



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



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

LUCAS SILVA GUTTERRES

**IMPLEMENTAÇÃO SIMPLIFICADA DE CONTROLE NÃO LINEAR EM FPGA IN
THE LOOP PARA CONVERSOR BOOST BASEADO NA TÉCNICA TAKAGI-
SUGENO FUZZY**

CAMPO GRANDE, MS

2018



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



LUCAS SILVA GUTTERRES

**IMPLEMENTAÇÃO SIMPLIFICADA DE CONTROLE NÃO LINEAR EM FPGA IN
THE LOOP PARA CONVERSOR BOOST BASEADO NA TÉCNICA TAKAGI-
SUGENO FUZZY**

Trabalho de Conclusão de Curso
apresentado à Faculdade de Engenharias,
Arquitetura e Urbanismo e Geografia da
Universidade Federal de Mato Grosso do Sul, como
requisito parcial para a obtenção do título de
Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Raymundo Cordero Garcia

CAMPO GRANDE, MS

2018



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



LUCAS SILVA GUTTERRES

**IMPLEMENTAÇÃO SIMPLIFICADA DE CONTROLE NÃO LINEAR EM FPGA IN
THE LOOP PARA CONVERSOR BOOST BASEADO NA TÉCNICA TAKAGI-
SUGENO FUZZY**

Trabalho de Conclusão de Curso
apresentado à Faculdade de Engenharias,
Arquitetura e Urbanismo e Geografia da
Universidade Federal de Mato Grosso do Sul como
requisito parcial para a obtenção do título de
Bacharel em Engenharia Elétrica.

Campo Grande, MS, ____ de _____ de 2018.

COMISSÃO EXAMINADORA

Prof. Dr. Raymundo Cordero Garcia
Orientador

Prof. Dr. Cristiano Quevedo Andrea
Avaliador

Prof. Dr. Edson Antonio Batista
Avaliador



AGRADECIMENTOS

A Deus pelo dom da vida, por seu amor incondicional e pelas oportunidades que me foram concedidas ao decorrer da vida, dentre elas, ingressar na Universidade Federal de Mato Grosso do Sul. Também, pela saúde e força concedidos a mim para a superação das dificuldades e conclusão deste trabalho.

Ao meu querido pai Edilton (*in memoriam*) que infelizmente partiu no meio dessa jornada de graduação. Pelo seu amor, incentivo, companheirismo e por todos os ensinamentos valiosíssimos que me foram repassados.

A minha mãe Oralisse e meu irmão Sânzio, pelo amor, incentivo e apoio incondicional.

A minha namorada Suzana pelo amor, apoio, carinho e paciência. Pela capacidade de me trazer paz nos dias mais turbulentos e por toda sua contribuição para que eu me torne uma pessoa melhor, através de seus conselhos e pelo seu esforço e dedicação ímpares, que servem de exemplo, não só para mim, mas para todos a sua volta.

Aos meus amigos pela parceira nos estudos e momentos de descontração.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior. Em especial a Prof. Dr. Luciana Cambraia Leite que não mediu esforços para auxiliar-me durante o curso.

Ao Prof. Dr. Cristiano Quevedo Andrea que dispôs de seu tempo para a participação na banca examinadora e auxílio em melhorias para este trabalho.

Ao Prof. Dr. Edson Antonio Batista pelo fornecimento do FPGA e orientações em relação ao FPGA *in the Loop*, sendo tais assistências fundamentais para o desenvolvimento do presente trabalho, bem como pela disponibilidade de compor a banca examinadora e pelas contribuições dadas acerca deste trabalho.

Ao meu orientador Prof. Dr. Raymundo Cordero García pela atenção concedida, confiança e incentivo para que eu me tornasse um profissional melhor. Por sua amizade, paciência e por todos os ensinamentos compartilhados.

Ao BATLAB pela disponibilidade de espaço e equipamentos essenciais para execução deste trabalho.

E a todos que direta ou indiretamente fizeram parte deste trabalho.



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



“Não confunda derrotas com fracasso nem vitórias com sucesso. Na vida de um campeão sempre haverá algumas derrotas, assim como na vida de um perdedor sempre haverá vitórias. A diferença é que, enquanto os campeões crescem nas derrotas, os perdedores se acomodam nas vitórias.”

(Roberto Shinyashiki)



RESUMO

O aumento da demanda de energia e da conscientização ambiental resultaram em um grande aumento nos estudos de fontes alternativas de energia. Neste cenário, a micro e mini geração distribuída vem ganhando força, com destaque para a geração de energia fotovoltaica. Desta maneira, o presente Trabalho de Conclusão de Curso propõe o controle não linear de um conversor CC-CC elevador de tensão, também conhecido como conversor CC-CC *Boost*, sendo este, largamente utilizado em sistemas fotovoltaicos. A fim de solucionar o problema decorrente das não linearidades presentes na modelagem matemática do conversor em questão, faz-se a proposta do desenvolvimento de um controle robusto a partir da metodologia Takagi-Sugeno *Fuzzy* (TSF) integrado ao Controlador de Modelagem Interna (CMI). Para a obtenção dos ganhos do controlador CMI-TSF foi adotada a técnica de Desigualdades Matriciais Lineares. A validação do controle proposto é realizada por meio da abordagem de teste *Hardware in the Loop* (HIL) aplicada ao FPGA, que recebe o nome de FPGA *in the Loop* (FIL). Entretanto, os controladores TSF requerem muitas operações matemáticas, que resultam no aumento do custo computacional e na redução de velocidade de execução do algoritmo de controle implementado. Por esse motivo, o trabalho aqui exposto, descreve uma simplificação matemática com o intuito de reduzir o número de operações matemáticas no algoritmo de controle. Com a finalidade de comprovar a robustez do controlador CMI-TSF proposto, as simulações foram realizadas para diversos tipos de cargas na saída do conversor *Boost* e além disso, foram consideradas perturbações na tensão de entrada do conversor. Os resultados obtidos evidenciam as vantagens da abordagem proposta.

Palavras-chave: controle não linear, conversor CC-CC *Boost*, Takagi-Sugeno *Fuzzy*, FPGA *in the Loop*.



ABSTRACT

This work proposes the non-linear control of a DC-DC converter to a higher voltage, also known as Boost or Step-up converter. To solve the recurring problem resulting from non-linearity in the mathematical modeling of this converter, it is done a proposal of the development of a robust control starting from the Takagi-Sugeno Fuzzy (TSF) model with integration to the Internal Model Control (IMC). For the obtaining of controller IMC-TSF's gains, it was adopted the Linear Matrix Inequality technique. The validation of the proposed control is accomplished through the approach of test Hardware in the Loop (HIL) applied to FPGA, that receives the name of FPGA in the Loop (FIL). However, the controllers TSF request many mathematical operations, that result in the increase of the cost computational and in the reduction of speed of execution of the control algorithm implemented in FPGA. For that reason, in this work is described a mathematical simplification in order to reducing the number of mathematical operations in the control algorithm. Based on the purpose of proving the controller's robustness of the IMC-TSF, the simulations were accomplished for several types of loads on the controller output and besides, disturbances in the controller input voltage were considered. The obtained results evidence the advantages of the proposed approach.

Key-words: non-linear control, DC-DC Boost converter, Takagi-Sugeno Fuzzy, FPGA in the Loop.



LISTA DE ILUSTRAÇÕES

Figura 2.1 - Conversor CC-CC Elevador de Tensão (<i>Boost</i>).....	20
Figura 2.2 – Circuito Equivalente ao Comportamento do Conversor <i>Boost</i>	21
Figura 2.3 - Primeira Etapa de Funcionamento do <i>Boost</i> em MCC.....	22
Figura 2.4 - Segunda Etapa de Funcionamento do <i>Boost</i> em MCC.....	24
Figura 2.5 - Principais Formas de Onda do Conversor <i>Boost</i> em MCC.	25
Figura 2.6 - Característica Ideal de Transferência Estática do Conversor <i>Boost</i>	27
Figura 2.7 - Conceito da metodologia da modelagem TSF. Erro! Indicador não definido.	
Figura 2.8 – Região de \mathcal{D} -estabilidade $\mathcal{S}(\gamma, r, \theta)$	47
Figura 2.9 – Sistema com Controlador de Modelagem Interna (CMI).	54
Figura 3.1 - Região $\mathcal{S}(\gamma)$ de Alocação dos Pólos Definida pelas LMIs.	68
Figura 3.2 – Diagrama de Blocos do Modelo do Controlador CMI-TSF Simplificado Proposto.....	71
Figura 3.3 - Manipulação Matemática Realizada no Termo $K_{ej}x_a$	75
Figura 4.1 - Tensão de Entrada do Conversor Boost Utilizada nas Simulações.	78
Figura 4.2 – Simulação MATLAB/SIMULINK: Razão Cíclica para $R_1 = 60\Omega$	79
Figura 4.3 – Simulação MATLAB/SIMULINK: Corrente no Indutor para $R_1 = 60\Omega$. ..	80
Figura 4.4 - Simulação MATLAB/SIMULINK: Tensão de Saída para $R_1 = 60\Omega$	80
Figura 4.5 - Simulação MATLAB/SIMULINK: Erro entre a Referência e a Saída para $R_1 = 60\Omega$	81
Figura 4.6 - Simulação MATLAB/SIMULINK: Razão Cíclica para $R_2 = 50\Omega$	81
Figura 4.7 – Simulação MATLAB/SIMULINK: Corrente no Indutor para $R_2 = 50\Omega$. ..	82
Figura 4.8 - Simulação MATLAB/SIMULINK: Tensão de Saída para $R_2 = 50\Omega$	82
Figura 4.9 - Simulação MATLAB/SIMULINK: Erro entre a Referência e a Saída para $R_2 = 50\Omega$	83
Figura 4.10 – Simulação MATLAB/SIMULINK: Razão Cíclica para $R_3 = 30\Omega$	83
Figura 4.11 - Simulação MATLAB/SIMULINK: Corrente no Indutor para $R_3 = 30\Omega$. ..	84
Figura 4.12 – Simulação MATLAB/SIMULINK: Tensão de Saída para $R_3 = 30\Omega$	84
Figura 4.13 - Simulação MATLAB/SIMULINK: Erro entre a Referência e a Saída para $R_3 = 30\Omega$	85



Figura 4.14 – Resultados da Compilação e do Custo Computacional do Algoritmo Proposto.....	86
Figura 4.15 – Selecionando o Assistente de Configuração do FIL.....	87
Figura 4.16 – Janela do Assistente de Configuração do FIL: Primeira Etapa.	88
Figura 4.17 - Janela do Assistente de Configuração do FIL: Segunda Etapa.	89
Figura 4.18 – Exemplo de Declaração de Variável como Ponto-Fixo Assinalado em VHDL.....	90
Figura 4.19 - Janela do Assistente de Configuração do FIL: Terceira Etapa.	91
Figura 4.20 - Janela do Assistente de Configuração do FIL: Quarta Etapa.	92
Figura 4.21 - Janela do Assistente de Configuração do FIL: Quinta Etapa.....	93
Figura 4.22 – Bloco FIL gerado com Conexões de Entrada e Saída no SIMULINK..	94
Figura 4.23 - Janela de Carregamento do Algoritmo no FPGA.	95
Figura 4.24 – Diagrama de Blocos.....	96
Figura 4.25 – Teste Experimental FIL em Bancada.	97
Figura 4.26 – Kit DE2-112 da Altera com FPGA Utilizado e Conexões Necessárias para o Teste FIL.	97
Figura 4.27 - Teste FIL: Razão Cíclica para $R_1 = 60\Omega$	98
Figura 4.28 – Teste FIL: Corrente no Indutor para $R_1 = 60\Omega$	99
Figura 4.29 - Teste FIL: Tensão de Saída para $R_1 = 60\Omega$	99
Figura 4.30 - Teste FIL: Erro entre a Referência e a Saída para $R_1 = 60\Omega$	100
Figura 4.31 - Teste FIL: Razão Cíclica para $R_2 = 50\Omega$	100
Figura 4.32 – Teste FIL: Corrente no Indutor para $R_2 = 50\Omega$	101
Figura 4.33 - Teste FIL: Tensão de Saída para $R_2 = 50\Omega$	101
Figura 4.34 - Teste FIL: Erro entre a Referência e a Saída para $R_2 = 50\Omega$	102
Figura 4.35 – Teste FIL: Razão Cíclica para $R_3 = 30\Omega$	102
Figura 4.36 - Teste FIL: Corrente no Indutor para $R_3 = 30\Omega$	103
Figura 4.37 – Teste FIL: Tensão de Saída para $R_3 = 30\Omega$	103
Figura 4.38 - Teste FIL: Erro entre a Referência e a Saída para $R_3 = 30\Omega$	104



Serviço Público Federal
Ministério da Educação

Fundação Universidade Federal de Mato Grosso do Sul



LISTA DE TABELAS

Tabela 4.1 – Variáveis Declaradas em Ponto-Fixo Assinalado com Respectivas Quantidades de Bits.	90
---------------------------------------------------------------------------------------------------------	----



LISTA DE ABREVIATURAS E SIGLAS

ABSOLAR	Associação Brasileira de Energia Solar Fotovoltaica
ANEEL	Agência Nacional de Energia Elétrica
CA	Corrente Alternada
CC	Corrente Contínua
CMI	Controlador de Modelagem Interna
CPD	Compensação Paralela Distribuída
DSP	<i>Digital Signal Processor</i>
FIL	<i>FPGA in the Loop</i>
FPGA	<i>Field Programmable Gate Array</i>
HIL	<i>Hardware in the Loop</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
LMI	<i>Linear Matrix Inequalities</i>
MCC	Modo de Condução Contínua
MPPT	<i>Maximum Power Point Tracking</i>
PID	Proporcional Integral Derivativo
PWM	<i>Pulse Width Modulation</i>
RN	Resolução Normativa
SISO	<i>Single Input Single Output</i>
TSF	Takagi-Sugeno <i>Fuzzy</i>
USB	<i>Universal Serial Bus</i>
VHDL	<i>VHSIC Hardware Description Language</i>
VHSIC	<i>Very High Speed Integrated Circuits</i>



SUMÁRIO

1	INTRODUÇÃO	14
1.1	DEFINIÇÃO DA PROBLEMÁTICA	16
1.2	OBJETIVOS	18
1.2.1	OBJETIVOS GERAIS	18
1.2.2	OBJETIVOS ESPECÍFICOS	18
1.3	ORGANIZAÇÃO DO TRABALHO	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	CONVERSOR CC-CC ELEVADOR DE TENSÃO	20
2.1.1	CONVERSOR <i>BOOST</i> EM MODO DE CONDUÇÃO CONTÍNUA 21	
2.1.2	GANHO ESTÁTICO	25
2.2	MODELAGEM MATEMÁTICA DO CONVERSOR CC-CC <i>BOOST</i> EM ESPAÇO DE ESTADOS	27
2.2.1	MODELO MÉDIO DO CONVERSOR CC-CC <i>BOOST</i>	28
2.3	SISTEMAS TAKAGI-SUGENO <i>FUZZY</i>	30
2.3.1	REPRESENTAÇÃO DE SISTEMAS NÃO LINEARES POR MEIO DO USO DE MODELOS LOCAIS	31
2.3.2	MODELAGEM TAKAGI-SUGENO <i>FUZZY</i>	32
2.4	TEOREMA DE ESTABILIDADE DE LYAPUNOV	40
2.4.1	ANÁLISE DE ESTABILIDADE SEGUNDO LYAPUNOV	41
2.5	PROJETO DE CONTROLADORES UTILIZANDO LMIS	43
2.6	PROJETO DE CONTROLADORES TAKAGI-SUGENO <i>FUZZY</i> UTILIZANDO LMIS	49



2.7	CONTROLADOR DE MODELAGEM INTERNA.....	53
2.8	FPGA <i>IN THE LOOP</i>	56
2.8.1	FPGA.....	56
2.8.2	<i>HARDWARE IN THE LOOP</i>	57
2.8.3	FPGA <i>IN THE LOOP</i>	58
3	ESTRUTURA DO CONTROLADOR PROPOSTO.....	60
3.1	MODELO TAKAGI-SUGENO <i>FUZZY</i> DO CONVERSOR <i>BOOST</i>	60
3.2	INTEGRAÇÃO ENTRE OS MODELOS TSF E CMI	63
3.3	DEFINIÇÃO DAS LMIS UTILIZADAS PARA O MODELO CMI-TSF ..	67
3.4	DETERMINAÇÃO DOS GANHOS <i>Kej</i> DO CONTROLADOR CMI-TSF	70
3.5	CONTROLADOR CMI-TSF SIMPLIFICADO PROPOSTO.....	70
3.5.1	SIMPLIFICAÇÕES NAS EQUAÇÕES DAS FUNÇÕES DE PERTINÊNCIA.....	71
3.5.2	DEFINIÇÃO DA LEI DE CONTROLE SIMPLIFICADA	74
4	RESULTADOS OBTIDOS.....	78
4.1	SIMULAÇÃO EM <i>SOFTWARE</i> – MATLAB/SIMULINK.....	79
4.2	TESTE FPGA <i>IN THE LOOP</i>	86
5	CONCLUSÃO	105
6	BIBLIOGRAFIA	106
	APÊNDICE I – Algoritmo de Controle Implementado no MATLAB	110
	APÊNDICE II – Algoritmo de Controle em VHDL	113
	APÊNDICE III – Diagrama de Blocos: Simulação SIMULINK/MATLAB	115
	APÊNDICE IV – Diagrama de Blocos: Teste FIL.....	116



1 INTRODUÇÃO

Devido ao aumento da demanda de energia e da conscientização ambiental, as fontes alternativas de energia, principalmente as renováveis, passaram a ser estudadas com grande relevância pelos especialistas em geração de energia. Como consequência, as pesquisas relacionadas a Eletrônica de Potência têm ganhado força na sociedade. O surgimento de inovações em conversores chaveados reforçou a produção de dispositivos mais modernos, seguros e acessíveis tanto ao pequeno como ao grande consumidor e produtor de energia, sob a ótica da cogeração, geração distribuída e *smart grids*.

Com o intuito de auxiliar o setor elétrico brasileiro a minimizar os impactos do crescimento elevado, como sobrecarga na geração e na transmissão de energia elétrica, em 17 de abril de 2012 entrou em vigor a Resolução Normativa da Agência Nacional de Energia Elétrica (ANEEL) - RN 482/2012, que trata da micro e mini geração distribuída e do sistema de compensação de energia elétrica. Esta, permite que os consumidores brasileiros em geral (residências, empresas, instituições, edifícios públicos, propriedades rurais) gerem energia renovável em sua própria unidade consumidora para seu consumo. Além disso, qualquer excedente produzido não é desperdiçado, podendo esse ser injetado na rede de distribuição (Agência Nacional de Energia Elétrica (ANEEL), 2012). A fim de aprimorar as diretrizes da RN 482/2012, surgiram também as RNs 687/2015 e 786/2017, o que oferece ao país um excelente arcabouço regulatório em favor da geração distribuída.

Dentre as fontes renováveis (eólica, fotovoltaica, célula de combustível, etc.), o sistema de geração de energia solar fotovoltaica vem crescendo em diversos setores, principalmente no residencial. Isto se dá pela sua implantação favorável no sistema de distribuição de energia, por ser de instalação simples, e adaptação em ambientes urbanos, bastando que o local de aplicação possua irradiação solar de qualquer intensidade (Dreher, 2012). Segundo a Associação Brasileira de Energia Solar Fotovoltaica (ABSOLAR), atualmente, a energia solar proveniente de placas fotovoltaicas representa 75,5% do total da potência instalada da micro e mini geração distribuída, liderando com folga este segmento com mais de 99,3% das instalações do país (Associação Brasileira de Energia Solar Fotovoltaica (ABSOLAR), 2018).



A conversão de energia solar em elétrica dá-se por meio de células fotovoltaicas, sendo geralmente, estas confeccionadas de silício cristalino. Uma célula fotovoltaica é capaz de produzir 1,5 W de potência, e são associadas de forma a formarem módulos com fornecimento, frequentemente, entre 100 e 500 W com uma tensão entre 18 e 46 V disponibilizada na forma de corrente e tensão contínua (Dreher, 2012).

Por apresentar geração em corrente e tensão contínua, a potência gerada pelos painéis fotovoltaicos não pode ser injetada diretamente na rede elétrica, devendo passar por um tratamento para tal. Primeiramente, o nível de tensão gerado deve ser adequado para conexão com a rede, passando por um conversor CC-CC, e posteriormente, por um inversor, para a transformação do sinal CC em CA. Após este procedimento, o sinal gerado deve ser posto em sincronia com o sistema de distribuição, para então, ser conectado na rede elétrica (Dreher, 2012).

Como o silício é um material que é susceptível às variações de temperatura e irradiação solar, o módulo fotovoltaico também tem seus valores de tensão e corrente gerados influenciados por tais fatores. Por esse motivo, os conversores para este tipo de aplicação devem trabalhar com MPPT (*Maximum Power Point Tracking*), que são implementados por meio de algoritmos que visam o rastreamento da máxima potência possível a ser extraída do painel. Esta técnica de rastreamento é de grande importância para que se tenha um melhor aproveitamento nas diferentes condições de corrente e tensão geradas. Como consequência, o retorno do investimento na instalação dessa tecnologia apresenta uma melhora significativa, tornando sua implantação em larga escala muito atrativa (Junior, 2011).

Os conversores estáticos de energia elétrica, para serem úteis nas mais diversas aplicações, devem ter suas variáveis elétricas, tais como, corrente, tensão e potência, adequadamente controladas (Barbi, 2015). Esta tarefa tem apresentado um aumento significativo do ponto de vista da sua aplicação como controle digital. Tal feito foi motivado, principalmente, pela evolução das estruturas da eletrônica de potência e microprocessadores, bem como pelo aperfeiçoamento da teoria de controle por meio do uso de técnicas de controle robusto. Adicionalmente, o uso de técnicas de simulação de projetos de controle vem ganhando força, já que é possível a



realização de testes em ambiente virtual antes de passar para os testes do sistema real, proporcionando assim uma melhor validação do sistema de controle projetado.

1.1 DEFINIÇÃO DA PROBLEMÁTICA

Muitas aplicações exigem o rastreamento de referências variantes no tempo. Projetos que visam o controle de temperatura, velocidade, vazão, nível, processos químicos e físicos, são exemplos de onde podemos encontrar esta exigência.

Considerando o cenário contemplado pelos sistemas fotovoltaicos, a proposta do presente trabalho tem enfoque destinado ao projeto de controle envolvendo o conversor CC-CC elevador de tensão, conhecido também na literatura como *Boost*. Neste conversor, a tensão média de saída é maior que a tensão de entrada, ou seja, a mínima tensão média de saída é teoricamente, igual a tensão de alimentação a qual o conversor está submetido. Suas principais aplicações são em retificadores com elevado fator de potência, fontes de alimentação, acionamento de motores de corrente contínua e, atualmente, como já explanado, muito estudado em aplicações envolvendo painéis fotovoltaicos (Martins & Barbi, 2006).

Uma das particularidades dos conversores estáticos CC-CC é a ocorrência, em um período de operação, de vários estágios topológicos, cujos circuitos equivalentes são lineares, que geralmente são descritos por equações diferenciais de primeira ou segunda ordem. Todavia, a nível macroscópico, suas respostas naturais, sob a ótica de valores médios quase instantâneos, comumente são não lineares (Barbi, 2015).

Para este tipo de sistema, muitas vezes são empregadas técnicas convencionais de controle linear, sendo a função de transferência da planta obtida matematicamente através de um ponto de operação específico. Esta metodologia é chamada de linearização. Para empregar a técnica de linearização do modelo médio não linear do conversor *Boost*, deve-se fixar seu ponto de operação ou a razão cíclica. Entretanto, em condições em que os estados do sistema de controle da planta não linear encontram-se afastados do ponto de operação projetado, a performance do conversor tende a ser comprometida, especialmente em períodos transitórios, visto que os requisitos de projeto não são satisfeitos. Assim sendo, é importante o desenvolvimento de metodologias de sistemas de controle que considerem tais não



linearidades, para que se obtenha um melhor desempenho do processo dinâmico a ser controlado.

Visando a obtenção de técnicas que pudessem representar estas não linearidades de forma mais apropriada, em (T. Takagi & M. Sugeno, 1985) propõe-se a modelagem Takagi-Sugeno *Fuzzy* (TSF), que tem como base a utilização da regra de implicação *Fuzzy* SE-ENTÃO. Esta técnica pode ser aplicada em diversas áreas, nas quais se têm o objetivo de alcançar um modelo mais próximo do real, pretendendo projetar controladores mais precisos considerando as não linearidades do sistema. Na literatura, foi constatado um crescimento significativo no número de trabalhos que tem como objetivo a garantia da estabilidade e o controle de sistemas não lineares (Tanaka, Ikeda, & Wang, 1998), (B. Chen, C. Tseng, & H. J. Uang, 2000), (H. D. Tuan, P. Apkarian, T. Narikiyo, & M. Kanota, 2004) e (C. S. Ting, P. Apkarian, T. Narikiyo, & M. Kanota, 2006).

Entretanto, é dificultoso definir as particularidades apresentadas pela resposta transitória, bem como garantir, mediante à perturbações e alterações nos parâmetros da planta, a estabilidade do sistema em malha fechada. Nesta situação, com o objetivo de facilitar a implementação da metodologia TSF, a técnica de Desigualdades Matriciais Lineares (*Linear Matrix Inequalities* – LMIs) surge como uma solução viável, posto que por meio de sua aplicação tem-se a possibilidade de estabelecer condições para a resposta em regime transitório e, além disso, definir condições de estabilidade robusta para os controladores. Desta maneira, empregou-se a técnica de LMIs para obtenção dos ganhos do controlador proposto. O uso das LMIs em projetos de controladores é largamente encontrado na literatura, como pode ser visto em (Andrea C. Q., 2007), (Teixeira, Pietrobom, & Assunção, 2000) e (Faria, 2009).

Fazendo complemento as técnicas utilizadas no projeto de controle, o uso de ferramentas de simulação é indispensável em situações que apresentam certa complexidade, como projetos destinados à sistemas dinâmicos e que fazem exigência de uma pequena margem para erros. Além disso as simulações trazem vantagens no tempo, custo e segurança, no desenvolvimento de controladores. Neste cenário, utiliza-se a abordagem de teste *FPGA in the Loop* para validação do algoritmo de controle proposto no presente trabalho, sendo esta uma ferramenta que proporciona a união das vantagens oferecidas pelas simulações realizadas em *software* e as



simulações realizadas em *hardware*. Destaca-se a utilização do FPGA *in the Loop* em diversas aplicações (Kumar, Ruber, & Jovanovic, 2017), (Gomez, 2013) e (Rogers, Kavasseri, & Smith, 2017).

1.2 OBJETIVOS

1.2.1 OBJETIVOS GERAIS

O objetivo geral deste Trabalho de Conclusão de Curso é a implementação de controle não linear em FPGA *in the Loop* para conversor *Boost*, baseado na técnica Takagi-Sugeno Fuzzy integrado ao Controlador de Modelagem Interna onde serão realizadas simplificações matemáticas no algoritmo de controle com a finalidade de reduzir do número de operações aritméticas e evitar problemas de representação no FPGA.

1.2.2 OBJETIVOS ESPECÍFICOS

Para a realização do objetivo geral deste trabalho destacam-se os seguintes objetivos específicos:

- Modelagem matemática do conversor *Boost* e representação em espaço de estados;
- Desenvolvimento do controle proposto utilizando as técnicas Takagi-Sugeno *Fuzzy* e Controle de Modelagem Interna;
- Aplicação de LMIs ao modelo de controle para determinação dos ganhos;
- Realização de simplificações matemáticas para determinação da lei de controle simplificada;
- Desenvolvimento do algoritmo de controle simplificado em linguagem VHDL para implementação em FPGA;
- Simulações em MATLAB/SIMULINK e validação do controle por teste FPGA *in the Loop*.
- Análise dos resultados obtidos e redação do Trabalho de Conclusão de Curso.



1.3 ORGANIZAÇÃO DO TRABALHO

No Capítulo 1 fez-se a contextualização do cenário brasileiro com enfoque na aplicação e controle do conversor CC-CC *Boost*, apresentando a problemática do controle não linear e fazendo a exposição de pesquisas realizadas na solução de tal. Além disso, são expostos os objetivos gerais e específicos do presente trabalho.

No Capítulo 2 é abordado o embasamento teórico para a realização do presente trabalho, onde são englobados os seguintes tópicos: estudo e modelagem do conversor CC-CC elevador de tensão (*Boost*), modelagem Takagi-Sugeno *Fuzzy*, projeto de controladores utilizando Desigualdades Matriciais Lineares baseados no teorema de estabilidade de Lyapunov, Controladores de Modelagem Interna e FPGA *in the Loop*.

No Capítulo 3 é exposto o desenvolvimento matemático realizado para obtenção do controlador proposto, baseado nos assuntos abordados no Capítulo 2. Ademais, será apresentada as simplificações matemáticas realizadas para a obtenção da lei de controle a ser implementada.

No Capítulo 4 são apresentados os resultados das simulações em *software* MATLAB/SIMULINK e, em seguida, com a finalidade de validar controle proposto, são expostos também os resultados dos testes FPGA *in the Loop*.

No Capítulo 5 faz-se as principais considerações relativas ao presente trabalho e apresentam-se as propostas referentes aos trabalhos futuros.

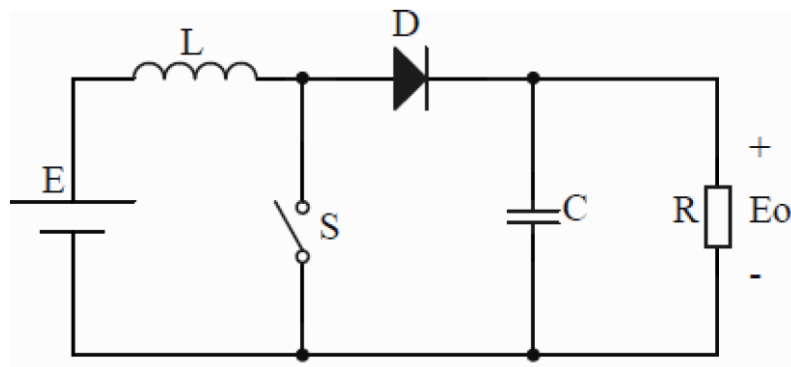
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, faz-se a abordagem do estudo teórico dos seguintes temas: Conversor Elevador de Tensão, também chamado de Conversor *Boost*, Sistemas Takagi-Sugeno *Fuzzy*, Projeto de Controladores Utilizando Desigualdades Matriciais Lineares, Controlador de Modelagem Interna e FPGA *in the Loop*.

2.1 CONVERSOR CC-CC ELEVADOR DE TENSÃO

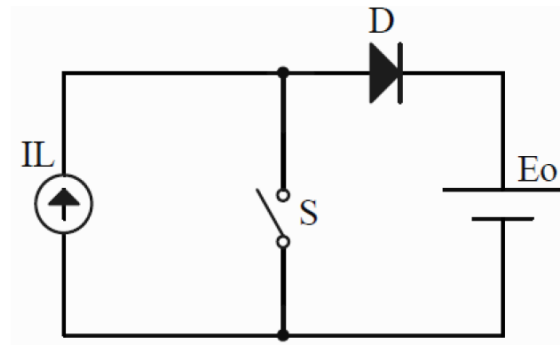
O conversor estático CC-CC elevador de tensão, ou conversor *Boost*, tem como principal característica a regulação da tensão média de saída para um nível igual ou superior ao de entrada. Sua estrutura básica é descrita a partir de uma fonte de tensão de entrada E , alocada em série com o indutor L , um dispositivo semiconductor representado pela chave S , um diodo D e um capacitor C em paralelo com a carga R . A topologia do conversor *Boost* pode ser visualizada conforme ilustrada na Figura 2.1.

Figura 2.1 - Conversor CC-CC Elevador de Tensão (*Boost*).



Fonte: (Assef, 2017).

Supondo o valor de C suficientemente grande, o comportamento da carga pode ser representado por uma f.e.m de valor E_o . Além disso, para frequências de chaveamento elevadas, a corrente no indutor pode ser considerada constante e igual a I_L (Martins & Barbi, 2006). Sendo assim, o circuito equivalente ao comportamento característico do elevador de tensão, pode ser representado pela Figura 2.2.

Figura 2.2 – Circuito Equivalente ao Comportamento do Conversor *Boost*.

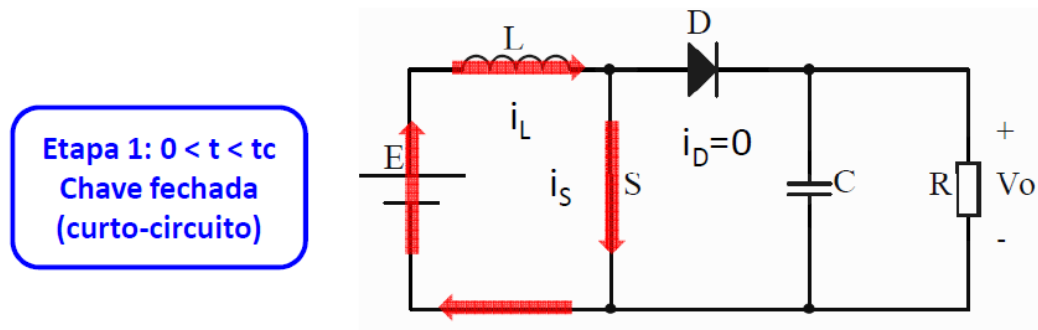
Fonte: (Assef, 2017).

2.1.1 CONVERSOR *BOOST* EM MODO DE CONDUÇÃO CONTÍNUA

No conversor *Boost*, a medida do grau de continuidade da corrente de entrada I_L , está relacionada com a quantidade de energia armazenada por meio do campo magnético do indutor de entrada L , durante o tempo em que a chave S se encontra em condução. A corrente no diodo D é sempre descontínua. Porém, a corrente I_L de entrada pode ser contínua ou descontínua. No modo de condução contínua (MCC), a corrente no indutor flui continuamente (Martins & Barbi, 2006).

Em MCC, o conversor *Boost* apresenta duas etapas de funcionamento. A primeira inicia-se quando a chave S é fechada ($0 \leq t < tc$), onde tc expressa tempo de carregamento. Neste período, o diodo D é polarizado inversamente, ou seja, a tensão do ânodo é mais negativa do que a tensão do cátodo, fazendo com que o mesmo não conduza, o que resulta na isolação entre a carga e a fonte de alimentação. Nestas condições, a fonte de tensão e o indutor são fechados em malha por meio da chave S , o que faz com que durante esse tempo ocorra o fornecimento de energia da fonte E para o indutor L , e conseqüentemente, o carregamento deste. De maneira semelhante, o capacitor C e a carga R também são fechados em malha, descarregando o capacitor sua energia armazenada na carga (Martins & Barbi, 2006).

Figura 2.3 - Primeira Etapa de Funcionamento do *Boost* em MCC.



Fonte: (Assef, 2017).

Para efeitos de simplificação, a análise dos circuitos será realizada levando em consideração que os elementos eletromagnéticos e semicondutores são ideais. Desta forma, por meio do estudo do circuito apresentado na Figura 2.3, que representa a primeira etapa de funcionamento do conversor *Boost* em MCC, têm-se para os valores das correntes (Pr. Dr. Moacyr de Brito, 2017):

$$I_L(t) = I_S(t) \quad (2.1)$$

$$I_D(t) = 0 \quad (2.2)$$

$$I_C(t) = -I_O(t) \quad (2.3)$$

$$I_L(t) = I_m + \frac{E}{L}(t) \quad (2.4)$$

$$I_O(t) = \frac{V_O}{R}(t) \quad (2.5)$$

Sendo:

- $I_L(t)$ a corrente no indutor L .
- $I_S(t)$ a corrente na chave S .
- $I_D(t)$ a corrente no diodo D .
- $I_C(t)$ a corrente no capacitor C .
- $I_O(t)$ a corrente na carga R .

E para os valores das tensões (Pr. Dr. Moacyr de Brito, 2017):



$$V_C(t) = V_O(t) = V_D(t) \quad (2.6)$$

$$E(t) = V_L(t) = \frac{LdI_L}{dt}(t) \quad (2.7)$$

Sendo:

- $E(t)$ a tensão da fonte E .
- $V_L(t)$ a tensão no indutor L .
- $V_C(t)$ a tensão no capacitor C .
- $V_D(t)$ a tensão no diodo D .
- $V_O(t)$ a tensão na carga R .

Como a corrente no indutor cresce linearmente a partir do momento em que a chave S é fechada, e sabe-se que o seu valor máximo é alcançado ao fim da primeira etapa em $t = tc$ (Martins & Barbi, 2006), então pode-se deduzir que a corrente máxima, com base na Equação (2.4), será:

$$I_M(t) = I_m + \frac{E}{L}(tc) \quad (2.8)$$

Sendo $I_M(t)$ a corrente máxima de entrada no indutor L .

Após decorrido o tempo tc de carregamento do indutor, inicia-se a segunda etapa no momento em que a chave S é aberta ($tc \leq t < T$), onde T expressa o período de chaveamento total. Por conseguinte, tem-se que o tempo que representa o tempo total de abertura é dado pela Equação (2.9) :

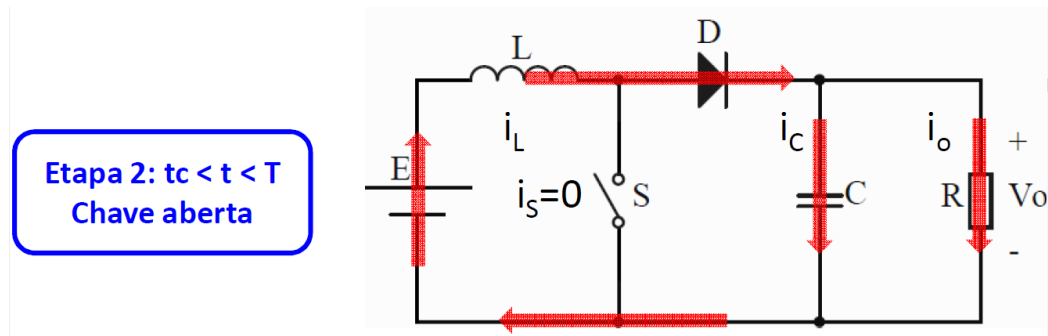
$$ta = T - tc \quad (2.9)$$

Sendo ta o tempo total de abertura da chave S .

Neste período, o diodo D é polarizado diretamente, ou seja, a tensão do ânodo é mais positiva do que a tensão do cátodo, fazendo com que o mesmo passe a conduzir e, como consequência, a fonte de corrente I_L passe a entregar energia a

fonte E_o (Martins & Barbi, 2006). Esta etapa é concluída quando a chave S é fechada, repetindo-se assim, o ciclo novamente. A segunda etapa pode ser visualizada conforme o circuito exposto na Figura 2.4.

Figura 2.4 - Segunda Etapa de Funcionamento do Boost em MCC.



Fonte: (Assef, 2017).

A partir da análise do circuito da Figura 2.4, são obtidas as seguintes equações para os valores das correntes (Pr. Dr. Moacyr de Brito, 2017):

$$I_S(t) = 0 \quad (2.10)$$

$$I_C(t) = I_L(t) - I_O(t) \quad (2.11)$$

$$I_D(t) = I_L(t) = I_M + \frac{(E - V_O)}{L}(t) \quad (2.12)$$

$$I_O(t) = \frac{V_O}{R}(t) \quad (2.13)$$

E para os valores das tensões (Pr. Dr. Moacyr de Brito, 2017):

$$E(t) - V_O(t) = V_L(t) \quad (2.14)$$

$$V_S(t) = V_O(t) = V_C(t) \quad (2.15)$$

$$V_L(t) = \frac{LdI_L}{dt}(t) \quad (2.16)$$

$$V_D(t) = 0 \quad (2.17)$$

Dado que nesta etapa o indutor entrega energia a carga, sua corrente decai linearmente a partir do momento em que a chave S é fechada em $t = t_c$, e seu valor

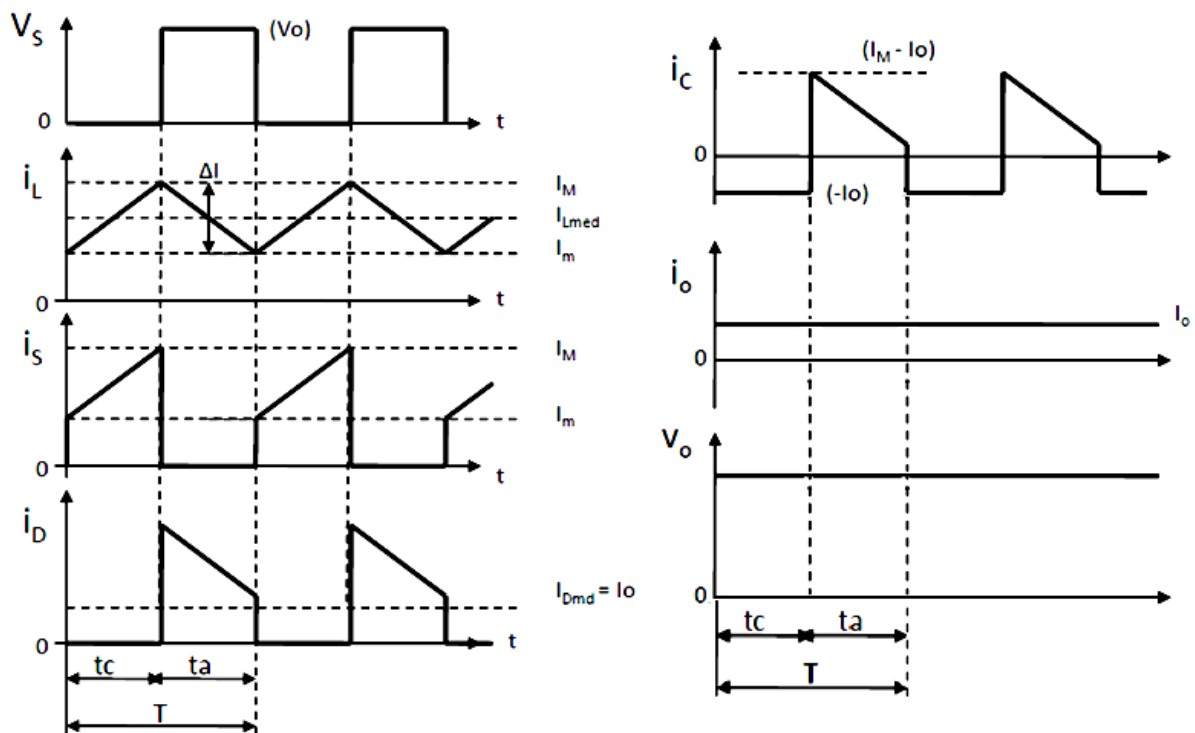
mínimo é alcançado ao final de t_a (Martins & Barbi, 2006). Portanto, a partir da Equação (2.12), a corrente mínima de entrada no indutor é dada por:

$$I_m(t) = I_M + \frac{(E - V_o)}{L} (ta) \quad (2.18)$$

Sendo $I_m(t)$ a corrente mínima de entrada no indutor L .

As principais formas de onda do conversor *Boost* em MCC são graficamente descritas na Figura 2.5.

Figura 2.5 - Principais Formas de Onda do Conversor *Boost* em MCC.



Fonte: (Assef, 2017).

2.1.2 GANHO ESTÁTICO



Para a obtenção do ganho estático G do conversor *Boost*, considera-se a chave S operando com frequência f fixa e razão cíclica D variável. Sendo o conceito de potência elétrica gerada definido pela quantidade de energia produzida em um dado intervalo de tempo, a energia W_E , fornecida pela fonte E , pode ser obtida por meio da Equação (2.19).

$$W_E = P_E \cdot T = E \cdot I_L \cdot T \quad (2.19)$$

Por outro lado, a energia W_O , recebida pela fonte E_O , é dada pela Equação (2.20).

$$W_O = P_O \cdot ta = E_O \cdot I_L \cdot ta \quad (2.20)$$

A partir da Equação (2.10), faz-se a substituição do valor de ta para que a energia W_O seja expressada em função do período de chaveamento total T . A Equação (2.21) é então alcançada.

$$W_O = P_O \cdot ta = E_O \cdot I_L \cdot (T - tc) \quad (2.21)$$

Considerando o sistema do conversor *Boost* como ideal, ou seja, onde não há perdas na transferência de energia, têm-se que toda a energia produzida é consumida. Sendo assim:

$$W_E = W_O \quad (2.22)$$

Logo, fazendo a substituição dos valores:

$$E \cdot I_L \cdot T = E_O \cdot I_L \cdot (T - tc) \quad (2.23)$$

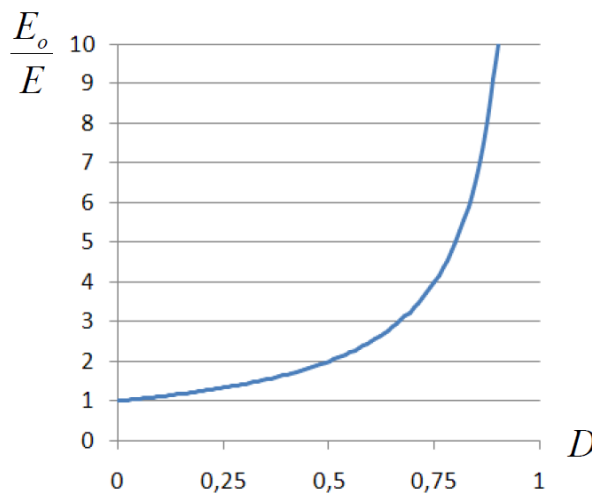
Desta maneira, o ganho estático G do conversor *Boost* pode ser obtido através da Equação (2.24).

$$G = \frac{E_o}{E} = \frac{1}{1 - D} \quad (2.24)$$

Sendo a razão cíclica D , a razão entre o tempo em que a chave S permanece ligada e o período de total de chaveamento, ou seja, $D = tc/T$, podendo esta grandeza variar de zero (chave aberta em todo o período T) ou um (chave fechada em todo período T).

A Equação (2.24), descreve a característica ideal de transferência do conversor *Boost* e está retratada graficamente na Figura 2.6.

Figura 2.6 - Característica Ideal de Transferência Estática do Conversor *Boost*.



Fonte: (Assef, 2017).

Analisando o gráfico apresentado na Figura 2.6, observa-se que, quando a razão cíclica D tende a unidade, a tensão de saída E_o encaminha-se, teoricamente, a um valor infinito, já que esta está alocada no numerador da fração que expressa o ganho estático. Além disso, pode ser observado que, a tensão mínima de saída, é igual a tensão de entrada, ocasionando um ganho estático unitário.

2.2 MODELAGEM MATEMÁTICA DO CONVERSOR CC-CC *BOOST* EM ESPAÇO DE ESTADOS



Como apresentado em (Carniato, 2009) o modelo instantâneo do conversor CC-CC *Boost* considera somente, o funcionamento do conversor baseado em seu chaveamento, ou seja, no momento em que a chave S está fechada, o circuito desenvolve um comportamento distinto em comparação com o período em que a chave S está aberta.

Sob a perspectiva do projeto de controle, a variável de interesse é a razão cíclica D , contudo, esta variável não é levada em consideração no modelo instantâneo. Sendo assim, emprega-se o modelo médio do conversor *Boost*, tendo em vista que esta metodologia é descrita considerando a variável de controle D .

2.2.1 MODELO MÉDIO DO CONVERSOR CC-CC *BOOST*

A técnica de modelagem a partir do modelo médio, como o próprio nome já expressa, leva em conta os valores médios das variáveis de controle e das variáveis de estado. Levando em consideração que um intervalo de comutação é definido por $[t - T, T]$ e tomando como base o modelo médio do conversor *Boost*, a razão cíclica média D é definida a partir da Equação (2.25).

$$D(t) = \frac{1}{T} \int_{t-T}^T p(\tau) d\tau \quad (2.25)$$

Sendo:

- $p(\tau)$ o posicionamento da chave S (fechada ou aberta), fazendo a diferenciação das duas etapas de funcionamento do conversor *Boost*. Além disso, $p(\tau)$ é periódica com período igual a T .

Da mesma forma, os valores médios da tensão no capacitor V_C e da corrente no indutor I_L , que são as variáveis de estados escolhidas para a modelagem, são estabelecidas, respectivamente, por meio das Equações (2.26) e (2.27).

$$\overline{V}_C(t) = \frac{1}{T} \int_{t-T}^T V_C(\tau) d\tau \quad (2.26)$$



$$\bar{I}_L(t) = \frac{1}{T} \int_{t-T}^T I_L(\tau) d\tau \quad (2.27)$$

Por conseguinte, tomando como base as Equações (2.26) e (2.27), é possível o levantamento das equações que expressam o modelo a partir dos valores médios de V_C e I_L , conforme segue nas Equações (2.28) e (2.29). Salienta-se que a partir daqui, que a tensão de entrada E do conversor *Boost* mencionada na seção 2.1 será denominada V_{in} .

$$\frac{d\bar{V}_C(t)}{dt} = \frac{1}{C} \left[\bar{I}_L(t) - \bar{p}\bar{I}_L(t) - \frac{\bar{V}_C(t)}{R} \right] \quad (2.28)$$

$$\frac{d\bar{I}_L(t)}{dt} = \frac{1}{L} [\bar{p}\bar{V}_C(t) - \bar{V}_C(t) + V_{in}(t)] \quad (2.29)$$

Durante um intervalo de comutação, os valores instantâneos da tensão no capacitor e da corrente no indutor, são aproximadamente, iguais aos seus valores médios. Consequentemente, podemos chegar as Equações (2.30) e (2.31).

$$\bar{p}\bar{V}_C(t) \cong \bar{p}(t) \cdot \bar{V}_C(t) = D(t) \cdot \bar{V}_C(t) \quad (2.30)$$

$$\bar{p}\bar{I}_L(t) \cong \bar{p}(t) \cdot \bar{I}_L(t) = D(t) \cdot \bar{I}_L(t) \quad (2.31)$$

Sendo assim, podemos fazer a transformação das Equações (2.28) e (2.29), respectivamente em (2.32) e (2.33).

$$\frac{d\bar{V}_C(t)}{dt} = \frac{1}{C} \left[(1 - D(t))\bar{I}_L(t) - \frac{\bar{V}_C(t)}{R} \right] \quad (2.32)$$

$$\frac{d\bar{I}_L(t)}{dt} = \frac{1}{L} [D(t) \bar{V}_C(t) - \bar{V}_C(t) + V_{in}(t)] \quad (2.33)$$

Desta forma, são consideradas as novas variáveis de estado como $\bar{I}_L(t)$ e $\bar{V}_C(t)$ e, portanto, o vetor de estado é definido pela Equação (2.34).

$$x(t) = [x_1(t) \ x_2(t)]^T = [\bar{I}_L(t) \ \bar{V}_C(t)]^T \quad (2.34)$$

Logo, as Equações (2.32) e (2.33) podem ser representadas através da modelagem em espaço de estados de diversas formas. Através de manipulações matemáticas, os modelos médios não lineares gerados podem apresentar mudanças quanto ao número de não linearidades de que dependem, sendo o objetivo, a redução desse número. Três modelos distintos são propostos em (Carniato, 2009) conforme segue:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} + \frac{V_{in}(t)}{Lx_2(t)} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{x_2(t)}{L} \\ -\frac{x_1(t)}{C} \end{bmatrix} D(t) \quad (2.35)$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{x_2(t)}{L} + \frac{V_{in}(t)}{LD(t)} \\ -\frac{x_1(t)}{C} \end{bmatrix} D(t) \quad (2.36)$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{RC} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} \frac{x_2(t)}{L} + \frac{V_{in}x_2(t)}{L(x_2(t) - V_{in}(t))} \\ -\frac{x_1(t)}{C} \end{bmatrix} D(t) \quad (2.37)$$

2.3 SISTEMAS TAKAGI-SUGENO FUZZY

A metodologia de modelagem Takagi-Sugeno *Fuzzy*, traz a possibilidade da modelagem de um sistema não linear, de maneira que este seja descrito como uma combinação de modelos lineares locais que não variam no tempo. Estes modelos locais tem a capacidade de descrever, de forma aproximada, o comportamento do sistema não linear em pontos diferentes de operação (Teixeira, Pietrobon, & Assunção, 2000).

Em diversas aplicações, o projeto de controladores fundamentados no mecanismo da modelagem TSF, é elaborado por meio do uso de Desigualdades Lineares Matriciais (*Linear Matrix Inequalities* – LMI em Inglês). Além disso, em busca

da obtenção de sistemas de controle mais robustos e com maior precisão, é feita a integração do CMI (Controlador de Modelagem Interna) com a técnica TSF.

2.3.1 REPRESENTAÇÃO DE SISTEMAS NÃO LINEARES POR MEIO DO USO DE MODELOS LOCAIS

Seja $g(x)$ uma função não linear a modelar. É possível realizar a linearização de $g(x)$ em torno de um ponto médio de operação x_0 . Para isso, são utilizadas séries de Taylor a fim de executar a aproximação desejada, conforme segue na Equação (2.38).

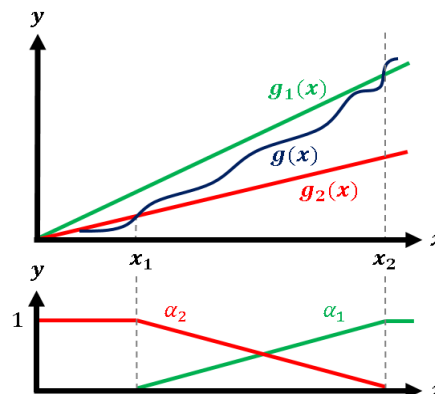
$$g(x) = g(x_0) + (x - x_0) \left. \frac{\partial g(x)}{\partial x} \right|_{x=x_0} + \dots \quad (2.38)$$

Contudo, como já dito, esta aproximação pode trazer um desempenho insatisfatório para pontos de operação diferentes de x_0 .

Sendo assim, faz-se a opção pela técnica de modelagem TSF para que se trate tais não linearidades. Na Figura 2.7 pode ser visualizado o conceito desta metodologia, onde, as funções lineares $g_1(x)$ e $g_2(x)$ delimitam uma faixa de operação para os valores de $g(x)$ de acordo com a Equação (2.39) e sendo $x \in [x_1, x_2]$.

$$g_2(x) \leq g(x) \leq g_1(x) \quad (2.39)$$

Figura 2.7 - Conceito da metodologia da modelagem TSF.



Fonte: Próprio Autor.



Seja $g_1(x)$ e $g_2(x)$ os modelos lineares locais, e $\alpha_1(x)$ e $\alpha_2(x)$ as funções de pertinência, pode-se descrever a função $g(x)$ com base na Figura 2.7 como:

$$g(x) = \alpha_1(x)g_1(x) + \alpha_2(x)g_2(x) \quad (2.40)$$

As funções de pertinência devem obedecer às seguintes propriedades:

$$\begin{cases} \alpha_i(x) \geq 0 \\ \sum_{i=1}^n \alpha_i(x) = 1 \end{cases} \quad (2.41)$$

Para a obtenção dos modelos locais bem como as funções de pertinência, será utilizada o método de modelagem exata que é largamente utilizado na literatura, vide (Carniato, 2009), (García, 2015), (Andrea, Pinto, Carniato, Godoy, & Rodrigues, 2009) e (Andrea, Batista, Leite, Pinto, & Coelho, 2014).

2.3.2 MODELAGEM TAKAGI-SUGENO *FUZZY*

A descrição de uma planta não linear a partir da técnica TSF é feita por um conjunto de implicações *Fuzzy* SE-ENTÃO, que fazem a representação local de relações lineares entre a entrada e a saída desse sistema. Tal representação local da planta dinâmica a ser controlada, está disponível nos termos dos modelos lineares locais, como segue a estrutura (T. Takagi & M. Sugeno, 1985):

$$\begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \quad (2.42)$$

sendo $i = 1, 2, \dots, r$ o i -ésimo modelo linear, o vetor de estados $x(t) \in \mathbb{R}^n$, o vetor de entrada $u(t) \in \mathbb{R}^m$, o vetor de saída $y(t) \in \mathbb{R}^q$, enquanto que $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times m}$ e $C_i \in \mathbb{R}^{q \times n}$ são as matrizes que definem o modelo local.



Deste modo, faz-se a junção da informação acima com as regras SE-ENTÃO existentes, onde a i -ésima regra pode ser estabelecida da seguinte maneira:

$$\begin{aligned} \text{Regra } i: & \text{ Se } z_1(t) \text{ é } \Psi_1^i \text{ e ... e Se } z_p(t) \text{ é } \Psi_p^i, \\ \text{ENTÃO } & \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \end{aligned} \quad (2.43)$$

sendo Ψ_j^i com $j = 1, 2, \dots, p$ o conjunto *Fuzzy* j da regra i e $z_1(t), \dots, z_p(t)$ são as variáveis premissas.

O modelo *Fuzzy* global do sistema é alcançado a partir da combinação *Fuzzy* dos modelos lineares locais. Fazendo a defuzzificação a partir da técnica de centro de gravidade, tem-se:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^r \alpha_i A_i x(t) + \sum_{i=1}^r \alpha_i B_i u(t) \\ y(t) = \sum_{i=1}^r \alpha_i C_i x(t) \end{cases} \quad (2.44)$$

sendo α_i a função de pertinência do i -ésimo modelo local.

A partir das propriedades previamente descritas na Equação (2.41), chega-se a seguinte constatação:

$$\begin{cases} A(x) = \sum_{i=1}^r \alpha_i A_i x(t) \\ B(x) = \sum_{i=1}^r \alpha_i B_i u(t) \\ C(x) = \sum_{i=1}^r \alpha_i C_i x(t) \end{cases} \quad (2.45)$$

Como explanado na seção 2.3.1, as funções de pertinência serão obtidas por meio da técnica de modelagem exata.



Primeiramente, para a obtenção das funções de pertinência, devem ser identificados, nas matrizes $A(x)$, $B(x)$ e $C(x)$ do modelo original, os termos que expressam as não linearidades, ou seja, os termos não constantes. Para h termos não constantes, sendo f_k ($k = 1, \dots, h$), então o modelo TSF da planta a ser controlada terá $r = 2^h$ modelos locais.

Em seguida, devem ser estabelecidos os valores máximos e mínimos previstos para os termos f_k como segue:

$$f_{k1} = \text{máx}(f_k) \quad (2.46)$$

$$f_{k2} = \text{mín}(f_k) \quad (2.47)$$

Feito isso, faz-se a definição das variáveis σ_{k1} e σ_{k2} conforme as Equações (2.48) e (2.49).

$$\sigma_{k1} = \frac{f_k - f_{k2}}{f_{k1} - f_{k2}} \quad (2.48)$$

$$\sigma_{k2} = \frac{f_{k1} - f_k}{f_{k1} - f_{k2}} \quad (2.49)$$

A respeito de σ_{k1} e σ_{k2} é possível a verificação das seguintes propriedades:

$$\begin{cases} \sigma_{k1} \geq 0 \\ \sigma_{k2} \geq 0 \end{cases} \quad (2.50)$$

$$\sigma_{k1} + \sigma_{k2} = \frac{f_k - f_{k2}}{f_{k1} - f_{k2}} + \frac{f_{k1} - f_k}{f_{k1} - f_{k2}} = 1 \quad (2.51)$$

$$\sigma_{k1}f_{k1} + \sigma_{k2}f_{k2} = \left(\frac{f_k - f_{k2}}{f_{k1} - f_{k2}}\right)f_{k1} + \left(\frac{f_{k1} - f_k}{f_{k1} - f_{k2}}\right)f_{k2} = f_k \quad (2.52)$$

$$\sigma_{k1} \prod_{\substack{l=1 \\ l \neq k}}^h (\sigma_{l1} + \sigma_{l2}) = \sigma_{k1} \quad (2.53)$$

$$\sigma_{k2} \prod_{\substack{l=1 \\ l \neq k}}^h (\sigma_{l1} + \sigma_{l2}) = \sigma_{k2} \quad (2.54)$$

Fazendo a substituição das Equações (2.53) e (2.54) na Equação (2.52), tem-se:

$$\sigma_{k1} \left[\prod_{\substack{l=1 \\ l \neq k}}^h (\sigma_{l1} + \sigma_{l2}) \right] f_{k1} + \sigma_{k2} \left[\prod_{\substack{l=1 \\ l \neq k}}^h (\sigma_{l1} + \sigma_{l2}) \right] f_{k2} = f_k \quad (2.55)$$

Substituindo cada termo não constante f_k ($k = 1, \dots, h$) na Equação (2.55) e fazendo as devidas manipulações matemáticas, o resultado faz equivalência à Equação (2.44). Entretanto, quando o número de modelos locais é muito grande, este mecanismo se torna muito difícil.

Desta forma, opta-se por uma metodologia mais atrativa para a obtenção das funções de pertinência e modelos locais, sendo que, os resultados alcançados a partir desta, são equivalentes aos atingidos conforme a Equação (2.55).

Conforme já dito anteriormente, a partir da análise da Figura 2.7, têm-se que as funções que fazem a delimitação da parte superior e inferior de uma função não linear podem ser utilizadas como modelos locais. Esse conceito, pode ser empregado com a finalidade de se obter os modelos locais e as funções de transferência da planta em estudo. Desta maneira, para que esta metodologia seja compreendida de uma forma mais simplificada, será desenvolvido um exemplo na sequência.

Seja a modelagem de uma planta em espaço de estados expresso na Equação (2.56).

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} -1 & f_1 \\ 2 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 2 \\ f_2 \end{bmatrix} u(t) \\ y(t) = [1 \quad 0] x(t) \end{cases} \quad (2.56)$$

Como pode ser observado, o sistema apresenta duas não linearidades geradas pelos termos f_1 e f_2 . A partir disso, define-se os valores máximos e mínimos para estes termos conforme segue:

$$\begin{cases} -1 \leq f_1 \leq 1 \\ -2 \leq f_2 \leq 2 \end{cases} \quad (2.57)$$



Desta forma, pode-se fazer a definição dos termos f_{11} , f_{12} , f_{21} e f_{22} conforme descrito nas Equações (2.46) e (2.47).

$$\begin{aligned} f_{11} &= \max(f_1) = 1 & f_{21} &= \max(f_2) = 2 \\ f_{12} &= \min(f_1) = -1 & f_{22} &= \min(f_2) = -2 \end{aligned} \quad (2.58)$$

Dando prosseguimento, é realizado o cálculo dos termos σ_{11} , σ_{12} , σ_{21} e σ_{22} com base nas Equações (2.48) e (2.49).

$$\begin{aligned} \sigma_{11} &= \frac{f_1 - (-1)}{1 - (-1)} = \frac{f_1 + 1}{2} \\ \sigma_{12} &= \frac{1 - f_1}{1 - (-1)} = \frac{1 - f_1}{2} \\ \sigma_{21} &= \frac{f_2 - (-2)}{2 - (-2)} = \frac{f_2 + 2}{4} \\ \sigma_{22} &= \frac{2 - f_2}{2 - (-2)} = \frac{2 - f_2}{4} \end{aligned} \quad (2.59)$$

Analisando a Equação (2.59), pode ser observado que $\sigma_{11} + \sigma_{12} = \sigma_{21} + \sigma_{22} = 1$, e que todos os termos são maiores ou iguais a zero, o que satisfaz as condições apresentadas nas Equações (2.50) e (2.51). Deste modo, podemos utilizar a propriedade descrita na Equação (2.55) para descrever os termos não constantes f_1 e f_2 , conforme segue:

$$\begin{aligned} f_1 &= \sigma_{11}(\sigma_{21} + \sigma_{22})f_{11} + \sigma_{12}(\sigma_{21} + \sigma_{22})f_{12} \\ f_1 &= \sigma_{11}\sigma_{21}f_{11} + \sigma_{11}\sigma_{22}f_{11} + \sigma_{12}\sigma_{21}f_{12} + \sigma_{12}\sigma_{22}f_{12} \\ f_1 &= \alpha_1 f_{11} + \alpha_2 f_{11} + \alpha_3 f_{12} + \alpha_4 f_{12} \end{aligned} \quad (2.60)$$

$$\begin{aligned} f_2 &= \sigma_{21}(\sigma_{11} + \sigma_{12})f_{21} + \sigma_{22}(\sigma_{11} + \sigma_{12})f_{22} \\ f_2 &= \sigma_{21}\sigma_{11}f_{21} + \sigma_{21}\sigma_{12}f_{21} + \sigma_{22}\sigma_{11}f_{22} + \sigma_{22}\sigma_{12}f_{22} \\ f_2 &= \alpha_1 f_{21} + \alpha_2 f_{22} + \alpha_3 f_{21} + \alpha_4 f_{22} \end{aligned} \quad (2.61)$$

Sendo:

$$\begin{aligned} \alpha_1 &= \sigma_{11}\sigma_{21} \\ \alpha_2 &= \sigma_{11}\sigma_{22} \end{aligned} \quad (2.62)$$



$$\alpha_3 = \sigma_{12} \sigma_{21}$$

$$\alpha_4 = \sigma_{12} \sigma_{22}$$

As funções de pertinência resultantes, expressadas na Equação (2.62), satisfazem as propriedades descritas nas Equação (2.41), conforme segue:

$$\begin{aligned} 0 \leq \sigma_{11} \wedge 0 \leq \sigma_{21} &\rightarrow \alpha_1 = \sigma_{11} \sigma_{21} \geq 0 \\ 0 \leq \sigma_{11} \wedge 0 \leq \sigma_{22} &\rightarrow \alpha_2 = \sigma_{11} \sigma_{22} \geq 0 \\ 0 \leq \sigma_{12} \wedge 0 \leq \sigma_{21} &\rightarrow \alpha_3 = \sigma_{12} \sigma_{21} \geq 0 \\ 0 \leq \sigma_{12} \wedge 0 \leq \sigma_{22} &\rightarrow \alpha_4 = \sigma_{12} \sigma_{22} \geq 0 \end{aligned} \quad (2.63)$$

$$\begin{aligned} \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 &= \sigma_{11} \sigma_{21} + \sigma_{11} \sigma_{22} + \sigma_{12} \sigma_{21} + \sigma_{12} \sigma_{22} \\ &= (\sigma_{11} \sigma_{12})(\sigma_{21} \sigma_{22}) = 1 \end{aligned} \quad (2.64)$$

Efetuada a substituição dos termos não constantes f_1 e f_2 , descritos nas Equações (2.60) e (2.61), na Equação (2.56), que representa o modelo em espaço de estados da planta, têm-se como resultado:

$$\mathbf{A} = \begin{bmatrix} -1 & f_1 \\ 2 & -2 \end{bmatrix} = \begin{bmatrix} -1 & \alpha_1 f_{11} + \alpha_2 f_{11} + \alpha_3 f_{12} + \alpha_4 f_{12} \\ 2 & -2 \end{bmatrix} \quad (2.65)$$

$$\mathbf{A} = \alpha_1 \begin{bmatrix} -1 & f_{11} \\ 2 & -2 \end{bmatrix} + \alpha_2 \begin{bmatrix} -1 & f_{11} \\ 2 & -2 \end{bmatrix} + \alpha_3 \begin{bmatrix} -1 & f_{12} \\ 2 & -2 \end{bmatrix} + \alpha_4 \begin{bmatrix} -1 & f_{12} \\ 2 & -2 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 2 \\ f_2 \end{bmatrix} = \begin{bmatrix} 2 \\ \alpha_1 f_{21} + \alpha_2 f_{22} + \alpha_3 f_{21} + \alpha_4 f_{22} \end{bmatrix} \quad (2.66)$$

$$\mathbf{B} = \alpha_1 \begin{bmatrix} 2 \\ f_{21} \end{bmatrix} + \alpha_2 \begin{bmatrix} 2 \\ f_{22} \end{bmatrix} + \alpha_3 \begin{bmatrix} 2 \\ f_{21} \end{bmatrix} + \alpha_4 \begin{bmatrix} 2 \\ f_{22} \end{bmatrix}$$

Dando prosseguimento, basta definir a matriz de saída \mathbf{C} do modelo. Já que esta matriz é constante, sendo $\mathbf{C} = [1 \ 0]$, pode ser realizada a seguinte manipulação a partir da propriedade das funções de pertinência descrita em (2.41):

$$\sum_{i=1}^n \alpha_i(x) = 1 \rightarrow \mathbf{C} = [1 \quad 0] = \sum_{i=1}^4 \alpha_i [1 \quad 0] \quad (2.67)$$

Finalmente, o modelo Takagi-Sugeno *Fuzzy* do sistema descrito em (2.56), tem sua composição dada pelas Equações (2.65), (2.66) e (2.67), onde, a partir da Equação (2.44), têm-se:

$$\begin{cases} \dot{\mathbf{x}}(t) = \sum_{i=1}^4 \alpha_i \mathbf{A}_i \mathbf{x}(t) + \sum_{i=1}^4 \alpha_i \mathbf{B}_i \mathbf{u}(t) \\ \mathbf{y}(t) = \sum_{i=1}^4 \alpha_i \mathbf{C}_i \mathbf{x}(t) \end{cases} \quad (2.68)$$

Sendo:

$$\mathbf{A}_1 = \begin{bmatrix} -1 & f_{11} \\ 2 & -2 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} -1 & f_{11} \\ 2 & -2 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} -1 & f_{12} \\ 2 & -2 \end{bmatrix}, \mathbf{A}_4 = \begin{bmatrix} -1 & f_{12} \\ 2 & -2 \end{bmatrix} \quad (2.69)$$

$$\mathbf{B}_1 = \begin{bmatrix} 2 \\ f_{21} \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 2 \\ f_{22} \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 2 \\ f_{21} \end{bmatrix}, \mathbf{B}_4 = \begin{bmatrix} 2 \\ f_{22} \end{bmatrix} \quad (2.70)$$

$$\mathbf{C}_1 = \mathbf{C}_2 = \mathbf{C}_3 = \mathbf{C}_4 = [1 \quad 0] \quad (2.71)$$

Sendo as funções de pertinência descritas conforme a Equação (2.62), é possível definir os modelos locais do sistema do seguinte modo:

- Modelo local 1:

$$\mathbf{A}_1 = \begin{bmatrix} -1 & f_{11} \\ 2 & -2 \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} 2 \\ f_{21} \end{bmatrix}, \mathbf{C}_1 = [1 \quad 0], \alpha_1 = \sigma_{11} \sigma_{21} \quad (2.72)$$

- Modelo local 2:

$$\mathbf{A}_2 = \begin{bmatrix} -1 & f_{11} \\ 2 & -2 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 2 \\ f_{22} \end{bmatrix}, \mathbf{C}_2 = [1 \quad 0], \alpha_2 = \sigma_{11} \sigma_{22} \quad (2.73)$$



- Modelo local 3:

$$\mathbf{A}_3 = \begin{bmatrix} -1 & f_{12} \\ 2 & -2 \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 2 \\ f_{21} \end{bmatrix}, \mathbf{C}_3 = [1 \quad 0], \alpha_3 = \sigma_{12} \sigma_{21} \quad (2.74)$$

- Modelo local 4:

$$\mathbf{A}_4 = \begin{bmatrix} -1 & f_{12} \\ 2 & -2 \end{bmatrix}, \mathbf{B}_4 = \begin{bmatrix} 2 \\ f_{22} \end{bmatrix}, \mathbf{C}_4 = [1 \quad 0], \alpha_4 = \sigma_{12} \sigma_{22} \quad (2.75)$$

Pode-se observar que as Equações (2.72), (2.73), (2.74) e (2.75), que definem os modelos locais, apresentam uma relação entre os índices das variáveis de máximo e mínimo dos termos não constantes f_{km} ($f, m = 1, 2$), que fazem a substituição dos termos f_k nas matrizes \mathbf{A} e \mathbf{B} , e os índices das funções σ_{km} , que compõem as funções de pertinência de cada modelo local. Esta mesma relação pode ser observada para modelos TSF que possuem um número maior de termos não lineares.

Desta maneira, um algoritmo proposto para o levantamento dos modelos locais e as funções de pertinência é dado conforme segue:

1ª Etapa: Representar o sistema não linear da planta em estudo em espaço de estados.

2ª Etapa: Identificar os termos f_k que contém as não linearidades nas matrizes $\mathbf{A}(x)$, $\mathbf{B}(x)$ e $\mathbf{C}(x)$ do modelo levantado na 1ª Etapa, sendo h o número de termos f_k identificados, onde $(k = 1, \dots, h)$.

3ª Etapa: Calcular o número de modelos locais r necessários, sendo $r = 2^h$.

4ª Etapa: Definir os valores máximos e mínimos dos termos f_k , conforme as Equações (2.46) e (2.47).

5ª Etapa: Definir as variáveis que compõem as funções de pertinência σ_{k1} e σ_{k2} conforme as Equações (2.48) e (2.49).

6ª Etapa: Calcular os termos f_k em função das funções de pertinência e dos seus valores máximos e mínimos, com base na Equação (2.55).



7ª Etapa: Levantar os modelos locais a partir da substituição dos termos f_k calculados na 6ª Etapa, sendo o número de modelos locais obtidos igual ao calculado na 3ª Etapa.

Por fim, constata-se que o i -ésimo modelo local, composto pelos termos f_{ka} , f_{kb} , ..., f_{kn} , onde $(a, b, \dots, n = 1, 2)$, terá como função de pertinência $\alpha_i = \sigma_{ka} * \sigma_{kb} * \dots * \sigma_{kn}$, conforme pode ser observado nas Equações do exemplo desenvolvido (2.72), (2.73), (2.74) e (2.75).

2.4 TEOREMA DE ESTABILIDADE DE LYAPUNOV

Visando a descrição da técnica de Desigualdades Matriciais Lineares que será abordada nas seções 2.4 e 2.5, deve-se entender, primeiramente, o conceito de estabilidade no sentido de Lyapunov.

Seja a região no espaço Euclidiano Ω e seja um vetor de estado $x(t) \in \Omega$, faz-se a definição das seguintes funções (Chen, 1999):

Função positiva definida: Uma função escalar $V(x)$ é positiva definida em uma região Ω se $V(x) > 0$ para todo $x \neq 0$ em Ω e $V(0) = 0$.

Função positiva semidefinida: Uma função escalar $V(x)$ é positiva semidefinida em uma região Ω se $V(x) \geq 0$ para $x \neq 0$ em Ω e $V(0) = 0$.

Função negativa definida: Uma função escalar $V(x)$ é negativa definida se $-V(x)$ for positiva definida.

Função negativa semidefinida: Uma função escalar $V(x)$ é negativa semidefinida se $-V(x)$ for positiva semidefinida.

Da mesma forma, por meio da forma quadrática $V(x) = x^T P x$ pode-se fazer a classificação da matriz real quadrada simétrica P ($P = P^T$) (Chen, 1999):

Matriz positiva definida: P é positiva definida ($P > 0$), se $x^T P x > 0$, para todo $x \neq 0$ e $x^T P x = 0$ somente quando $x = 0$ (vetor nulo).

Matriz positiva semidefinida: P é positiva semidefinida ($P \geq 0$), se $x^T P x \geq 0$, para todo x e existem alguns vetores $x \neq 0$ tais que $x^T P x = 0$.



Matriz negativa definida: P é negativa definida ($P < 0$) se $-P$ for positiva definida.

Matriz negativa semidefinida: P é negativa semidefinida ($P \leq 0$) se $-P$ for positiva semidefinida.

A definição positiva da forma quadrática ou da forma hermitiana $V(x)$ pode ser obtida a partir da análise dos autovalores de P . Se os autovalores de P forem todos positivos então, pode concluir-se que, P é positiva definida. Isso pode ser facilmente feito utilizando o *software* MATLAB, sendo os autovalores de uma matriz obtidos usando a função “eig”.

2.4.1 ANÁLISE DE ESTABILIDADE SEGUNDO LYAPUNOV

Seja um sistema $\dot{x} = f(x)$, de acordo com o segundo teorema de Lyapunov, este sistema será assintoticamente estável caso exista uma função positiva definida $V(x)$ em que a derivada em relação ao tempo seja uma função negativa definida, conforme exposto na Equação (2.76):

$$\begin{aligned} V(x) &> 0, \\ \dot{V}(x) &< 0 \end{aligned} \quad (2.76)$$

Considere o modelo em espaço de estados com entrada nula conforme a Equação (2.77):

$$\dot{x} = Ax(t) \quad (2.77)$$

Sendo x o vetor de estados e A uma matriz constante. Se A é não-singular, então o único estado de equilíbrio é a origem $x = 0$.

Nestas condições é definida uma função positiva definida conforme segue:

$$V(x) = x^T P x > 0 \text{ (positiva definida, } P > 0) \quad (2.78)$$



A derivada de $V(x)$ em relação ao tempo é:

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} \quad (2.79)$$

Fazendo a substituição da Equação (2.77) em (2.79), tem-se:

$$\begin{aligned} \dot{V}(x) &= (Ax)^T P x + x^T P (Ax) \\ &= x^T A^T P x + x^T P A x \\ &= x^T (A^T P + P A) x \end{aligned} \quad (2.80)$$

Como $V(x)$ foi escolhida sendo definida positiva, para se alcançar a estabilidade assintótica é necessário que $\dot{V}(x)$ seja definida negativa:

$$\dot{V}(x) = x^T (A^T P + P A) x < 0 \quad (2.81)$$

O que implica em:

$$A^T P + P A < 0 \text{ (com } P \text{ positiva definida, } P > 0) \quad (2.82)$$

Considerando agora, um sistema linear controlado por meio da realimentação de estados, conforme a Equação (2.83).

$$\begin{aligned} x(t) &= Ax(t) + Bu(t), \\ u(t) &= -Kx(t) \end{aligned} \quad (2.83)$$

Deste modo, é obtido:

$$\begin{aligned} x(t) &= Ax(t) - BKx(t) \\ &= (A - BK)x(t) \end{aligned} \quad (2.84)$$



Sendo assim, de acordo com o critério de estabilidade de Lyapunov apresentado anteriormente, o sistema definido na Equação (2.83) será assintoticamente estável se:

$$(A - BK)^T P + P(A - BK) < 0 \text{ (com } P \text{ positiva definida, } P > 0) \quad (2.85)$$

Analisando a Equação (2.85), objetiva-se fazer a estimativa da matriz de realimentação de estados K , o que é possível a partir da aplicação da técnica de LMIs, que será explicada na seção 2.4.

2.5 PROJETO DE CONTROLADORES UTILIZANDO LMIS

Muitas metodologias são utilizadas para o projeto de controladores em sistemas lineares, podendo-se citar: lugar das raízes, diagramas de Bode, PID em avanço e atraso, entre outras. Tais técnicas possibilitam estabelecer controladores com algumas características definidas (percentual de *overshoot*, erro em regime estacionário, tempo de assentamento, etc.). Entretanto, tais variáveis dificilmente são aplicadas quando o sistema em questão é não linear, ou até mesmo quando os parâmetros da planta se modificam. Uma alternativa seria a aplicação de técnicas com embasamento em inteligência artificial, pois estas proporcionam um ótimo desempenho para sistemas com não linearidades. Porém, assegurar a estabilidade do sistema em malha fechada à perturbações e alterações nos parâmetros da planta e definir as características da resposta em regime transitório, se tornam tarefas difíceis de realizar.

Neste contexto, a técnica de Desigualdades Matriciais Lineares (*Linear Matrix Inequalities* – LMIs) se apresenta como uma alternativa viável, já que permite estabelecer condições de estabilidade robusta por meio de controladores com base em realimentação estados, visando a teoria de estabilidade segundo Lyapunov, explanada anteriormente na seção 2.4 Além disso, podem ser definidas condições para a resposta em regime transitório dos controladores alcançados.

As LMIs são ferramentas matemáticas largamente aplicadas na teoria de controle. Seu surgimento advento provavelmente se deu há mais de cem anos com



as pesquisas de Lyapunov. Nos últimos anos, o desenvolvimento de algoritmos numéricos sofisticados tem possibilitado a solução numérica das LMIs com eficiência e confiabilidade, tornando a sua utilização crescente (Skogestad & Postlethwaite, 2005).

Uma das principais utilidades das LMIs é a possibilidade de seu uso para a resolução de problemas que englobam numerosas variáveis matriciais, sendo que diversas estruturas podem ser impostas a tais variáveis. Além disso, as LMIs apresentam uma metodologia flexível na resolução de problemáticas na engenharia de controle, sendo que seu uso, em muitos casos, trazem a possibilidade da eliminação de restrições associadas às metodologias convencionais, bem como o auxílio na generalização de algumas tipologias de problemas (Skogestad & Postlethwaite, 2005).

A técnica de projeto a partir das LMIs, consiste em obter a solução de um conjunto de desigualdades definidas pela somatória de produtos matriciais, conforme segue:

$$\sum k_i \mathbf{A}_i \mathbf{M}_i \mathbf{B}_i < 0 \quad (2.86)$$

Sendo k_i uma constante, \mathbf{A}_i e \mathbf{B}_i matrizes conhecidas e \mathbf{M}_i uma matriz desconhecida. Desta maneira, o controlador desejado é uma função linear de \mathbf{M}_i . No geral, os problemas envolvendo LMIs podem ter várias matrizes desconhecidas, porém estas devem ser descritas como a somatória dos termos lineares $\mathbf{A}_i \mathbf{M}_i \mathbf{B}_i$.

Uma grande vantagem no uso das LMIs é a possibilidade de seu uso em conjunto com técnicas de modelagem e controle não linear, como a técnica TSF abordada neste trabalho.

Analisando a Equação (2.85) levantada anteriormente na seção 2.4, tem-se que:

$$\begin{aligned} & (\mathbf{A} - \mathbf{BK})^T \mathbf{P} + \mathbf{P}(\mathbf{A} - \mathbf{BK}) < 0 \\ & = (\mathbf{A}^T - \mathbf{K}^T \mathbf{B}^T) \mathbf{P} + \mathbf{PA} - \mathbf{PBK} < 0 \\ & = \mathbf{A}^T \mathbf{P} - \mathbf{K}^T \mathbf{B}^T \mathbf{P} + \mathbf{PA} - \mathbf{PBK} < 0 \end{aligned} \quad (2.87)$$



Neste caso, deve-se resolver a Equação (2.87) a partir da obtenção de K . Porém, nota-se que a Equação (2.87) não é LMI, pois apresenta os termos $K^T B^T P$ e PBK (os quais possuem duas variáveis desconhecidas: K e P). Sendo assim, se torna necessário uma manipulação matemática para resolver este problema. Multiplicando-se ambos os lados da Equação (2.87) por P^{-1} , onde $P > 0$, vem:

$$\begin{aligned} P^{-1}(A^T P - K^T B^T P + PA - PBK)P^{-1} &< 0 \\ P^{-1}A^T - P^{-1}K^T B^T + AP^{-1} - BPKP^{-1} &< 0 \\ QA^T - QK^T B^T + AQ - BKQ &< 0 \end{aligned} \quad (2.88)$$

Sendo $Q = P^{-1}$ (com $Q > 0$) e $G = KP^{-1} = KQ$.

Observando o termo QK^T da Equação (2.88) e tendo que $G^T = (KQ)^T = Q^T K^T$, pode-se dizer que $G^T = QK^T$, pois $P = P^T \rightarrow Q = Q^T$. Desta forma, a Equação (2.88) pode ser reescrita como:

$$\begin{aligned} QA^T - G^T B^T + AQ - BG &< 0 \\ = AQ + QA^T - (BG + G^T B^T) &< 0, \quad Q > 0 \end{aligned} \quad (2.89)$$

Finalmente, tem-se que a Equação (2.89) é descrita na forma de LMIs, resolvendo o problema apresentado na Equação (2.87). Deste modo, se torna possível o uso de ferramentas matemáticas como o MATLAB para solucionar a Equação (2.89), resultando na obtenção das matrizes Q e G . Após feito isso, basta calcular a matriz de realimentação K , sendo $G = KQ \rightarrow K = GQ^{-1}$.

A LMI expressa na Equação (2.89) possibilita a obtenção de uma matriz de realimentação K que assegura a estabilidade assintótica da malha fechada, contudo, não estabelece a dinâmica de controle em malha fechada.

As condições de \mathcal{D} -estabilidade podem ser aplicadas a fim de estipular a posição dos pólos do sistema por meio de restrição do tipo LMI. Por exemplo, considere a seguinte resposta de um sistema de segunda-ordem sem zeros e com pólos



$$\lambda = x \pm yj = -\zeta\omega_n \pm \omega_d j \quad (2.90)$$

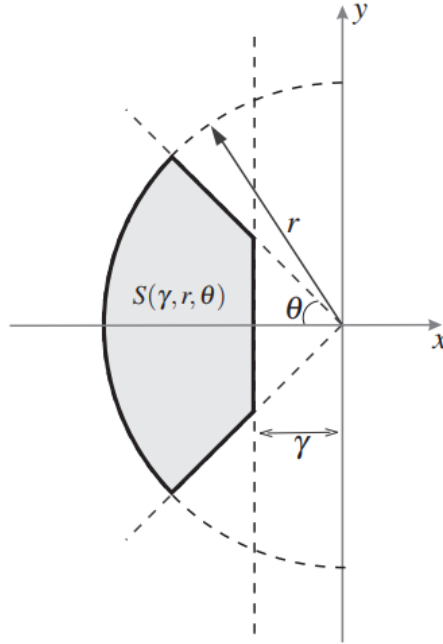
sendo, $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ e $0 < \zeta < 1$. Pode-se caracterizar tal resposta em termos dos seguintes parâmetros: frequência natural não amortecida $\omega_n = |\lambda|$, do coeficiente de amortecimento ζ e da frequência amortecida ω_d (Ogata, 2011) (Dorf & Bishop, 2001). Restringindo λ a uma determinada região do semi-plano esquerdo complexo, é possível definir limites nesses parâmetros, assegurando uma resposta transitória satisfatória.

Fazendo as seguintes substituições de variáveis: $r = \omega_n$, $\zeta = \cos(\theta)$ e $\gamma = \zeta\omega_n$ (Chilali & Gahinet, 1996). Neste contexto, um conjunto de interesse na área de engenharia de controle é o conjunto $S(\gamma, r, \theta)$ de números complexos $x + jy$ que possui as seguintes características:

$$S(\gamma, r, \theta) = \begin{cases} x < -\gamma < 0, & (a) \\ \sqrt{x^2 + y^2} < r, & (b) \\ \tan(\theta) x < -|y|, & (c) \end{cases} \quad (2.91)$$

Na Equação (2.91) a restrição (a) expressa um semi-plano à esquerda da reta vertical que passa pelo ponto $(-\gamma, 0)$ com $\gamma > 0$; (b) expressa um disco de centro na origem e raio r ; (c) expressa o limite do argumento θ dos elementos do conjunto. A ilustração da região $S(\gamma, r, \theta)$ pode ser vista na Figura 2.8.

Figura 2.8 – Região de \mathcal{D} -estabilidade $S(\gamma, r, \theta)$.



Fonte: (Faria, 2009).

Um sistema de segunda ordem é dito \mathcal{D} -estável se os pólos deste sistema pertencerem a região $S(\gamma, r, \theta)$ e, neste caso, o sistema possuirá: taxa de decaimento γ mínima, coeficiente de amortecimento $\zeta = \cos(\theta)$ mínimo e frequência amortecida $\omega_d = r \sin(\theta)$ máxima (Chilali & Gahinet, 1996).

Considere um sistema linear autônomo

$$\dot{x}(t) = A_f x(t). \quad (2.92)$$

O sistema descrito na Equação (2.92) será \mathcal{D} -estável se e somente se, existir uma matriz $Q \in \mathbb{R}^{n \times n}$, tal que as LMIs sejam satisfeitas:

$$A_f Q + Q A_f^T + 2\gamma Q < 0 \quad (2.93)$$

$$\begin{bmatrix} -rQ & A_f Q \\ Q A_f^T & -rQ \end{bmatrix} < 0 \quad (2.94)$$



$$\begin{bmatrix} \sin(\theta)(A_f Q + Q A_f^T) & \cos(\theta)(A_f Q - Q A_f^T) \\ \cos(\theta)(Q A_f^T - A_f Q) & \sin(\theta)(A_f Q + Q A_f^T) \end{bmatrix} < 0 \quad (2.95)$$

$$Q > 0 \quad (2.96)$$

Considerando a matriz de estado em malha fechada A_f , sendo $A_f = A - BK$, tem-se:

$$A_f Q = (A - BK)Q = AQ - BKQ = AQ - BG \quad (2.97)$$

Fazendo a substituição da Equação (2.97) nas Equações (2.93), (2.94) e (2.95), teremos como resultado:

$$AQ + QA^T - BG - G^T B^T + 2\gamma Q < 0 \quad (2.98)$$

$$\begin{bmatrix} -rQ & AQ - BG \\ QA^T - G^T B^T & -rQ \end{bmatrix} < 0 \quad (2.99)$$

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} < 0$$

$$\begin{aligned} a_1 &= \sin(\theta)[AQ + QA^T - (BG + G^T B^T)] \\ a_2 &= \cos(\theta)[AQ + G^T B^T - (QA^T + BG)] \\ a_3 &= \cos(\theta)[QA^T + BG - (AQ + G^T B^T)] \\ a_4 &= \sin(\theta)[AQ + QA^T - (BG + G^T B^T)] \end{aligned} \quad (2.100)$$

As Equações (2.98), (2.99) e (2.100) podem ser utilizadas para a obtenção de controladores robustos assegurando a estabilidade assintótica dos mesmos e garantindo a alocação dos pólos em malha fechada na região $S(\gamma, r, \theta)$.

Deste modo, conforme a abordagem apresentada, a possibilidade da utilização de diversas restrições em paralelo, faz o uso das LMIs ser um importante instrumento no projeto de controladores em malha fechada.

2.6 PROJETO DE CONTROLADORES TAKAGI-SUGENO *FUZZY* UTILIZANDO LMIS

A Compensação Paralela Distribuída – CPD (do inglês, *Parallel Distributed Compensation*), é um procedimento amplamente utilizado para melhorar as condições de estabilidade no projeto de controladores baseados na técnica TSF. A CPD consiste em projetar controladores para cada um dos modelos locais e o controlador global, que é não-linear em geral, é obtido por meio da combinação *Fuzzy* (conjunto de sentenças SE-ENTÃO) dos controladores locais (Tanaka, Ikeda, & Wang, 1998). Cada implicação faz a descrição de um controlador local com base na realimentação de estados:

$$\begin{aligned} \text{Regra } j: & \text{Se } z_1(t) \text{ é } \Psi_1^j \text{ e ... e Se } z_p(t) \text{ é } \Psi_p^j, \text{ então:} \\ & \mathbf{u}_j(t) = -\mathbf{K}_j \mathbf{x}(t) \end{aligned} \quad (2.101)$$

Sendo \mathbf{K}_j a matriz local de realimentação. Fazendo a defuzzificação a partir da técnica de centro de gravidade, tem-se:

$$\mathbf{u}(t) = -\sum_{j=1}^r \alpha_j \mathbf{K}_j \mathbf{x}(t) \quad (2.102)$$

Sendo α_j ($j = 1, \dots, r$) as funções de pertinência obtidas do modelo TSF. Fazendo a substituição da Equação (2.102) na Equação (2.44), tem-se:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \sum_{i=1}^r \alpha_i \mathbf{A}_i \mathbf{x}(t) - \sum_{i=1}^r \alpha_i \mathbf{B}_i \left(\sum_{j=1}^r \alpha_j \mathbf{K}_j \mathbf{x}(t) \right) \\ &= \sum_{i=1}^r \alpha_i \mathbf{A}_i \mathbf{x}(t) - \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \mathbf{B}_i \mathbf{K}_j \mathbf{x}(t) \\ &= \sum_{i=1}^r \left(\alpha_i \mathbf{A}_i \mathbf{x}(t) - \sum_{j=1}^r \alpha_i \alpha_j \mathbf{B}_i \mathbf{K}_j \mathbf{x}(t) \right) \end{aligned} \quad (2.103)$$

Pela propriedade descrita na Equação (2.41), faz-se a seguinte manipulação:

$$\sum_{j=1}^r \alpha_j = 1 \rightarrow \alpha_i \mathbf{A}_i \mathbf{x}(t) = \sum_{j=1}^r \alpha_j (\alpha_i \mathbf{A}_i \mathbf{x}(t)) = \sum_{j=1}^r \alpha_i \alpha_j \mathbf{A}_i \mathbf{x}(t) \quad (2.104)$$

Substituindo a Equação (2.103) na Equação (2.104), obtém-se:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \sum_{i=1}^r \left(\sum_{j=1}^r \alpha_i \alpha_j \mathbf{A}_i \mathbf{x}(t) \right) - \sum_{i=1}^r \alpha_i \mathbf{B}_i \left(\sum_{j=1}^r \alpha_j \mathbf{K}_j \mathbf{x}(t) \right) \\ &= \sum_{i=1}^r \sum_{j=1}^r \left(\alpha_i \alpha_j \mathbf{A}_i \mathbf{x}(t) - \alpha_i \alpha_j \mathbf{B}_i \mathbf{K}_j \mathbf{x}(t) \right) \\ &= \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j (\mathbf{A}_i - \mathbf{B}_i \mathbf{K}_j) \mathbf{x}(t) \\ &= \sum_{i=1}^r \sum_{j=1}^r \alpha_i \alpha_j \mathbf{H}_{ij} \mathbf{x}(t) \end{aligned} \quad (2.105)$$

Tendo-se a seguinte definição:

$$\mathbf{H}_{ij} = \mathbf{A}_i - \mathbf{B}_i \mathbf{K}_j \quad (2.106)$$

Pode-se reescrever a Equação (2.105) da seguinte forma:

$$\dot{\mathbf{x}}(t) = \sum_{i=1}^r \alpha_i^2 \mathbf{H}_{ii} \mathbf{x}(t) + 2 \sum_{\substack{i,j=1 \\ i < j}}^r \alpha_i \alpha_j \left(\frac{\mathbf{H}_{ij} + \mathbf{H}_{ji}}{2} \right) \mathbf{x}(t) \quad (2.107)$$

O sistema apresentado na Equação (2.107) será assintoticamente estável se existir uma matriz positiva definida \mathbf{Q} ($\mathbf{Q} > 0$), tal que (Faria, 2009):



$$\begin{aligned} H_{ii}Q + QH_{ii}^T &< 0, (\text{para } i = j) \\ \left(\frac{H_{ij} + H_{ji}}{2}\right)Q + Q\left(\frac{H_{ij} + H_{ji}}{2}\right)^T &< 0, (\text{para } i < j) \end{aligned} \quad (2.108)$$

Além disso, o sistema exposto na Equação (2.107) será dito \mathcal{D} -estável, ou seja, seus pólos estarão alocados na região $\mathcal{S}(\gamma, r, \theta)$, caso as seguintes LMIs sejam satisfeitas (Faria, 2009):

Para $i = j$:

$$H_{ii}Q + QH_{ii}^T + 2\gamma Q < 0 \quad (2.109)$$

$$\begin{bmatrix} -rQ & H_{ii}Q \\ QH_{ii}^T & -rQ \end{bmatrix} < 0 \quad (2.110)$$

$$\begin{bmatrix} \sin(\theta)(H_{ii}Q + QH_{ii}^T) & \cos(\theta)(H_{ii}Q - QH_{ii}^T) \\ \cos(\theta)(QH_{ii}^T - H_{ii}Q) & \sin(\theta)(H_{ii}Q + QH_{ii}^T) \end{bmatrix} < 0 \quad (2.111)$$

Para $i < j$:

$$\left(\frac{H_{ij} + H_{ji}}{2}\right)Q + Q\left(\frac{H_{ij} + H_{ji}}{2}\right)^T + 2\gamma Q < 0 \quad (2.112)$$

$$\begin{bmatrix} -rQ & \left(\frac{H_{ij} + H_{ji}}{2}\right)Q \\ Q\left(\frac{H_{ij} + H_{ji}}{2}\right)^T & -rQ \end{bmatrix} < 0 \quad (2.113)$$

$$\begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} < 0$$

$$b_1 = \sin(\theta) \left[\left(\frac{H_{ij} + H_{ji}}{2}\right)Q + Q\left(\frac{H_{ij} + H_{ji}}{2}\right)^T \right] \quad (2.114)$$

$$b_2 = \cos(\theta) \left[\left(\frac{H_{ij} + H_{ji}}{2}\right)Q - Q\left(\frac{H_{ij} + H_{ji}}{2}\right)^T \right]$$

$$b_3 = \cos(\theta) \left[\mathbf{Q} \left(\frac{\mathbf{H}_{ij} + \mathbf{H}_{ji}}{2} \right)^T - \left(\frac{\mathbf{H}_{ij} + \mathbf{H}_{ji}}{2} \right) \mathbf{Q} \right]$$

$$b_4 = \sin(\theta) \left[\left(\frac{\mathbf{H}_{ij} + \mathbf{H}_{ji}}{2} \right) \mathbf{Q} + \mathbf{Q} \left(\frac{\mathbf{H}_{ij} + \mathbf{H}_{ji}}{2} \right)^T \right]$$

Da Equação (2.106) e definindo $\mathbf{G}_j = \mathbf{K}_j \mathbf{Q}$, tem-se:

$$\mathbf{H}_{ij} \mathbf{Q} = (\mathbf{A}_i - \mathbf{B}_i \mathbf{K}_j) \mathbf{Q} = \mathbf{A}_i \mathbf{Q} - \mathbf{B}_i \mathbf{K}_j \mathbf{Q} = \mathbf{A}_i \mathbf{Q} - \mathbf{B}_i \mathbf{G}_j \quad (2.115)$$

Fazendo a substituição da Equação (2.115) nas Equações (2.109), (2.110), (2.111), (2.112), (2.113) e (2.114), tem-se:

Para $i = j$:

$$\mathbf{A}_i \mathbf{Q} + \mathbf{Q} \mathbf{A}_i^T - (\mathbf{B}_i \mathbf{G}_i + \mathbf{G}_i^T \mathbf{B}_i^T) + 2\gamma \mathbf{Q} < 0 \quad (2.116)$$

$$\mathbf{A}_i \mathbf{Q} + \mathbf{Q} \mathbf{A}_i^T - (\mathbf{B}_i \mathbf{G}_i + \mathbf{G}_i^T \mathbf{B}_i^T) < 0 \quad (2.117)$$

$$\begin{bmatrix} -r\mathbf{Q} & \mathbf{A}_i \mathbf{Q} - \mathbf{B}_i \mathbf{G}_i \\ \mathbf{Q} \mathbf{A}_i^T - \mathbf{G}_i^T \mathbf{B}_i^T & -r\mathbf{Q} \end{bmatrix} < 0 \quad (2.118)$$

$$\begin{bmatrix} c_1 & c_2 \\ c_3 & c_4 \end{bmatrix} < 0$$

$$\begin{aligned} c_1 &= \sin(\theta) [\mathbf{A}_i \mathbf{Q} + \mathbf{Q} \mathbf{A}_i^T - (\mathbf{B}_i \mathbf{G}_i + \mathbf{G}_i^T \mathbf{B}_i^T)] \\ c_2 &= \cos(\theta) [\mathbf{A}_i \mathbf{Q} + \mathbf{G}_i^T \mathbf{B}_i^T - (\mathbf{B}_i \mathbf{G}_i + \mathbf{Q} \mathbf{A}_i^T)] \\ c_3 &= \cos(\theta) [\mathbf{Q} \mathbf{A}_i^T + \mathbf{B}_i \mathbf{G}_i - (\mathbf{A}_i \mathbf{Q} + \mathbf{G}_i^T \mathbf{B}_i^T)] \\ c_4 &= \sin(\theta) [\mathbf{A}_i \mathbf{Q} + \mathbf{Q} \mathbf{A}_i^T - (\mathbf{B}_i \mathbf{G}_i + \mathbf{G}_i^T \mathbf{B}_i^T)] \end{aligned} \quad (2.119)$$

Para $i < j$:

$$\frac{\mathbf{A}_i \mathbf{Q} + \mathbf{A}_j \mathbf{Q} + \mathbf{Q} \mathbf{A}_i^T + \mathbf{Q} \mathbf{A}_j^T - (\mathbf{B}_i \mathbf{G}_j + \mathbf{B}_j \mathbf{G}_i + \mathbf{G}_i^T \mathbf{B}_j^T + \mathbf{G}_j^T \mathbf{B}_i^T)}{2} + 2\gamma \mathbf{Q} < 0 \quad (2.120)$$

$$\frac{A_i Q + A_j Q + Q A_i^T + Q A_j^T - (B_i G_j + B_j G_i + G_i^T B_j^T + G_j^T B_i^T)}{2} < 0 \quad (2.121)$$

$$\begin{bmatrix} -rQ & \frac{A_i Q + A_j Q - (B_i G_j + B_j G_i)}{2} \\ \frac{Q A_i^T + Q A_j^T - (G_i^T B_j^T + G_j^T B_i^T)}{2} & -rQ \end{bmatrix} < 0 \quad (2.122)$$

$$\begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix} < 0$$

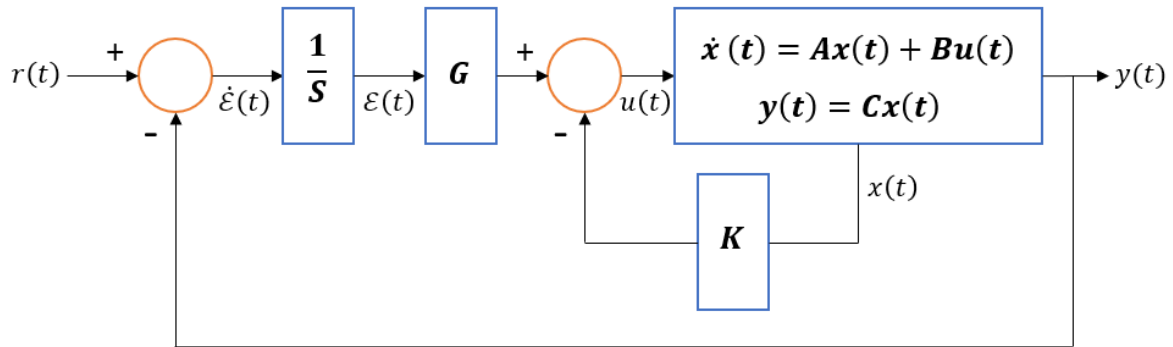
$$\begin{aligned} d_1 &= \sin(\theta) \left[\frac{A_i Q + A_j Q + Q A_i^T + Q A_j^T - (B_i G_j + B_j G_i + G_i^T B_j^T + G_j^T B_i^T)}{2} \right] \\ d_2 &= \cos(\theta) \left[\frac{A_i Q + A_j Q + G_i^T B_j^T + G_j^T B_i^T - (Q A_i^T + Q A_j^T + B_i G_j + B_j G_i)}{2} \right] \\ d_3 &= \cos(\theta) \left[\frac{Q A_i^T + Q A_j^T + B_i G_j + B_j G_i - (A_i Q + A_j Q + G_i^T B_j^T + G_j^T B_i^T)}{2} \right] \\ d_4 &= \sin(\theta) \left[\frac{A_i Q + A_j Q + Q A_i^T + Q A_j^T - (B_i G_j + B_j G_i + G_i^T B_j^T + G_j^T B_i^T)}{2} \right] \end{aligned} \quad (2.123)$$

2.7 CONTROLADOR DE MODELAGEM INTERNA

Seja um sistema SISO, ou seja, um sistema que apresente somente uma entrada e uma saída. O Controlador de Modelagem Interna (CMI) é definido como um regulador linear embasado no princípio de modelagem interna. Considerando uma referência $R(s)$ aplicada a uma planta $G(s)$ a controlar, um controlador em cascata $G_c(s)$ e uma saída $Y(s)$, o princípio de modelagem interna estabelece que, se o produto $G(s)G_c(s)$ contiver $R(s)$, então $y(t)$ rastreará $r(t)$ assintoticamente, fazendo com que o erro em regime permanente entre $Y(s)$ e $R(s)$ seja nulo (Dorf & Bishop, 2001).

Existem várias maneiras diferentes para a aplicação do CMI. A topologia utilizada neste trabalho de conclusão de curso é composta pela inserção de um controlador integral de função de transferência $1/s$ em cascata com a planta $G(s)$ e o controlador $G_c(s)$, onde é realizada a realimentação de estados. A proposição deste sistema tem a finalidade de realizar o rastreamento de referências tipo degrau. Vale salientar que para sistemas que possuam pólos na origem, a inserção do controlador integral se torna inviável. O diagrama exposto na Figura 2.9 faz a ilustração da topologia utilizada no presente trabalho.

Figura 2.9 – Sistema com Controlador de Modelagem Interna (CMI).



Fonte: Próprio Autor.

Observa-se na Figura 2.9 que o CMI insere uma nova variável de estado $\varepsilon(t)$ definida como a integral do erro entre a referência $r(t)$ e a saída $y(t)$. O sistema em questão pode ser definido por meio da Equação (2.124):

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) \\ u(t) &= -\mathbf{K}\mathbf{x}(t) + G_c \varepsilon(t) \\ \dot{\varepsilon}(t) &= r(t) - y(t) = r(t) - \mathbf{C}\mathbf{x}(t) \end{aligned} \tag{2.124}$$

Supondo que a planta em questão possua estados completamente controláveis e que não possua zero na origem, a fim de evitar a possibilidade de cancelamento



com o integrador inserido, é possível descrever a dinâmica do sistema pela Equação (2.125), conforme segue (Ogata, 2011):

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathcal{E}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathcal{E}(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \quad (2.125)$$

Para o projeto de um sistema assintoticamente estável, deseja-se obter valores constantes para $\mathbf{x}(\omega)$, $\mathcal{E}(\omega)$ e $u(\omega)$, sendo que, em regime permanente, $\dot{\mathcal{E}}(t) = 0$ e $y(\omega) = r$.

Modificando a Equação (2.125) para o regime permanente, vem:

$$\begin{bmatrix} \dot{\mathbf{x}}(\omega) \\ \dot{\mathcal{E}}(\omega) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(\omega) \\ \mathcal{E}(\omega) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} u(\omega) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(\omega) \quad (2.126)$$

Fazendo a subtração da Equação (2.125) da Equação (2.126) e sabendo que $r(t)$ é uma referência em degrau, sendo $r(\omega) = r(t) = r$ (constante) para $t > 0$, tem-se:

$$\begin{bmatrix} \dot{\mathbf{x}}(t) - \dot{\mathbf{x}}(\omega) \\ \dot{\mathcal{E}}(t) - \dot{\mathcal{E}}(\omega) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) - \mathbf{x}(\omega) \\ \mathcal{E}(t) - \mathcal{E}(\omega) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} [u(t) - u(\omega)] \quad (2.127)$$

Sendo:

$$\begin{aligned} \mathbf{x}(t) - \mathbf{x}(\omega) &= \mathbf{x}_e(t) \\ \mathcal{E}(t) - \mathcal{E}(\omega) &= \mathcal{E}_e(t) \\ u(t) - u(\omega) &= u_e(t) \end{aligned} \quad (2.128)$$

Desta maneira, pode-se reescrever a Equação (2.127) em termos das novas variáveis definidas na Equação (2.128):

$$\begin{bmatrix} \dot{\mathbf{x}}_e(t) \\ \dot{\mathcal{E}}_e(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_e(t) \\ \mathcal{E}_e(t) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix} [u_e(t)] \quad (2.129)$$



Sendo:

$$u_e(t) = -\mathbf{K}\mathbf{x}_e(t) + G_c\mathcal{E}_e(t) \quad (2.130)$$

Definindo um novo vetor de erro $\mathbf{e}(t) = \begin{bmatrix} \mathbf{x}_e(t) \\ \mathcal{E}_e(t) \end{bmatrix}$, então tem-se que a Equação (2.129) resulta em:

$$\dot{\mathbf{e}}(t) = \mathbf{A}_e\mathbf{e}(t) + \mathbf{B}_eu_e(t) \quad (2.131)$$

Sendo: $\mathbf{A}_e = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix}$ e $\mathbf{B}_e = \begin{bmatrix} \mathbf{B} \\ 0 \end{bmatrix}$.

Desta forma, pode-se reescrever a Equação (2.130) como:

$$u_e(t) = -\mathbf{K}_e\mathbf{e}(t) \quad (2.132)$$

Sendo: $\mathbf{K}_e = [\mathbf{K} \quad -G_c]$.

Finalmente, substituindo a Equação (2.132) na Equação (2.131), é obtida a equação de estado do erro, conforme segue:

$$\dot{\mathbf{e}}(t) = (\mathbf{A}_e - \mathbf{B}_e\mathbf{K}_e)\mathbf{e}(t) \quad (2.133)$$

Considerando a Equação (2.133), deve-se selecionar adequadamente o vetor de ganhos \mathbf{K}_e a fim de que o sistema proposto seja assintoticamente estável.

Os autovalores da matriz $(\mathbf{A}_e - \mathbf{B}_e\mathbf{K}_e)$ (ou seja, os pólos desejados em malha fechada) devem possuir parte real negativa para que o erro do sistema se torna nulo em regime permanente e a saída do sistema se estabilize na referência desejada (Chen, 1999).

2.8 FPGA IN THE LOOP

2.8.1 FPGA



Os FPGAs (*Field Programmable Gate Array*), são chips reprogramáveis que utilizam blocos lógicos programáveis pré-construídos, onde pode-se fazer a implementação das características de *hardware* a partir de sua configuração. Os modelos matemáticos são feitos em *software* e são compilados para gerar um arquivo de configuração que possui as diretrizes sobre como os componentes internos do FPGA devem ser conectados entre si. Ademais, os FPGAs são totalmente reconfiguráveis possibilitando a recompilação de novos modelos, podendo assim assumir diferentes configurações de circuito (National Instruments, 2016).

Os blocos lógicos dos FPGAs podem ser configurados como um circuito único, ou como circuitos em paralelo. Esta configuração de circuitos em paralelo visa a redução do tempo de processamento, característica essencial em determinados tipos de aplicações. Um exemplo de sua importância é na aplicação em conversores, em que são utilizadas técnicas de controle complexas, com altas frequências de chaveamento que exigem um número de cálculos impraticáveis em um ciclo de controle por um processador digital de sinal, mais conhecido como DSP (Gomez, 2013).

2.8.2 HARDWARE IN THE LOOP

Atualmente existem diversas metodologias para o desenvolvimento de sistemas digitais, sendo que a maioria dos *drivers* de potência são baseados nestes tipos de sistemas. Tais metodologias requerem processos de ajuste, validação e verificação de sistemas (Altera, 2013).

Geralmente, os testes de desenvolvimento dos sistemas digitais são executados fazendo o uso de duas abordagens: o teste de *software* e o teste de *hardware*.

No teste de *software*, faz-se a análise e o ajuste do desempenho e da funcionalidade do sistema, a partir do uso de ferramentas de simulação de *software*. A confiabilidade da simulação de *software* é específica de cada aplicação e depende da precisão da modelagem da planta em estudo e do período de duração dos testes



realizados. Já no teste de *hardware*, o projeto é implementado em uma plataforma experimental com a finalidade de verificar a funcionalidade e fazer medições da performance em condições mais realistas.

O teste de *software* possibilita a realização de testes considerando diversos cenários, porém boas simulações em geral levam muito tempo e os resultados obtidos dependem da modelagem da planta, que difere de uma planta real. Por outro lado, os testes de *hardware* proporcionam resultados mais rápidos e mais realistas do que os testes de *software*, entretanto, o número de possíveis cenários de testes é limitado e além disso, apresentam maior dificuldade na implementação quando comparados às simulações de *software* (Altera, 2013).

Neste contexto, o *Hardware in the Loop (HIL)* é uma abordagem de teste que vem ganhando espaço, pois oferece as vantagens de testes de *software* em conjunto com os testes de *hardware* no desenvolvimento de sistemas embarcados complexos. No *HIL* o controlador é implementado em *hardware*, e a modelagem da planta, perturbações, ruídos e outros estímulos, são implementados em *software* (Altera, 2013) (Gomez, 2013).

2.8.3 FPGA IN THE LOOP

Devido aos avanços relacionados aos testes e simulações com HIL, surgiram o desenvolvimento ferramentas de testes HIL aplicados à FPGA, advindo assim, o FPGA in the Loop (FIL). Como o FPGA é um equipamento reconfigurável e de rápida prototipagem, destaca-se algumas vantagens notáveis proporcionadas pelo FIL no tocante a eficiência no desenvolvimento dos sistemas embarcados, como (Gomez, 2013):

- Melhoria na qualidade dos testes: além de apresentar resultados com maior fidelidade e mais realísticos em comparação com as simulações tradicionais, o FIL possibilita que qualquer cenário possa ser testado com o *hardware*, desde que modelado no simulador, permitindo a previsão comportamental da planta em situações de adversas.



- Desenvolvimento dos sistemas a curto período: o desenvolvimento e testes dos sistemas utilizando a técnica FIL são realizados em um tempo menor em diversos setores da indústria, permitindo a realização de testes antes do protótipo para testes reais estar concluído.
- Desenvolvimento mais econômico e seguro: grande parte das plantas envolvidas do desenvolvimento de sistemas embarcados apresentam alto custo e, dada sua submissão para a realização de testes, existe a possibilidade de danificá-las, ocasionando grande prejuízo. Desta forma, por meio do *FIL*, é possível a obtenção de um controlador de forma mais econômica e segura, já que alguns testes são realizados manualmente, garantindo o máximo de confiabilidade possível antes da execução dos testes reais.



3 ESTRUTURA DO CONTROLADOR PROPOSTO

Neste capítulo, será exposta a metodologia aplicada no presente trabalho para o controle do conversor *Boost*, que tem como base a modelagem Takagi-Sugeno *Fuzzy* (TSF) integrada ao Controlador de Modelagem Interna (CMI) e as Desigualdades Lineares Matriciais (LMIs). Adicionalmente será apresentada a simplificação matemática realizada a fim de efetuar a implementação do controlador em FPGA.

3.1 MODELO TAKAGI-SUGENO FUZZY DO CONVERSOR BOOST

O modelo *Takagi-Sugeno Fuzzy* do conversor *Boost* utilizado no presente trabalho, é descrito conforme a Equação (3.1) exposta em (García, Suemitsu, & Pinto, 2013), conforme segue:

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}_i \mathbf{x}(t) + \mathbf{B}_i \mathbf{u}(t) + \mathbf{P} V_{in} \\ \mathbf{y}(t) = V_{out} = V_C = \mathbf{C}_i \mathbf{x}(t) \end{cases} \quad (3.1)$$

Para o modelo dado na Equação (3.1) tem-se que a tensão de entrada V_{in} é considerada como uma perturbação no sistema, enquanto \mathbf{P} é a matriz de perturbação. Considera-se que a perturbação será rejeitada pelo controlador CMI proposto. Desta forma, o efeito de \mathbf{P} foi desconsiderado no projeto do controlador CMI-TSF (García, Suemitsu, & Pinto, 2013).

De acordo com as Equações (2.32) e (2.33) definidas na seção 2.2.1, pode-se chegar à diversos modelos em espaço de estados para o conversor *Boost*, como por exemplo os modelos apresentados nas Equações (2.35), (2.36) e (2.37). Para o projeto do controlador retratado neste trabalho, o modelo em espaço de estados foi embasado no modelo presente em (Garcia, Pinto, Soares, Galotto, & Onizuka, 2016), sendo o sistema utilizado descrito pelas Equações (3.3) e (3.4). Foram consideradas como variáveis de estado $x_1(t) = V_C(t)$ e $x_2(t) = I_L(t)$, resultando em um vetor de estado conforme exposto na Equação (3.2).

$$\mathbf{x}(t) = [x_1(t) \ x_2(t)]^T = \begin{bmatrix} V_C(t) \\ I_L(t) \end{bmatrix} \quad (3.2)$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} \dot{V}_C(t) \\ \dot{I}_L(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix} \begin{bmatrix} V_C(t) \\ I_L(t) \end{bmatrix} + \begin{bmatrix} -\frac{I_L(t)}{C} \\ \frac{V_C(t)}{L} \end{bmatrix} D(t) + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} V_{in} \quad (3.3)$$

$$y = V_C(t) = [1 \ 0] \begin{bmatrix} V_C(t) \\ I_L(t) \end{bmatrix} \quad (3.4)$$

Sendo, V_{in} a tensão de entrada, V_C a tensão no capacitor que é igual a tensão de saída V_{out} , I_L a corrente no indutor, L a indutância, R a resistência da carga, C a capacitância e D o *duty cycle* do conversor *Boost* escolhido como a lei de controle.

Observa-se que o modelo em espaço de estados contém dois termos não lineares, definidos como $b_1 = -\frac{I_L(t)}{C}$ e $b_2 = \frac{V_C(t)}{L}$ presentes na matriz \mathbf{B} . De acordo com metodologia de modelagem explicada na seção 2.3.2, o modelo *Takagi-Sugeno Fuzzy* definido pelas Equações (3.3) e (3.4) terá $2^2 = 4$ modelos locais.

A partir disso, define-se os valores máximos e mínimos b_{11} , b_{12} , b_{21} e b_{22} , conforme segue:

$$\begin{aligned} b_{11} &= \max(b_1) & b_{21} &= \max(b_2) \\ b_{12} &= \min(b_1) & b_{22} &= \min(b_2) \end{aligned} \quad (3.5)$$

Dando prosseguimento a metodologia apresentada na seção 2.3.2, são definidos os termos σ_{11} , σ_{12} , σ_{21} e σ_{22} com base nas Equações (2.48) e (2.49):

$$\begin{aligned} \sigma_{11} &= \frac{b_1 - b_{12}}{b_{11} - b_{12}} \\ \sigma_{12} &= \frac{b_{11} - b_1}{b_{11} - b_{12}} \\ \sigma_{21} &= \frac{b_2 - b_{22}}{b_{21} - b_{22}} \\ \sigma_{22} &= \frac{b_{21} - b_2}{b_{21} - b_{22}} \end{aligned} \quad (3.6)$$



Desta maneira, pode-se descrever as funções de pertinência do modelo proposto tal qual como a Equação (2.62) da seção 2.3.2.

Os modelos locais da modelagem *Takagi-Sugeno Fuzzy* do sistema proposto são apresentados conforme as Equações (3.7), (3.8) e (3.9) e (3.10), como mostrado a seguir:

- Modelo local 1:

$$A_1 = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix}, B_1 = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}, P_1 = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{L} \end{bmatrix}, C_1 = [1 \quad 0], \alpha_1 = \sigma_{11} \sigma_{21} \quad (3.7)$$

- Modelo local 2:

$$A_2 = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix}, B_2 = \begin{bmatrix} b_{11} \\ b_{22} \end{bmatrix}, P_2 = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{L} \end{bmatrix}, C_2 = [1 \quad 0], \alpha_2 = \sigma_{11} \sigma_{22} \quad (3.8)$$

- Modelo local 3:

$$A_3 = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix}, B_3 = \begin{bmatrix} b_{12} \\ b_{21} \end{bmatrix}, P_3 = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{L} \end{bmatrix}, C_3 = [1 \quad 0], \alpha_3 = \sigma_{12} \sigma_{21} \quad (3.9)$$

- Modelo local 4:

$$A_4 = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix}, B_4 = \begin{bmatrix} b_{12} \\ b_{22} \end{bmatrix}, P_4 = \begin{bmatrix} 0 \\ 1 \\ \frac{1}{L} \end{bmatrix}, C_4 = [1 \quad 0], \alpha_4 = \sigma_{12} \sigma_{22} \quad (3.10)$$

Pode-se observar que as matrizes A , C e P dos modelos locais são matrizes constantes, já que os termos não lineares se encontram somente na matriz B . O modelo *Fuzzy* global do sistema é descrito pela Equação (3.11) a seguir:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^4 \alpha_i A_i x(t) + \sum_{i=1}^4 \alpha_i B_i u(t) + \sum_{i=1}^4 \alpha_i P_i V_{in}(t) \\ y(t) = \sum_{i=1}^4 \alpha_i C_i x(t) \end{cases} \quad (3.11)$$

Sendo A_i as matrizes de estado, B_i as matrizes de entrada, C_i as matrizes de saída e P_i as matrizes de perturbação do i -ésimo modelo local. Como salientado anteriormente o efeito da perturbação produzido por V_{in} é considerado rejeitado pelo controlador CMI proposto. Sendo assim, o modelo *Fuzzy* global do sistema pode ser redefinido como:

$$\begin{cases} \dot{x}(t) \cong \sum_{i=1}^4 \alpha_i A_i x(t) + \sum_{i=1}^4 \alpha_i B_i u(t) \\ y(t) = \sum_{i=1}^4 \alpha_i C_i x(t) \end{cases} \quad (3.12)$$

3.2 INTEGRAÇÃO ENTRE OS MODELOS TSF E CMI

Seja o sistema TSF definido na seção 3.1. O presente trabalho propõe a integração deste modelo TSF com o Controlador de Modelagem Interna (CMI), apresentado na seção 2.7. A partir das definições realizadas para um sistema com CMI conforme exposto na Equação (2.124) e considerando que as matrizes de

estados e saída do modelo TSF são constantes, ou seja, $A_i = A = \begin{bmatrix} -\frac{1}{RC} & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{bmatrix}$ e $C_i =$

$C = [1 \ 0]$ conforme dito na seção 3.1, define-se um novo vetor $x_a(t)$ de estados aumentado com a inclusão da nova variável $\mathcal{E}(t)$, sendo:

$$\mathbf{x}_a(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathcal{E}(t) \end{bmatrix} \quad (3.13)$$

Desta maneira, fazendo algumas manipulações algébricas, pode-se reescrever $\mathbf{A}_i \mathbf{x}(t)$ e $\mathbf{C}_i \mathbf{x}(t)$ do modelo TSF proposto em termos de $\mathbf{x}_a(t)$, conforme segue:

$$\mathbf{A}_i \mathbf{x}(t) = \mathbf{A} \mathbf{x}(t) = [\mathbf{A} \quad 0] \begin{bmatrix} \mathbf{x}(t) \\ \mathcal{E}(t) \end{bmatrix} = [\mathbf{A} \quad 0] \mathbf{x}_a(t) \quad (3.14)$$

$$\mathbf{C}_i \mathbf{x}(t) = \mathbf{C} \mathbf{x}(t) = [\mathbf{C} \quad 0] \begin{bmatrix} \mathbf{x}(t) \\ \mathcal{E}(t) \end{bmatrix} = [\mathbf{C} \quad 0] \mathbf{x}_a(t) \quad (3.15)$$

Substituindo as Equações (3.14) e (3.15) no modelo TSF da Equação (3.12), tem-se:

$$\begin{cases} \dot{\mathbf{x}}(t) = \sum_{i=1}^r \alpha_i [\mathbf{A} \quad 0] \mathbf{x}_a(t) + \sum_{i=1}^r \alpha_i \mathbf{B}_i u(t) \\ \mathbf{y}(t) = \sum_{i=1}^r \alpha_i [\mathbf{C} \quad 0] \mathbf{x}_a(t) \end{cases} \quad (3.16)$$

Além disso, também pode-se reescrever $\dot{\mathcal{E}}(t)$ em termos de $\mathbf{x}_a(t)$:

$$\dot{\mathcal{E}}(t) = r(t) - \mathbf{y}(t) = r(t) + [-\mathbf{C} \quad 0] \mathbf{x}_a(t) + 0u(t) \quad (3.17)$$

Aplicando a propriedade das funções de pertinência descrita na Equação (2.41) da seção 2.3.1 $\sum_{i=1}^n \alpha_i(x) = 1$, obtém-se:

$$\dot{\mathcal{E}}(t) = r(t) + \sum_{i=1}^r \alpha_i [-\mathbf{C} \quad 0] \mathbf{x}_a(t) + \sum_{i=1}^r \alpha_i 0u(t) \quad (3.18)$$

Desta forma, combinando as Equações (3.16) e (3.18), pode-se definir um novo modelo TSF integrado ao CMI:



$$\begin{aligned}\dot{\mathbf{x}}_a(t) &= \begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{e}}(t) \end{bmatrix} = \sum_{i=1}^r \alpha_i \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix} \mathbf{x}_a(t) + \sum_{i=1}^r \alpha_i \begin{bmatrix} \mathbf{B}_i \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \\ \mathbf{y}(t) &= \sum_{i=1}^r \alpha_i [\mathbf{C} \quad 0] \mathbf{x}_a(t)\end{aligned}\quad (3.19)$$

Pode-se reescrever a Equação (3.19) a partir da definição das novas matrizes \mathbf{A}_a , \mathbf{B}_{ai} e \mathbf{C}_a , conforme segue:

$$\begin{aligned}\dot{\mathbf{x}}_a(t) &= \sum_{i=1}^r \alpha_i \mathbf{A}_a \mathbf{x}_a(t) + \sum_{i=1}^r \alpha_i \mathbf{B}_{ai} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t) \\ \mathbf{y}(t) &= \sum_{i=1}^r \alpha_i \mathbf{C}_a \mathbf{x}_a(t)\end{aligned}\quad (3.20)$$

Sendo:

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{A} & 0 \\ -\mathbf{C} & 0 \end{bmatrix}, \mathbf{B}_{ai} = \begin{bmatrix} \mathbf{B}_i \\ 0 \end{bmatrix} \text{ e } \mathbf{C}_a = [\mathbf{C} \quad 0] \quad (3.21)$$

A lei de controle para um sistema com CMI apresentada na Equação (2.124) pode ser redefinida em função de $\mathbf{x}_a(t)$, conforme segue:

$$\begin{aligned}u(t) &= [-\mathbf{K} \quad G_c] \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{e}(t) \end{bmatrix} = -\mathbf{K}_e \mathbf{x}_a(t), \\ \mathbf{K}_e &= [\mathbf{K} \quad -G_c]\end{aligned}\quad (3.22)$$

Partindo das Equações (2.102) e (3.22), é possível projetar um controlador que faça a integração da lei de controle de um CMI com as características da modelagem *Takagi-Sugeno Fuzzy*. Deste modo, a lei de controle para o sistema apresentado na Equação (3.20) é definida conforme segue:

$$\mathbf{u}(t) = - \sum_{j=1}^r \alpha_j \mathbf{K}_{ej} \mathbf{x}_a(t) \quad (3.23)$$



Sendo K_{ej} a j -ésima matriz de realimentação do CMI local.

Assim como explanado na seção 2.7, para a Equação (3.23) deve-se selecionar adequadamente o vetor de ganhos K_{ej} a fim de que o sistema proposto seja assintoticamente estável.

Ademais, com a garantia de que o sistema em malha fechada é assintoticamente estável, o sinal de referência $r(t)$ pode ser considerado como uma perturbação a qual o controlador CMI rejeitará. Deste modo, tomando a Equação (3.20) pode-se desconsiderar $r(t)$ no projeto do controlador CMI-TSF. Portanto, a modelagem proposta para o sistema resulta em:

$$\begin{aligned}\dot{x}_a(t) &\cong \sum_{i=1}^r \alpha_i A_a x_a(t) + \sum_{i=1}^r \alpha_i B_{ai} u(t) \\ y(t) &= \sum_{i=1}^r \alpha_i C_a x_a(t)\end{aligned}\tag{3.24}$$

A Equação (3.24) também pode reescrita conforme a Equação (3.25) a seguir, a partir da premissa que as matrizes A e C do modelo são constantes juntamente com a propriedade $\sum_{i=1}^n \alpha_i(x) = 1$ das funções de pertinência.

$$\begin{aligned}\dot{x}_a(t) &\cong A_a x_a(t) + \sum_{i=1}^r \alpha_i B_{ai} u(t) \\ y(t) &= C_a x_a(t)\end{aligned}\tag{3.25}$$

Analisando as Equações (3.23) e (3.24), observa-se que as mesmas apresentam semelhanças estruturais quando comparadas com as Equações (2.102) e (2.44), respectivamente. Desta maneira, pode-se aplicar analogamente a técnica de LMIs, conforme apresentada nas seções 2.5 e 2.6, para realizar o projeto do controlador a partir da obtenção do vetor de ganhos K_{ej} .

O presente trabalho propõe a aplicação de LMIs embasadas no Teorema de Estabilidade de Lyapunov. A resolução destas LMIs por meio de programas computacionais, garantem um vetor de ganhos K_{ej} para que o sistema tenha



estabilidade assintótica, ou seja, garantem que pólos desejados do sistema em malha fechada tenham parte real negativa e conseguem restringir os mesmos a uma região delimitada, fazendo com que o erro do sistema $\hat{\mathcal{E}}(t)$ se torne nulo em regime permanente e a saída do sistema $y(t)$ se estabilize na referência $r(t)$ desejada. Portanto:

$$\hat{\mathcal{E}}(t) = r(t) - y(t) \rightarrow 0 \Rightarrow y(t) \rightarrow r(t), (para\ t \rightarrow \infty) \quad (3.26)$$

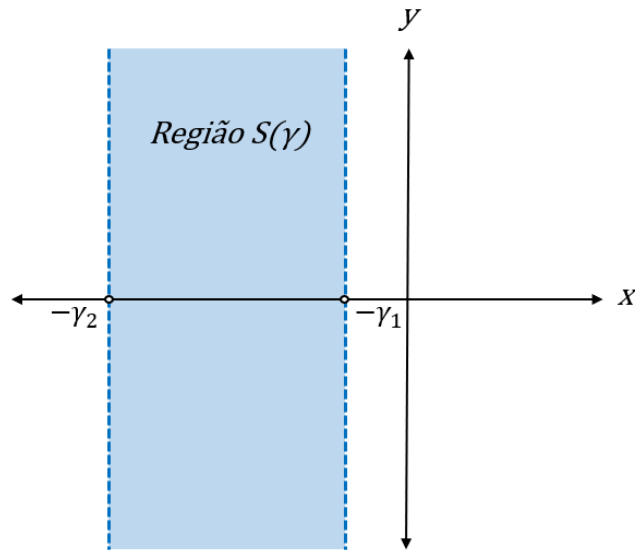
3.3 DEFINIÇÃO DAS LMIS UTILIZADAS PARA O MODELO CMI-TSF

Tomando os resultados da seção 3.2, o sistema CMI-TSF proposto para o presente trabalho é apresentado na Equação (3.27), conforme segue:

$$\begin{aligned} \dot{\mathbf{x}}_a(t) &\cong \sum_{i=1}^4 \alpha_i \mathbf{A}_a \mathbf{x}_a(t) + \sum_{i=1}^4 \alpha_i \mathbf{B}_{ai} u(t) \\ y(t) &= \sum_{i=1}^4 \alpha_i \mathbf{C}_a \mathbf{x}_a(t) \\ u(t) &= - \sum_{j=1}^4 \alpha_j \mathbf{K}_{ej} \mathbf{x}_a(t) \end{aligned} \quad (3.27)$$

Como já salientado, o sistema CMI-TSF obtido possui uma estrutura adequada para a aplicação da técnica de LMIs. O presente trabalho propõe delimitação da região de alocação de pólos a partir das restrições γ_1 e γ_2 , que definem um semiplano delimitado pelas retas verticais que passam pelos pontos $(-\gamma_1; 0)$ e $(-\gamma_2; 0)$ (com $\gamma_1, \gamma_2 > 0$). A Figura 3.1 ilustra a região $S(\gamma)$ definida pelas restrições impostas no conjunto dos números complexos $x + jy$.

Figura 3.1 - Região $S(\gamma)$ de Alocação dos Pólos Definida pelas LMIs.



Fonte: Próprio Autor.

Desta forma, basta fazer as substituições definidas na Equação (3.28) nas LMIs apresentadas na seção 2.6 expressas pelas Equações (2.116), (2.117), (2.120) e (2.121).

$$\begin{aligned}
 A_i &= A_a = A_j \\
 B_i &= B_{ai}, B_j = B_{aj} \\
 C_i &= C_a = C_j \\
 K_i &= K_{ei}, K_j = K_{ej} \\
 G_i &= G_{ai} = K_{ei}Q, G_j = G_{aj} = K_{ej}Q
 \end{aligned} \tag{3.28}$$

Sendo $Q > 0$ positiva definida e $i, j = 1, \dots, 4$.

Portanto, as LMIs propostas para a alocação de pólos no controlador CMI-TSF proposto são descritas pelas Equações (3.29), (3.30), (3.31), (3.32), (3.33), (3.34) e (3.35).

Para $i = j$:



LMI para estabilidade assintótica segundo Lyapunov

$$A_a Q + Q A_a^T - (B_{ai} G_{ai} + G_{ai}^T B_{ai}^T) < 0 \quad (3.29)$$

LMI com restrição imposta pela reta limitante referente à γ_1

$$A_a Q + Q A_a^T - (B_{ai} G_{ai} + G_{ai}^T B_{ai}^T) + 2\gamma_1 Q < 0 \quad (3.30)$$

LMI com restrição imposta pela reta limitante referente à γ_2

$$A_a Q + Q A_a^T - (B_{ai} G_{ai} + G_{ai}^T B_{ai}^T) + 2\gamma_2 Q > 0 \quad (3.31)$$

Para $i < j$:

LMI para estabilidade assintótica segundo Lyapunov

$$A_a Q + Q A_a^T - \frac{(B_{ai} G_{aj} + B_{aj} G_{ai} + G_{ai}^T B_{aj}^T + G_{aj}^T B_{ai}^T)}{2} < 0 \quad (3.32)$$

LMI com restrição imposta pela reta limitante referente à γ_1

$$A_a Q + Q A_a^T - \frac{(B_{ai} G_{aj} + B_{aj} G_{ai} + G_{ai}^T B_{aj}^T + G_{aj}^T B_{ai}^T)}{2} + 2\gamma_1 Q < 0 \quad (3.33)$$

LMI com restrição imposta pela reta limitante referente à γ_2

$$A_a Q + Q A_a^T - \frac{(B_{ai} G_{aj} + B_{aj} G_{ai} + G_{ai}^T B_{aj}^T + G_{aj}^T B_{ai}^T)}{2} + 2\gamma_2 Q > 0 \quad (3.34)$$

Para Q positiva definida:

LMI para estabelecer Q como positiva definida

$$Q > 0 \quad (3.35)$$



3.4 DETERMINAÇÃO DOS GANHOS K_{ej} DO CONTROLADOR CMI-TSF

O presente trabalho, traz a proposta da implementação do controle CMI-TSF para um conversor *Boost* com os seguintes parâmetros: Capacitância $C = 50 \mu F$, Indutância $L = 2,4 mH$, Resistência $R = 50 \Omega$, Frequência de Chaveamento $f_s = 50 kHz$, Tensão de Entrada $V_{in} = 9 V_{dc}$. Para os máximos e mínimos das variáveis de estado foi considerado que os valores esperados estão nas seguintes faixas: $I_L \rightarrow 0,01A$ à $1,5A$ e para $V_C \rightarrow 9V_{dc}$ à $27V_{dc}$. Além disso, foram adotados $\gamma_1 = 68$ e $\gamma_2 = 4500$, definindo dessa forma, a região $S(\gamma)$ como um semiplano delimitado pelas retas verticais que passam pelos pontos $(-68; 0)$ e $(-4500; 0)$.

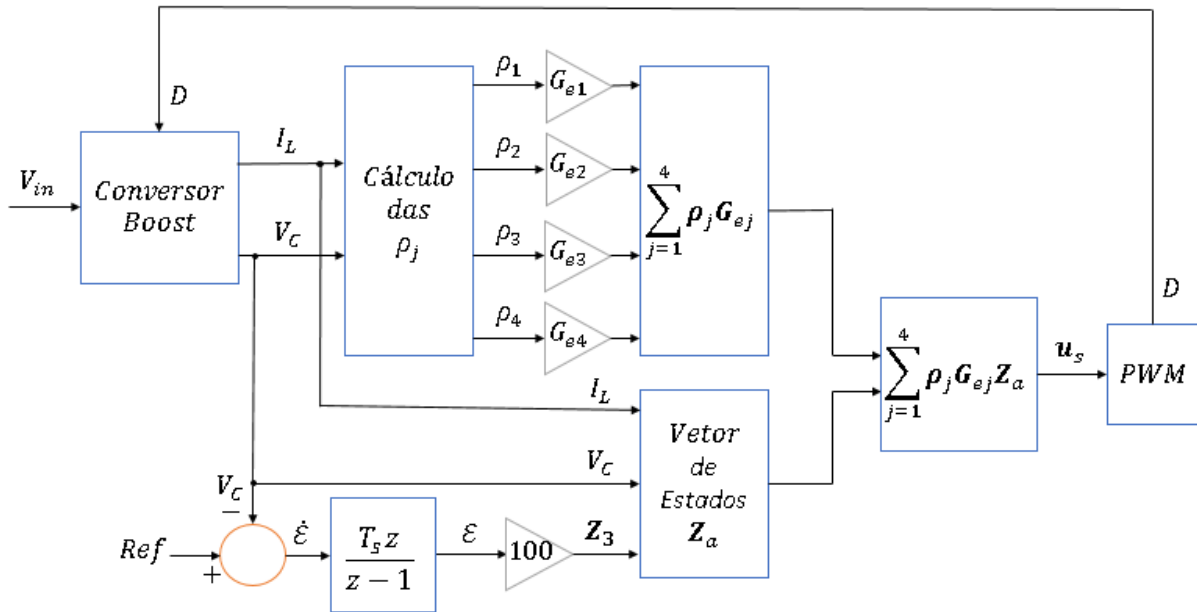
Aplicando as LMIs apresentadas na seção 3.3, tem-se como resultado os seguintes valores para os ganhos K_{ej} do controlador proposto, conforme a Equação (3.36):

$$K_{ej} (j = 1, \dots, 4) \rightarrow \begin{matrix} K_{e1} = [0,0001 & 0,3596 & -11,88] \\ K_{e2} = [0,0007 & 1,0803 & -35,84] \\ K_{e3} = [0,0140 & 0,4045 & -16,22] \\ K_{e4} = [0,0396 & 0,9143 & -34,51] \end{matrix} \quad (3.36)$$

3.5 CONTROLADOR CMI-TSF SIMPLIFICADO PROPOSTO

Analisando os ganhos K_{ej} obtidos apresentados na Equação (3.36) da seção 3.4, juntamente com os equacionamentos presentes na modelagem TSF exposta na seção 3.1, decidiu-se realizar algumas simplificações a partir de manipulações matemáticas, com a finalidade de facilitar a implementação do controle em FPGA a partir da redução dos números de operações presentes nos equacionamentos e, consequentemente, resultando em uma quantidade menor de tempo para obtenção da lei de controle, reduzindo assim, o custo computacional. Na Figura 3.2 ilustra-se o modelo simplificado proposto em diagrama de blocos para melhor entendimento.

Figura 3.2 – Diagrama de Blocos do Modelo do Controlador CMI-TSF Simplificado Proposto.



Fonte: Próprio Autor.

As seções 3.5.1 e 3.5.2 a seguir, fazem o detalhamento das simplificações realizadas para a obtenção do modelo apresentado na Figura 3.2.

Após a obtenção do sinal da lei de controle simplificada u_s , basta aplicar a técnica de PWM, comparando o sinal u_s com uma onda triangular para gerar o sinal da razão cíclica D e enviá-lo para o chaveamento do conversor *Boost*, conforme pode ser visualizado na Figura 3.2.

3.5.1 SIMPLIFICAÇÕES NAS EQUAÇÕES DAS FUNÇÕES DE PERTINÊNCIA

A partir da Equação (3.6) definida na seção 3.1 para os termos σ_{11} , σ_{12} , σ_{21} e σ_{22} e da Equação (2.62) exposta seção 2.3.2 para determinação das funções de pertinências α_i do modelo TSF, chega-se as seguintes equações para as funções de pertinência:

$$\begin{aligned}\alpha_1 &= \sigma_{11} \sigma_{21} \rightarrow \alpha_1 = \left(\frac{b_1 - b_{12}}{b_{11} - b_{12}} \right) \left(\frac{b_2 - b_{22}}{b_{21} - b_{22}} \right) \\ \alpha_2 &= \sigma_{11} \sigma_{22} \rightarrow \alpha_2 = \left(\frac{b_1 - b_{12}}{b_{11} - b_{12}} \right) \left(\frac{b_{21} - b_2}{b_{21} - b_{22}} \right) \\ \alpha_3 &= \sigma_{12} \sigma_{21} \rightarrow \alpha_3 = \left(\frac{b_{11} - b_1}{b_{11} - b_{12}} \right) \left(\frac{b_2 - b_{22}}{b_{21} - b_{22}} \right) \\ \alpha_4 &= \sigma_{12} \sigma_{22} \rightarrow \alpha_4 = \left(\frac{b_{11} - b_1}{b_{11} - b_{12}} \right) \left(\frac{b_{21} - b_2}{b_{21} - b_{22}} \right)\end{aligned}\tag{3.37}$$

Observa-se na Equação (3.37) que o denominador é comum para todas as equações das funções de pertinência. Portanto, definiu-se uma variável D_c conforme exposto na Equação (3.38), a fim de representar os denominadores em comum e reduzir os cálculos, já que com esta manipulação, o número de vezes que o denominador é calculado passa quatro para somente um.

$$D_c = \frac{1}{(b_{11} - b_{12})(b_{21} - b_{22})}\tag{3.38}$$

Substituindo as variáveis não lineares $b_1 = -\frac{I_L(t)}{C}$ e $b_2 = \frac{V_C(t)}{L}$ nas equações das funções de pertinência e aplicando a simplificação proposta na Equação (3.38), tem-se:

$$\begin{aligned}\alpha_1 &= -(I_L + b_{12}C)(V_C - b_{22}L) \frac{D_c}{LC} \\ \alpha_2 &= -(I_L + b_{12}C)(b_{21}L - V_C) \frac{D_c}{LC} \\ \alpha_3 &= (b_{11}C + I_L)(V_C - b_{22}L) \frac{D_c}{LC} \\ \alpha_4 &= (b_{11}C + I_L)(b_{21}L - V_C) \frac{D_c}{LC}\end{aligned}\tag{3.39}$$

Analisando a Equação (3.39), pode-se observar agora que ainda existem alguns termos que se repetem em duas ou mais equações. Dessa forma as seguintes variáveis são definidas para efeito de simplificação:



$$\begin{aligned}n_1 &= -b_{12}C \\n_2 &= -b_{11}C \\n_3 &= b_{22}L \\n_4 &= b_{21}L\end{aligned}\tag{3.40}$$
$$S = \frac{D_c}{LC} = \frac{1}{(b_{11} - b_{12})(b_{21} - b_{22})LC}$$

Vale salientar que as variáveis definidas na Equação (3.40) resultarão em valores constantes, sendo assim, podem ser calculadas em modo *offline*, ou seja, estas variáveis tem seus valores previamente obtidos antes da execução do FPGA *in the Loop* (FIL).

Finalmente, as funções de pertinências α_i do modelo TSF proposto podem ser reescritas conforme a Equação (3.41).

$$\begin{aligned}\alpha_1 &= -S(I_L - n_1)(V_C - n_3) \\ \alpha_2 &= -S(I_L - n_1)(n_4 - V_C) \\ \alpha_3 &= -S(n_2 - I_L)(V_C - n_3) \\ \alpha_4 &= -S(n_2 - I_L)(n_4 - V_C)\end{aligned}\tag{3.41}$$

Considerou-se para a implementação do controlador, que o sinal negativo que acompanha o somatório da lei de controle definida na seção 3.3 pela Equação (3.27) se anula com o sinal negativo embutido nas equações das funções de pertinência, sendo implementadas ρ_j ($j = 1, \dots, 4$), conforme segue:

$$\begin{aligned}\alpha_1 &= -S\rho_1 \\ \alpha_2 &= -S\rho_2 \\ \alpha_3 &= -S\rho_3 \\ \alpha_4 &= -S\rho_4\end{aligned}\tag{3.42}$$
$$\begin{aligned}\rho_1 &= (I_L - n_1)(V_C - n_3) \\ \rho_2 &= (I_L - n_1)(n_4 - V_C) \\ \rho_3 &= (n_2 - I_L)(V_C - n_3) \\ \rho_4 &= (n_2 - I_L)(n_4 - V_C)\end{aligned}$$

Desta forma, a nova lei de controle com as simplificações realizadas é definida como:

$$\mathbf{u}_1(t) = \sum_{j=1}^4 \rho_j S \mathbf{K}_{ej} \mathbf{x}_a(t) \quad (3.43)$$

3.5.2 DEFINIÇÃO DA LEI DE CONTROLE SIMPLIFICADA

Primeiramente, realizou-se a multiplicação da lei de controle definida na seção 3.5.1 pela Equação (3.43) por 100, com a finalidade de tornar os resultados mais homogêneos, tendo como consequência a redução dos números de bits e, além disso, expressar a nova lei de controle em uma faixa que varie de 0 a 100 por cento. Desta forma, a nova lei de controle foi definida como \mathbf{u}_2 , conforme expressa a Equação (3.42):

$$\mathbf{u}_2 = 100 \sum_{j=1}^4 \rho_j S \mathbf{K}_{ej} \mathbf{x}_a \quad (3.44)$$

Sabendo-se que os sinais do vetor de estado \mathbf{x}_a serão amostrados, é necessário que o integrador inserido na composição do controlador digital proposto a partir da metodologia CMI apresentada na seção 2.7, tenha sua função de transferência discretizada para reconhecer o sinal de erro amostrado, onde a saída do integrador discreto é $y(n) = y(n-1) + T_s r(n)$, sendo $r(n)$ a entrada, $T_s = 1/f_s$ o tempo de amostragem com f_s sendo a frequência de chaveamento do conversor *Boost*.

Dessa maneira, a função de transferência discreta do integrador, realizada a partir da Transformada Z, pode ser definida da seguinte forma (Dorf & Bishop, 2013):

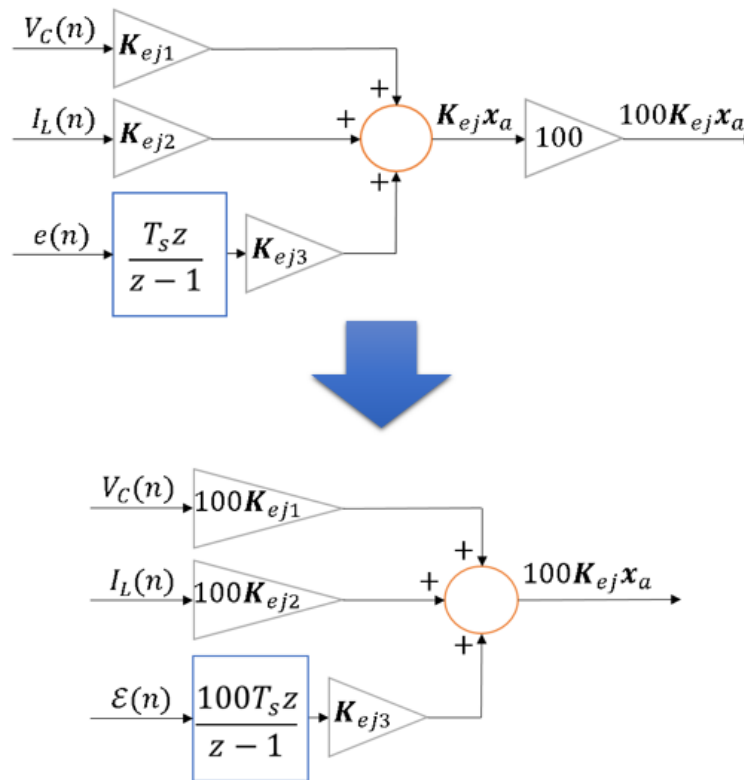
$$\frac{y(z)}{r(z)} = \frac{T_s z}{z - 1} \quad (3.45)$$

Observando os ganhos \mathbf{K}_{ej} obtidos apresentados na Equação (3.36) da seção 3.4, observa-se que os valores dos termos que ocupam as duas primeiras posições no vetor \mathbf{K}_{ej} ($j = 1, \dots, 4$) são pequenos quando comparados aos valores dos termos

que ocupam a terceira posição. Esta não homogeneidade entre tais termos faz com que a representação dos ganhos digitalmente necessite de uma quantidade maior de bits, o que aumenta o custo computacional do algoritmo. Além disso, tem-se que a frequência de chaveamento do conversor definido para o controle é de 50kHz, o que acarreta em um tempo de amostragem $T_s = 20\mu s$, número esse muito pequeno que necessitaria de pelo menos 20 bits para ter uma representação com boa precisão.

Portanto, a fim de minimizar tais desvantagens, fez-se a seguinte manipulação matemática para o termo $K_{ej}x_a$ da lei de controle, conforme exposto no digrama da Figura 3.3:

Figura 3.3 - Manipulação Matemática Realizada no Termo $K_{ej}x_a$.



Fonte: Próprio Autor.

Observa-se que a multiplicação por 100 na função de pertinência u_2 da Equação (3.44) foi deslocada para o termo $K_{ej}x_a$, não alterando a equação matematicamente. Analisando a Figura 3.3, constata-se que após a manipulação, os valores dos termos que ocupam as duas primeiras posições no vetor K_{ej} ($j = 1, \dots, 4$)



foram multiplicados por 100, enquanto que o ganho do terceiro foi deslocado para o último termo do vetor de estados x_a , não alterando a equação original, conforme segue:

$$100 * S K_{ej} x_a = S [100 K_{ej1} \quad 100 K_{ej2} \quad K_{ej3}] \begin{bmatrix} V_C \\ I_L \\ 100 \mathcal{E} \end{bmatrix} \quad (3.46)$$

A partir da manipulação apresentada na Equação (3.46) verifica-se que, os termos do vetor de ganhos K_{ej} ficaram mais homogêneos e além disso, o termo \mathcal{E} engloba o tempo de amostragem T_s , onde $100T_s$ pode ser implementado com maior facilidade. Tais manipulações reduziram a quantidade de bits utilizados para a representação em ponto-fixa das variáveis envolvidas.

Portanto, a partir das Equações (3.44) e (3.46), a lei de controle u_s simplificada, proposta para implementação, é apresentada a seguir:

$$\begin{aligned} u_s &= \sum_{j=1}^4 \rho_j G_{ej} z_a \\ G_{ej} &= S [100 K_{ej1} \quad 100 K_{ej2} \quad K_{ej3}] \\ z_a &= \begin{bmatrix} V_C \\ I_L \\ z_3 \end{bmatrix}, z_3 = 100 \mathcal{E} \\ \rho_1 &= h_1 h_2 \\ \rho_2 &= h_1 h_3 \\ \rho_3 &= h_4 h_2 \\ \rho_4 &= h_4 h_3 \\ h_1 &= (I_L - n_1) \\ h_2 &= (V_C - n_3) \\ h_3 &= (n_4 - V_C) \\ h_4 &= (n_2 - I_L) \end{aligned} \quad (3.47)$$

A implementação da Equação (3.47) necessita, portanto, de um número menor de operações matemáticas e uma quantidade menor de bits para representação em ponto-fixa comparada a lei de controle definida antes da simplificação. Os novos resultados para os vetores de ganho de acordo com a Equação (3.47) são:



$$\begin{aligned} \mathbf{G}_{e1} &= [0,0005 \quad 1,3406 \quad -0,4431] \\ \mathbf{G}_{e2} &= [0,0027 \quad 4,0280 \quad -1,3362] \\ \mathbf{G}_{e3} &= [0,0521 \quad 1,5080 \quad -0,6048] \\ \mathbf{G}_{e4} &= [0,1476 \quad 3,4089 \quad -1,2868] \end{aligned} \quad \mathbf{G}_{ej} (j = 1, \dots, 4) \rightarrow \quad (3.48)$$

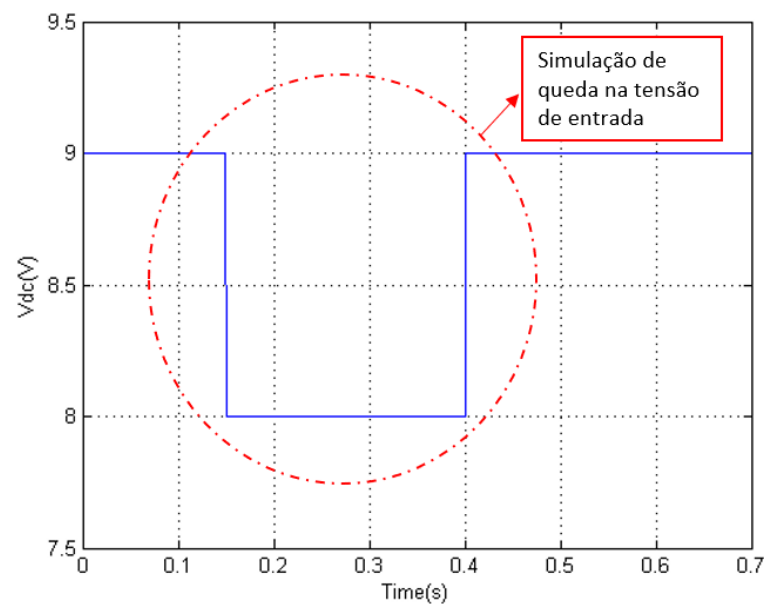
Nota-se que de fato, os valores obtidos para os vetores \mathbf{G}_{ej} são mais homogêneos do que os valores obtidos para os vetores \mathbf{K}_{ej} antes das manipulações matemáticas apresentados na Equação (3.36).

4 RESULTADOS OBTIDOS

Com a finalidade de verificar e analisar o controlador simplificado CMI-TSF proposto apresentado no Capítulo 3 do presente trabalho, fizeram-se simulações em *software* e após, realizou-se a validação do modelo por meio do teste FIL. O controlador proposto foi aplicado ao conversor *Boost* de características descritas na seção 3.4.

Para ambos os casos, foi considerada uma perturbação na tensão de entrada do conversor *Boost*, conforme mostrada na Figura 4.1, a fim de testar a performance do controlador proposto.

Figura 4.1 - Tensão de Entrada do Conversor Boost Utilizada nas Simulações.



Fonte: Próprio Autor.

Além disso, testou-se a robustez do controlador para diversas cargas aplicadas ao conversor. Serão expostos no presente trabalho, os resultados obtidos para os testes com cargas dos seguintes valores: $R_1 = 60\Omega$, $R_2 = 50\Omega$ (*carga nominal*) e $R_3 = 30\Omega$.

A forma de onda utilizada para a referência de saída estipulada foi comparada com a forma de onda apresentada na saída do conversor *Boost*. A variação da

referência imposta tem a finalidade de testar a rastreabilidade do controlador, onde o erro em regime permanente deve ser nulo.

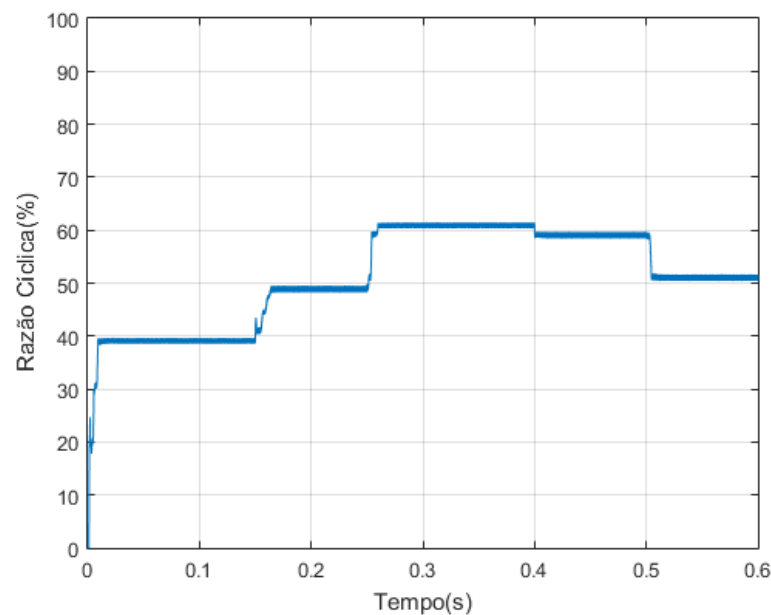
Na seção 4.1 serão apresentados os resultados obtidos para a simulação em *software* e na seção 4.2 serão apresentados os resultados obtidos para o teste FIL. Para ambas simulações, serão expostas as formas de ondas da razão cíclica D (0 a 100%) do conversor, da corrente no indutor I_L , da tensão de saída, que é a tensão no capacitor V_C e do erro entre a saída e a referência resultantes, considerando as simulações realizadas.

4.1 SIMULAÇÃO EM SOFTWARE – MATLAB/SIMULINK

Primeiramente, implementou-se tanto o modelo do conversor *Boost*, quanto a modelagem do controlador CMI-TSF simplificado proposto utilizando o *software* MATLAB na versão 2012 em conjunto com sua ferramenta SIMULINK, conforme a ilustração em diagrama de blocos na Figura 3.2, presente na seção 3.5.

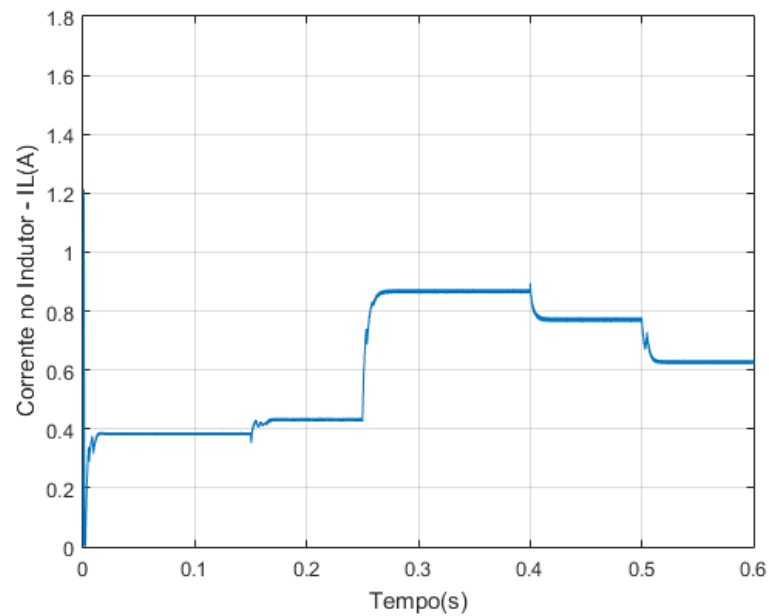
Os resultados de simulação para a carga $R_1 = 60\Omega$ são apresentados nas Figuras 4.2, 4.3, 4.4 e 4.5.

Figura 4.2 – Simulação MATLAB/SIMULINK: Razão Cíclica para $R_1 = 60\Omega$.



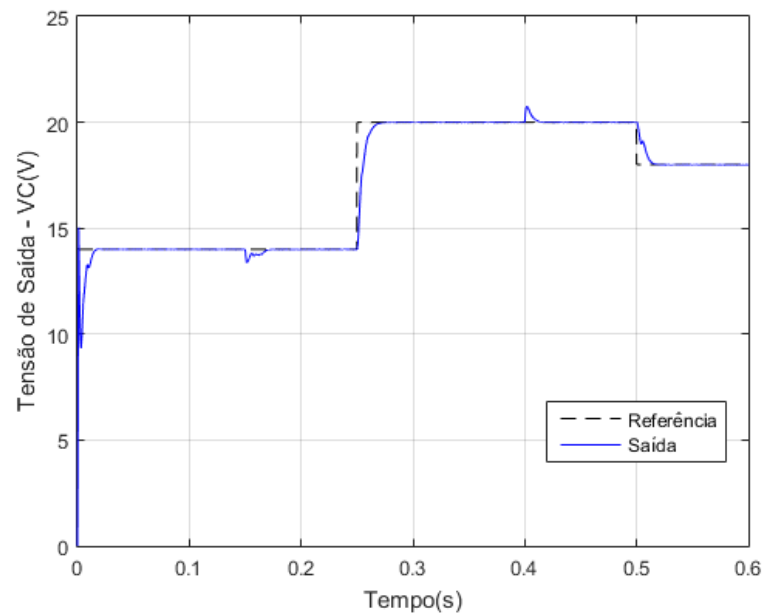
Fonte: Próprio Autor.

Figura 4.3 – Simulação MATLAB/SIMULINK: Corrente no Indutor para $R_1 = 60\Omega$.



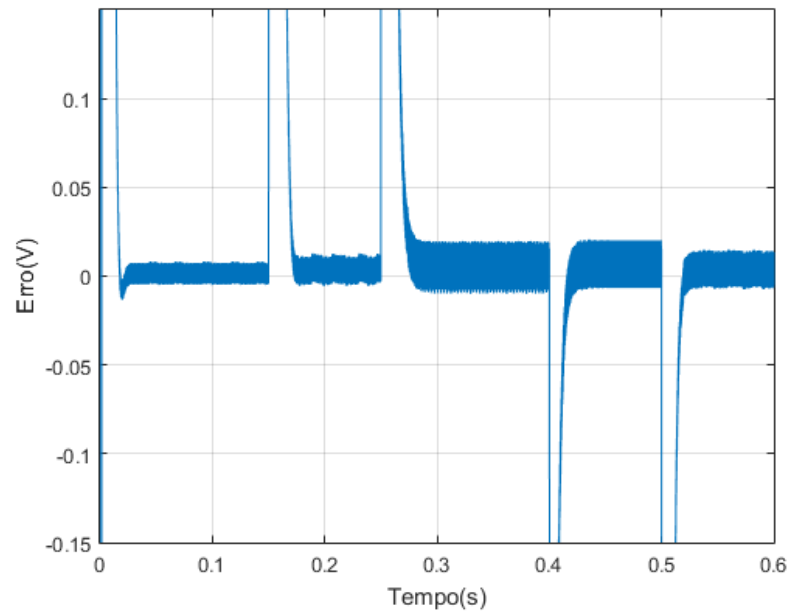
Fonte: Próprio Autor.

Figura 4.4 - Simulação MATLAB/SIMULINK: Tensão de Saída para $R_1 = 60\Omega$.



Fonte: Próprio Autor.

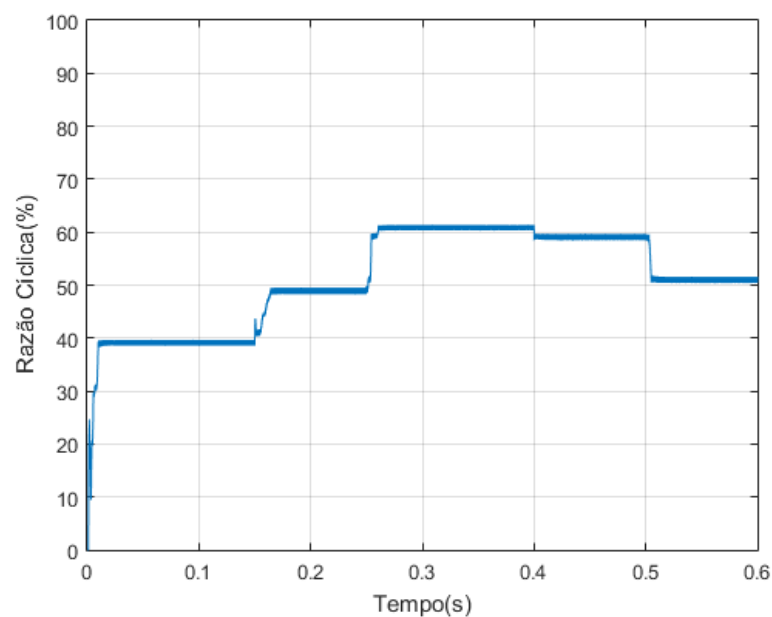
Figura 4.5 - Simulação MATLAB/SIMULINK: Erro entre a Referência e a Saída para $R_1 = 60\Omega$.



Fonte: Próprio Autor.

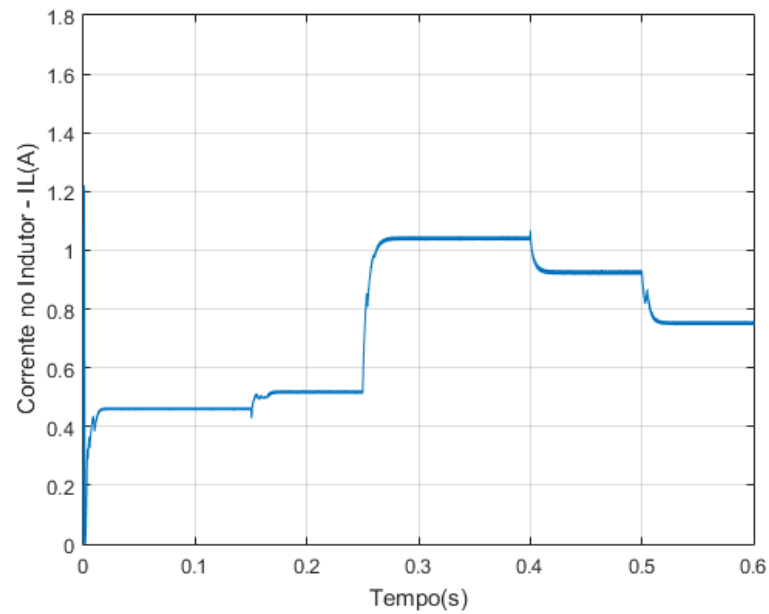
Os resultados de simulação para a carga nominal $R_2 = 50\Omega$ são apresentados nas Figuras 4.6, 4.7, 4.8 e 4.9.

Figura 4.6 - Simulação MATLAB/SIMULINK: Razão Cíclica para $R_2 = 50\Omega$.



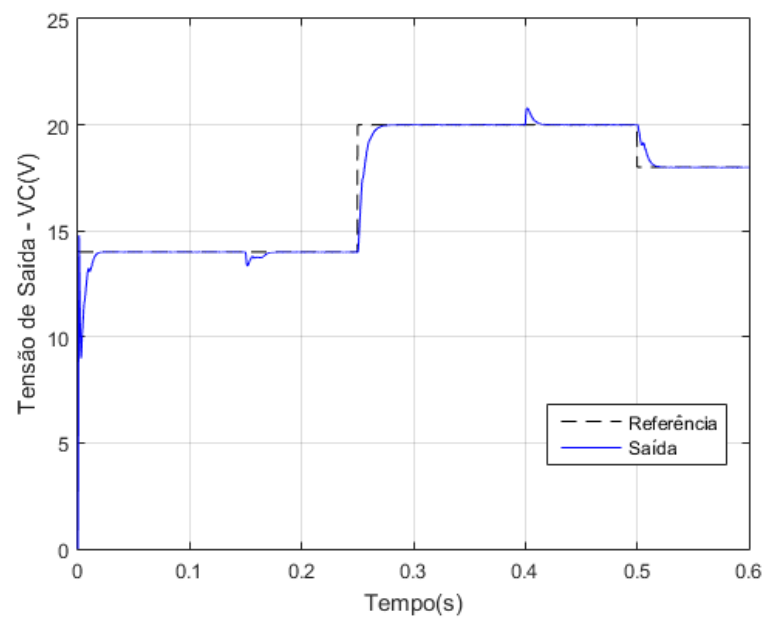
Fonte: Próprio Autor.

Figura 4.7 – Simulação MATLAB/SIMULINK: Corrente no Indutor para $R_2 = 50\Omega$.



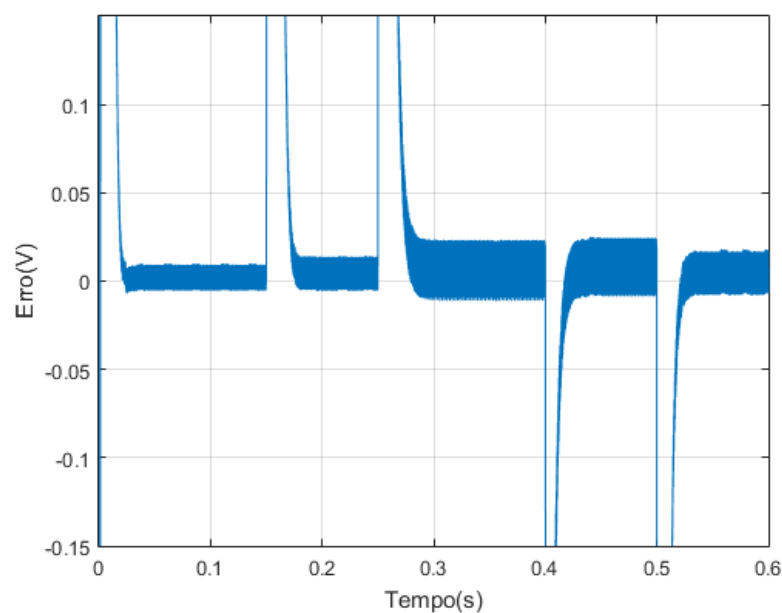
Fonte: Próprio Autor.

Figura 4.8 - Simulação MATLAB/SIMULINK: Tensão de Saída para $R_2 = 50\Omega$.



Fonte: Próprio Autor.

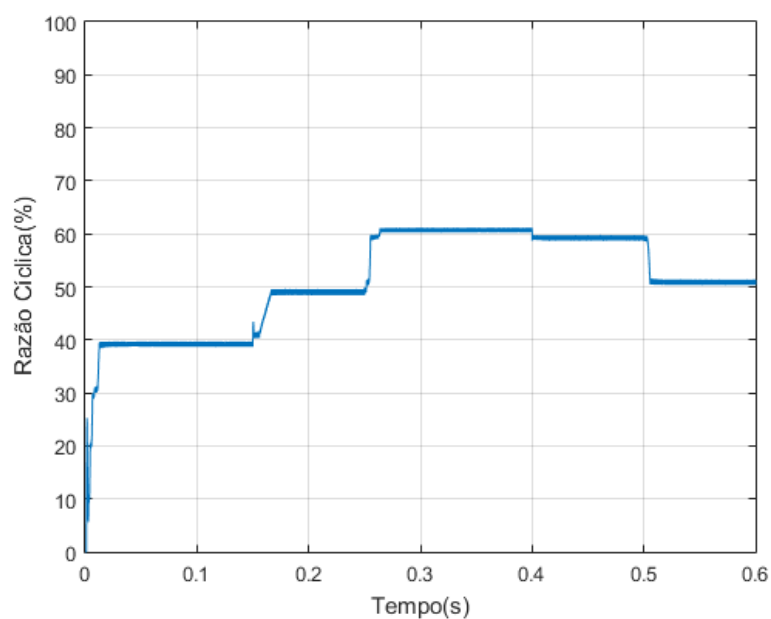
Figura 4.9 - Simulação MATLAB/SIMULINK: Erro entre a Referência e a Saída para $R_2 = 50\Omega$.



Fonte: Próprio Autor.

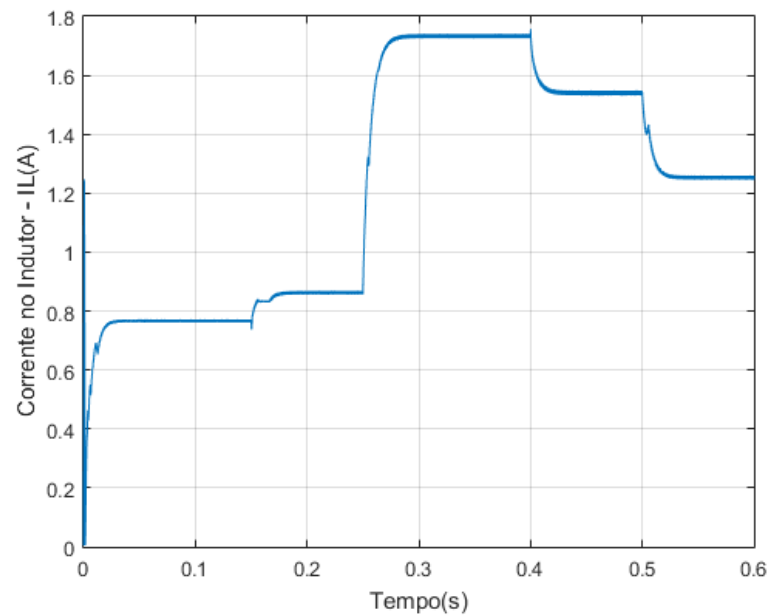
Os resultados de simulação para a carga $R_3 = 30\Omega$ são apresentados nas Figuras 4.10, 4.11, 4.12 e 4.13.

Figura 4.10 – Simulação MATLAB/SIMULINK: Razão Cíclica para $R_3 = 30\Omega$.



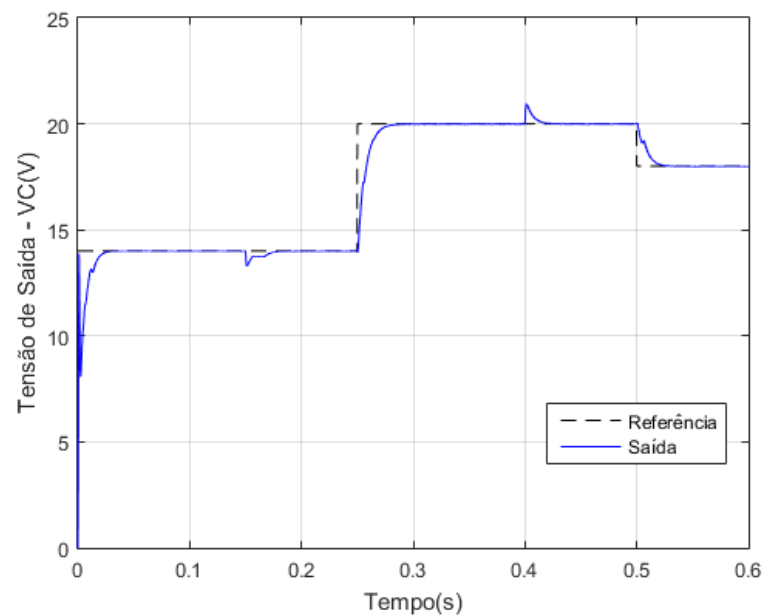
Fonte: Próprio Autor.

Figura 4.11 - Simulação MATLAB/SIMULINK: Corrente no Indutor para $R_3 = 30\Omega$.



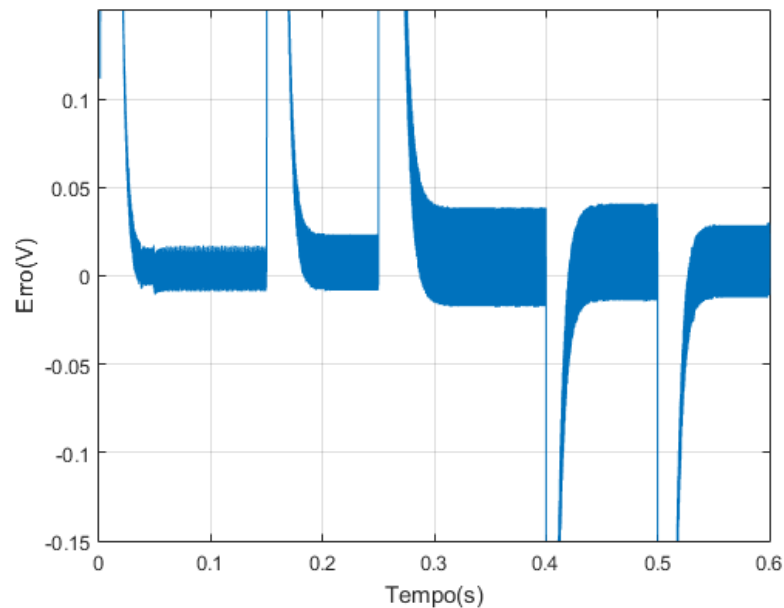
Fonte: Próprio Autor.

Figura 4.12 – Simulação MATLAB/SIMULINK: Tensão de Saída para $R_3 = 30\Omega$.



Fonte: Próprio Autor.

Figura 4.13 - Simulação MATLAB/SIMULINK: Erro entre a Referência e a Saída para $R_3 = 30\Omega$.



Fonte: Próprio Autor.

Pode-se observar em relação aos resultados obtidos que a tensão do indutor I_L aumentou proporcionalmente conforme o aumento da carga aplicada à saída do conversor *Boost*, o que já era esperado. Observou-se também, que para todos os casos simulados, foram identificadas pequenas oscilações na tensão de saída do conversor (aproximadamente $0,9V$) iniciadas em $t = 0,15s$ e $t = 0,4s$, provenientes da queda de tensão na entrada aplicada para a simulação. Tais oscilações foram rapidamente corrigidas pelo controlador proposto em um tempo de assentamento de aproximadamente $15ms$. A tensão de saída do conversor rastreou com sucesso a referência estipulada, apresentando um *overshoot* desprezível e erro em regime permanente tendendo à zero, com tempo de assentamento em torno de $20ms$.

Os resultados alcançados para a simulação em software SIMULINK/MATLAB foram satisfatórios, mostrando que o controlador digital projetado foi adequado para a aplicação. Deste modo, o próximo passo realizado, foi o desenvolvimento do algoritmo de controle simplificado em VHDL para realização de sua validação por meio da abordagem de teste FPGA *in the Loop*. O teste FIL realizado foi detalhado na seção 4.2 a seguir.



4.2 TESTE FPGA IN THE LOOP

Posteriormente à simulação em MATLAB/SIMULINK do controlador proposto realizada na seção 4.1, foi desenvolvido um algoritmo do modelo do controlador proposto para a implementação em *hardware*, a fim de validar experimentalmente o controlador aplicado ao conversor *Boost*.

O desenvolvimento do algoritmo proposto se deu em VHDL - VHSIC (*Very High Speed Integrated Circuits*) *Hardware Description Language*. O VHDL é uma linguagem específica que visa a descrição das estruturas e do comportamento do *hardware* que tem o objetivo de facilitar o projeto de circuitos digitais programáveis. É padronizada pelo IEEE - *Institute of Electrical and Electronic Engineers*. O processo descrito em VHDL opera em paralelo, fazendo conformidade ao funcionamento dos dispositivos de *hardware* digital (Prof.^a Luiza M. R. Codá).

O algoritmo desenvolvido para o controle foi compilado utilizando o *software* Altera Quartus II, sendo este compatível com o modelo do FPGA. Os resultados da compilação e as informações referentes ao custo computacional do algoritmo são mostrados na Figura 4.14.

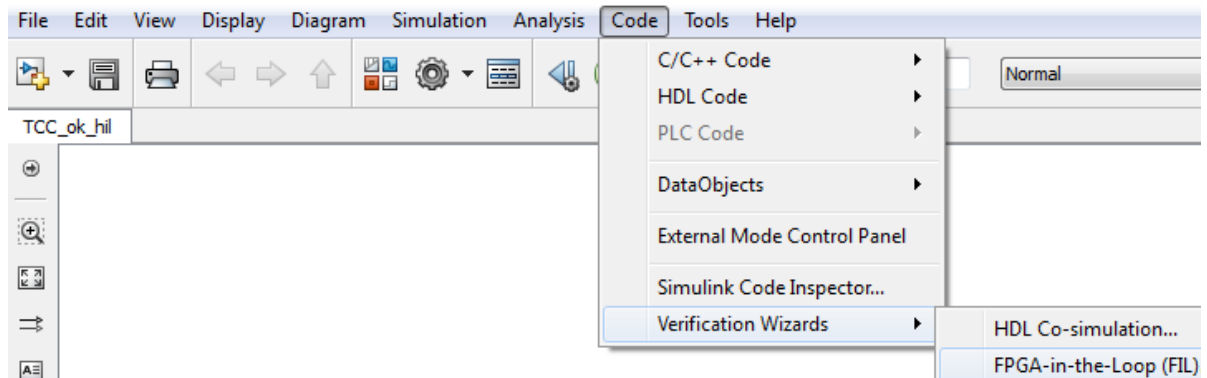
Figura 4.14 – Resultados da Compilação e do Custo Computacional do Algoritmo Proposto

Flow Summary	
Flow Status	Successful - Thu Nov 24 14:54:33 2016
Quartus II 64-Bit Version	12.1 Build 177 11/07/2012 SJ Web Edition
Revision Name	exemplo_hil
Top-level Entity Name	tcc_lucas
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	2,567 / 114,480 (2 %)
Total combinational functions	2,566 / 114,480 (2 %)
Dedicated logic registers	215 / 114,480 (< 1 %)
Total registers	215
Total pins	67 / 529 (13 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	71 / 532 (13 %)
Total PLLs	0 / 4 (0 %)

Fonte: Próprio Autor.

Primeiramente, para realizar o FIL, deve-se selecionar na barra de ferramentas do SIMULINK o menu “Code” e, em seguida, “Verification Wizards” > “FPGA-in-the-Loop (FIL)”, conforme pode ser visto da Figura 4.15.

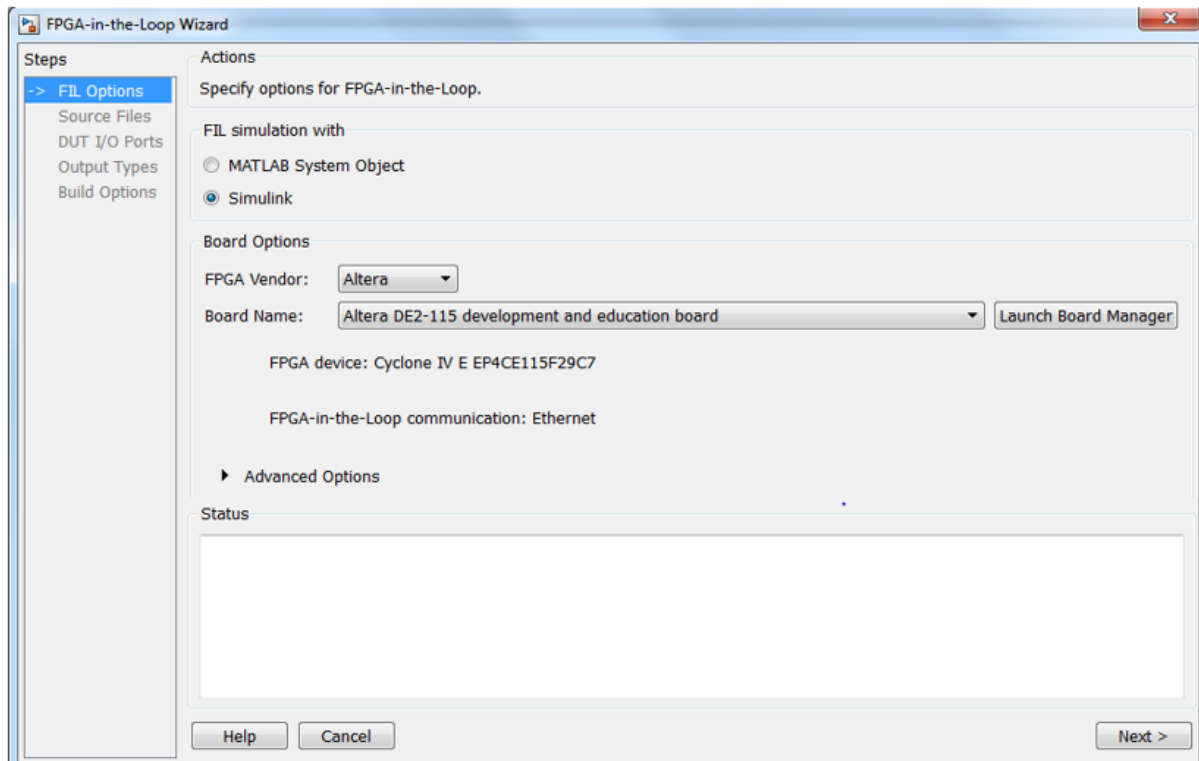
Figura 4.15 – Selecionando o Assistente de Configuração do FIL.



Fonte: Próprio Autor.

Após selecionar “FPGA-in-the-Loop (FIL)”, será aberta uma janela para início da configuração do FIL, conforme ilustrada na Figura 4.16. A configuração do FIL é composta por cinco etapas. Na primeira etapa, em “FIL Options”, indica-se onde ocorrerá o FIL, que no caso do presente trabalho, foi realizado no SIMULINK. Em seguida, indica-se as especificações do FPGA utilizado, onde, para a execução do FIL foi utilizado o kit educacional e de desenvolvimento DE2-115 da Altera com FPGA Cyclone IV EP4CE115F29C7N.

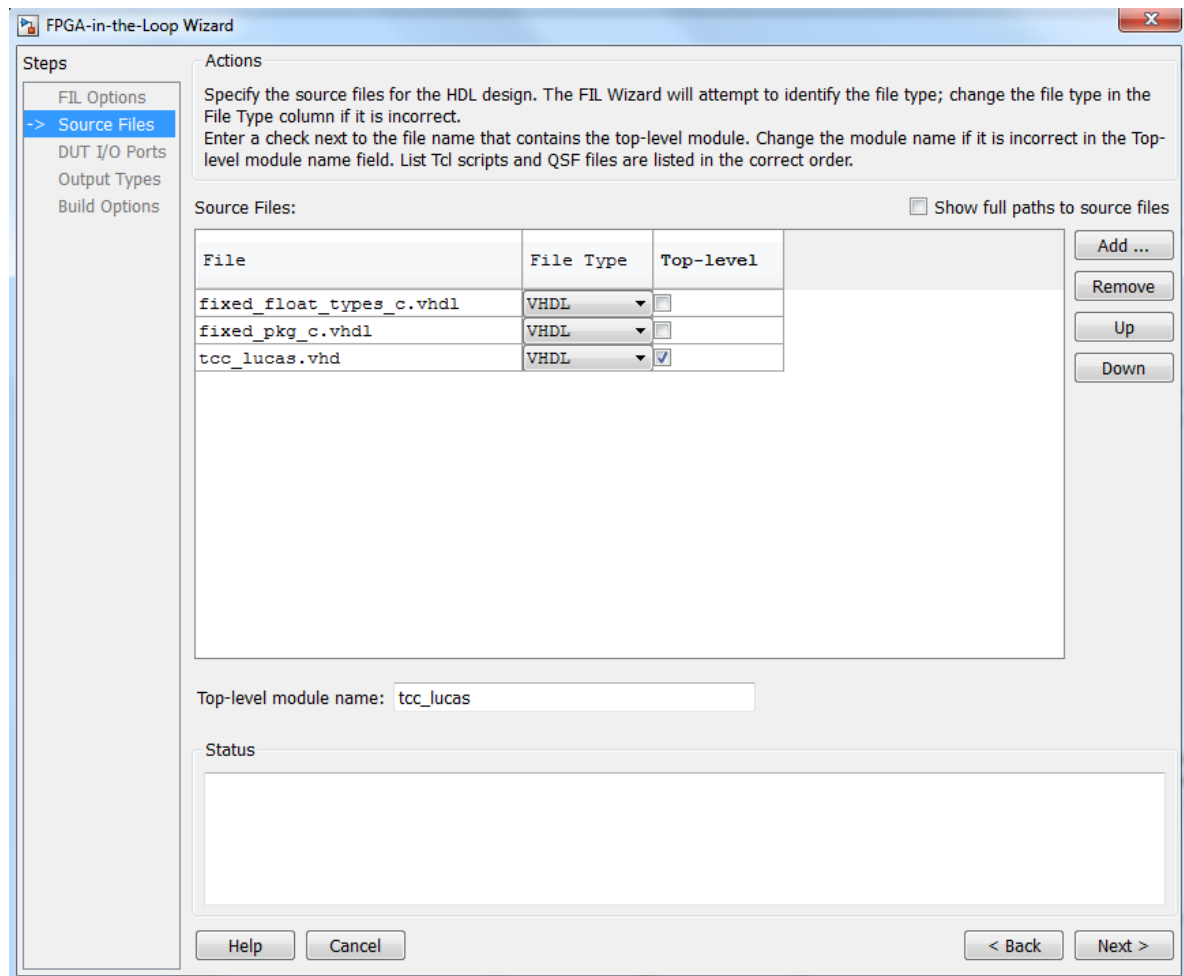
Figura 4.16 – Janela do Assistente de Configuração do FIL: Primeira Etapa.



Fonte: Próprio Autor.

Na segunda etapa, em “*Source Files*”, deve-se inserir todos os arquivos utilizados correspondentes ao algoritmo desenvolvido em VHDL. Os valores das variáveis presentes no algoritmo foram declarados em formato *sfixed* (*signed fixed point*) ou ponto-fixo assinalado. No entanto, para que seja possível a sintetização dos valores definidos em ponto-fixo, faz-se a necessidade da inserção de algumas bibliotecas junto ao arquivo do algoritmo. Na Figura 4.17 estão todos os arquivos VHDL utilizados. Destaca-se que é fundamental que as bibliotecas inseridas estejam acima do arquivo do algoritmo, e além disso, deve-se assinalar o arquivo do algoritmo desenvolvido como arquivo principal (*top-level*), caso contrário, as variáveis em ponto-fixo tem a possibilidade de não serem sintetizadas.

Figura 4.17 - Janela do Assistente de Configuração do FIL: Segunda Etapa.

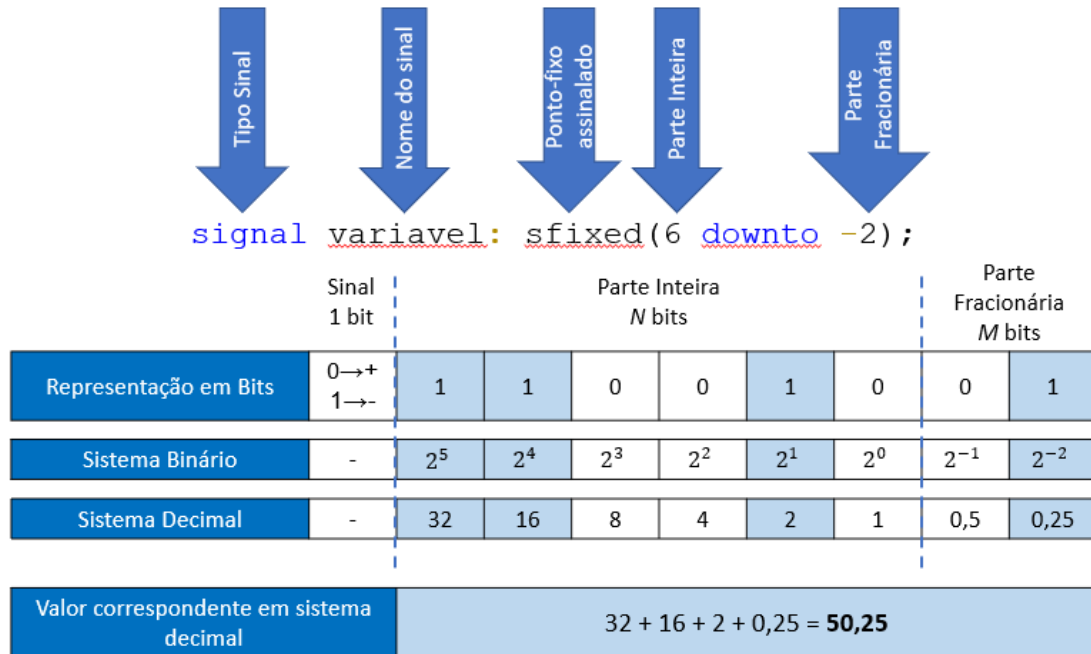


Fonte: Próprio Autor.

Uma variável em ponto-fixo assinalado ou *sfixed*, tem sua representação por N números de bits para a parte inteira, M números de bits para a parte fracionária e 1 bit para o sinal, sendo 0 para sinais positivos ou 1 para sinais negativos. Dessa maneira, cabe ao projetista determinar o tamanho de bits necessários para a representação correta das variáveis de trabalho em ponto-fixo, devendo, portanto, ao realizar operações matemáticas, estar atento aos seus resultados, reconhecendo separadamente as quantidades de bits necessários para as partes inteiras e fracionárias.

Na Figura 4.18 faz-se a ilustração de um exemplo de declaração de uma variável como ponto-fixo assinalado utilizando a linguagem VHDL, onde sabe-se que seu valor é igual a 50,25.

Figura 4.18 – Exemplo de Declaração de Variável como Ponto-Fixo Assinalado em VHDL.



Fonte: Próprio Autor.

Na Tabela 4.1 a seguir, estão expostas as variáveis utilizadas, bem como a quantidade de bits definida para cada uma delas, referentes ao algoritmo de controle CMI-TSF proposto.

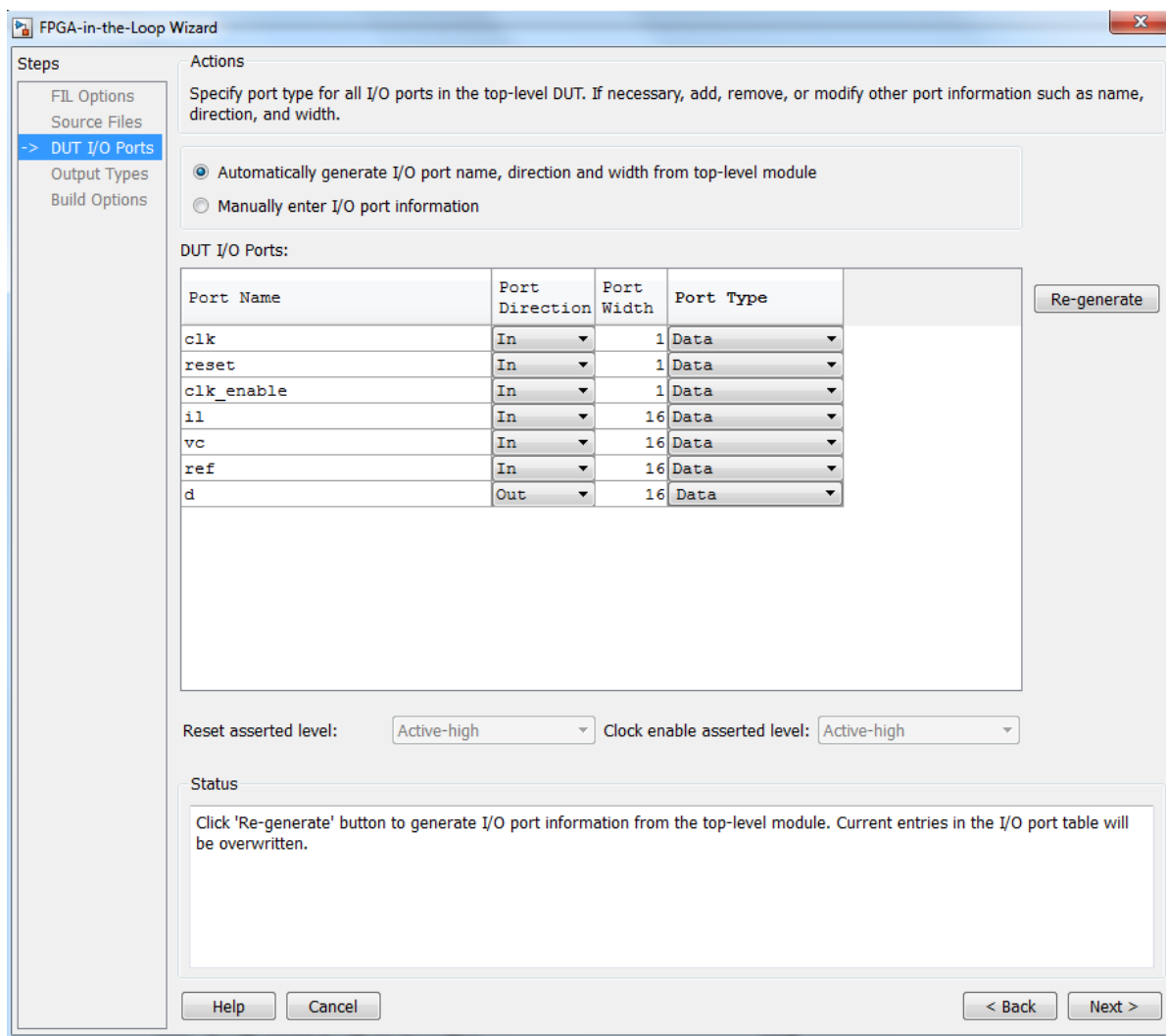
Tabela 4.1 – Variáveis Declaradas em Ponto-Fixo Assinalado com Respectivas Quantidades de Bits.

Variáveis Declaradas	Nº de Bits Sinal	Nº de Bits Parte Inteira	Nº de Bits Parte Fracionária	Total de Bits Utilizados
E	1	7	8	16
ILs, VCs, Refs	1	7	8	16
h1, h2, h3, h4	1	15	16	32
p1, p2, p3, p4	1	7	8	16
i_n, i_n1	1	7	16	24
U	1	8	7	16
F1, F2, F3	1	8	12	21
n1, n2, n3, n4	1	6	11	18
ts100	1	0	14	15
G11, G12, G13, G21, G22, G23, G31, G32, G33, G41, G42, G43	1	4	27	32

Fonte: Próprio Autor.

Na terceira etapa de configuração, em “*DUT I/O Ports*”, deve-se especificar o tipo de cada variável de entrada e saída do FIL, presentes no algoritmo VHDL. Para o presente trabalho o tipo das variáveis foi definido como “*Data*”, pois os valores destes dados foram adquiridos do modelo desenvolvido em MATLAB/SIMULINK. As configurações realizadas em “*DUT I/O Ports*” podem ser visualizadas na Figura 4.19.

Figura 4.19 - Janela do Assistente de Configuração do FIL: Terceira Etapa.



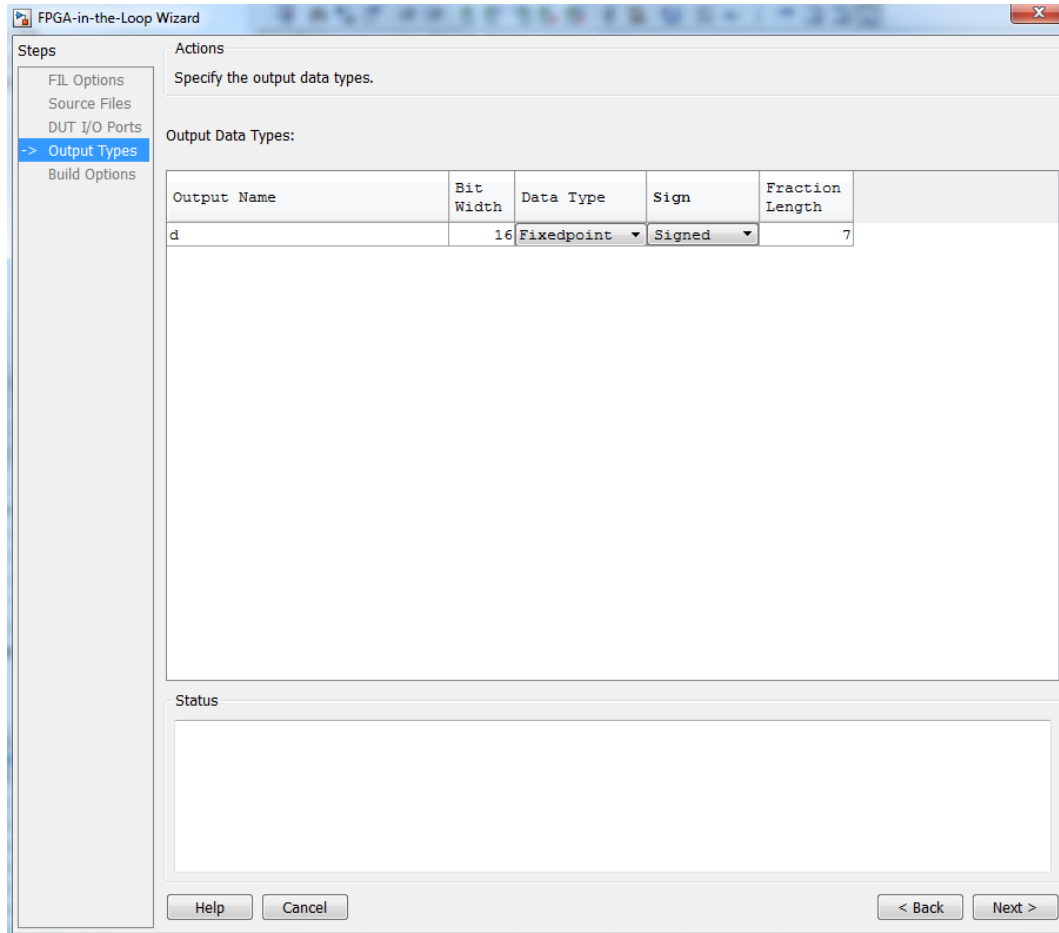
Port Name	Port Direction	Port Width	Port Type
clk	In	1	Data
reset	In	1	Data
clk_enable	In	1	Data
il	In	16	Data
vc	In	16	Data
ref	In	16	Data
d	Out	16	Data

Fonte: Próprio Autor.

Na quarta etapa, em “*Output Types*”, deve-se especificar o tipo de dado utilizado para as variáveis de saída, e também, o número de bits utilizados para a parte fracionária. Como já dito anteriormente, no presente trabalho as variáveis foram

declaradas em ponto-fixo assinalado. A quarta etapa de configuração pode ser observada na Figura 4.20.

Figura 4.20 - Janela do Assistente de Configuração do FIL: Quarta Etapa.

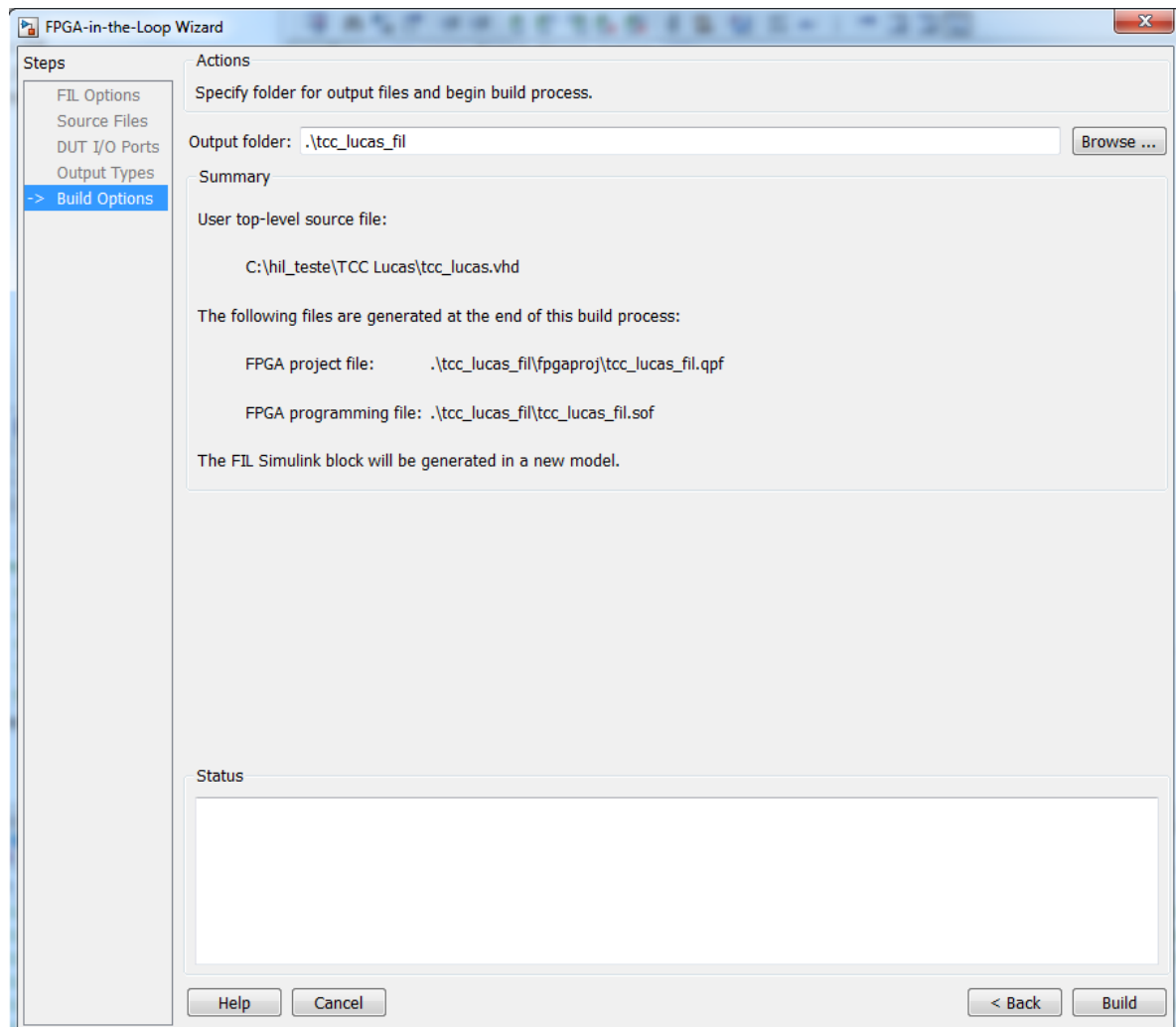


Output Name	Bit Width	Data Type	Sign	Fraction Length
d	16	Fixedpoint	Signed	7

Fonte: Próprio Autor.

Na quinta e última etapa, em “*Build Options*”, deve-se especificar a pasta de destino para os arquivos de saída a serem criados pelo assistente FIL (vide Figura 4.21) e ao clicar em “*Build*”, inicia-se o processo de geração dos arquivos “*FPGA Project file*” em formato “.qpf” e “*FPGA programming file*” em formato “.sof” e, além disso, dá-se início a geração do bloco FIL do SIMULINK.

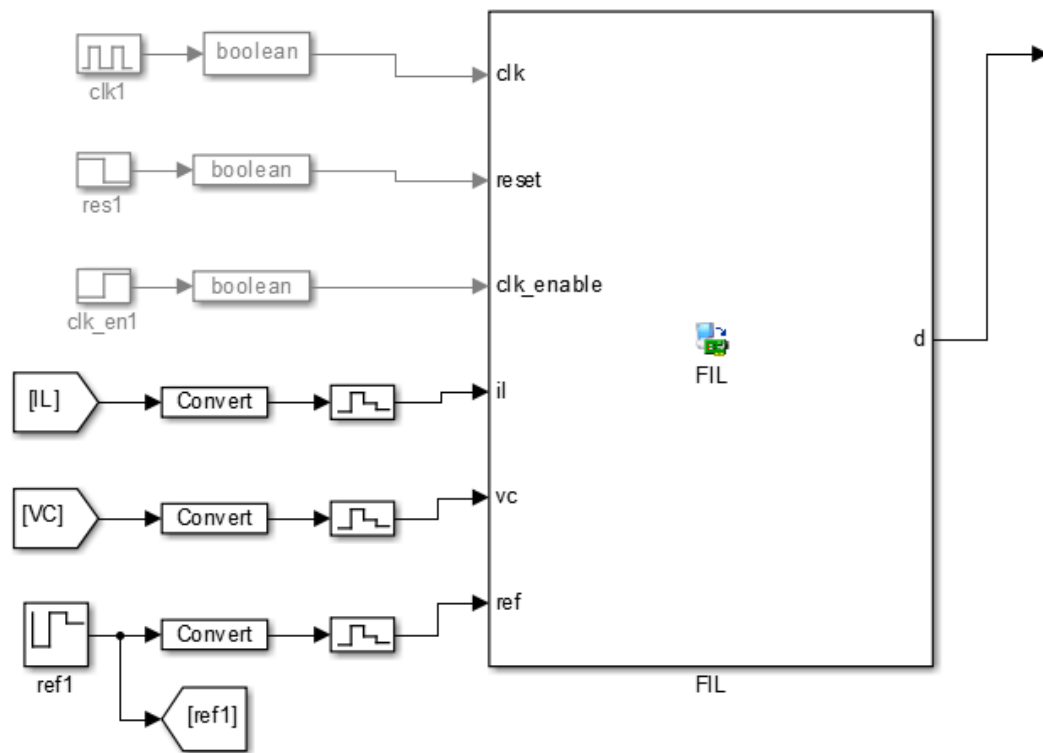
Figura 4.21 - Janela do Assistente de Configuração do FIL: Quinta Etapa.



Fonte: Próprio Autor.

Ao final do processo, o bloco FIL gerado deve ser inserido no *software* MATLAB/SIMULINK, onde se encontra o modelo da planta e conectado corretamente com tempos de amostragem e tipos de variável projetados. O bloco FIL gerado com as entradas e saídas já conectadas no modelo do SIMULINK é ilustrado na Figura 4.22.

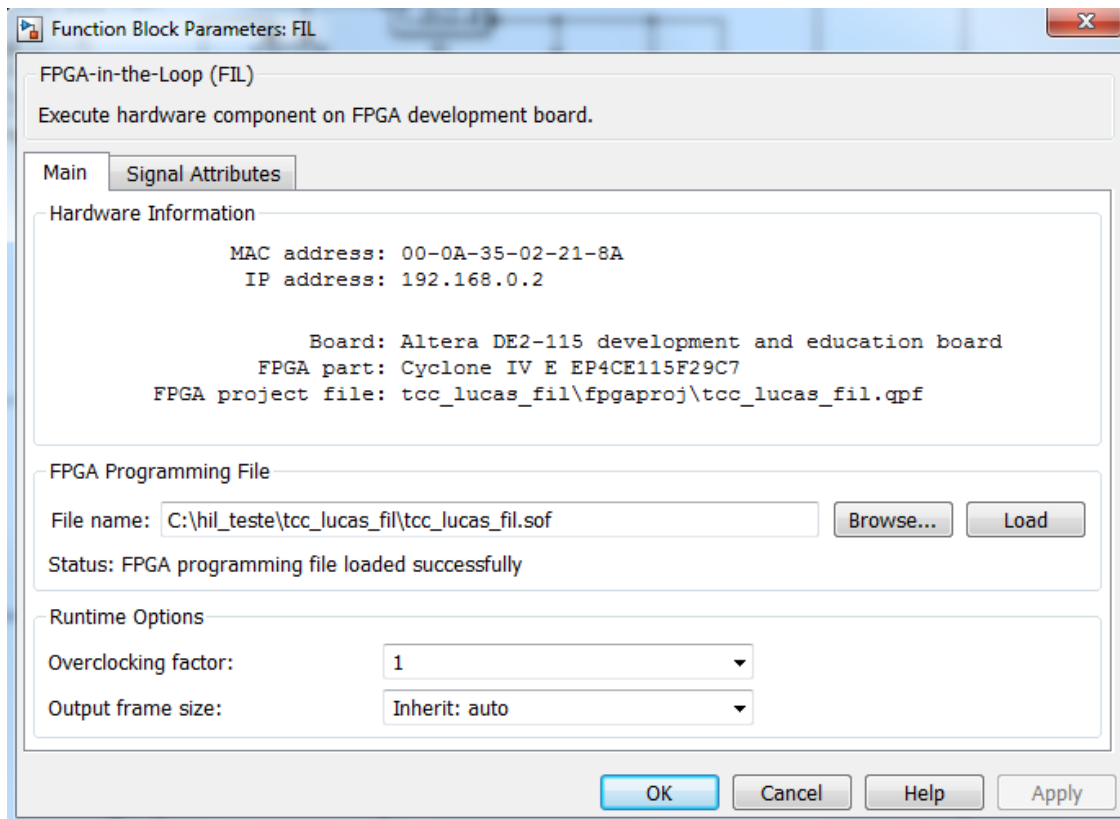
Figura 4.22 – Bloco FIL gerado com Conexões de Entrada e Saída no SIMULINK.



Fonte: Próprio Autor.

Para que o teste FIL possa ser iniciado, é preciso ainda que se faça o carregamento do arquivo “*.sof*”, gerado pelo assistente, via USB no FPGA. Isto pode ser feito ao clicar duas vezes sobre o bloco FIL, onde a janela para efetuar o carregamento é aberta. Deve-se verificar se a porta *Ethernet* do computador está de acordo com a porta de comunicação estipulada para o FIL. A janela de carregamento do algoritmo no FPGA pode ser visualizada na Figura 4.23.

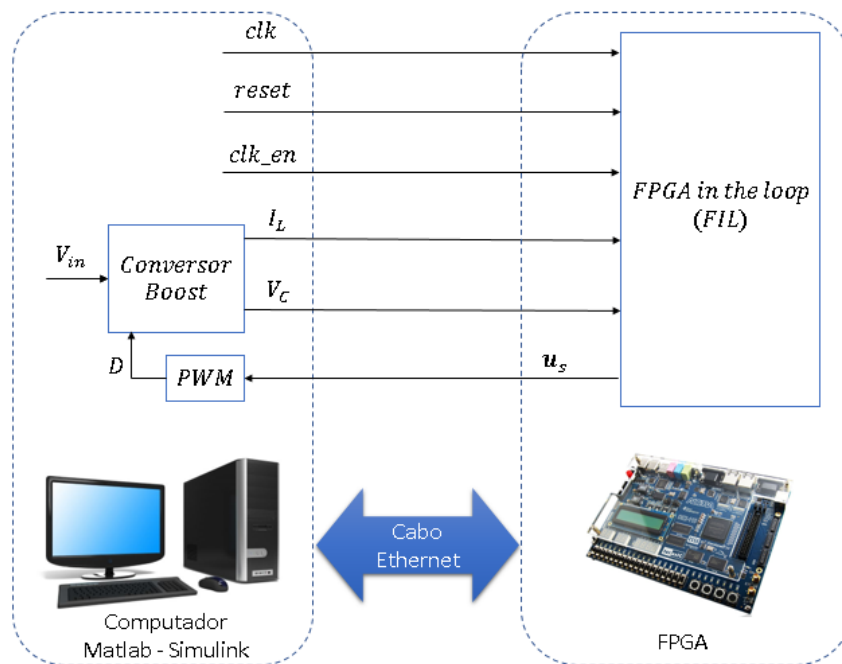
Figura 4.23 - Janela de Carregamento do Algoritmo no FPGA.



Fonte: Próprio Autor.

Após o carregamento do algoritmo desenvolvido no FPGA, finalmente é realizado o teste experimental *FPGA in the Loop*. A Figura 4.24 expõe um diagrama de blocos para o teste FIL realizado.

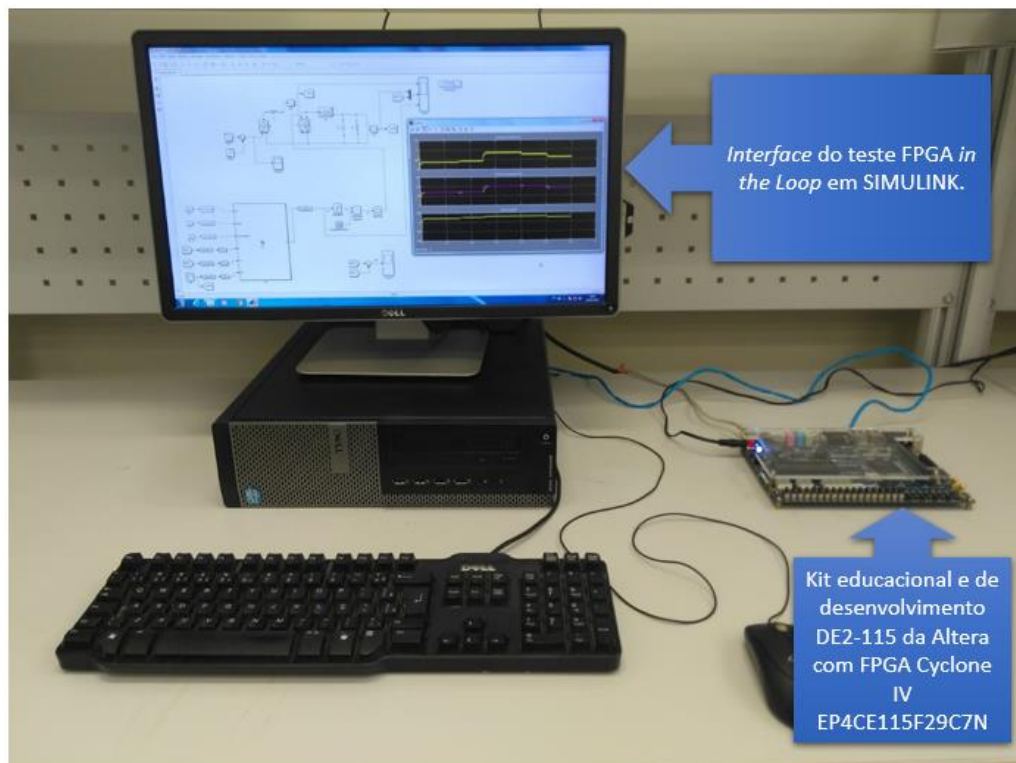
Figura 4.24 – Diagrama de Blocos



Fonte: Próprio Autor.

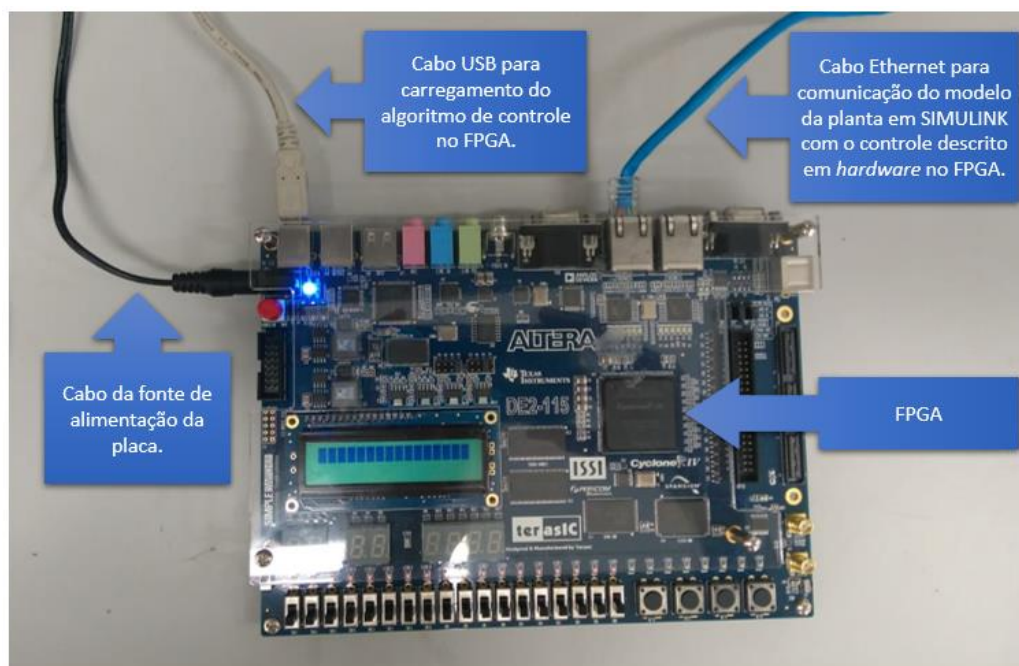
O teste de bancada realizado para o presente trabalho pode ser visualizado na Figura 4.25 e o kit educacional e de desenvolvimento DE2-115 da Altera com FPGA Cyclone IV EP4CE115F29C7N, com as conexões necessárias para a realização do teste, pode ser observado na Figura 4.26.

Figura 4.25 – Teste Experimental FIL em Bancada.



Fonte: Próprio Autor.

Figura 4.26 – Kit DE2-112 da Altera com FPGA Utilizado e Conexões Necessárias para o Teste FIL.

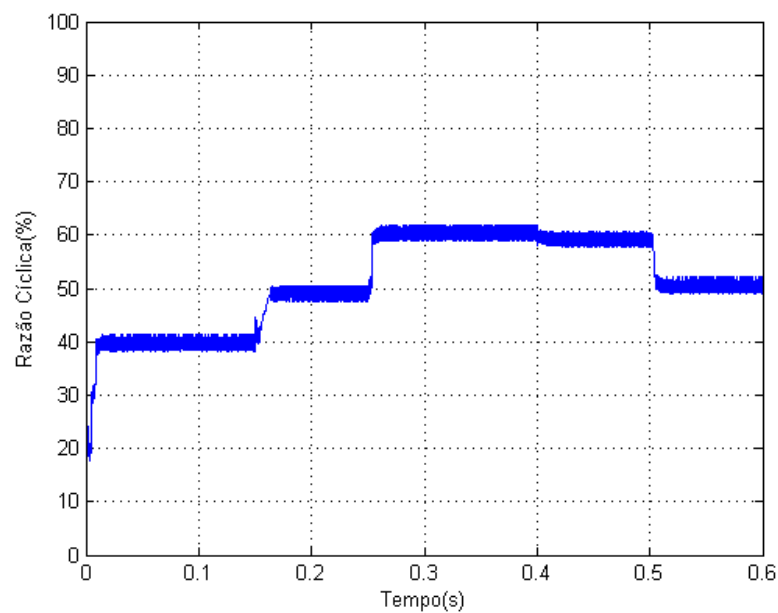


Fonte: Próprio Autor.

Os resultados obtidos do FIL foram organizados de forma análoga aos resultados da simulação em *software* MATLAB/SIMULINK presentes na seção 4.1.

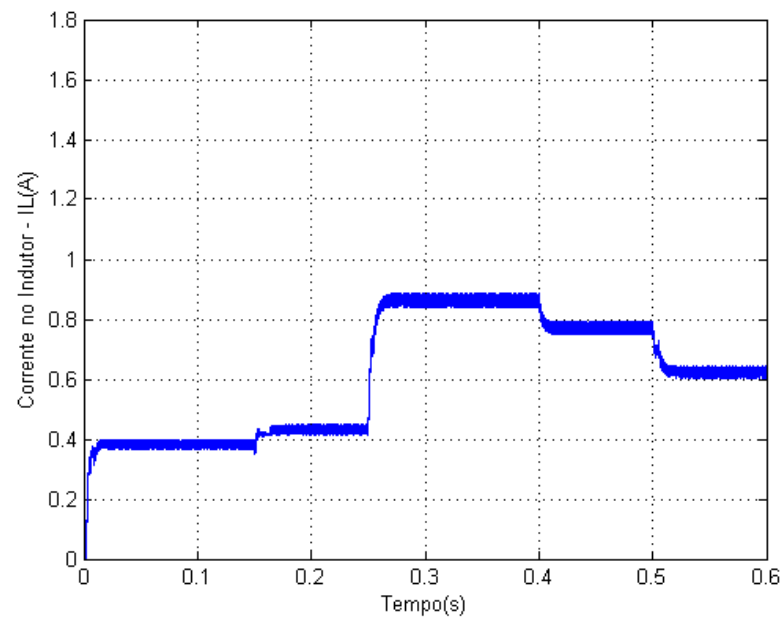
Para a carga $R_1 = 60\Omega$, os resultados obtidos foram apresentados nas Figuras 4.27, 4.28, 4.29 e 4.30.

Figura 4.27 - Teste FIL: Razão Cíclica para $R_1 = 60\Omega$.



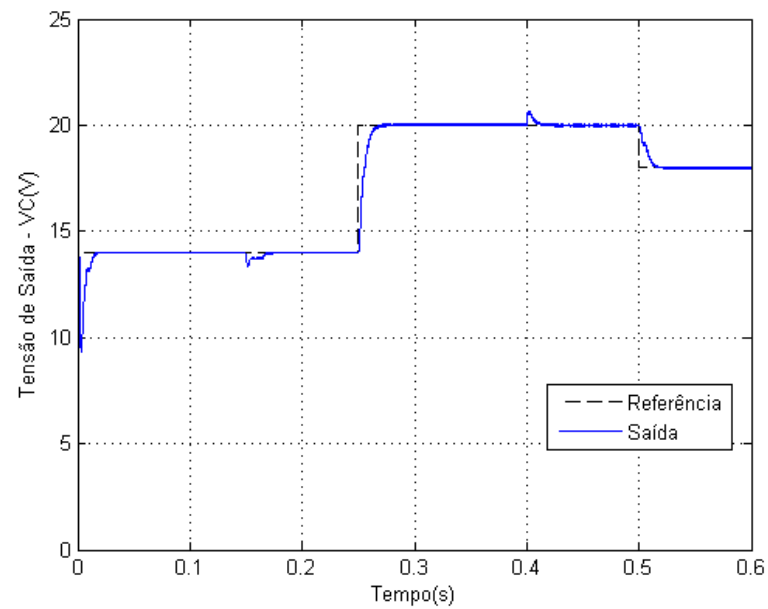
Fonte: Próprio Autor.

Figura 4.28 – Teste FIL: Corrente no Indutor para $R_1 = 60\Omega$.



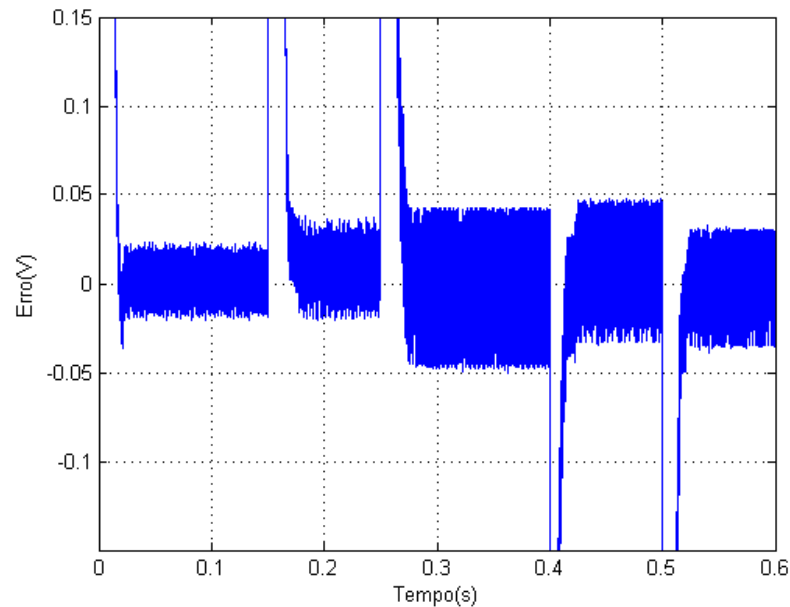
Fonte: Próprio Autor.

Figura 4.29 - Teste FIL: Tensão de Saída para $R_1 = 60\Omega$.



Fonte: Próprio Autor.

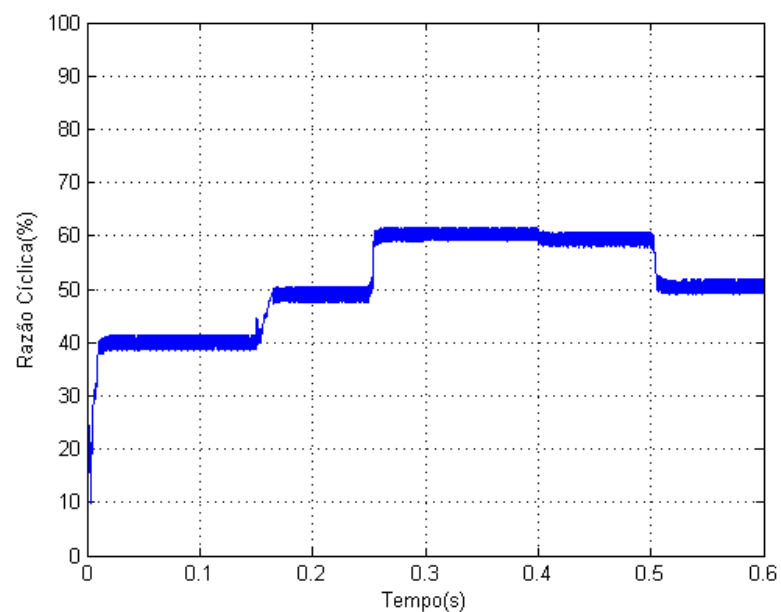
Figura 4.30 - Teste FIL: Erro entre a Referência e a Saída para $R_1 = 60\Omega$.



Fonte: Próprio Autor.

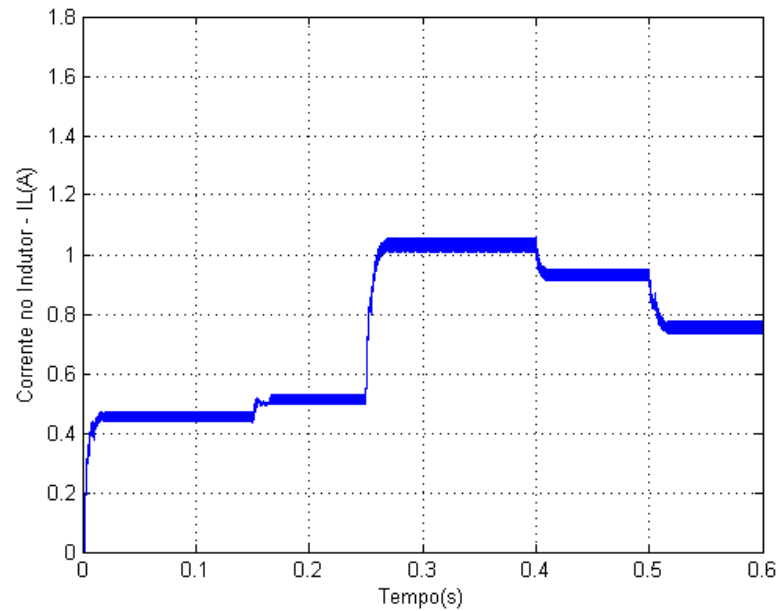
Os resultados de simulação para a carga nominal $R_2 = 50\Omega$ foram apresentados nas Figuras 4.31, 4.32, 4.33 e 4.34.

Figura 4.31 - Teste FIL: Razão Cíclica para $R_2 = 50\Omega$.



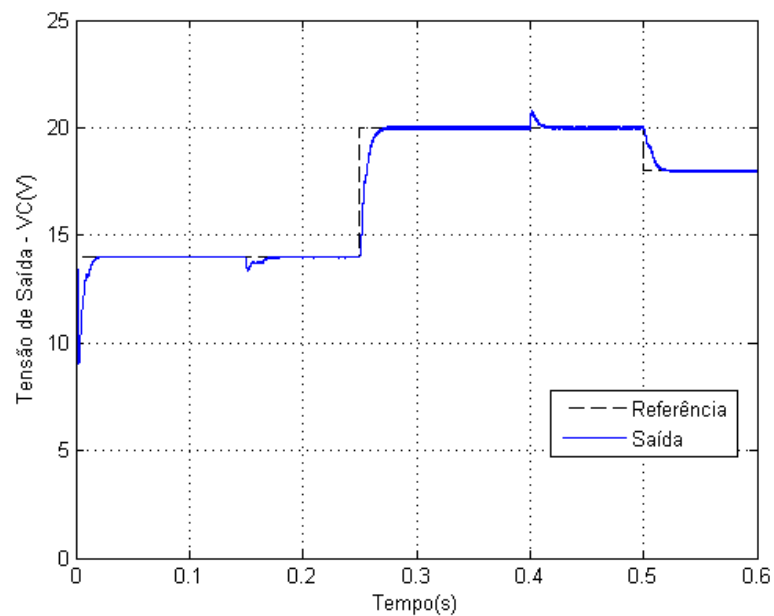
Fonte: Próprio Autor.

Figura 4.32 – Teste FIL: Corrente no Indutor para $R_2 = 50\Omega$.



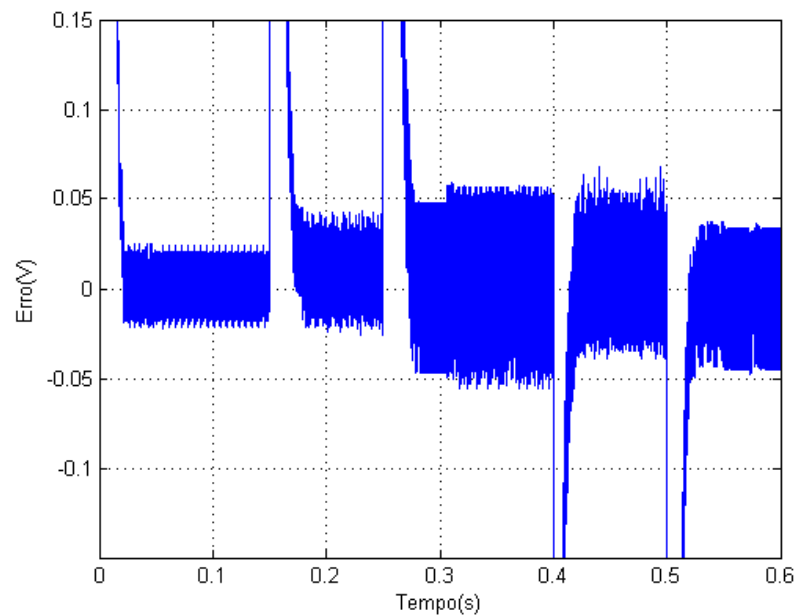
Fonte: Próprio Autor.

Figura 4.33 - Teste FIL: Tensão de Saída para $R_2 = 50\Omega$.



Fonte: Próprio Autor.

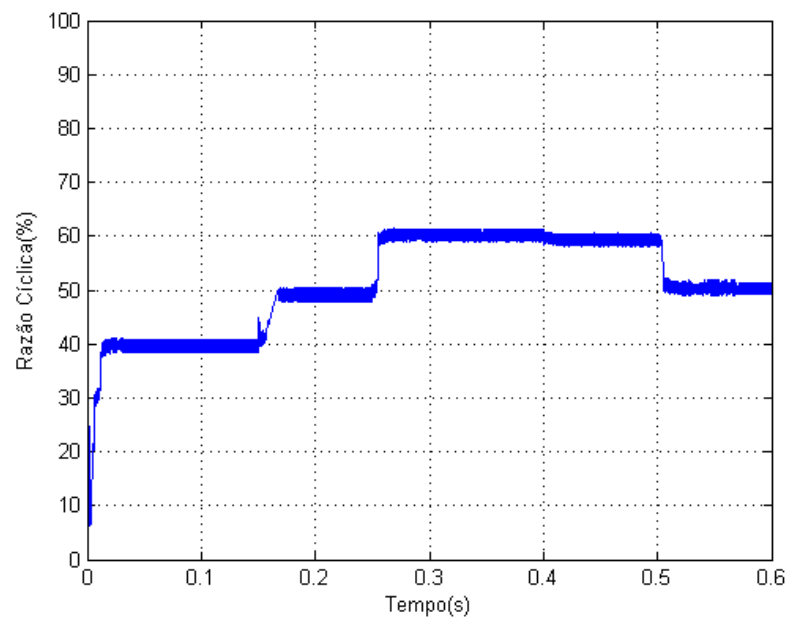
Figura 4.34 - Teste FIL: Erro entre a Referência e a Saída para $R_2 = 50\Omega$.



Fonte: Próprio Autor.

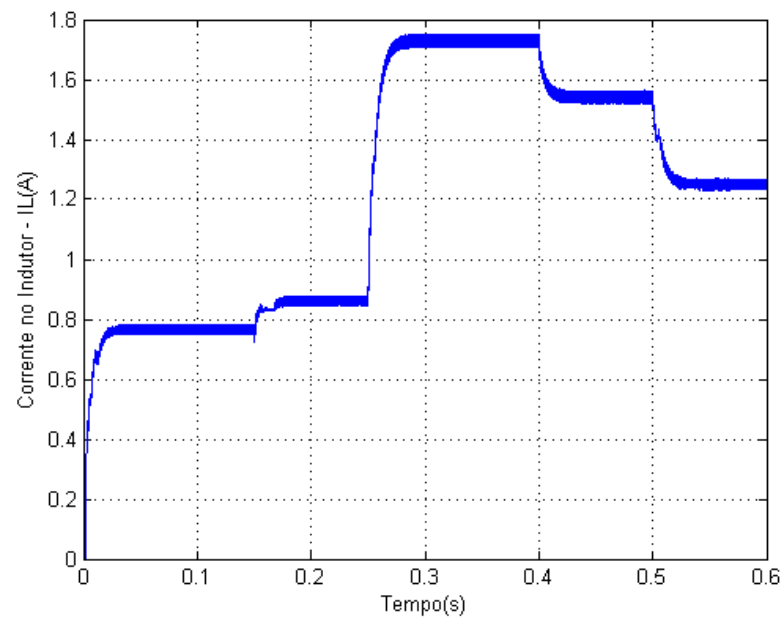
Os resultados de simulação para a carga $R_3 = 30\Omega$ foram apresentados nas Figuras 4.35, 4.36, 4.37 e 4.38.

Figura 4.35 – Teste FIL: Razão Cíclica para $R_3 = 30\Omega$.



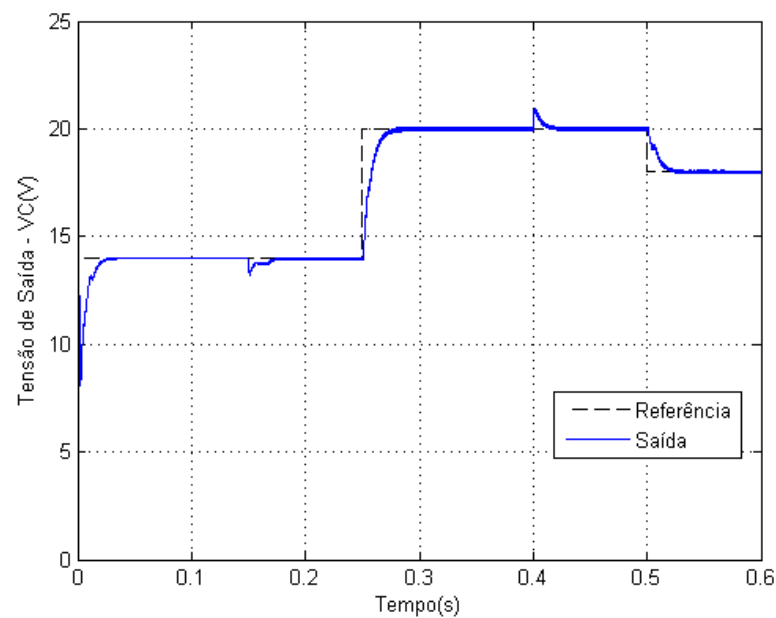
Fonte: Próprio Autor.

Figura 4.36 - Teste FIL: Corrente no Indutor para $R_3 = 30\Omega$.



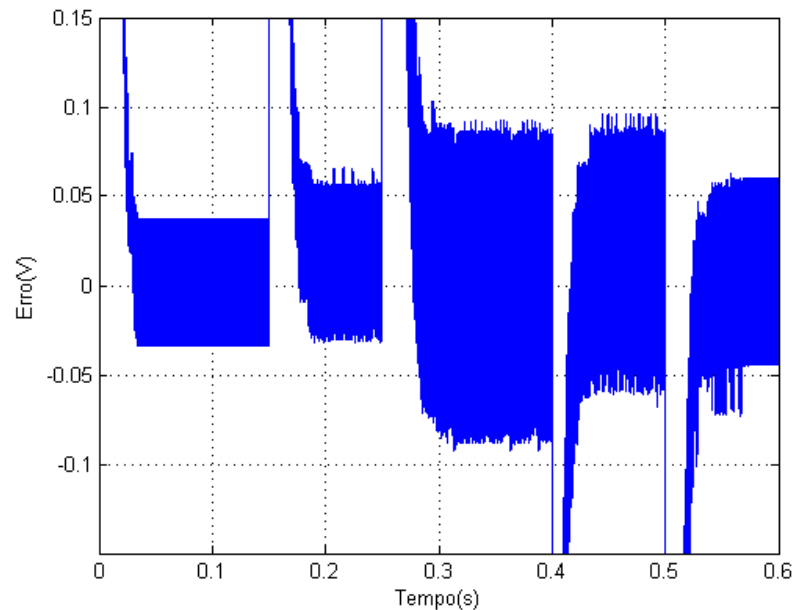
Fonte: Próprio Autor.

Figura 4.37 – Teste FIL: Tensão de Saída para $R_3 = 30\Omega$.



Fonte: Próprio Autor.

Figura 4.38 - Teste FIL: Erro entre a Referência e a Saída para $R_3 = 30\Omega$.



Fonte: Próprio Autor.

Os resultados obtidos pelo FIL foram muito próximos aos resultados anteriormente obtidos para a simulação MATLAB/SIMULINK. Destaca-se assim que a tensão de saída do conversor *Boost* fez o rastreamento da referência definida em ambas as simulações, com erro em regime permanente tendendo à zero e apresentando um *overshoot* desprezível, tanto para as perturbações impostas pela tensão de entrada, quanto pela variação das cargas na saída do conversor controlado.

Desta forma, como os resultados alcançados para o teste FIL foram favoráveis, valida-se assim o controlador proposto implementado em *hardware*, cumprindo-se os objetivos propostos para o presente trabalho.



5 CONCLUSÃO

O presente Trabalho de Conclusão de Curso retrata a implementação de um controle não linear para um conversor *Boost*, baseado na técnica Takagi-Sugeno Fuzzy integrado ao Controlador de Modelagem Interna. Adicionalmente, foram realizadas algumas simplificações matemáticas para a obtenção de uma nova lei de controle simplificada para o controlador CMI-TSF.

A técnica Takagi-Sugeno *Fuzzy* integrada ao Controlador de Modelagem Interna, permite o desenvolvimento de controladores com boa precisão e robustez. Além disso, a dinâmica da planta em malha fechada pode ser descrita pela região de \mathcal{D} -estabilidade, em que se faz a alocação dos pólos por meio de LMIs baseadas na teoria de estabilidade segundo Lyapunov.

O controlador CMI-TSF proposto, com as devidas simplificações realizadas, foi inicialmente implementado e simulado em MATLAB/SIMULINK, onde foram definidas uma perturbação na tensão de entrada, representada por uma queda de tensão, e variações das cargas resistivas aplicadas à saída do conversor *Boost*. Os resultados obtidos para a simulação MATLAB/SIMULINK foram satisfatórios.

Posteriormente, com o intuito de fazer a validação do controlador CMI-TSF, foi desenvolvido um algoritmo em VHDL correspondente ao controlador CMI-TSF simplificado, onde realizou-se sua implementação em FPGA e, juntamente com o modelo da planta descrito em SIMULINK, aplicou-se o teste experimental FPGA *in the Loop*. Analogamente à simulação MATLAB/SIMULINK realizada, foram mantidas a perturbação na tensão de entrada e as variações das cargas na saída do conversor *Boost*. Os resultados alcançados por meio do FIL, evidenciaram a robustez e precisão do controlador CMI-TSF proposto.

Propõe-se para trabalhos futuros o uso do controlador CMI-TSF proposto em um conversor *Boost* físico e também o desenvolvimento de novos controladores CMI-TSF que considerem diferentes regiões de \mathcal{D} -estabilidade.



6 BIBLIOGRAFIA

- Agência Nacional de Energia Elétrica (ANEEL). (17 de 04 de 2012). *RESOLUÇÃO NORMATIVA Nº 482*. Acesso em 29 de Maio de 2018, disponível em <http://www2.aneel.gov.br/cedoc/ren2012482.pdf>
- Altera. (Setembro de 2013). Hardware in the Loop from the MATLAB/Simulink Environment. *Altera Corporation*.
- Andrea, C. Q. (2007). *Modificação de Zeros em Sistemas de Controle Robusto Utilizando LMIs*. Tese de Doutorado, UNESP, Departamento de Engenharia Elétrica, Ilha Solteira.
- Andrea, C. Q. (2007). *Modificação de Zeros em Sistemas de Controle Robusto Utilizando LMIs*. Tese de Doutorado, UNESP, Departamento de Engenharia Elétrica, Ilha Solteira.
- Andrea, C. Q., Pinto, J. P., Carniato, A., Godoy, R., & Rodrigues, D. (2009). Modelagem Exata do Conversor CC-CC Boost via Modelos Fuzzy Takagi-Sugeno. *SBAI 2009 IX Simpósio Brasileiro de Automação Inteligente XII Competição Brasileira de Robótica (CBR) III Olimpíada Brasileira de Robótica*.
- Andrea, C., Batista, E., Leite, L., Pinto, J., & Coelho, M. (Setembro de 2014). Modelagem do Conversor CC-CC Buck-Boost via Fuzzy Takagi-Sugeno: Sistema de Controle e Simulação Computacional. *Anais do XX Congresso Brasileiro de Automática*, 925-930.
- Assef, A. (2017). *Conversor CC-CC Elevador de Tensão Boost*. Acesso em 01 de Junho de 2018, disponível em Universidade Tecnológica Federal do Paraná: http://paginapessoal.utfpr.edu.br/amauriassef/disciplinas/eletronica-de-potencia/apresentacoes/Eletr_Pot1_24.pdf/view
- Associação Brasileira de Energia Solar Fotovoltaica (ABSOLAR). (14 de 05 de 2018). *BRASIL TEM MARCA HISTÓRICA COM MICRO E MINIGERAÇÃO DE ENERGIA SOLAR*. Acesso em 30 de Abril de 2018, disponível em <http://www.absolar.org.br/noticia/noticias-externas/energia-solar-fotovoltaica-atinge-marca-historica-no-brasil.html>



- B. Chen, C. Tseng, & H. J. Uang. (2000). Mixed H2 and H1 Fuzzy output feedback control design for nonlinear dynamic systems: an LMI approach. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 8(3), 249-265.
- Barbi, I. (2015). *Modelagem de Conversores CC-CC Empregando Modelo Médio em Espaço de Estados*. Florianópolis, Santa Catarina, Brasil: Edição do Autor.
- C. S. Ting, P. Apkarian, T. Narikiyo, & M. Kanota. (2006). Stability Analysis and Design of Takagi-Sugeno Fuzzy Systems. *INFORMATION SCIENCES*, 176(19), 2817-2845.
- Carniato, A. A. (2009). *Modelagem Exata do Conversor CC-CC Boost Via Modelos Fuzzy Takagi-Sugeno*. Dissertação de Mestrado, UFMS, Departamento de Engenharia Elétrica, Campo Grande.
- Chen, C. -T. (1999). *Linear System Theory and Design* (3ª ed.). New York: Oxford University Press.
- Chilali, M., & Gahinet, P. (1996). H_∞ design with pole placement constraints: an LMI approach. *IEEE Transactions on Automatic Control*, v.41, n.3, 358-367.
- Dorf, R. C., & Bishop, R. H. (2001). *Sistemas de Controle Modernos* (8ª ed.). Rio de Janeiro: LTC.
- Dorf, R. C., & Bishop, R. H. (2013). *Sistemas de Controle Modernos* (12ª ed.). LTC.
- Dreher, J. R. (2012). *Conversores Boost-Flyback Integrados para Aplicações com Alto Ganho de Tensão*. Dissertação de Mestrado, UTFPR, Departamento de Engenharia Elétrica, Pato Branco - PR.
- Faria, F. A. (2009). *Projetos de Controladores Baseados em LMI usando Realimentação da Derivada dos Estados*. Tese de Doutorado, UNESP, Programa de Pós-Graduação em Engenharia Elétrica, Ilha Solteira - SP.
- García, R. C. (2015). *Controle de Velocidade de Motor Síncrono de Ímã Permanente Utilizando Redes Neurais Artificiais e Multiplexação em Frequência*. Tese de Doutorado, UFRJ / COPPE, Departamento de Engenharia Elétrica, Rio de Janeiro.
- Garcia, R. C., Pinto, J. P., Soares, J. O., Galotto, L. G., & Onizuka, M. V. (2016). Experimental Validation of a New Robust Tracking Controller for Nonlinear SISO Plants Based on Type-II IMC and TS Fuzzy Systems. *IEEE Conference Publications*.



- García, R. C., Suemitsu, W. I., & Pinto, J. O. (2013). Takagi-Sugeno Fuzzy Model and Control of a Boost Converter Using Type-I Internal Model Control. *IECON - 39th Annual Conference of the IEEE Industrial Electronics Society*, 3794-3799.
- Gomez, R. C. (2013). *Projeto de um Hardware de Controle em FPGA para Filtro Ativo Paralelo Trifásico com Fuzzy e PWM a 400 kHz*. Dissertação de Mestrado, UFMS, Departamento de Engenharia Elétrica, Campo Grande.
- H. D. Tuan, P. Apkarian, T. Narikiyo, & M. Kanota. (2004). New Fuzzy Control Model and Dynamic Output Feedback Parallel Distributed Compensation. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 12(1), 13-21.
- Junior, L. G. (2011). *Inversores Integrados Monofásicos Aplicados em Sistemas Fotovoltaicos com Conexão à Rede de Distribuição de Energia Elétrica*. Tese de Doutorado, UNESP, Departamento de Engenharia Elétrica, Ilha Solteira - SP.
- Kumar, M., Ruber, L., & Jovanovic, M. (2017). Verification of Control Design and Implementation for Power Supplies by FPGA-in-the-loop Simulation. *Applied Power Electronics Conference and Exposition - APEC*, 3114-3121.
- Martins, D. C., & Barbi, I. (2006). *Eletrônica de Potência: Conversores CC-CC Básicos Não Isolados*. Florianópolis, Santa Catarina, Brasil: Edição dos Autores.
- National Instruments. (Dezembro de 2016). *Introdução à tecnologia FPGA*. Acesso em Novembro de 2018, disponível em Site da National Instruments: <http://www.ni.com/white-paper/6984/pt/>
- Ogata, K. (2011). *Engenharia de Controle Moderno* (5ª ed.). Pearson.
- Pr. Dr. Moacyr de Brito. (2017). *Eletrônica de Potência - Aula 05 - Conversor Boost*. Fonte: Notas de Aula.
- Prof.^a Luiza M. R. Codá. (s.d.). *Apostila de Introdução a VHDL*. Acesso em Novembro de 2018, disponível em Departamento de Engenharia Elétrica e de Computação - EESC / USP: https://edisciplinas.usp.br/pluginfile.php/530833/mod_resource/content/1/Apostila%20de%20Introdu%C3%A7%C3%A3o%20a%20VHDL_2014.pdf
- Rogers, P., Kavasseri, R., & Smith, S. (2017). An FPGA-in-the-loop Approach for HDL Motor Controller Verification. *International Conference on ReConfigurable Computing and FPGAs - ReConFig*, 1-6.



- Skogestad, S., & Postlethwaite, I. (2005). *Multivariable Feedback Control: Analysis and Design* (2ª ed.). WILEY.
- T. Takagi, & M. Sugeno. (1985). Fuzzy Identification of a System and its Application to Modeling and Control. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 15(1), 116-132.
- Tanaka, K., Ikeda, T., & Wang, H. O. (1998). Fuzzy Regulators and Fuzzy Observers: Relaxed Satability Conditions and LMI-Based Designs. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 6(2), 250-265.
- Teixeira, M., Pietrobon, H., & Assunção, E. (2000). Novos Resultados sobre a Estabilidade e Controle de Sistemas Não-Lineares utilizando Modelos Fuzzy e LMI. *Controle & Automação*, 11(1), 37-48.



APÊNDICE I – Algoritmo de Controle Implementado no MATLAB

```
clc; close all;
tclk = 1/200000;

%Parâmetros Conversor Boost-----
ts=1/50000; %tempo de amostragem (fs=50kHz)
r = 50;
L = 2.4e-3;
C = 50e-6;
Vin=9;

%Barreiras da Região definida pelas LMIs-----
y1=68;
y2=4500;

%Corrente no Indutor-----
iL=0.01:0.01:1.5;
b1=-1*iL/C;
b11=max(b1); b12=min(b1);

%Tensão no Capacitor-----
vc=Vin:0.1:3*Vin;
b2=vc/L;
b21=max(b2); b22=min(b2);

%Constantes para Cálculo das Funções de Pertinência-----
Dc = (1/(b11-b12))*(1/(b21-b22));
S = Dc/(L*C);
n1 = -b12*C;
n2 = -b11*C;
n3 = b22*L;
n4 = b21*L;

%Modelos Locais TSF-----
A= [-1/(r*C) 1/C; -1/L 0];
C0=[1 0];
B1= [b11; b21];
B2= [b11; b22];
B3= [b12; b21];
B4= [b12; b22];

%Modelos Locais com Integração CMI-TSF-----
Aa=[A [0;0]; -C0 0];
Ba1=[B1;0];
Ba2=[B2;0];
Ba3=[B3;0];
Ba4=[B4;0];
MB=[Ba1 Ba2 Ba3 Ba4];

%Definição das LMIs-----
setlmis([]);
Q=lmivar(1, [3 1]);
M1=lmivar(2, [1 3]); %Mj, j=1
M2=lmivar(2, [1 3]); %Mj, j=2
M3=lmivar(2, [1 3]); %Mj, j=3
M4=lmivar(2, [1 3]); %Mj, j=4
```



```
MM=[M1 M2 M3 M4];
n=0;
%Para i=j-----
for i=1:4

    Aai = Aa;
    if (i==1)
        Bai=Ba1;
        Mi = M1;
    elseif (i==2)
        Bai=Ba2;
        Mi = M2;
    elseif (i==3)
        Bai=Ba3;
        Mi = M3;
    else %i=4
        Bai=Ba4;
        Mi = M4;
    end;

    %LMIs para i=j-----
    %LMI - Estabilidade Segundo Lyapunov
    n=n+1;
    lmiterm([n,1,1,Q],Aai,1,'s');
    lmiterm([n,1,1,Mi],Bai,-1,'s');
    %LMI - Barreira na Direita (y1)
    n=n+1;
    lmiterm([n,1,1,Q],Aai,1,'s');
    lmiterm([n,1,1,Mi],Bai,-1,'s');
    lmiterm([n,1,1,Q],2*y1,1);
    %LMI - Barreira na Esquerda (y2)
    n=n+1;
    lmiterm([-n,1,1,Q],Aai,1,'s');
    lmiterm([-n,1,1,Mi],Bai,-1,'s');
    lmiterm([-n,1,1,Q],2*y2,1);
end

%Para i<j-----
for i=1:4
    for j=1:4
        if (i<j)
            Bai = MB(:,i);
            Baj = MB(:,j);
            Mi = MM(i);
            Mj = MM(j);
            %LMI - Estabilidade Segundo Lyapunov
            n=n+1;
            lmiterm([n,1,1,Q],Aa,1,'s');
            lmiterm([n,1,1,Mi],Bai,-0.5,'s');
            lmiterm([n,1,1,Mj],Baj,-0.5,'s');
            %LMI - Barreira na Direita (y1)
            n=n+1;
            lmiterm([n,1,1,Q],Aa,1,'s');
            lmiterm([n,1,1,Mi],Bai,-0.5,'s');
            lmiterm([n,1,1,Mj],Baj,-0.5,'s');
            lmiterm([n,1,1,Q],2*y1,1);
```



```
%LMI - Barreira na esquerda (y2)
n=n+1;
lmiterm([-n,1,1,Q],Aa,1,'s');
lmiterm([-n,1,1,Mi],Bai,-0.5,'s');
lmiterm([-n,1,1,Mj],Baj,-0.5,'s');
lmiterm([-n,1,1,Q],2*y2,1);

end

end

end

%LMI - Definição de Q>0 (positiva definida)
n=n+1;
lmiterm([-n,1,1,Q],1,1);

%Solução das LMIs-----
solucao=getlmis;
[tmin, xfeasp]=feasp(solucao); % Cálculo de Mi e Q
if (tmin < 0)
    %Decodificar Q
    Q = dec2mat(solucao,xfeasp,Q);
    %Decodificar Mi
    M1 = dec2mat(solucao,xfeasp,M1);
    M2 = dec2mat(solucao,xfeasp,M2);
    M3 = dec2mat(solucao,xfeasp,M3);
    M4 = dec2mat(solucao,xfeasp,M4);
    %Cálculo dos Ganhos Kej
    Ke1=M1*inv(Q);
    Ke2=M2*inv(Q);
    Ke3=M3*inv(Q);
    Ke4=M4*inv(Q);
    disp([Ke1;Ke2;Ke3;Ke4])
    %Cálculo dos Ganhos Gej (Ganhos Simplificados)
    G1 = Ke1*S;
    G2 = Ke2*S;
    G3 = Ke3*S;
    G4 = Ke4*S;

    Ge1=[100 100 1].*G1;
    Ge2=[100 100 1].*G2;
    Ge3=[100 100 1].*G3;
    Ge4=[100 100 1].*G4;
    disp([G1;G2;G3;G4])
    disp([Ge1;Ge2;Ge3;Ge4])
else
    disp('não existe solução')
end

hdlsetuptoolpath('ToolName','Altera Quartus II','ToolPath','C:\altera\12.✓
1\quartus\bin\quartus.exe');
```



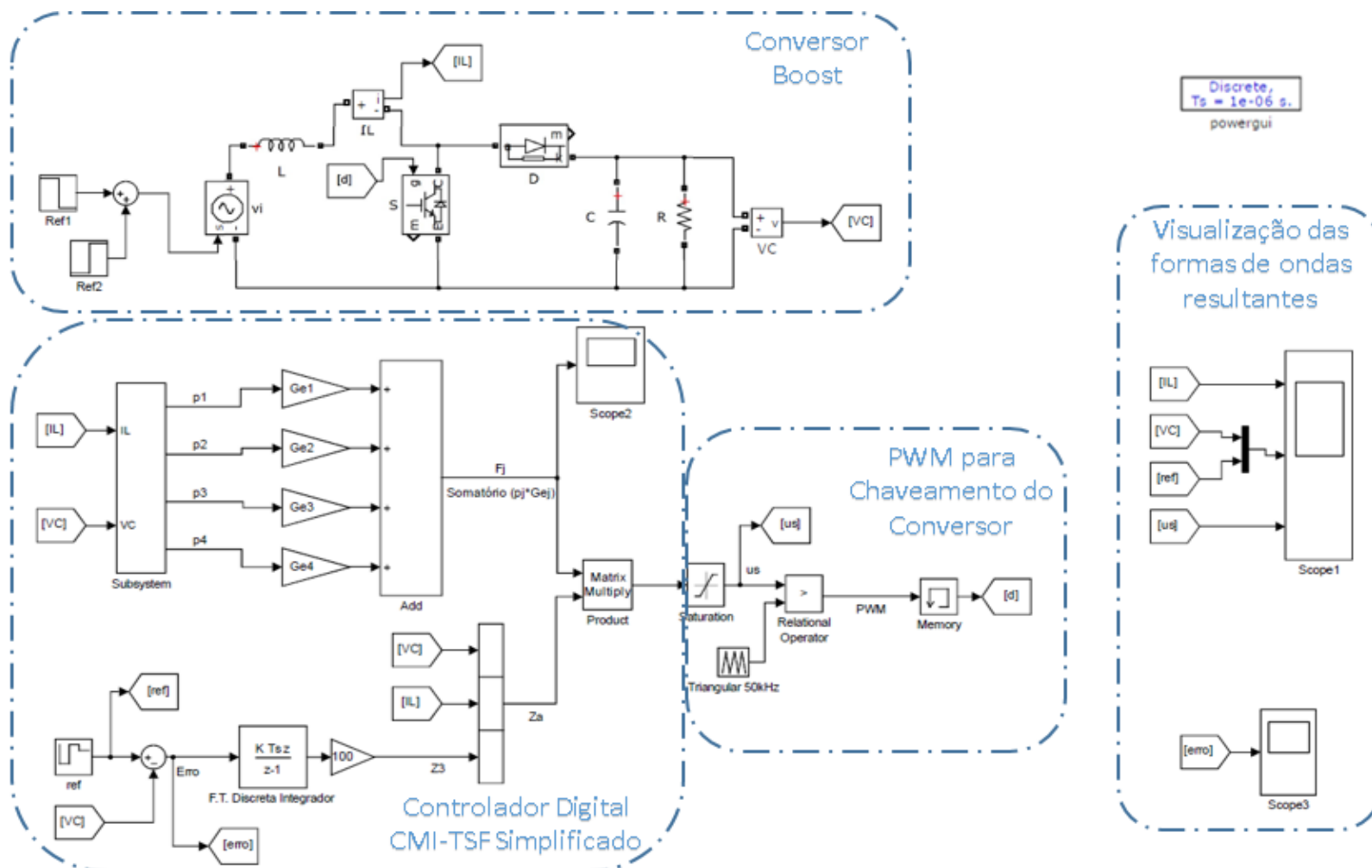

APÊNDICE II – Algoritmo de Controle em VHDL

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  use ieee.std_logic_unsigned.all;
5  use ieee.numeric_std.all;
6  use ieee.fixed_float_types.all;
7  use ieee.fixed_pkg.all;
8
9  entity tcc_lucas is port (
10     clk       : in  std_logic;
11     reset     : in  std_logic;
12     clk_enable : in  std_logic;
13
14     IL        : in  std_logic_vector(15 downto 0);
15     VC        : in  std_logic_vector(15 downto 0);
16     Ref       : in  std_logic_vector(15 downto 0);
17     D         : out std_logic_vector(15 downto 0)
18
19 ); end entity;
20
21 architecture arq of tcc_lucas is
22     --Variavel para contagem
23     signal cont : std_logic_vector(7 downto 0);
24     --Declaracao em ponto fixo das variaveis auxiliares
25     signal e : sfixed(7 downto -8);
26     signal ILs, VCs, Refs: sfixed(7 downto -8);
27     signal h1, h2, h3, h4: sfixed(15 downto -16);
28     signal p1, p2, p3, p4: sfixed(7 downto -8);
29     signal i_n, i_n1: sfixed(7 downto -16);
30     signal u: sfixed(8 downto -7);
31     signal F1, F2, F3 : sfixed(8 downto -12);
32     -- Declaracao em ponto-fixo das constantes
33     signal n1, n2, n3, n4 : sfixed(6 downto -11);
34     signal ts100 : sfixed(0 downto -14);
35     signal G11,G12,G13,G21,G22,G23,G31,G32,G33,G41,G42,G43 : sfixed(4 downto -27);
36
37     begin
38         -- conversao para ponto fixo
39         ILs <= to_sfixed(IL, ILs);
40         VCs <= to_sfixed(VC, VCs);
41         Refs <= to_sfixed(Ref, Refs);
42
43         process (clk, clk_enable, reset) is
44
45         begin
46             if (reset = '1') then
47                 -- Declaracao dos valores das constantes
48                 D <= (others => '0'); -- D = 0 (zera a saida)
49                 n1 <= to_sfixed(1.5,n1);
50                 n2 <= to_sfixed(0.01,n2);
51                 n3 <= to_sfixed(9,n3);
52                 n4 <= to_sfixed(27,n4);
53                 ts100 <= to_sfixed(0.002,ts100);
54                 -- Declaracao dos valores dos Ganhos
55                 G11 <= to_sfixed(0.000474727003537,G11);
56                 G12 <= to_sfixed(1.340622962880009,G12);
57                 G13 <= to_sfixed(-0.443137315055666,G13);
58                 G21 <= to_sfixed(0.002743603858124,G21);
59                 G22 <= to_sfixed(4.028016223663143,G22);
60                 G23 <= to_sfixed(-1.336196714325886,G23);
61                 G31 <= to_sfixed(0.052143094638523,G31);
62                 G32 <= to_sfixed(1.508042474708270,G32);
63                 G33 <= to_sfixed(-0.604767045916095,G33);
64                 G41 <= to_sfixed(0.147644994196773,G41);
65                 G42 <= to_sfixed(3.408889756914117,G42);
66                 G43 <= to_sfixed(-1.286849253568274,G43);
```



```
67
68      -- Zerando os valores das variaveis
69      h1   <= (others =>'0');
70      h2   <= (others =>'0');
71      h3   <= (others =>'0');
72      h4   <= (others =>'0');
73      e    <= (others =>'0');
74      i_n   <= (others =>'0');
75      i_n1  <= (others =>'0');
76      p1 <= (others =>'0');
77      p2 <= (others =>'0');
78      p3 <= (others =>'0');
79      p4 <= (others =>'0');
80      F1   <= (others =>'0');
81      F2   <= (others =>'0');
82      F3   <= (others =>'0');
83      u    <= (others =>'0');
84
85  elsif (rising_edge(clk)) then
86      if (clk_enable = '1') then
87          -- Contador para processos sequenciais
88          if (cont = 3) then
89              cont <= (others =>'0');
90          else
91              cont <= cont + 1;
92          end if;
93          -- Algoritmo de controle
94          --Calculo dos funcoes de pertinencia simplificadas
95          if (cont=0) then
96              h1 <= resize((ILs-n1),h1);
97              h2 <= resize((VCs-n3),h2);
98              h3 <= resize((n4-VCs),h3);
99              h4 <= resize((n2-ILs),h4);
100             e <= resize(Refs-VCs,e);
101         elsif (cont=1) then
102             p1 <= resize((h1*h2),p1);
103             p2 <= resize((h1*h3),p2);
104             p3 <= resize((h4*h2),p3);
105             p4 <= resize((h4*h3),p4);
106             -- Integrador por meio da transformada Z
107             i_n <= resize((i_n1+ts100*e),i_n);
108         elsif (cont=2) then
109             i_n1 <= i_n;
110             -- Soma da Multiplicacao dos pj pelos Gj:
111             F1 <= resize((p1*G11+p2*G21+p3*G31+p4*G41),F1);
112             F2 <= resize((p1*G12+p2*G22+p3*G32+p4*G42),F2);
113             F3 <= resize((p1*G13+p2*G23+p3*G33+p4*G43),F3);
114         elsif (cont=3) then
115             -- razao ciclica
116             u <= resize(((VCs*F1)+(ILs*F2)+(i_n*F3)),u);
117             D <= to_slv(resize(((VCs*F1)+(ILs*F2)+(i_n*F3)),u));
118         end if;
119     end if;
120     --*****
121 end if;
122 end process;
123
124 end;
```

APÊNDICE III – Diagrama de Blocos: Simulação SIMULINK/MATLAB



APÊNDICE IV – Diagrama de Blocos: Teste FIL

