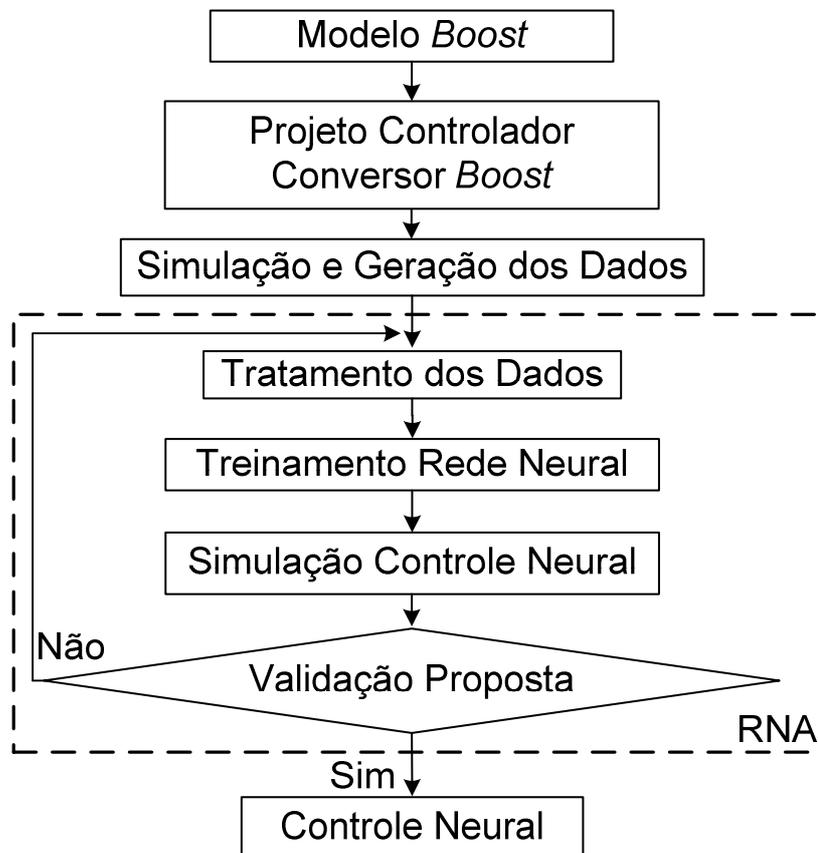


Figura 21 - Diagrama de treinamento da rede neural.

Fonte: Autoria própria.

Através da Figura 21 é possível observar que após encontrar o modelo do conversor *boost*, é possível projetar um controle para o conversor e executar simulações para a geração dos dados. Após a coleta dos dados, o processo entra em um ciclo composto pelo tratamento desses dados, o treinamento de uma topologia de rede neural e, após a convergência da rede, a simulação da proposta. Se o controle neural apresentou melhoria no controle do conversor, a proposta é validada e a rede neural treinada pode ser utilizada como controlador do conversor. Caso a resposta não atenda às expectativas, os dados são processados novamente ou a topologia da rede é alterada, até que a resposta do controle traga melhorias em relação ao controle convencional.

A proposta deste trabalho é fazer com que a rede neural trabalhe como controlador, otimizando o processo transitório da saída de tensão do conversor. Dessa forma, os dados que contém o modo de operação em regime transitório podem ser reduzidos, dando prioridade aos dados de operação em regime permanente. Fazendo isso, a rede aprenderá o padrão de funcionamento do regime permanente de maneira satisfatória e fará uma aproximação da saída quando estiver operando em regime transitório.

Assim, os dados foram organizados de maneira que contivessem um número maior de pontos do regime permanente, dando ênfase para esse modo de operação, e reduzido número de amostras para o regime transitório. A Tabela 5 apresenta uma síntese de como ficou os dados de treinamento.

Tabela 5 – Amostras para rede neural.

Tipo de amostra	Quantidade	Número de pontos	Tipo de pontos
Amostras para treinamento	40	102 cada	80%-Permanente – 20%Transitório
Amostras para validação	4	102 cada	80%-Permanente – 20%Transitório

Fonte: Autoria própria.

Com essas amostras geradas a partir do sistema simulado com o controle convencional, a rede neural pode ser treinada e validada para operação.

Como a busca da rede neural é uma saída que represente uma função, a topologia escolhida para esse projeto é a Rede *Perceptron* Multicamadas. Como a rede trabalhará com duas entradas, ela possui uma camada de entrada, composta por 2 neurônios. A saída dessa rede, que é a razão cíclica para o conversor, é obtida pela camada de saída composta por um neurônio. A Figura 22 apresenta a topologia da rede escolhida.

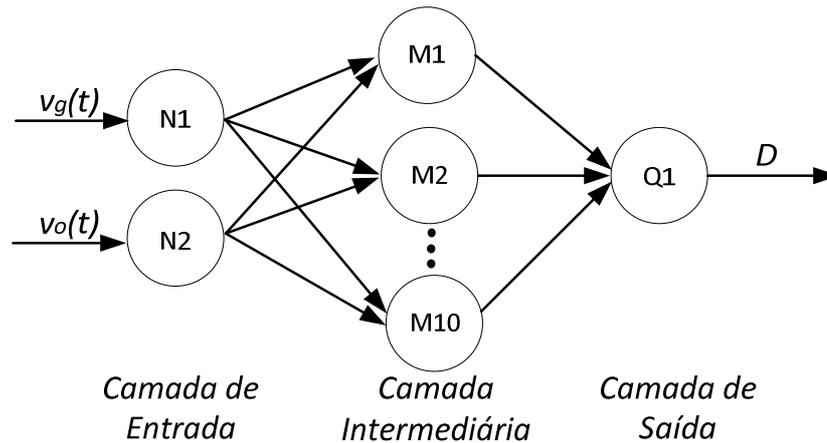


Figura 22 – Topologia da rede neural utilizada como controle do conversor.

Fonte: Autoria própria.

O computador utilizado para a simulação possui um processador Core2Duo com velocidade de 1.8Ghz e memória RAM de 4Gb. A Tabela 6 apresenta as informações referentes à rede.

Tabela 6 – Dados da rede neural.

Topologia da rede	PMC
Neurônios na camada de entrada	2
Neurônios na camada intermediária	10
Neurônios na camada de saída	1
Tolerância de erro	2.10^{-6}
Método de Otimização	Levenberg-Marquardt
Épocas para convergência	6200
Tempo de Convergência	4m:13s
Função de ativação (camada intermediária)	Tangente Hiperbólica
Função de ativação (camada saída)	Rampa Simétrica
Matriz de treinamento	4.080x3
Matriz de validação	408x3

Fonte: Autoria própria.

Com estes parâmetros, a rede neural foi treinada para o padrão de funcionamento com as tensões de entradas mencionadas acima. A rede apresentou um tempo de convergência de aproximadamente 5 minutos. Isso porque a matriz de treinamento e operação foi reduzida a fim de fornecer somente os dados suficientes para o aprendizado da rede neural.

4.3 SIMULAÇÃO DO CONVERSOR COM CONTROLE NEURAL

A simulação do sistema para validação do controle neural foi executada utilizando o mesmo procedimento utilizado na simulação do sistema com controlador convencional.

O controlador neural foi simulado considerando todos os neurônios com os respectivos pesos encontrados pelo processo de aprendizagem e a aplicação da função de ativação nos neurônios. Em todas as simulações, é possível perceber como que a mesma planta apresenta diferentes respostas para diferentes controladores.

Como o projeto do controlador convencional foi executado considerando a tensão de entrada de 200V, a primeira simulação consiste em verificar a resposta do sistema com a tensão de 200V. Dessa forma, a Figura 23 apresenta a resposta do sistema para essa tensão de entrada.

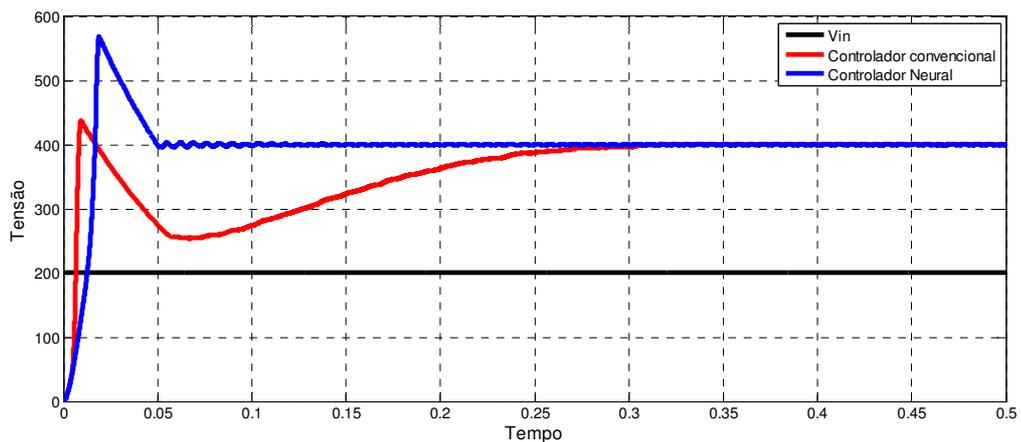


Figura 23 - Resposta do sistema para uma tensão de entrada de 200V.

Fonte: Autoria própria.

A curva em preto representa a tensão de entrada do sistema. A curva em vermelho representa a tensão de saída do conversor atuando através do controlador. A curva em azul representa a tensão de saída do conversor controlado pelo controlador neural. Observando a curva da tensão do conversor controlado pelo controlador neural, é possível perceber que o tempo de subida utilizando este tipo de controle é inferior ao tempo de subida conversor controlado pelo método convencional. O tempo de assentamento da tensão de saída utilizando o controle neural é inferior ao o tempo utilizado pela outra topologia. A Tabela 7 apresenta o índice ITAE para a simulação com tensão de entrada de 200V.

Tabela 7 – Índice ITAE para tensão de entrada de 200V.

Tipo de controle	ITAE
Convencional	2.0975
Neural	0.1686

Fonte: Autoria própria.

O controle neural trouxe uma melhoria na resposta do sistema em relação ao controle convencional. Ressaltando que a tensão de 200V de entrada não fazia parte do grupo de dados utilizados no treinamento da rede. Por isso, além de trazer a melhora na resposta do sistema, pode-se concluir também que a rede está sendo capaz de generalizar sua saída para valores de tensão de entrada que não foram utilizados no treinamento. Isso mostra que o sistema composto pelo controlador neural apresenta vantagens em relação ao controle convencional.

Para a validação completa do sistema, outras simulações foram executadas considerando diferentes tensões de entrada. De forma comparativa, os valores utilizados como tensão de entrada são os mesmos utilizados no capítulo anterior. Figura 24 apresenta a resposta do sistema para uma tensão de entrada de 350V.

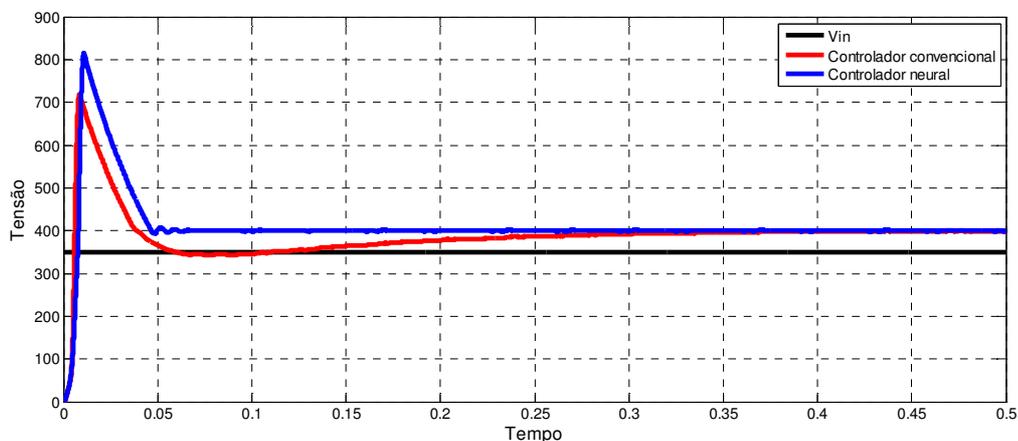


Figura 24 - Resposta do sistema para uma tensão de entrada de 350V.

Fonte: Autoria própria.

Da mesma forma que a imagem anterior, a curva em preto representa a tensão de entrada do conversor, a curva em vermelho representa a tensão de saída do conversor controlado pelo método convencional e a curva em azul representa a saída do sistema controlado pelo controle neural. As vantagens encontradas na simulação anterior podem ser observadas neste ponto de operação. A Tabela 8 apresenta o índice ITAE para a tensão de entrada de 350V.

Tabela 8 – Índice ITAE para tensão de entrada de 350V.

Tipo de controle	ITAE
Convencional	1.3748
Neural	0.2191

Fonte: Autoria própria.

A Figura 25 apresenta as tensões de saída para uma tensão de entrada de 280V.

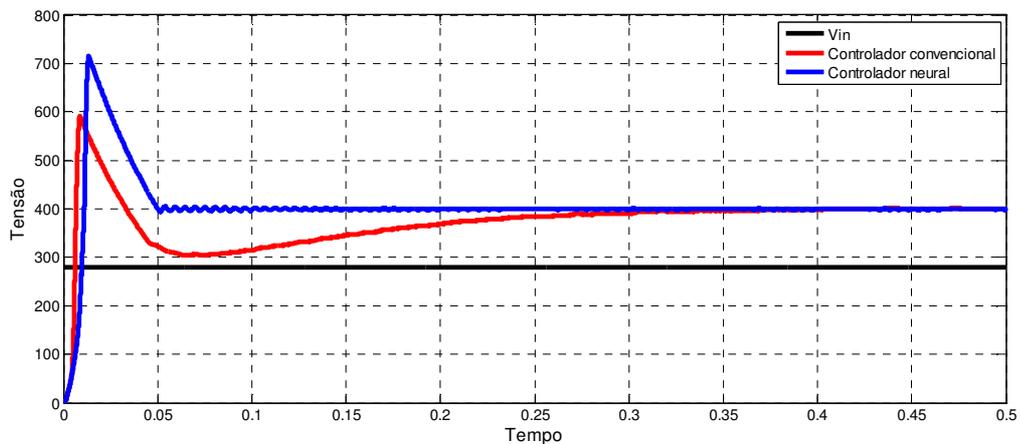


Figura 25 - Resposta do sistema para uma tensão de entrada de 280V.

Fonte: Autoria própria.

A Tabela 9 apresenta o índice ITAE para esta tensão de operação.

Tabela 9 – Índice ITAE para tensão de entrada de 280V.

Tipo de controle	ITAE
Convencional	1.7628
Neural	0.2265

Fonte: Autoria própria.

As Figuras 26 e 27 apresentam, respectivamente, os resultados de simulação para tensões de entrada de 210V, e 140V.

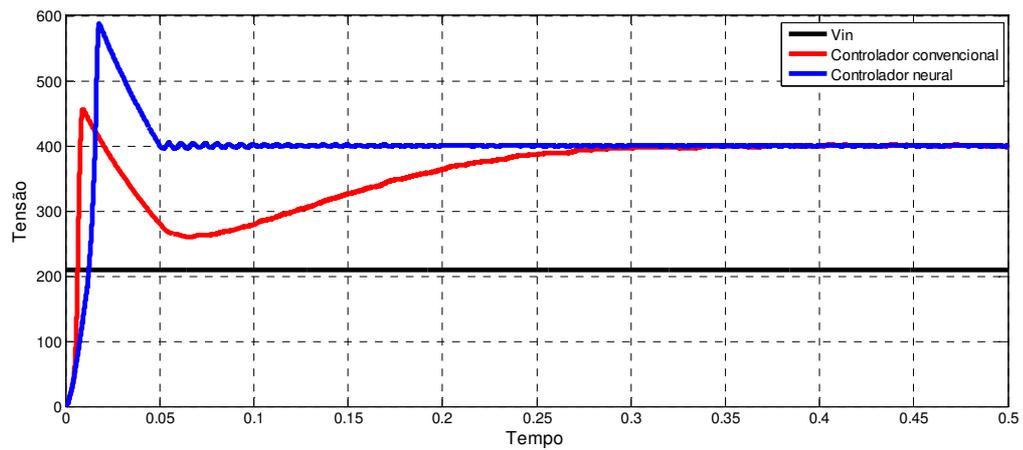


Figura 26 - Resposta do sistema para uma tensão de entrada de 210V.

Fonte: Autoria própria.

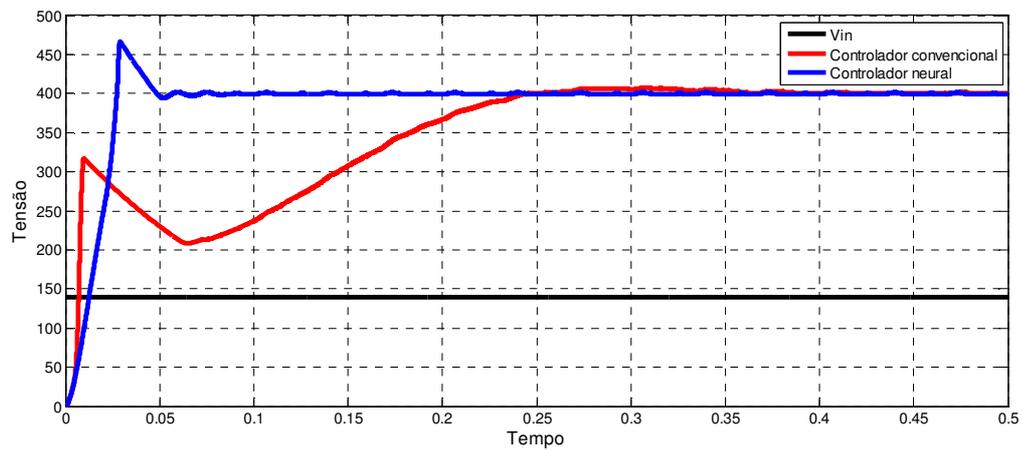


Figura 27 - Resposta do sistema para uma tensão de entrada de 140V.

Fonte: Autoria própria.

As Tabelas 10 e 11 apresentam, respectivamente, o índice ITAE para tensões de entrada de 210V, e 140V.

Tabela 10 – Índice ITAE para tensão de entrada de 210V.

Tipo de controle	ITAE
Convencional	2.0805
Neural	0.1682

Fonte: Autoria própria.

Tabela 11 – Índice ITAE para tensão de entrada de 140V.

Tipo de controle	ITAE
Convencional	2.4910
Neural	0.1775

Fonte: Autoria própria.

As Figuras 28 e 29 apresentam, respectivamente, os resultados de tensão de saída para tensão de entrada de 70V e 50V.

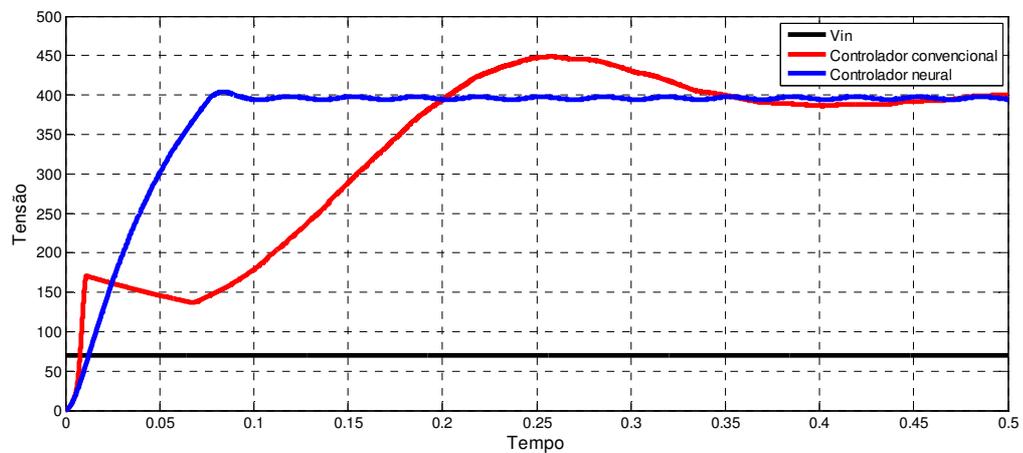


Figura 28 - Resposta do sistema para uma tensão de entrada de 70V.

Fonte: Autoria própria.

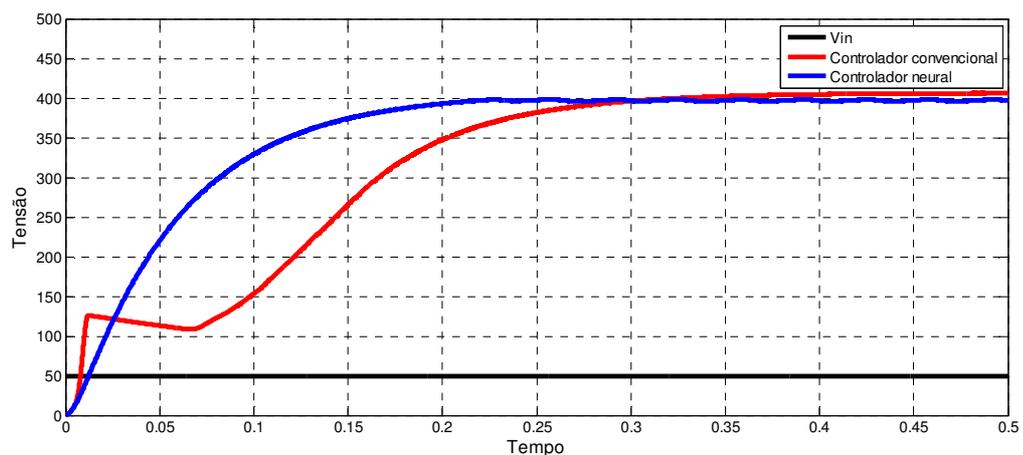


Figura 29 - Resposta do sistema para uma tensão de entrada de 50V.

Fonte: Autoria própria.

As Tabelas 12 e 13 apresentam o índice ITAE para as tensões de entrada de 70V e 50V.

Tabela 12 – Índice ITAE para tensão de entrada de 70V.

Tipo de controle	ITAE
Convencional	4.3733
Neural	0.7936

Fonte: Autoria própria.

Tabela 13 – Índice ITAE para tensão de entrada de 50V.

Tipo de controle	ITAE
Convencional	4.2406
Neural	1.4931

Fonte: Autoria própria.

Através destas simulações, é possível perceber uma melhora no tempo de estabilização da tensão de saída para toda a faixa de operação, quando o controlador neural é utilizado. Como a proposta deste trabalho era melhorar este aspecto da rede, o sistema agora está trabalhando de forma mais satisfatória. Todas as simulações apresentaram um valor do índice ITAE menor para o controle convencional. Isso indica a melhoria que o sistema obteve com o controlador neural aplicado no conversor.

A validação final da proposta foi feita através de simulações, com as tensões de entrada e saída apresentadas na Figura 30, onde há uma variação na tensão de entrada do conversor na forma de rampa. Isso porque não existe uma rajada de vento suficiente para tão forte que seja capaz de produzir uma variação na tensão de entrada do conversor na forma de rampa.

Estas condições representam uma situação mais próxima da realidade que este conversor encontrará durante seu funcionamento, uma vez que a inércia o sistema turbina-gerador não permitirá variações de tensão na forma de degrau, e a força do vento não irá garantir um funcionamento constante, variando sempre a tensão de saída do gerador.

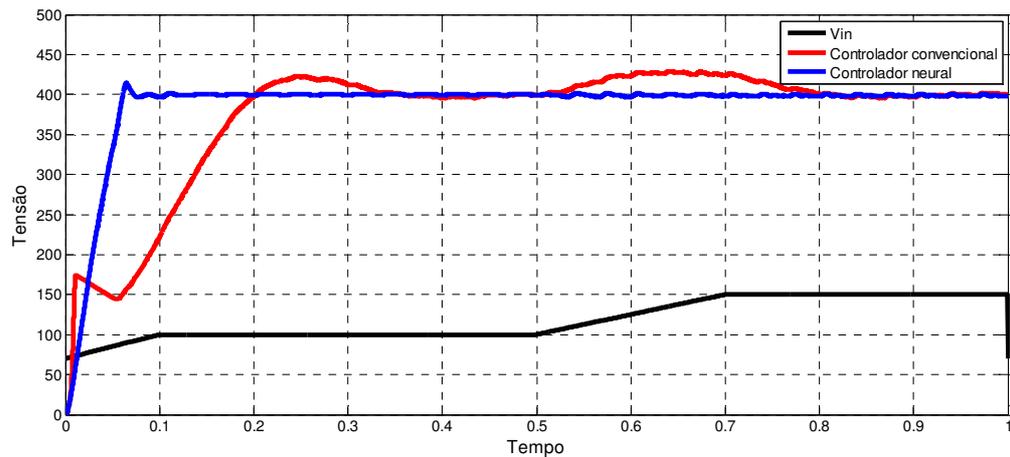


Figura 30 - Resposta do sistema para uma tensão de entrada em rampa.

Fonte: Autoria própria.

A simulação representada pela Figura 30 apresenta uma situação mais próxima da realidade desse conversor, devido a algumas características do aerogerador. Dessa forma, a simulação apresenta um início de operação do sistema na forma de degrau de uma tensão de 70V. A rampa continuou até o valor de 100V. Neste período, o controlador neural já conseguiu garantir a tensão de 400V desejada na saída, enquanto o sistema operando pelo controlador convencional ainda não tinha alcançado a tensão desejada.

Após operar um período com tensão constante de 100V, o sistema volta a sofrer uma variação de tensão na forma de rampa, alcançando a tensão e 150V, e continuando nesta tensão até o fim da simulação. É possível perceber que o conversor controlado pelo controlador neural não gera nenhum erro na tensão de saída, enquanto que o controle convencional sofre uma variação de na tensão de saída, necessitando de um tempo até alcançar a tensão desejada.

A Tabela 14 apresenta o índice ITAE desta simulação.

Tabela 14 – Índice ITAE para tensão de entrada em rampa.

Tipo de controle	ITAE
Convencional	6.4269
Neural	0.7861

Fonte: Autoria própria.

A Figura 31 apresenta uma simulação em que a tensão de entrada sobre uma variação maior do que na simulação anterior, e posteriormente sofre um decréscimo de tensão.

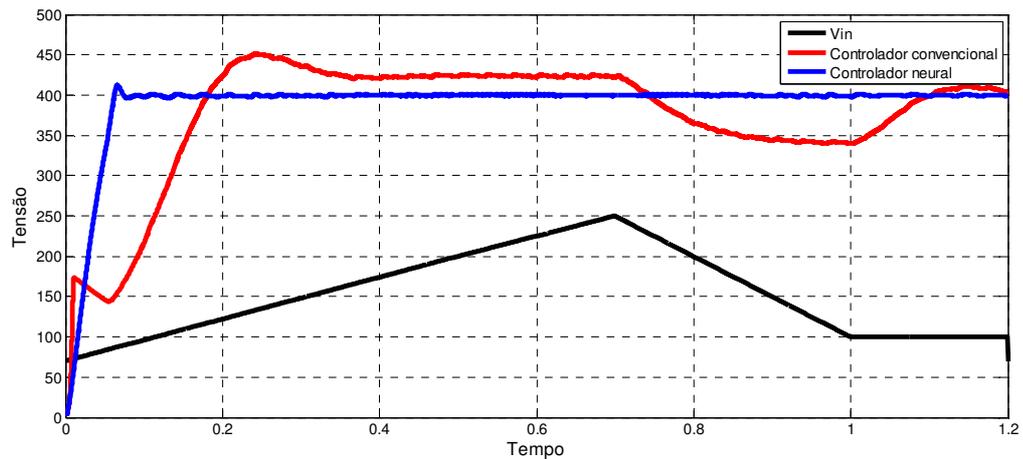


Figura 31 - Resposta do sistema para uma tensão de entrada em rampa.

Fonte: Autoria própria.

Esta simulação apresentou a tensão de saída onde o conversor entrou em operação com uma tensão de 70V, sofrendo um acréscimo até a tensão de 250V. Durante este modo de operação, o controlador convencional não conseguiu garantir a tensão de saída, ou seja, o tempo de distúrbio na tensão foi maior do que o tempo que o controlador precisava para estabilizar a tensão de saída. Antes que fosse capaz de buscar a tensão de referência, a tensão de entrada sofreu um decréscimo, estabilizando a tensão em 100V. Novamente o controle neural conseguiu garantir a tensão desejada na saída, mesmo diante destes distúrbios. A Tabela 15 apresenta o índice ITAE para este modo de operação.

Tabela 15 – Índice ITAE para tensão de entrada em rampa.

Tipo de controle	ITAE
Convencional	22.7477
Neural	0.6941

Fonte: Autoria própria.

Esta simulação apresentou a tensão de saída onde o conversor entrou em operação com uma tensão de 70V, sofrendo um acréscimo até a tensão de 250V. Durante este modo de operação, o controlador

5 CONCLUSÃO

5.1 CONCLUSÃO

Este trabalho apresentou o estudo, projeto e simulação de um conversor CC-CC tipo *boost* que será aplicado a um sistema de geração de energia eólica. A característica desse sistema de geração eólica é o fato de que a velocidade e a potência do gerador são variáveis de acordo com a velocidade do vento incidente na turbina eólica. Nesse caso, a potência mecânica no eixo da turbina não é constante, o que gera na saída do gerador tensões de amplitude variável de acordo com a velocidade do gerador. Por isso, a utilização do conversor para estabilizar a tensão de saída é vital para que o sistema possa servir como fonte para alimentação de cargas. Foi estudado e comprovado através de simulações que um controle pode ser projetado para controlar o conversor. Entretanto, o projeto do controlador é executado em um ponto de operação, trazendo funcionamento com dinâmica diferente para diferentes faixas de operação.

A proposta foi utilizar um sistema de controle neural, capaz de aprender os melhores pontos de operação do sistema, buscando um funcionamento otimizado para todas as faixas de operação.

O controle proposto atendeu as expectativas e mostrou-se como solução para o problema. Ele foi capaz de substituir o controlador convencional e trouxe uma melhoria na resposta para as faixas de operação em que o sistema convencional não conseguiu. Além disso, se mostrou melhor no tempo de resposta e melhor resposta em regime permanente. Dessa forma, a proposta foi alcançada e o sistema pode substituir o controle convencional.

5.2 TRABALHOS FUTUROS

Uma proposta para trabalho futuro é implementar este controle no sistema para a validação dos resultados de simulação.

5.3 PUBLICAÇÃO ASSOCIADA AO TRABALHO

TONON, T.; VOLTOLINI, H; GOEDEL, A; Controle Neural Aplicado a um Conversor *Boost* Utilizado em Aerogeradores de Baixa Potência. XX Congresso Brasileiro de Automática. Setembro 2014.

REFERÊNCIAS

DHIVYA, B.S.; KRISHNAN, V.; RAMAPRABHA, R., **Neural network controller for boost converter**. Circuits, Power and Computing Technologies (ICCPCT), 2013 International Conference on , vol., no., pp.246,251, 20-21 March 2013.

DORF, R. C.; BISHOP, R. H.; **Sistemas de Controle Modernos**. Rio de Janeiro, RJ: LTC Editora 2001.

HAGAN, M.; MENHAJ, M. **Training feedforward networks with the marquardt algorithm**. IEEE Transactions on Neural Networks, v. 5, n. 6, p. 989 –993, nov 1994. ISSN 1045-9227.

HASSAM, M.Y.; KOTHAPALLI, G., **Comparison between Neural Network based PI and PID controllers**. Systems Signals and Devices (SSD), 2010 7th International Multi-Conference on , vol., no., pp.1,6, 27-30 June 2010.

JAWHAR, J.; MARIMUTHU, N.S.; SINGH, A.; **An Neuro-Fuzzy Controller for a Non Linear Power Electronic Boost Converter**. Information and Automation, 2006. ICIA 2006.International Conference on, December 2006.

JOHANSSON, B.; **Improved Models for DC-DC Converters**. Lund University: Sweden, 2003.

KABOLI, S.H.A.; MANSOURI, M.; SELVARAJ, J.; RAHIM, N.B.A., **A hybrid adaptive Neural-Fuzzy tuned P.I. controller based Unidirectional Boost P.F.C. converter feeds B.L.D.C. drive**. Power Electronics, Drive Systems and Technologies Conference (PEDSTC), 2013 4th , vol., no., pp.176,181, 13-14 Feb. 2013.

KOTHAPALLI, G; HASSAN, M. Y.; **Design of a Neural Network based intelligent PI controller for a pneumatic system**. IAENG International Journal of Computer Science 01/2008 IJCS, 2008.

MAHDAVI J.; M. R. NASIRI; A. AGAH; A. EMADI; **Application of neural networks and state-space averaging to DC/DC PWM converters in sliding-mode operation**. IEEE Transaction on Mechatronics, vol. 10, No.1. pp. 60 - 67, February 2005.

NUNES, I; SPATTI, D. H.; FLAUZINO, R.A. **Redes neurais Artificiais para Engenharia e Ciências Aplicadas**. São Paulo, SP: Artliber, 2010).

OGATA, K. **Engenharia de Controle Moderno**. 4 ed.. ed. São Paulo, SP: Prentice-Hall, 2003.

ROSTAMI, H.; KHANURI, D.A.; **Neural networks controlling for both the DC boost and AC output voltage of Z-source inverter**. Power Electronic & Drive Systems & Technologies Conference (PEDSTC), 2010 1st. pp. 135-140 17-18 Feb 2010.

UTOMO, W.M.; BAKAR, A.; AHMAD, M.; TAUFIK, T.; HERIANSYAH, R., **Online Learning Neural Network Control of Buck-Boost Converter**. Information Technology: New Generations (ITNG), 2011 Eighth International Conference on , vol., no., pp.485,489, 11-13 April 2011 4th Power Electronics, Drive Systems & Technologies Conference (PEDSTC2013), Feb 13-14, 2013, Tehran, Iran.