

UNIVERSIDADE DE SÃO PAULO – USP
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Daniel Silva de Castro

**Modelagem de Conversores de Potência para
Implementação em *Hardware-in-the-Loop* com
MCUs de Alto Desempenho**

Daniel Silva de Castro

Modelagem de Conversores de Potência para
Implementação em *Hardware-in-the-Loop* com
MCUs de Alto Desempenho

Dissertação de mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Escola de Engenharia de São Carlos como parte dos requisitos para a obtenção do título de Mestre em Ciências.

Área de concentração: Sistemas Dinâmicos

Orientador: Prof. Dr. Ricardo Quadros Machado

São Carlos – SP

2021

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o Programa de Pós-Graduação de Engenharia Elétrica.

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

S355m Silva de Castro, Daniel
Modelagem de Conversores de Potência para
Implementação em Hardware-in-the-Loop com MCUs de Alto
Desempenho / Daniel Silva de Castro; orientador Ricardo
Quadros Machado. São Carlos, 2021.

Dissertação (Mestrado) - Programa de
Pós-Graduação em Engenharia Elétrica e Área de
Concentração em Sistemas Dinâmicos -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2021.

1. Eletrônica de Potência. 2. Simulação em
Tempo Real. 3. Hardware-in-the-Loop. 4. HIL. I. Título.

FOLHA DE JULGAMENTO

Candidato: Engenheiro **DANIEL SILVA DE CASTRO**.

Título da dissertação: “Modelagem de conversores de potência para implementação em Hardware-in-the-Loop com MCUs de alto desempenho”.

Data da defesa: 27/08/2021.

Comissão Julgadora

Resultado

Prof. Associado **Ricardo Quadros Machado**
(Orientador)

(Escola de Engenharia de São Carlos – EESC/USP)

Aprovado

Prof. Dr. **Pedro Gomes Barbosa**

(Universidade Federal de Juiz de Fora/UFJF)

_ Aprovado _

Prof. Dr. **Marcelo Suetake**

(Universidade Federal de São Carlos/UFSCar)

_ Aprovado _

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:

Prof. Dr. **João Bosco Augusto London Junior**

Presidente da Comissão de Pós-Graduação:

Prof. Titular **Murilo Araujo Romero**

Agradecimentos

Primeiramente, gostaria de agradecer a Deus pelas oportunidades concedidas e por ter me dado forças para seguir adiante.

Agradeço à minha esposa Jamilly por me apoiar e por sempre ter estado ao meu lado, inclusive durante os momentos mais difíceis e desafiadores da minha trajetória.

Agradeço aos meus pais, José Francisco e Edilce, pelo encorajamento e por terem me incentivado a estudar, e às minhas duas irmãs, Andréia e Adriana, por sempre terem me apoiado.

Agradeço aos amigos do LAFAPE, pela parceria e pelos muitos momentos de descontração em meio as turbulências da vida acadêmica, e aos amigos da Tecumseh do Brasil, pelas muitas "cavalgadas" e pelas dicas quentes do café com bolsa.

Gostaria também de expressar os meus sinceros agradecimentos ao meu orientador, Professor Doutor Ricardo Quadros Machado, e também à Professora Doutora Vilma Alves de Oliveira, pela amizade, pelos ensinamentos e por terem me auxiliado de forma excepcional durante o período da minha supervisão.

Por fim, agradeço à CAPES pelo apoio financeiro prestado durante o desenvolvimento deste trabalho de mestrado.

“For God so loved the world, that He gave his only begotten Son, that whosoever believeth in him should not perish, but have everlasting life.”

– John 3:16

Resumo

CASTRO, D. S. **Modelagem de Conversores de Potência para Implementação em *Hardware-in-the-Loop* com MCUs de Alto Desempenho.** 125 p. Dissertação de mestrado – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2021.

Neste trabalho, é abordada a modelagem de conversores de potência para implementação em *Hardware-in-the-Loop* (HIL) com o uso de unidades microcontroladas (MCUs) de alto desempenho. Duas técnicas de modelagem de circuitos elétricos chaveados são discutidas, as técnicas por discretização de circuito elétrico associado (ADC) e por espaço de estado, nos quais são apresentados os métodos dos subcircuitos e do modelo equivalente de chave. Os cenários de teste incluem o conversor CC–CC *boost*, e quatro configurações diferentes de conversor CC–CA, as versões monofásicas em ponte completa do VSI, nos modos ilhado e conectado, do ZSI e do Quasi–ZSI. As simulações foram realizadas por meio do *software* Simulink[®], da Mathworks[®], em que foram testados todos os modelos dinâmicos dos conversores mencionados. Devido à necessidade de se empregar tempos de amostragem menores que 100 ns para manter alta precisão numérica nos resultados, tanto a técnica ADC quanto o método do modelo equivalente de chave não são adequados para implementações em MCUs de alto desempenho. O método dos subcircuitos, em contrapartida, proporciona resultados com alta precisão numérica a um tempo de amostragem de 1 μ s. Com base no método dos subcircuitos, o comportamento dinâmico e o erro relativo de todas as variáveis de estado dos conversores de potência considerados são discutidos. Nesse sentido, os resultados mostram que a implementação de um procedimento de simulação em tempo real a partir de MCUs de alto desempenho é factível para um tempo de amostragem alvo de 1 μ s que, inclusive, garante uma precisão numérica satisfatória. A simulação em HIL dos conversores de potência considerados foi desenvolvida com o uso do MCU STM32H723ZGT6U da STMicroelectronics[®], que utiliza processador Arm[®] Cortex[®]-M7 de 32 *bits* em 550 MHz.

Palavras-chave: Eletrônica de Potência, Simulação em Tempo Real, *Hardware-in-the-Loop*, HIL.

Abstract

CASTRO, D. S. **Modeling of Power Converters for Hardware-in-the-Loop Implementation in High-performance MCUs.** 125 p. Master Thesis – São Carlos School of Engineering, University of São Paulo, 2021.

In this thesis, the modeling and implementation of power converters for Hardware-in-the-Loop (HIL) real-time simulation by using high performance microcontroller units (MCUs) are approached. In this context, two modeling techniques for switched electric circuits are studied. The first technique is based on the associate discrete circuit (ADC), while the other requires the state space formulation and yields both the subcircuits and the equivalent switch model methods. The test scenarios are the dc–dc boost converter, and four different dc–ac power converter configurations, the single-phase full-bridge versions of the VSI, in both standalone and grid-tied modes, the ZSI and the Quasi–ZSI converters. The simulations were performed by using Simulink[®] software from Mathworks[®], in which all the calculated dynamic models were effectively tested. Owing to a low sampling time requirement that is as low as 100 ns to keep good numerical precision in the results, both the ADC technique and the equivalent switch model method are not suitable for HIL implementation in high performance MCUs. The subcircuits method, on the other hand, provides good numerical precision given a sampling time of 1 μ s. For the subcircuits method, both the dynamic behavior and the relative error obtained for all the state variables of the considered power converters are presented. The results show that it is possible to implement a HIL simulation procedure for the considered power convertes by using high performance MCUs with a sampling time target of 1 μ s and achieve sufficient numerical precision. The HIL simulation of the considered power converters has been developed by using the 32-bit 550 MHz Arm[®] Cortex[®]-M7 STM32H723ZGT6U MCU from STMicroelectronics[®].

Keywords: Power Electronics, Real-Time Simulation, Hardware-in-the-Loop, HIL.

Lista de Figuras

Figura 2.1	Modelos ADC para indutores, capacitores e chaves.	33
Figura 2.2	Fluxograma de simulação de um circuito elétrico chaveado por meio da técnica ADC.	35
Figura 2.3	Circuito elétrico chaveado para exemplificação da redução de subcircuitos pela consideração de subintervalos de chaveamento factíveis.	37
Figura 2.4	Fluxograma de simulação de um circuito elétrico chaveado por meio do método dos subcircuitos.	38
Figura 2.5	Modelo equivalente dos semicondutores para a representação em espaço de estados.	39
Figura 2.6	Procedimento de introdução de indutores virtuais para facilitar o equacionamento de correntes em capacitores.	40
Figura 2.7	Fluxograma de simulação de um circuito elétrico chaveado por meio do método do modelo equivalente de chave.	42
Figura 2.8	Sincronismo entre o relógio de tempo real e o tempo de processamento numérico para a implementação em tempo real de modelos dinâmicos em HIL.	43
Figura 2.9	Procedimento de simulação computacional genérico para a validação de modelos dinâmicos de conversores.	46
Figura 3.1	Circuito elétrico do conversor <i>boost</i>	50
Figura 3.2	Circuito elétrico do modelo ADC do conversor <i>boost</i>	51
Figura 3.3	Subcircuitos referentes aos subintervalos de chaveamento factíveis do conversor <i>boost</i>	53
Figura 3.4	Circuito elétrico do conversor <i>boost</i> com a substituição das chaves para a aplicação da técnica do modelo equivalente de chave.	55
Figura 3.5	Estrutura básica do conversor VSI trifásico.	57
Figura 3.6	Configuração esquemática genérica para montagem de conversores CC-CA.	57

Figura 3.7	Interface CA composta por um filtro LC alimentando uma carga R no modo ilhado.	58
Figura 3.8	Estrutura do conversor VSI com filtro LC no modo ilhado.	58
Figura 3.9	Subcircuitos referentes aos subintervalos de chaveamento factíveis do conversor VSI na configuração monofásica em ponte completa.	59
Figura 3.10	Circuito elétrico do conversor VSI na configuração monofásica em ponte completa com a substituição das chaves para a aplicação da técnica do modelo equivalente de chave.	61
Figura 3.11	Interface CA composta por um filtro LC alimentando uma carga R no modo conectado.	63
Figura 3.12	Estrutura do conversor VSI com filtro LC no modo conectado.	65
Figura 3.13	Interface CC do conversor ZSI.	66
Figura 3.14	Estrutura do conversor ZSI no modo ilhado.	66
Figura 3.15	Subcircuito do VSI monofásico em ponte completa durante a operação em <i>shoot-through</i> , subintervalo de chaveamento $S_{st} = 1$	67
Figura 3.16	Subcircuitos referentes aos subintervalos de chaveamento factíveis da malha Z.	68
Figura 3.17	Circuito elétrico da malha Z com a substituição da chave para a aplicação da técnica do modelo equivalente de chave.	70
Figura 3.18	Interface CC do conversor Quasi-ZSI.	73
Figura 3.19	Estrutura do conversor Quasi-ZSI no modo ilhado.	73
Figura 3.20	Subcircuitos referentes aos subintervalos de chaveamento factíveis da malha Quasi-Z.	74
Figura 3.21	Circuito elétrico da malha Quasi-Z com a substituição da chave para a aplicação da técnica do modelo equivalente de chave.	76
Figura 4.1	Resposta de i_{L_1} obtida do modelo ADC do conversor <i>boost</i> para diferentes tempos de amostragem.	81
Figura 4.2	Resposta de v_{C_1} obtida do modelo do conversor <i>boost</i> pelo método do modelo equivalente de chave com integração numérica via BDF2 para diferentes tempos de amostragem.	81
Figura 4.3	Resposta de i_{L_1} obtida do modelo do ZSI monofásico em ponte completa pelo método do modelo equivalente de chave com integração numérica via BDF2 para diferentes tempos de amostragem.	82
Figura 4.4	Resposta de v_{C_2} obtida do modelo do Quasi-ZSI monofásico em ponte completa pelo método do modelo equivalente de chave com integração numérica via BDF2 para diferentes tempos de amostragem.	82
Figura 4.5	Respostas de i_{L_1} e v_{C_1} obtidas na simulação do conversor <i>boost</i>	85
Figura 4.6	Erros relativos entre as respostas i_{L_1} e v_{C_1} do modelo dinâmico do conversor <i>boost</i> e do circuito elétrico de referência.	85

Figura 4.7	Respostas de i_{L_a} e v_{C_a} obtidas na simulação do VSI monofásico em ponte completa no modo ilhado.	87
Figura 4.8	Erros relativos entre as respostas i_{L_a} e v_{C_a} do modelo dinâmico do VSI monofásico em ponte completa no modo ilhado e do circuito elétrico de referência.	87
Figura 4.9	Respostas de i_{L_a} , i_{L_A} e v_{C_a} obtidas na simulação do VSI monofásico em ponte completa no modo conectado.	89
Figura 4.10	Erros relativos entre as respostas i_{L_a} , i_{L_A} e v_{C_a} do modelo dinâmico do VSI monofásico em ponte completa no modo conectado e do circuito elétrico de referência.	90
Figura 4.11	Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} obtidas na simulação do ZSI monofásico em ponte completa.	92
Figura 4.12	Erros relativos entre as respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} do modelo dinâmico do ZSI monofásico em ponte completa e do circuito elétrico de referência.	93
Figura 4.13	Respostas de i_{L_a} e v_{C_a} obtidas na simulação do ZSI monofásico em ponte completa.	94
Figura 4.14	Erros relativos entre as respostas de i_{L_a} e v_{C_a} do modelo dinâmico do ZSI monofásico em ponte completa e do circuito elétrico de referência.	94
Figura 4.15	Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} obtidas na simulação do Quasi-ZSI monofásico em ponte completa.	96
Figura 4.16	Erros relativos entre as respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} do modelo dinâmico do Quasi-inversor tipo fonte de impedância (ZSI) monofásico em ponte completa e do circuito elétrico de referência.	97
Figura 4.17	Respostas de i_{L_a} e v_{C_a} obtidas na simulação do Quasi-ZSI monofásico em ponte completa.	98
Figura 4.18	Erros relativos entre as respostas de i_{L_a} e v_{C_a} do modelo dinâmico do Quasi-ZSI monofásico em ponte completa e do circuito elétrico de referência.	98
Figura 5.1	Diagrama para implementação experimental do procedimento de simulação em HIL.	104
Figura 5.2	<i>Setup</i> de testes para a simulação em HIL do conversor <i>boost</i>	105
Figura 5.3	Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor <i>boost</i> em comparação com o tempo de interrupção.	106
Figura 5.4	Respostas de i_{L_1} e v_{C_1} obtidas na simulação em HIL do conversor <i>boost</i>	106
Figura 5.5	<i>Setup</i> de testes para a simulação em HIL do conversor VSI no modo ilhado.	107

Figura 5.6	Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor VSI no modo ilhado em comparação com o tempo de interrupção.	107
Figura 5.7	Respostas de i_{L_a} e v_{C_a} obtidas na simulação em HIL do VSI monofásico em ponte completa no modo ilhado.	108
Figura 5.8	<i>Setup</i> de testes para a simulação em HIL do conversor VSI no modo conectado.	108
Figura 5.9	Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor VSI no modo conectado em comparação com o tempo de interrupção.	109
Figura 5.10	Respostas de i_{L_a} , i_{L_A} e v_{C_a} obtidas na simulação em HIL do VSI monofásico em ponte completa no modo conectado.	110
Figura 5.11	<i>Setup</i> de testes para a simulação em HIL do lado CC do conversor ZSI.	111
Figura 5.12	<i>Setup</i> de testes para a simulação em HIL do lado CA do conversor ZSI.	111
Figura 5.13	Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor ZSI em comparação com o tempo de interrupção.	112
Figura 5.14	Respostas de i_{L_a} e v_{C_a} obtidas na simulação em HIL do ZSI monofásico em ponte completa.	112
Figura 5.15	Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} obtidas na simulação em HIL do ZSI monofásico em ponte completa.	113
Figura 5.16	<i>Setup</i> de testes para a simulação em HIL do lado CC do conversor Quasi-ZSI.	114
Figura 5.17	<i>Setup</i> de testes para a simulação em HIL do lado CA do conversor Quasi-ZSI.	114
Figura 5.18	Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor Quasi-ZSI em comparação com o tempo de interrupção.	115
Figura 5.19	Respostas de i_{L_a} e v_{C_a} obtidas na simulação em HIL do Quasi-ZSI monofásico em ponte completa.	115
Figura 5.20	Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} obtidas na simulação em HIL do Quasi-ZSI monofásico em ponte completa.	116

Lista de Tabelas

Tabela 2.1	Lógicas de comutação para diferentes tipos de chaves na técnica ADC.	34
Tabela 2.2	Lógicas de comutação para diferentes tipos de chaves na técnica de modelagem em espaço de estados pelo método por modelo equivalente de chave.	41
Tabela 2.3	Matrizes para a implementação dos métodos de integração numérica BDF1, BDF2 e TR.	45
Tabela 4.1	Parâmetros utilizados na simulação do conversor <i>boost</i>	84
Tabela 4.2	Parâmetros utilizados na simulação do VSI monofásico em ponte completa no modo ilhado.	86
Tabela 4.3	Parâmetros utilizados na simulação do VSI monofásico em ponte completa no modo conectado.	88
Tabela 4.4	Parâmetros utilizados na simulação do ZSI monofásico em ponte completa.	91
Tabela 4.5	Parâmetros utilizados na simulação do Quasi-ZSI monofásico em ponte completa.	95

Lista de Siglas

ADC discretização de circuito elétrico associado

ASP processador específico da aplicação

BDF1 *backward* Euler de primeira ordem

BDF2 *backward* Euler de segunda ordem

CA corrente alternada

CC corrente contínua

CMSIS *Common Microcontroller Software Interface Standard*

DSP processador digital de sinais

DI *digital input*

DO *digital output*

DAC *digital-to-analog converter*

FE *forward* Euler

FPGA *Field-programmable Gate Array*

FPU *floating-point unit*

HIL *Hardware-in-the-Loop*

HVDC alta tensão em corrente contínua

IGBT *Insulated-gate Bipolar Transistor*

IRENA Agência Internacional de Energia Renovável

MCU unidade microcontrolada

PWM modulação por largura de pulso

TR trapezoidal

UPS fonte de energia ininterrupta

VSI inversor tipo fonte de tensão

ZSI inversor tipo fonte de impedância

Lista de Símbolos

\mathbb{N} indica o conjunto dos números naturais

\mathbb{Z} indica o conjunto dos números inteiros

\mathbb{R} indica o conjunto dos números reais

x notação para grandezas variantes no tempo

\mathbf{x} notação para vetores coluna

\mathbf{X} notação para matrizes

h tempo de amostragem

f_h frequência de amostragem ($f_h = 1/h$)

f_s frequência de chaveamento

$R_{\{\bullet\}}$ notação para resistência

$G_{\{\bullet\}}$ notação para condutância ($G_{\{\bullet\}} = 1/R_{\{\bullet\}}$)

$r_{\{\bullet\}}$ notação para perda resistiva

$L_{\{\bullet\}}$ notação para indutância

$C_{\{\bullet\}}$ notação para capacitância

$S_{\{\bullet\}}$ notação para variável booleana

$\bar{S}_{\{\bullet\}}$ indica a operação de negação da variável booleana $S_{\{\bullet\}}$

$\{\bullet\}^{k+M}$ indica a amostra $k + M$, com $k \in \mathbb{N}$ e $M \in \mathbb{Z}$

$\{\bullet\}^T$ indica a operação de transposição de vetores e matrizes

Sumário

1	Introdução	25
1.1	Motivação	25
1.2	Objetivos	27
1.3	Contribuições do Trabalho	28
1.4	Estrutura do Texto	28
2	Implementação em Tempo Real de Circuitos Elétricos Chaveados	31
2.1	Introdução	31
2.2	Técnica de Modelagem via Discretização de Circuito Elétrico Associado	32
2.3	Técnicas de Modelagem via Espaço de Estado	36
2.3.1	Método dos Subcircuitos	36
2.3.2	Método do Modelo Equivalente de Chave	39
2.4	Métodos Numéricos para Simulações em Tempo Real	43
2.5	Validação de Modelos Dinâmicos e Análise de Resultados	45
2.6	Considerações Finais	47
3	Obtenção de Modelos Dinâmicos	49
3.1	Introdução	49
3.2	Conversor CC–CC <i>Boost</i>	50
3.2.1	Modelagem pela Técnica ADC	50
3.2.2	Modelagem pelo Método dos Subcircuitos	53
3.2.3	Modelagem pelo Método do Modelo Equivalente de Chave	54
3.3	Conversores CC–CA	56
3.3.1	Conversor VSI com filtro <i>LC</i> no Modo Ilhado	58
3.3.2	Conversor VSI com filtro <i>LC</i> no Modo Conectado	62
3.3.3	Conversor ZSI	65
3.3.4	Conversor Quasi–ZSI	72
3.4	Considerações Finais	78

4	Resultados Simulados	79
4.1	Introdução	79
4.2	Considerações Preliminares	80
4.2.1	Seleção do <i>Hardware</i> de Processamento	80
4.2.2	Seleção da Técnica de Modelagem para a Implementação em Tempo Real	80
4.2.3	Seleção do Método Numérico	83
4.3	Validação de Modelos Dinâmicos	83
4.3.1	Conversor <i>Boost</i>	84
4.3.2	Conversor VSI no Modo Ilhado	86
4.3.3	Conversor VSI no Modo Conectado	88
4.3.4	Conversor ZSI	90
4.3.5	Conversor Quasi-ZSI	95
4.4	Considerações Finais	99
5	Simulação em <i>Hardware-in-the-Loop</i>	101
5.1	Introdução	101
5.2	Compilação dos Modelos Dinâmicos	102
5.3	Recursos Arm [®] CMSIS-DSP	103
5.4	Simulação em Tempo Real	104
5.4.1	Conversor <i>Boost</i>	105
5.4.2	Conversor VSI no Modo Ilhado	107
5.4.3	Conversor VSI no Modo Conectado	108
5.4.4	Conversor ZSI	111
5.4.5	Conversor Quasi-ZSI	114
5.5	Considerações Finais	117
6	Considerações Finais	119
6.1	Conclusões	119
6.2	Publicações	121
6.2.1	Trabalhos Completos Publicados em Anais de Eventos	121
6.2.2	Trabalhos Completos Publicados em Periódicos	121
	Referências Bibliográficas	123

Introdução

1.1 Motivação

No contexto econômico atual, a constante utilização de plantas de geração baseadas em recursos fósseis e nucleares é uma questão de grande relevância em todas as esferas da sociedade, principalmente pelo fato de tais recursos serem não sustentáveis e promoverem a degradação do meio ambiente. Nesse sentido, a demanda global por soluções de geração baseadas em fontes de energia que são limpas e sustentáveis, ou renováveis, tais como o sol, o vento e a biomassa, vem crescendo rapidamente ao longo dos anos (Qazi et al., 2019; Fuzato et al., 2016).

Segundo mostram os relatórios anuais da Agência Internacional de Energia Renovável (IRENA), por exemplo, no final de 2018 a capacidade de potência instalada de painéis fotovoltaicos no mundo alcançou cerca de 480 MW, o que representou um crescimento de 24,4% em relação ao ano de 2017. Na geração eólica, em contrapartida, os resultados para 2018 apontaram 565 MW de potência instalada, um crescimento por volta de 9,54% em relação ao total observado no ano de 2017. No que diz respeito à biomassa, o total foi de 116 MW de potência instalada em 2018, superando em 5,22% o valor de 2017. (WHITEMAN et al., 2019).

Como pode ser visto, as estatísticas revelam que a penetração das fontes renováveis no mundo encontra-se em rápida expansão. Por esse motivo, tanto na academia quanto nos grandes centros de pesquisa industriais há uma forte busca pelo aprimoramento de tecnologias voltadas a melhorar a qualidade operacional de plantas renováveis (Forouzesh et al., 2017). Uma dessas tecnologias, por exemplo, são os dispositivos eletrônicos chaveados, ou conversores de potência, que são estruturas eletrônicas responsáveis por realizar todo o processamento da energia proveniente das fontes renováveis, condicionando-a tanto para corrente contínua (CC) quanto para corrente alternada (CA) (Erickson; Maksimović, 2001).

As fontes renováveis possuem uma característica intrínseca de fornecer energia elétrica em baixa tensão, além de serem intermitentes. Por isso, normalmente são utilizados conversores do tipo CC–CC para um pré-estágio de processamento de energia, cujo

objetivo é manter a tensão dessas fontes em níveis mais elevados, permitindo o acoplamento em barramentos CC (Li; Liu, 2019). Além disso, os conversores CC–CC são utilizados como interfaces que viabilizam a associação de bancos de baterias para suprir demandas de energia durante intervalos de intermitência das fontes renováveis. Em termos de distribuição de energia elétrica, é possível então utilizar conversores CC–CC para elevar a tensão do barramento CC a níveis adequados de operação em alta tensão em corrente contínua (HVDC), ou utilizar conversores do tipo CC–CA com filtros passivos para o acoplamento com a rede CA básica de distribuição (Teodorescu; Liserre; Rodríguez, 2011).

A partir disso, é fácil notar que a introdução de conversores no processamento de energia de fontes renováveis envolve a aplicação de estruturas de gerenciamento e malhas de controle específicas e muito bem projetadas (Li et al., 2018; Cornea et al., 2017). Caso contrário, o sistema de geração pode ser comprometido e entrar em colapso, podendo ocasionar acidentes, destruição de componentes e interrupção de fornecimento. Por esse motivo, todo projeto de acionamento dos conversores precisa ser devidamente testado e validado antes da implementação real.

Uma das formas mais convencionais de testar o comportamento dos conversores é por meio de *softwares* de simulação, tais como o PSIM[®], da Powersim Technologies Inc., e o Simulink[®], da MathWorks[®]. A partir de diagramas esquemáticos, esses simuladores são capazes de reproduzir o comportamento dinâmico de sistemas complexos com muita versatilidade. Com isso, eles permitem a execução de testes em malha aberta e em malha fechada com a simples introdução de blocos adicionais no diagrama esquemático, o que os tornam mecanismos atrativos para o teste de controladores projetados para aplicações em conversores de potência.

O problema dos *softwares* de simulação ocorre quando o diagrama esquemático se torna excessivamente complexo, tornando o cômputo das respostas temporais de uma determinada planta, no intervalo de alguns poucos milissegundos, uma tarefa computacional que é realizada em vários minutos. Além do mais, eles não permitem testar a verdadeira influência do *hardware* de controle na resposta da planta analisada, pois não desenvolvem respostas em tempo real e, por isso, são incapazes de incorporar interfaces de integração com o mundo real (Castro et al., 2019; Gregoire; Al-Haddad; Nanjundaiah, 2011).

Foi pensando nisso que empresas como a OPAL-RT e a Typhoon HIL Inc. desenvolvem plataformas de simulação em tempo real construídas com *Field-programmable Gate Arrays* (FPGAs) conhecidas como *Hardware-in-the-Loop* (HIL), que implementam com sucesso a integração entre o ambiente virtual de simulação e o mundo real. Sendo assim, o produto comercializado por tais empresas são módulos com interfaces de programação capazes de implementar modelos dinâmicos e reproduzir as suas respostas em tempo real. Toda a parte de acionamento pode ser sintetizada fora do ambiente virtual criado pelo HIL, e implementada por meio de *hardwares* de controle, tornando o ambiente de simulação mais próximo da realidade e atrativo para execução de testes sofisticados envolvendo

dispositivos de potência (Castro et al., 2019).

Apesar da facilidade proporcionada pelos módulos HIL dessas empresas, o custo de aquisição envolvido é elevado para muitos grupos de pesquisa. Por esse motivo, foi visto na literatura um esforço considerável de alguns pesquisadores em apresentar tanto a implementação de HILs independentes como também novas técnicas que ajudam a melhorar o desempenho do processamento em tempo real dos modelos dinâmicos de dispositivos eletrônicos.

No trabalho desenvolvido por Wang et al. (2019), por exemplo, os autores utilizam um FPGA para a simulação em tempo real de um conversor CC-CA trifásico. Para que isso seja feito, eles também apresentam um método generalizado de modelagem por discretização de circuito elétrico associado que melhora a estabilidade numérica da planta simulada. Já Liu et al. (2018) mostram uma forma de maximizar o paralelismo de modelos de conversores CC-CA trifásicos em implementações com FPGA, de modo a atingir maior desempenho de processamento em tempo real. Por fim, é mencionado o trabalho desenvolvido por Huang e Dinavahi (2019), que mostra a implementação em tempo real de um conversor CC-CA multinível em FPGA a partir de uma técnica de modelagem que visa melhorar a qualidade da resposta dinâmica durante o cômputo da solução de modelos dinâmicos de conversores.

Conforme é observado, grande parte dos trabalhos são focados em implementações para FPGAs. Nesse sentido, este trabalho se apresenta como uma forma de sistematizar as principais técnicas de modelagem utilizadas na implementação de conversores de potência em tempo real, porém introduzindo a utilização de unidades microcontroladas (MCUs) de alto desempenho no desenvolvimento de um HIL não comercial para testes e validação de respostas dinâmicas em tempo real, com o intuito de tornar esse método de simulação ainda mais acessível na academia.

1.2 **Objetivos**

A execução deste trabalho tem como objetivo geral o estudo de técnicas de modelagem e de implementação de conversores de potência em HIL para aplicações de tempo real, visando a utilização de MCUs de alto desempenho. Os objetivos específicos podem ser resumidos de acordo com os seguintes pontos:

- Estudar as principais técnicas de modelagem de conversores de potência utilizadas no contexto de implementação em tempo real;
- Pesquisar por métodos numéricos viáveis para o processamento em tempo real dos modelos dinâmicos;
- Aplicar as técnicas de modelagem estudadas em topologias de conversores de potência conhecidas na literatura;

- Validar os modelos dinâmicos por meio de simulações em *software* específico;
- Avaliar a precisão dos métodos numéricos empregados na solução dos modelos dinâmicos considerados;
- Pesquisar por formas de processar modelos dinâmicos em tempo real com a utilização de MCUs de alto desempenho.

1.3 Contribuições do Trabalho

Como contribuições, este trabalho visa:

- Compilar os principais métodos empregados na modelagem de conversores de potência para a realização de procedimentos de simulação computacional no domínio do tempo;
- Apresentar de forma sistematizada uma metodologia de implementação de conversores de potência que pode ser utilizada no desenvolvimento de um HIL não comercial;
- Desenvolver um procedimento de simulação em tempo real de conversores de potência por meio de MCUs de alto desempenho;
- Implementar modelos dinâmicos de conversores de potência em HIL por meio de MCUs de alto desempenho e avaliar os resultados.

1.4 Estrutura do Texto

A estrutura pela qual o texto está organizado neste trabalho é apresentada a seguir.

No Capítulo 1, a contextualização geral do tema, os objetivos definidos para a execução deste trabalho e as contribuições do mesmo são apresentadas. Além disso, é mostrada a estrutura na qual este texto é organizado.

No Capítulo 2, serão mostradas as principais técnicas de modelagem de circuitos elétricos chaveados encontradas na literatura. A partir disso, são discutidas as estratégias para a elaboração de métodos numéricos para o processamento em tempo real dos modelos dinâmicos, bem como o método a ser utilizado para a validação e as análises dos resultados.

No Capítulo 3, são apresentados os conversores de potência a serem utilizados como cenários de teste para a implementação do HIL. Em seguida, é realizado o levantamento dos modelos dinâmicos associados a cada topologia considerada.

No Capítulo 4, os resultados simulados são mostrados. Para isso, os métodos numéricos discutidos no Capítulo 2 são utilizados para resolver os modelos dinâmicos calculados no Capítulo 3.

No Capítulo 5, o procedimento de simulação em HIL é implementado, e os resultados experimentais obtidos para os conversores de potência discutidos no Capítulo 3 são mostrados.

Por fim, as considerações finais são apresentadas no Capítulo 6. Adicionalmente, os artigos em que o autor deteve participação são listados.

Implementação em Tempo Real de Circuitos Elétricos Chaveados

2.1 Introdução

A busca por representações matemáticas de circuitos elétricos chaveados para fins de simulação assistida por computador tornou-se um tema de pesquisa em meados da década de 80 (Pong; Jackson, 1985). O grande problema enfrentado na época estava concentrado em encontrar modelos adequados que fossem capazes de descrever com precisão o comportamento dinâmico chaveado dos diferentes tipos de semicondutores, ou chaves, disponíveis até então.

As primeiras contribuições relevantes ficaram registradas nos trabalhos publicados por Hui e Christopoulos (1990), Pejović e Maksimović (1994) e Hui e Morrall (1994), em que foi descrita uma forma de representar qualquer combinação de chave como sendo apenas um indutor de baixa indutância no modo de condução e um capacitor de baixa capacitância no modo de bloqueio. Com isso, a partir de um método difundido na época, que permitia discretizar elementos armazenadores de energia para simulações de linhas de transmissão, foi possível simular, computacionalmente, circuitos elétricos com chaves no domínio do tempo, por meio de uma formulação nodal que ficou conhecida como discretização de circuito elétrico associado (ADC).

Em termos de aplicações de tempo real em HIL, diversos trabalhos mostraram uma implementação bem sucedida do método ADC com o uso de FPGAs, conforme é descrito, por exemplo, nos trabalhos apresentados por Matar e Iravani (2010), Ould-Bachir et al. (2012) e Herrera et al. (2015).

Uma outra técnica interessante para modelar circuitos elétricos chaveados é por meio da representação em espaço de estado, apesar de quase não ser encontrada em implementações de tempo real. Contudo, nessa técnica dois métodos podem ser seguidos.

O primeiro método se resume em obter equações dinâmicas associadas aos elementos armazenadores de energia do circuito. Não se faz a modelagem da chave, entretanto,

um conjunto de equações dinâmicas precisa ser extraído individualmente para cada possível subcircuito relacionado à combinação das chaves e, por isso, é necessário ter um conhecimento prévio acerca da operação do circuito elétrico a ser modelado (Erickson; Maksimović, 2001).

O segundo método foi introduzido por Blanchette, Ould-Bachir e David (2012) já no contexto de simulação em tempo real, sendo a ideia básica substituir as chaves por modelos equivalentes dados por uma resistência variável em paralelo com um capacitor. Diferentemente do método dos subcircuitos, as equações dinâmicas associadas são extraídas de um único circuito elétrico, e o efeito de comutação das chaves é incorporado nas resistências variáveis, que assumem baixa impedância no modo de condução e alta impedância no modo de bloqueio. Assim, não é necessário ter um conhecimento prévio da operação do circuito elétrico a ser modelado.

A seguir, as técnicas de modelagem ADC e por espaço de estado são mostradas com detalhes. Na sequência, os possíveis métodos numéricos para implementações em tempo real são discutidos, e a técnica de validação de modelos e análise de resultados é apresentada.

2.2 Técnica de Modelagem via Discretização de Circuito Elétrico Associado

Para realizar a modelagem por meio da técnica ADC, os elementos armazenadores de energia e as chaves precisam ser substituídos por modelos equivalentes discretizados. O processo de discretização é apresentado a seguir.

Considerando um indutor L e um capacitor C lineares e constantes, as expressões da tensão sobre o indutor, v_L , e a corrente no capacitor, i_C , são

$$v_L = L \frac{d}{dt} i_L \quad \text{e} \quad i_C = C \frac{d}{dt} v_C,$$

respectivamente.

Dessa forma, é possível definir

$$f(t, i_L) := \frac{d}{dt} i_L = \frac{1}{L} v_L \quad (2.1)$$

e

$$f(t, v_C) := \frac{d}{dt} v_C = \frac{1}{C} i_C. \quad (2.2)$$

Assumindo um tempo de amostragem $h = t^{k+1} - t^k$, em que $k + 1$ indica a amostra atual, é possível aplicar o método implícito de integração *backward* Euler de primeira ordem (BDF1), a saber

$$x^{k+1} \approx x^k + h f(t^{k+1}, x^{k+1}),$$

nas expressões (2.1) e (2.2), o que resulta em

$$\begin{aligned} i_L^{k+1} &\approx i_L^k + \frac{h}{L} v_L^{k+1} \\ &= -j_L^{k+1} + G_L v_L^{k+1} \end{aligned} \quad (2.3)$$

e

$$\begin{aligned} i_C^{k+1} &\approx -\frac{C}{h} v_C^k + \frac{C}{h} v_C^{k+1} \\ &= -j_C^{k+1} + G_C v_C^{k+1}, \end{aligned} \quad (2.4)$$

que são os modelos discretos do indutor L e do capacitor C , respectivamente, obtidos via BDF1. Note que o mesmo procedimento poderia ser realizado para outros métodos de integração, tais como trapezoidal (TR) e *backward* Euler de segunda ordem (BDF2).

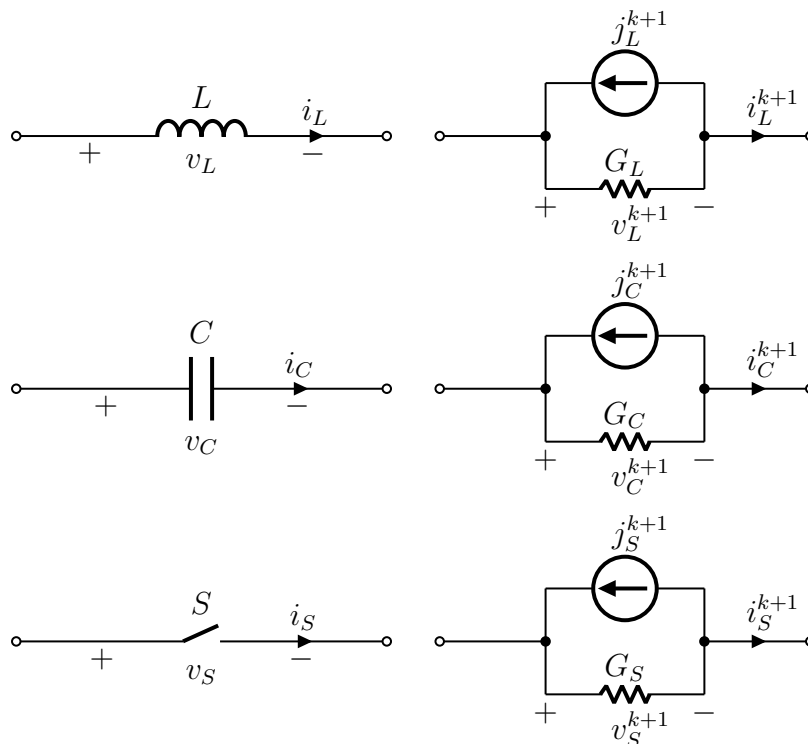
No método ADC, o semicondutor é considerado como um pequeno indutor L_S no modo de condução ($S = 1$), e um pequeno capacitor C_S no modo de bloqueio ($S = 0$). Nesse sentido, por (2.3) e (2.4),

$$\begin{aligned} i_S^{k+1} &\approx \begin{cases} i_S^k + \frac{h}{L_S} v_S^{k+1}, & \text{se } S^{k+1} = 1 \\ -\frac{C_S}{h} v_S^k + \frac{C_S}{h} v_S^{k+1}, & \text{se } S^{k+1} = 0 \end{cases} \\ &= -j_S^{k+1} + G_S v_S^{k+1}, \end{aligned} \quad (2.5)$$

em que $G_S = C_S/h = h/L_S$ é constante e o valor de j_S^{k+1} indica o estado da chave.

Observando (2.3), (2.4) e (2.5), nota-se que os modelos ADC podem ser representados conforme mostrado na Figura 2.1.

Figura 2.1 – Modelos ADC para indutores, capacitores e chaves.



Fonte: Próprio autor.

Ao realizar a substituição dos elementos armazenadores de energia e das chaves pelos seus respectivos modelos ADC, o circuito elétrico resultante pode ser completamente equacionado via análise nodal, resultando em um sistema de equações do tipo

$$\mathbf{H} \mathbf{x}^{k+1} \approx \mathbf{b}^{k+1}, \quad (2.6)$$

em que $\mathbf{H} \in \mathbb{R}^{M \times M}$ é uma matriz constante e $\mathbf{x} \in \mathbb{R}^{M \times 1}$ é o vetor das incógnitas (tensões nos nós e correntes nos ramos), com $M \in \mathbb{N}$ sendo a soma dos números de nós e ramos do circuito (Najm, 2010).

O vetor $\mathbf{b} \in \mathbb{R}^{M \times 1}$ é formado pelo conjunto de entradas do circuito, de tal forma que

$$\mathbf{b}^{k+1} = \begin{bmatrix} \mathbf{j}_S^{k+1} \\ \mathbf{u}^{k+1} \end{bmatrix},$$

em que $\mathbf{j}_S \in \mathbb{R}^{P \times 1}$ é um vetor que inclui as fontes de corrente referentes ao modelo ADC das chaves, e $\mathbf{u} \in \mathbb{R}^{Q \times 1}$ é um vetor que inclui tanto as fontes de corrente referentes ao modelo ADC dos elementos armazenadores de energia quanto as fontes de tensão e de corrente referentes às verdadeiras entradas do circuito, com $\{P, Q\} \in \mathbb{N} / (P + Q) = M$.

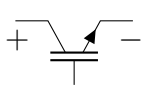
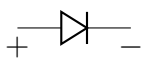
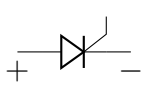
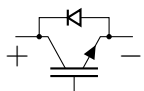
Vale ressaltar que, como \mathbf{H} é uma matriz constante, a solução do sistema, dada por

$$\mathbf{x}^{k+1} \approx \mathbf{H}^{-1} \mathbf{b}^{k+1}, \quad (2.7)$$

tem baixo esforço computacional para simulações no domínio do tempo, visto que \mathbf{H}^{-1} pode ser pré-calculada de tal forma a evitar operações de inversão a cada passo da simulação.

As chaves de um circuito elétrico podem ser controladas, por meio de um sinal lógico de comando d , ou não controladas, com comutação natural. Dessa forma, cada tipo diferente de semicondutor possui uma lógica de comutação característica.

Tabela 2.1 – Lógicas de comutação para diferentes tipos de chaves na técnica ADC.

Chave	S^{k+1}
	d^{k+1}
	$S^k (i_S^k \geq 0) + \bar{S}^k (v_S^k \geq 0)$
	$d^{k+1} (v_S^k \geq 0) + S^k (i_S^k > 0)$
	$d^{k+1} + S^k (i_S^k \leq 0) + \bar{S}^k (v_S^k < 0)$

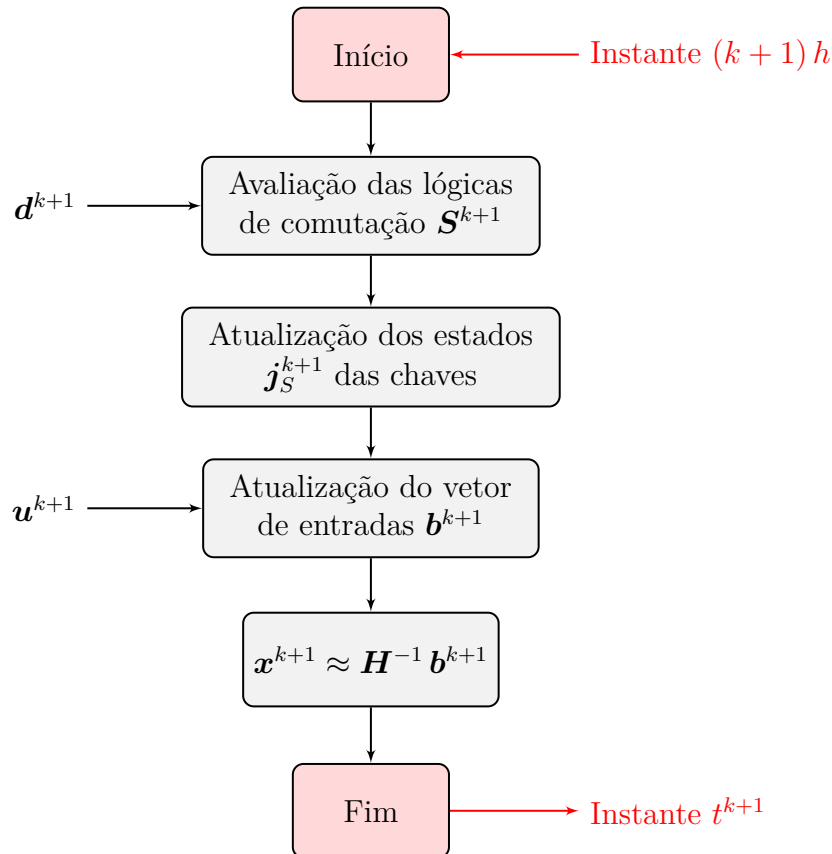
Fonte: Adaptado de Pejović e Maksimović (1994).

Os semicondutores mais comuns na literatura são os diodos, os tiristores e os transístores, por exemplo os *Insulated-gate Bipolar Transistors* (IGBTs), com diodo antiparalelo. Por esse motivo, a Tabela 2.1 mostra as lógicas de comutação desses tipos de chaves que precisam ser consideradas para a simulação no domínio do tempo pela técnica ADC.

O procedimento que deve ser seguido para realizar a simulação no domínio do tempo de um dado circuito elétrico chaveado por meio da técnica ADC é resumido na Figura 2.2.

No instante $(k + 1)h$, o processamento é iniciado a partir da leitura dos sinais lógicos de comando no vetor \mathbf{d}^{k+1} , que permite a avaliação das lógicas de comutação \mathbf{S}^{k+1} de todas as chaves do circuito. Por meio dos resultados em \mathbf{S}^{k+1} , o vetor \mathbf{j}_S^{k+1} é atualizado com os novos estados das chaves. Na sequência, o vetor \mathbf{b}^{k+1} é atualizado a partir de \mathbf{j}_S^{k+1} e de \mathbf{u}^{k+1} , e calcula-se \mathbf{x}^{k+1} por meio da matriz inversa pre-calculada \mathbf{H}^{-1} , encerrando o passo de simulação no instante t^{k+1} .

Figura 2.2 – Fluxograma de simulação de um circuito elétrico chaveado por meio da técnica ADC.



Fonte: Próprio autor.

Apesar do baixo esforço computacional, muito tem sido discutido na literatura acerca da eficiência da técnica de modelagem ADC em termos de simulação em tempo real, principalmente pelo fato dela incorporar oscilações numéricas na solução, criando perdas de potência virtuais devido à interpretação indutor-capacitor das chaves (Song et al., 2014). Nesse sentido, alguns trabalhos, tais como o desenvolvido por Wang et al. (2019), apareceram discutindo acerca da melhor forma de sintonizar os parâmetros C_S e L_S ,

portanto G_S , dado um tempo de amostragem h , visando a minimização de oscilações numéricas na solução.

Vale ressaltar que o método ADC ainda é atrativo para a grande maioria das implementações de conversores de potência em HIL, visto que o mesmo é constantemente discutido e amplamente considerado em trabalhos publicados recentemente em importantes veículos de publicação da área, conforme é visto em Herrera et al. (2015), Ould-Bachir, Blanchette e Al-Haddad (2015), Hadizadeh et al. (2019) e Huang e Dinavahi (2019).

2.3 Técnicas de Modelagem via Espaço de Estado

Na técnica de modelagem por espaço de estado, as equações de estado, obtidas a partir da aplicação das leis de Kirchhoff nos elementos armazenadores de energia do circuito elétrico, precisam ser representadas conforme

$$\mathbf{K}\dot{\mathbf{x}} = \mathbf{A}_K \mathbf{x} + \mathbf{B}_K \mathbf{u}, \text{ com } \dot{\mathbf{x}} := \frac{d}{dt} \mathbf{x},$$

ou ainda

$$\begin{aligned} \dot{\mathbf{x}} &= (\mathbf{K}^{-1} \mathbf{A}_K) \mathbf{x} + (\mathbf{K}^{-1} \mathbf{B}_K) \mathbf{u} \\ &= \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u}, \end{aligned} \tag{2.8}$$

em que $\mathbf{x} \in \mathbb{R}^{M \times 1}$ é o vetor de estados (correntes nos indutores e tensões nos capacitores), $\mathbf{A} = \mathbf{K}^{-1} \mathbf{A}_K \in \mathbb{R}^{M \times M}$ é a matriz da planta a ser simulada, $\mathbf{B} = \mathbf{K}^{-1} \mathbf{B}_K \in \mathbb{R}^{M \times N}$ é a matriz de entrada e $\mathbf{u} \in \mathbb{R}^{N \times 1}$ é o vetor de entradas (fontes de tensão e corrente), com $\{M, N\} \in \mathbb{N}$ sendo os números de estados e de entradas, respectivamente (Erickson; Maksimović, 2001).

A partir disso, as duas formas conhecidas de modelar um circuito elétrico chaveado por meio da representação (2.8) são discutidas a seguir.

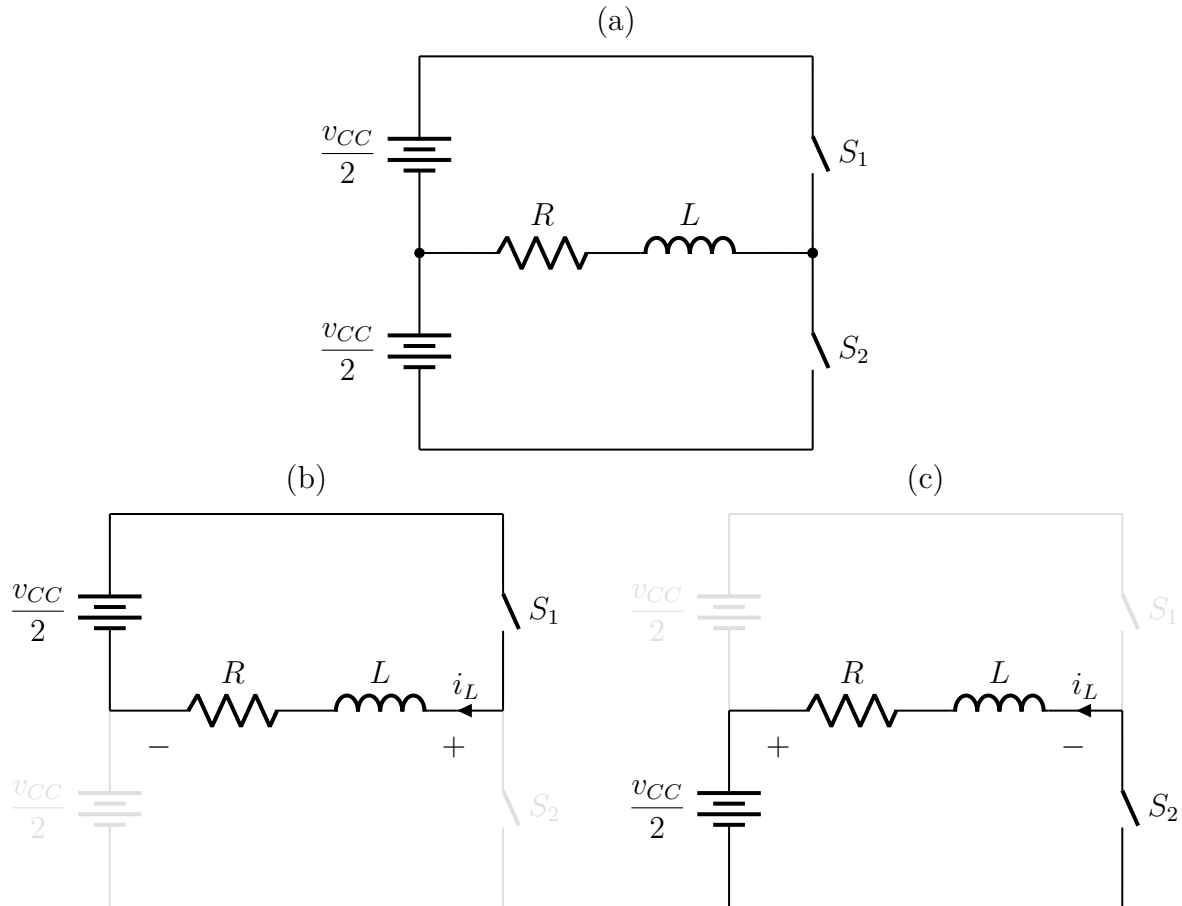
2.3.1 Método dos Subcircuitos

Nessa metodologia, a quantidade observável de possíveis subcircuitos está relacionada com o número de semicondutores utilizados no circuito elétrico a ser modelado. Como toda chave possui dois possíveis estados, $S = 1$ ou $S = 0$, uma quantidade $P \in \mathbb{N}$ de chaves resultará em 2^P subcircuitos.

Ao ser considerado um exemplo no qual $P = 3$ chaves são utilizadas, é possível inferir que será necessário obter um conjunto de equações no formato (2.8) para cada um dos $2^3 = 8$ subcircuitos, o que é uma quantidade considerável e resultará em um exaustivo esforço de modelagem. Por esse motivo, é fundamental conhecer o princípio básico de operação da planta, visto que, em alguns casos, a maioria dos subintervalos de chaveamento não são factíveis, o que minimiza a quantidade de subcircuitos que precisarão ser modelados para uma completa descrição dinâmica da planta.

A possibilidade de redução de subcircuitos pela consideração de subintervalos de chaveamento factíveis pode ser compreendida por meio do circuito elétrico chaveado básico da Figura 2.3(a), que apresenta $P = 2$ chaves, S_1 e S_2 , com $2^2 = 4$ possíveis subcircuitos.

Figura 2.3 – Circuito elétrico chaveado para exemplificação da redução de subcircuitos pela consideração de subintervalos de chaveamento factíveis. (a) Representação do circuito elétrico com as chaves S_1 e S_2 , (b) subcircuito referente ao subintervalo de chaveamento $S = 1$ e (c) subcircuito referente ao subintervalo de chaveamento $S = 0$.



Fonte: Próprio autor.

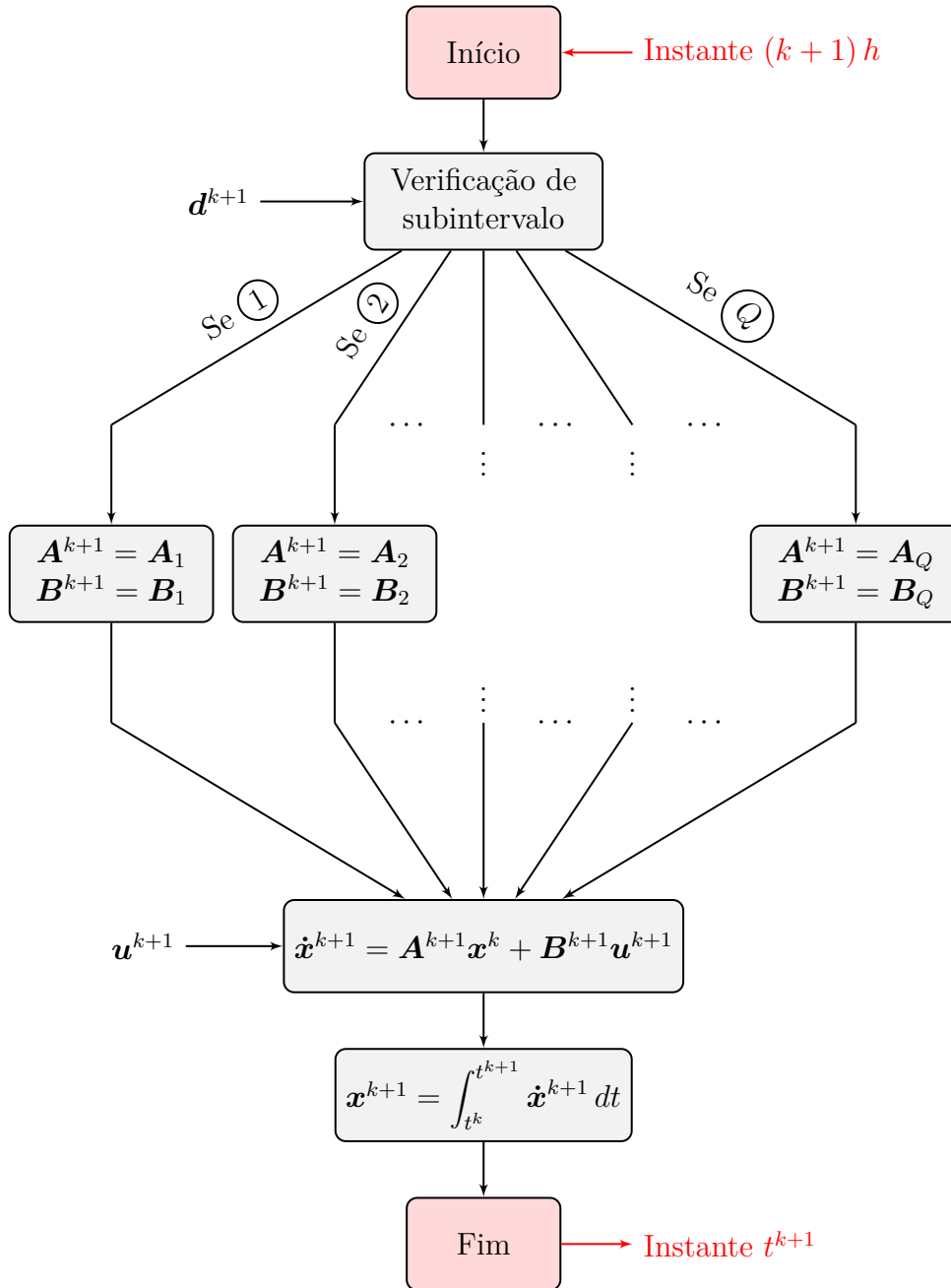
Analisando o circuito elétrico chaveado da Figura 2.3(a), é fácil notar que, em termos práticos, se $S_1 = S_2 = 1$, ambas as fontes de tensão $v_{CC}/2$ entram em curto-circuito. Em contrapartida, se $S_1 = S_2 = 0$, a carga RL fica flutuando. Sendo assim, só é válido $S_1 = \bar{S}_2 = S$, de modo que apenas os subcircuitos referentes aos subintervalos definidos por $S = 1$, Figura 2.3(b), e $S = 0$, Figura 2.3(c), são factíveis.

Em termos de simulação no domínio do tempo, a Figura 2.4 resume o procedimento que deve ser seguido para implementar um dado circuito elétrico chaveado com $Q \in \mathbb{N}$ subintervalos factíveis.

No instante $(k + 1)h$, o processamento é inicializado a partir da leitura dos sinais lógicos de comando \mathbf{d}^{k+1} . Em seguida, é verificado qual subintervalo de chaveamento deverá ser considerado na formação das matrizes \mathbf{A}^{k+1} e \mathbf{B}^{k+1} . Por fim, $\dot{\mathbf{x}}^{k+1}$ é obtido a partir de \mathbf{x}^k e \mathbf{u}^{k+1} , e, somente então, é feita a integração no domínio do tempo para a

obtenção de \mathbf{x}^{k+1} , o que encerra o passo de simulação no instante t^{k+1} .

Figura 2.4 – Fluxograma de simulação de um circuito elétrico chaveado por meio do método dos subcircuitos.



Fonte: Próprio autor.

Em geral, é possível dizer que utilizar o método dos subcircuitos para modelar conversores de potência para HIL é uma estratégia de implementação simples e requer baixo custo computacional. Entretanto, conversores com um grande número de elementos armazenadores de energia e de chaves tendem a aumentar de forma significativa a complexidade do equacionamento matemático pelas leis de Kirchhoff e o número de subcircuitos associados.

Vale ressaltar que a desconsideração de um modelo equivalente de chave introduz uma característica negativa no método, relacionada ao modo de condução de semicondutores não controlados. Nesse sentido, é desejável que diodos não entrem em condução de maneira

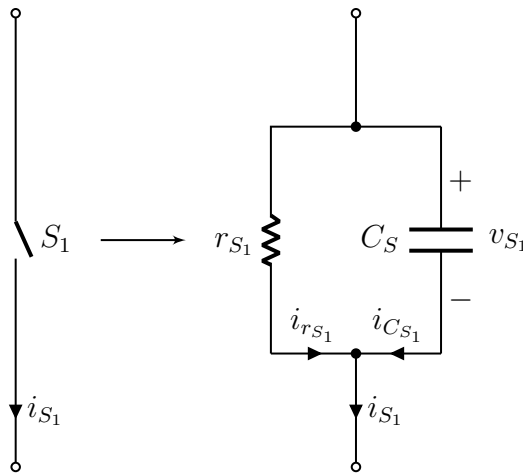
imprevisível no regime de operação analisado, o que torna necessário que eles sejam relacionados de alguma forma com os sinais lógicos de comando aplicados. Caso contrário, subcircuitos não previstos por subintervalos de chaveamento entram em vigor e, com isso, lógicas computacionais, nem sempre triviais, precisam ser incorporadas no procedimento de simulação, para que o modelo dinâmico seja capaz de representar todo o comportamento do circuito elétrico chaveado em questão (Castro et al., 2019).

2.3.2 Método do Modelo Equivalente de Chave

No método por modelo equivalente de chave, todo o semicondutor do circuito elétrico é substituído por uma resistência variável em paralelo com um capacitor de baixa capacitância, conforme ilustrado na Figura 2.5, em que uma chave S_1 é substituída por uma resistência variável r_{S_1} em paralelo com um capacitor de capacitância C_S .

O funcionamento do modelo equivalente ocorre quando $r_{S_1} = r_{ON}$ no modo de condução ($S_1 = 1$) e $r_{S_1} = r_{OFF}$ no modo de bloqueio ($S_1 = 0$), em que $r_{ON} = r_S$ e $r_{OFF} = r_\infty$ são resistências série equivalentes de condução e de bloqueio, respectivamente. Em condições ideais, $r_{ON} = 0 \Omega$ e $r_{OFF} = \infty \Omega$.

Figura 2.5 – Modelo equivalente dos semicondutores para a representação em espaço de estados.



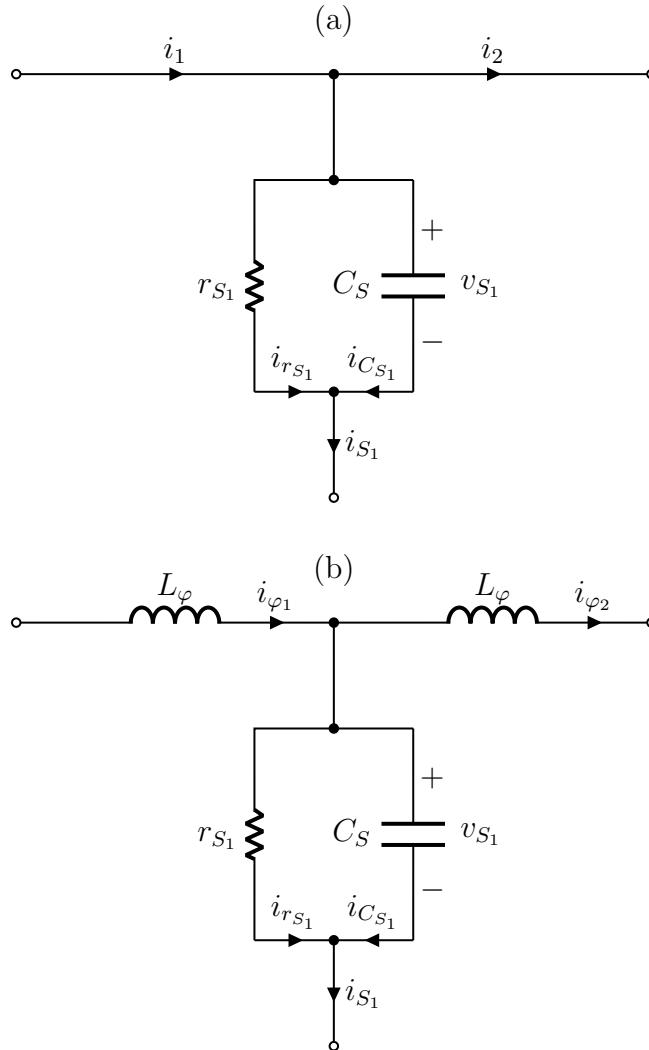
Fonte: Adaptado de Blanchette, Ould-Bachir e David (2012).

Após a substituição das chaves pelos modelos equivalentes, são empregadas as leis de Kirchhoff para obter as equações de estado de todos os elementos armazenadores de energia do circuito, inclusive aquelas referentes aos capacitores adicionais inseridos pelos modelos equivalentes.

Pelo fato de a quantidade de capacitores no circuito aumentar, em alguns casos torna-se complexo encontrar a equação da corrente que flui em um capacitor em função das variáveis de estado. Por esse motivo, um artifício que o método introduz é a inserção de indutores virtuais de baixa indutância L_φ nos ramos adjacentes aos capacitores em que as correntes são desconhecidas, conforme mostrado na Figura 2.6. Uma interpretação física razoável

desse procedimento reside na consideração de um pequeno fluxo de dispersão, do tipo $\varphi = L_\varphi i_\varphi$, quer circula em ramos cujo as correntes são descorrelacionadas com aquelas de indutores físicos (Blanchette; Ould-Bachir; David, 2012).

Figura 2.6 – Procedimento de introdução de indutores virtuais de baixa indutância para facilitar o equacionamento de correntes em capacitores. (a) Ramos adjacentes sem indutores virtuais e (b) ramos adjacentes com indutores virtuais.



Fonte: Próprio autor.

Pela análise do circuito elétrico mostrado na Figura 2.6(a), nota-se que a relação entre a corrente da chave, i_{S_1} , e as correntes dos ramos adjacentes, i_1 e i_2 , é dada por

$$i_{S_1} = i_1 - i_2,$$

logo, a corrente no capacitor, $i_{C_{S_1}}$, é expressa conforme

$$\begin{aligned} i_{C_{S_1}} &= C_S \frac{d}{dt} v_{S_1} = i_{S_1} - i_{r_{S_1}} \\ &= i_1 - i_2 - \frac{v_{S_1}}{r_{S_1}}. \end{aligned}$$

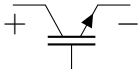
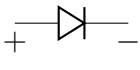
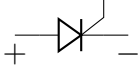
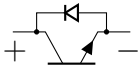
Considerando os indutores virtuais L_φ nos ramos adjacentes, conforme Figura 2.6(b), encontra-se

$$C_S \frac{d}{dt} v_{S_1} = i_{\varphi_1} - i_{\varphi_2} - \frac{v_{S_1}}{r_{S_1}},$$

que é uma equação de estado válida, visto que as correntes dos indutores, i_{φ_1} e i_{φ_2} , são variáveis de estado.

Para a simulação no domínio do tempo, a Tabela 2.2 mostra as lógicas de comutação que devem ser adotadas para cada uma das principais chaves utilizadas na literatura. Além disso, é considerado que as chaves controladas são acionadas por meio de um sinal lógico de comando d .

Tabela 2.2 – Lógicas de comutação para diferentes tipos de chaves na técnica de modelagem em espaço de estados pelo método por modelo equivalente de chave.

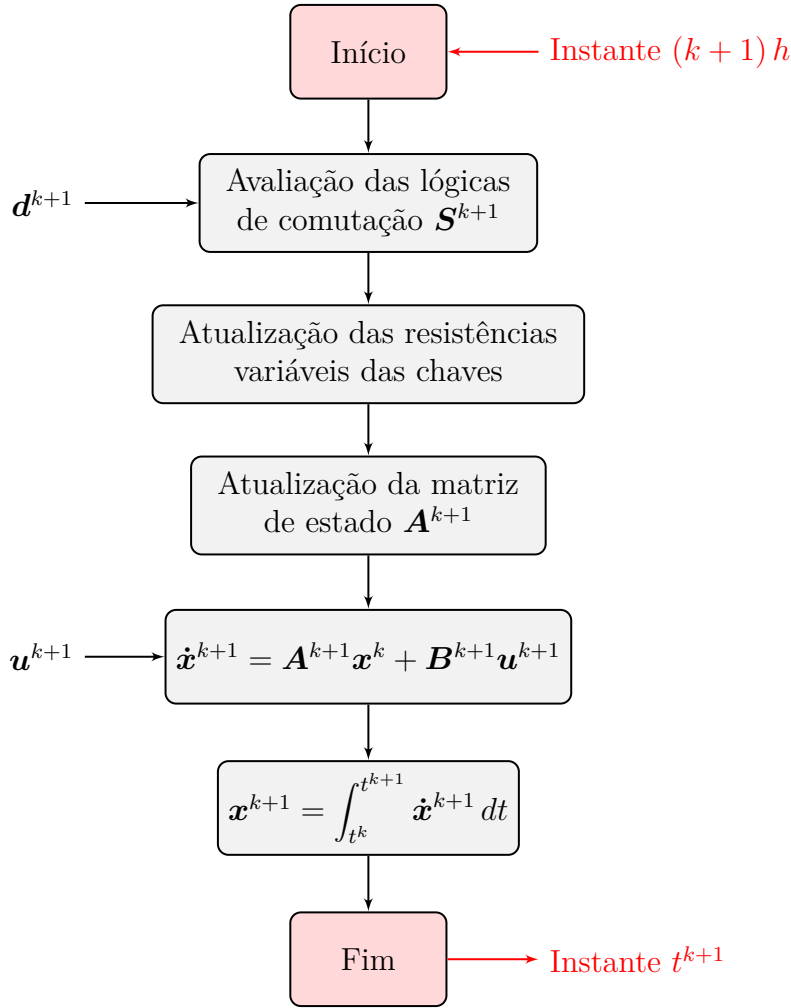
Chave	S^{k+1}
	d^{k+1}
	$(v_S^k \geq 0)$
	$d^{k+1} (v_S^k \geq 0) + (i_S^k > 0)$
	$d^{k+1} + (v_S^k \leq 0)$

Fonte: Adaptado de Blanchette, Ould-Bachir e David (2012).

O procedimento que deve ser seguido para realizar a simulação no domínio do tempo de um dado circuito elétrico chaveado por meio do método do modelo equivalente de chave é resumido na Figura 2.7.

No instante $(k + 1)h$, o processamento é inicializado a partir da leitura dos sinais lógicos de comando d^{k+1} , que permite a avaliação das lógicas de comutação S^{k+1} de todas as chaves do circuito. Em seguida, os estados das chaves são atualizados por meio da mudança das resistências imposta pelos resultados em S^{k+1} . A matriz de estado A^{k+1} é atualizada e, então, $\dot{\mathbf{x}}^{k+1}$ é calculado a partir de \mathbf{x}^k e de \mathbf{u}^{k+1} . Por fim, é realizada a integração no domínio do tempo para a obtenção de \mathbf{x}^{k+1} , encerrando o passo de simulação no instante t^{k+1} .

Figura 2.7 – Fluxograma de simulação de um circuito elétrico chaveado por meio do método do modelo equivalente de chave.



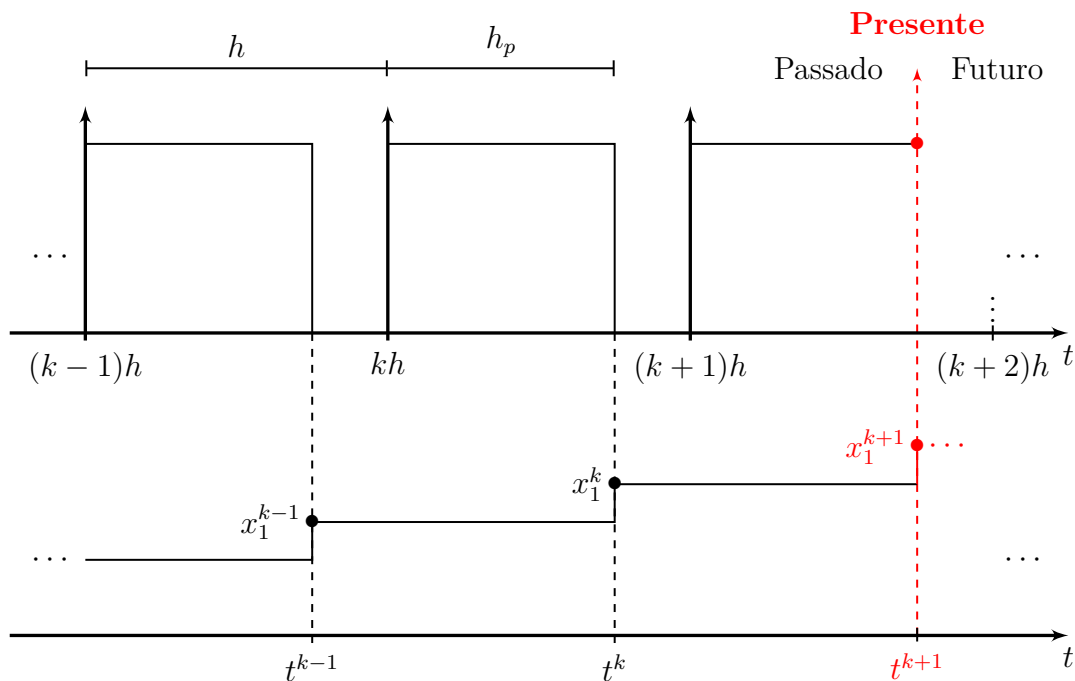
Em termos de representação dinâmica para a simulação computacional no domínio do tempo, o método do modelo equivalente de chave é uma solução interessante para a implementação de conversores de potência em HIL, principalmente por ser capaz de representar toda a dinâmica de operação dos conversores sem quaisquer restrições, independente de qual tipo de chave está sendo utilizada.

O grande problema do método do modelo equivalente de chave, em comparação com o dos subcircuitos, é o mal-condicionamento da matriz de estado \mathbf{A} , causado pela adição de elementos parasitas no modelo, o que exige um tempo de amostragem h muito pequeno para que métodos de integração numérica convencionais sejam estáveis e mantenham uma precisão razoável (Hadizadeh et al., 2019). Com isso, é necessário considerar métodos de integração numérica mais sofisticados e que demandam maior custo computacional, dificultando a implementação do procedimento de simulação em tempo real por meio de MCUs de alto desempenho.

2.4 Métodos Numéricos para Simulações em Tempo Real

Em *softwares* simuladores, não existe restrição quanto ao tempo gasto pelo computador para resolver um dado conjunto de equações. Por mais complexa e demorada que seja a simulação, os resultados referentes ao tempo de resposta especificado serão entregues assim que se encerrar todo o processamento numérico envolvido. Nas aplicações de tempo real em HIL, por outro lado, todo o processamento numérico do modelo precisa estar sincronizado com um relógio de tempo real que avança a passos de h unidades de tempo, conforme ilustrado na Figura 2.8.

Figura 2.8 – Sincronismo entre o relógio de tempo real e o tempo de processamento numérico para a implementação em tempo real de modelos dinâmicos em HIL.



Fonte: Próprio autor.

No instante $(k + M)h$, com $M \in \mathbb{Z}$, é inicializado o processamento numérico do modelo. Após h_p unidades de tempo, esse processamento é encerrado, e todas as variáveis de estado, a saber

$$\mathbf{x}^{k+M} = [x_1^{k+M} \quad x_2^{k+M} \quad \dots \quad x_N^{k+M}]^T,$$

com $N \in \mathbb{N}$, são atualizadas em $t^{k+M} = (k + M)h + h_p$. Passado o tempo ocioso de $h - h_p$ unidades de tempo, é iniciado um novo processamento numérico do modelo no instante $(k + M + 1)h$. A partir disso, é possível verificar que o critério $h_p < h$, que é a restrição de tempo real, precisa ser atendido. Caso contrário, a execução da simulação em tempo real é comprometida por uma perda de sincronia (Castro et al., 2019).

Diante desse contexto, a aplicação de um método numérico para se resolver um determinado modelo para a simulação em HIL deve ser capaz de entregar a solução sem ocasionar a perda de sincronia com o relógio de tempo real. Na técnica de modelagem

ADC, a inversa \mathbf{H}^{-1} é calculada uma única vez e a aplicação do método numérico se resume em resolver (2.7) a cada passo de simulação. Nas técnicas de modelagem por espaço de estados, em contrapartida, é necessário considerar a aplicação de métodos de integração numérica para se resolver

$$\mathbf{x}^{k+1} = \int_{t^k}^{t^{k+1}} \dot{\mathbf{x}}^{k+1} dt.$$

Com a imposição da restrição de tempo real, a ideia mais razoável é selecionar métodos de integração numérica simples e com baixo esforço computacional. Dessa forma, a primeira tentativa é a utilização de algoritmos explícitos de passo único, tal como o método Adams-Bashford de primeira ordem,

$$\mathbf{x}^{k+1} \approx \mathbf{x}^k + h f(t^k, \mathbf{x}^k), \quad (2.9)$$

também conhecido como *forward* Euler (FE).

Sabendo que

$$f(t^{k+M}, \mathbf{x}^{k+M}) := \dot{\mathbf{x}}^{k+M}, \quad (2.10)$$

e considerando $M = 0$, a integração numérica via FE fornece

$$\mathbf{x}^{k+1} \approx (\mathbf{I} + h \mathbf{A}^k) \mathbf{x}^k + h \mathbf{B}^k \mathbf{u}^k, \quad (2.11)$$

em que \mathbf{I} é uma matriz identidade.

Conforme pode ser facilmente notado, a integração numérica por meio do método FE é uma solução que proporciona baixo esforço computacional, por não envolver inversão de matriz. Contudo, métodos explícitos são conhecidos por não apresentarem boa estabilidade numérica quando o modelo a ser integrado apresenta características de mal-condicionamento. Nesse sentido, o benefício em atingir baixo esforço computacional é contraposto pela necessidade de ser adotado um tempo de amostragem h muito pequeno para manter a solução do método precisa (Cellier; Kofman, 2006; Najm, 2010).

Pensando nisso, a próxima tentativa é a utilização de algoritmos implícitos, tal como o método *backward* Euler de primeira ordem (BDF1),

$$\mathbf{x}^{k+1} \approx \mathbf{x}^k + h f(t^{k+1}, \mathbf{x}^{k+1}), \quad (2.12)$$

ou ainda os métodos *backward* Euler de segunda ordem (BDF2) e trapezoidal (TR), que proporcionam melhor resolução em troca de um maior esforço computacional devido à ordem mais elevada,

$$\mathbf{x}^{k+1} \approx \frac{4}{3} \mathbf{x}^k - \frac{1}{3} \mathbf{x}^{k-1} + \frac{2}{3} h f(t^{k+1}, \mathbf{x}^{k+1}), \quad (2.13)$$

e

$$\mathbf{x}^{k+1} \approx \mathbf{x}^k + \frac{h}{2} [f(t^{k+1}, \mathbf{x}^{k+1}) + f(t^k, \mathbf{x}^k)], \quad (2.14)$$

respectivamente.

Os algoritmos implícitos podem ser representados no formato

$$\mathbf{W}^{k+1} \mathbf{x}^{k+1} \approx \mathbf{w}^{k+1}. \quad (2.15)$$

Substituindo-se (2.10) em (2.12), (2.13) e (2.14), a partir de $M = 0$ e $M = 1$, as expressões de \mathbf{W} e \mathbf{w} referentes aos métodos BDF1, BDF2 e TR são mostradas por meio da Tabela 2.3.

Tabela 2.3 – Matrizes para a implementação dos métodos de integração numérica BDF1, BDF2 e TR.

Método	\mathbf{W}^{k+1}	\mathbf{w}^{k+1}
BDF1	$\mathbf{I} - h \mathbf{A}^{k+1}$	$\mathbf{x}^k + h \mathbf{B}^{k+1} \mathbf{u}^{k+1}$
BDF2	$\mathbf{I} - \frac{2}{3} h \mathbf{A}^{k+1}$	$\frac{4}{3} \mathbf{x}^k - \frac{1}{3} \mathbf{x}^{k-1} + \frac{2}{3} h \mathbf{B}^{k+1} \mathbf{u}^{k+1}$
TR	$\mathbf{I} - \frac{h}{2} \mathbf{A}^{k+1}$	$\left(\mathbf{I} + \frac{h}{2} \mathbf{A}^k \right) \mathbf{x}^k + \frac{h}{2} \mathbf{B}^{k+1} \mathbf{u}^{k+1} + \frac{h}{2} \mathbf{B}^k \mathbf{u}^k$

Fonte: Próprio autor.

Como se pode notar, a aplicação de algoritmos implícitos, apesar de tratar o problema de sistemas mal-condicionados, requer o cômputo de \mathbf{W}^{-1} a cada passo de simulação, caso \mathbf{W} seja variante no tempo. Tal procedimento não é viável, visto que um grande esforço computacional é demandado. Contudo, existem algumas estratégias de solução computacional que viabilizam a simulação em tempo real de conversores em que os modelos dinâmicos são mal-condicionados.

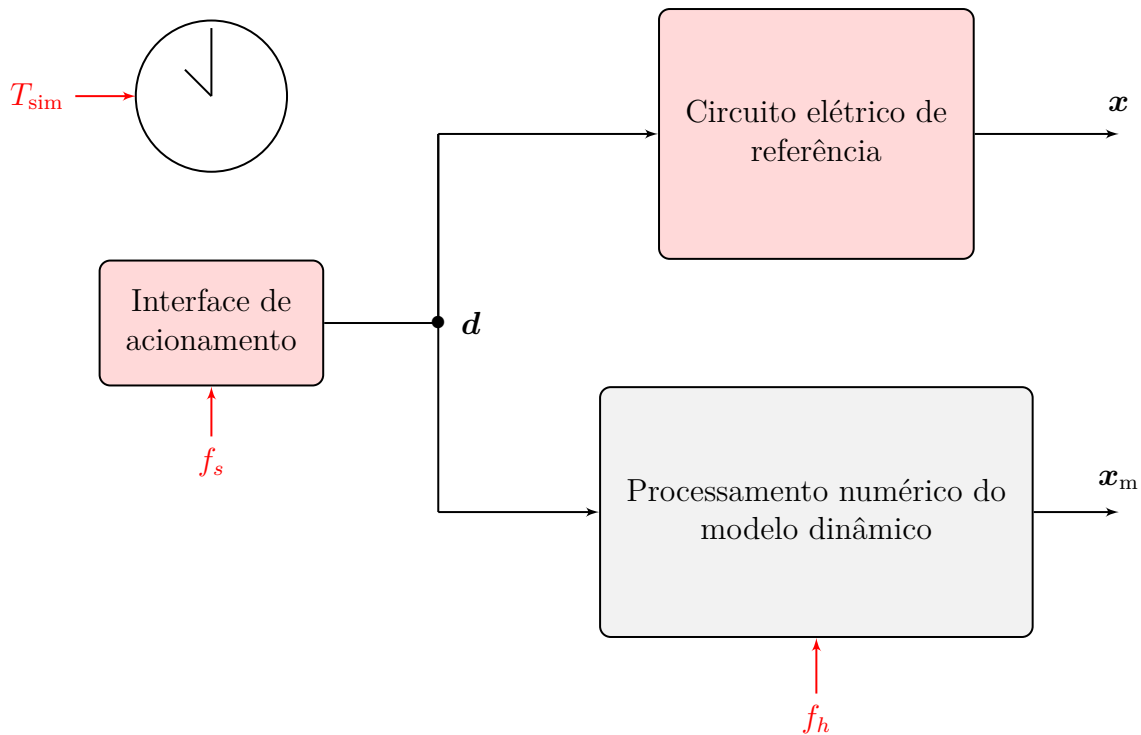
A solução mais simples, entretanto com utilização de alta capacidade de memória, é por meio do pré-cálculo e alocação de todas as possíveis \mathbf{W}^{-1} (Dufour; Mahseredjian; Belanger, 2011). Outra alternativa possível, porém com maior aplicação para implementações em FPGAs, devido à possibilidade de paralelismo, seria por meio do método iterativo de Gauss–Seidel, com restrição ao número máximo de iterações (Blanchette; Ould-Bachir; David, 2012). Por fim, é possível investigar as formulas de Sherman–Morrison–Woodbudy (Hager, 1989) para se realizar atualizações consecutivas de \mathbf{W}^{-1} a partir de uma matriz inversa inicial pré-calculada (Hadizadeh et al., 2019).

2.5 Validação de Modelos Dinâmicos e Análise de Resultados

Para realizar a validação de modelos dinâmicos de conversores, em geral, é implementado o procedimento de simulação computacional de acordo com o diagrama esquemático

apresentado por meio da Figura 2.9.

Figura 2.9 – Procedimento de simulação computacional genérico para a validação de modelos dinâmicos de conversores.



Fonte: Próprio autor.

Com a elaboração de uma interface de acionamento, os sinais lógicos de comando \mathbf{d} são implementados a uma frequência de chaveamento f_s especificada. Com isso, é realizado o acionamento do circuito elétrico de referência, que é construído a partir da topologia do conversor a ser analisado. O modelo dinâmico, que é processado na frequência de amostragem $f_h = 1/h$, é então acionado de forma síncrona com o circuito elétrico de referência.

A simulação é feita para um tempo de resposta T_{sim} , e todos os parâmetros selecionados para o circuito elétrico de referência são utilizados na descrição do modelo dinâmico. Dessa forma, as respostas desenvolvidas pelo circuito elétrico de referência e pelo processamento numérico do modelo dinâmico, \mathbf{x} e \mathbf{x}_m , respectivamente, podem ser comparadas no tempo por meio da superposição dos sinais.

Como métrica de avaliação de desempenho para análise das soluções numéricas, é possível realizar o cálculo do erro relativo, $e_r[\bullet]$, entre as respostas desenvolvidas pelo circuito elétrico de referência e pelo processamento numérico do modelo dinâmico. Para grandezas CC, os erros relativos são ponderados com relação aos valores médios das respostas do circuito elétrico de referência, ao passo que, em grandezas CA, a ponderação pode ser feita a partir dos respectivos valores eficazes (rms).

2.6 Considerações Finais

Neste capítulo, foi discutido de forma generalizada a implementação em tempo real de circuitos elétricos chaveados, com ênfase para o desenvolvimento de aplicações em tempo real para simulação de conversores de potência em HIL. Sendo assim, as principais técnicas de modelagem encontradas na literatura, via ADC e por espaço de estados, foram apresentadas com detalhes.

Dessa forma, soluções numéricas que podem ser utilizadas no contexto de implementações em tempo real de circuitos elétricos chaveados foram apresentadas. Dessa forma, foi discutida a solução numérica da técnica de modelagem ADC, bem como os principais métodos de integração numérica para solução da técnica de modelagem por espaço de estados, sendo FE para modelos dinâmicos sem problemas de condicionamento, e BDF1, BDF2 e TR para modelos dinâmicos com mal-condicionamento.

O capítulo se encerra com a apresentação do método de validação de modelos dinâmicos, seguida de uma breve discussão acerca da métrica de avaliação de desempenho da solução numérica aplicada.

Obtenção de Modelos Dinâmicos

3.1 Introdução

A primeira tarefa executada na grande maioria dos *softwares* de simulação, durante a construção de uma rotina de simulação computacional no domínio do tempo, é a representação matemática da planta por meio da obtenção do respectivo modelo dinâmico associado. Uma vez que o modelo dinâmico está disponível, os *softwares* de simulação passam para a etapa de emulação temporal a partir da definição do tempo de simulação, permitindo que o modelo dinâmico seja processado numericamente ao longo da faixa de tempo de análise. Os resultados podem, então, ser entregues durante a emulação temporal, ou após finalizado todo o processamento numérico do modelo dinâmico (Chua; Lin, 1975; Najm, 2010).

Para a implementação do procedimento de simulação em tempo real, assim como em *softwares* de simulação, a primeira tarefa a ser executada é a obtenção do modelo dinâmico. A grande diferença ocorre na etapa de emulação temporal, que não é realizada quando a abordagem é de tempo real (Castro et al., 2019). Como foi visto no Capítulo 2, mais precisamente na Seção 2.4, Figura 2.8, o processamento numérico do modelo dinâmico ocorre em sincronia com um relógio de tempo real e, diferentemente dos *softwares* de simulação, os resultados são entregues em tempo real, o que possibilita a interação entre o mundo real e o ambiente virtual.

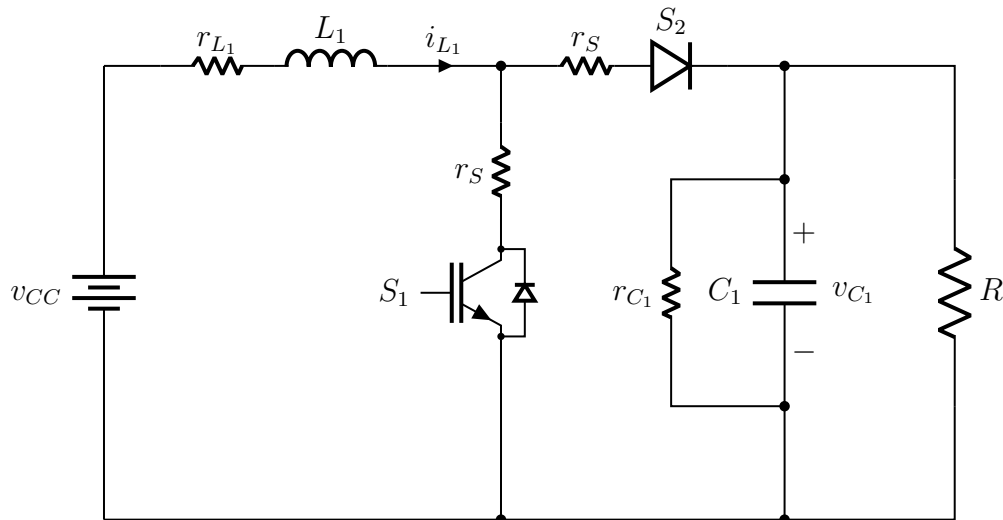
Tendo em vista a importância dos modelos dinâmicos durante o processo de implementação do procedimento de simulação no domínio do tempo, este capítulo aborda o levantamento de modelos dinâmicos de algumas topologias de conversores de potência amplamente estudadas na literatura. O primeiro conversor a ser considerado é o conversor CC-CC *boost*. Em seguida, serão analisadas quatro configurações diferentes de conversores CC-CA, as versões monofásica em ponte completa dos conversores inversor tipo fonte de tensão (VSI), em seus modos ilhado e conectado, o ZSI e a versão Quasi-ZSI.

3.2 Conversor CC–CC *Boost*

O conversor *boost* é um dispositivo eletrônico chaveado clássico e, por isso, amplamente discutido em diversos livros da literatura, tais como Erickson e Maksimović (2001), Hart (2011) e Rashid, Kumar e Kulkarni (2014). As aplicações desse conversor e de suas muitas derivações são vastas, dentre as quais são mencionados o processamento de energia em painéis fotovoltaicos e em microrredes CC, a elevação de tensão CC em linhas de transmissão HVDC e o condicionamento de tensão em trens elétricos e em aeronaves (Forouzesh et al., 2017).

O circuito elétrico do conversor *boost* com perdas resistivas é mostrado por meio da Figura 3.1, em que é observada a existência de dois elementos armazenadores de energia, o indutor L_1 e o capacitor C_1 , uma chave controlada, S_1 , que é considerada como sendo um IGBT, e um diodo, S_2 . O seu comportamento básico é que, dada uma tensão de entrada v_{CC} , ao se aplicar um sinal lógico de comando d em S_1 , por meio da modulação por largura de pulso (PWM) com razão cíclica D , a tensão v_{C_1} sobre o capacitor de saída será igual ou superior ao valor de v_{CC} ao longo da faixa de operação do conversor.

Figura 3.1 – Circuito elétrico do conversor *boost*.



Fonte: Adaptado de Rashid, Kumar e Kulkarni (2014)

Na sequência, os modelos dinâmicos do conversor *boost* são obtidos por meio das técnicas de modelagem apresentadas no Capítulo 2.

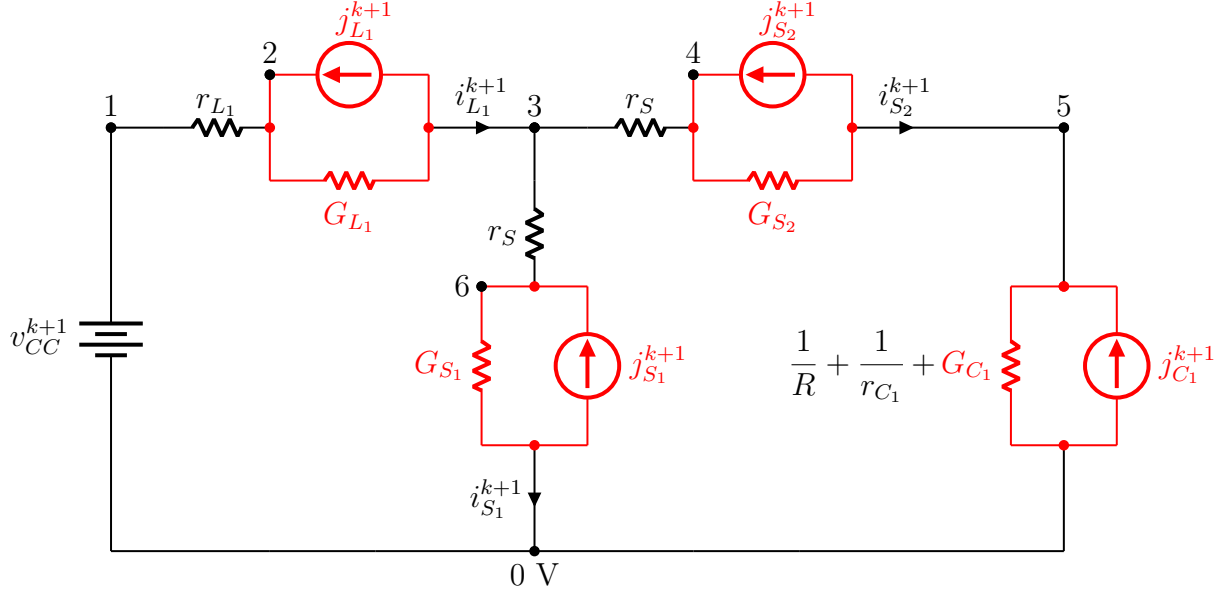
3.2.1 Modelagem pela Técnica ADC

Substituindo os elementos armazenadores de energia e os semicondutores do conversor *boost* pelos respectivos modelos ADC da Figura 2.1, é encontrada a versão ADC do circuito elétrico da Figura 3.1, que é ilustrada na Figura 3.2.

Assim, o modelo dinâmico obtido por meio da técnica ADC é extraído por meio da formulação nodal. Sendo assim, é verificado que o circuito elétrico ADC resultante apresenta

seis nós de tensão e três ramos de corrente, de tal forma que $\mathbf{H} \in \mathbb{R}^{9 \times 9}$ e $\{\mathbf{x}, \mathbf{b}\} \in \mathbb{R}^{9 \times 1}$. Além disso, como o conversor *boost* contém dois semicondutores, $\mathbf{j}_S \in \mathbb{R}^{2 \times 1}$, logo $\mathbf{u} \in \mathbb{R}^{7 \times 1}$.

Figura 3.2 – Circuito elétrico do modelo ADC do conversor *boost*.



Fonte: Próprio autor.

Aplicando a lei de Kirchhoff das correntes nos nós do circuito elétrico da Figura 3.2, é encontrado o conjunto de equações nodais

$$\begin{cases} -i_{S_1}^{k+1} + G_{S_1} v_6^{k+1} - G_{S_1} v_7^{k+1} = j_{S_1}^{k+1}, \\ -i_{S_2}^{k+1} + G_{S_2} v_4^{k+1} - G_{S_2} v_5^{k+1} = j_{S_2}^{k+1}, \\ -i_{L_1}^{k+1} + G_{L_1} v_2^{k+1} - G_{L_1} v_3^{k+1} = j_{L_1}^{k+1}, \\ -i_{S_2}^{k+1} + \left(\frac{1}{r_{C_1}} + \frac{1}{R} + G_{C_1} \right) v_5^{k+1} = j_{C_1}^{k+1}, \\ v_1^{k+1} = v_{CC}^{k+1}, \\ -i_{L_1} + \frac{1}{r_{L_1}} v_1^{k+1} - \frac{1}{r_{L_1}} v_2^{k+1} = 0, \\ -i_{S_1} + \frac{1}{r_S} v_3^{k+1} - \frac{1}{r_S} v_6^{k+1} = 0, \\ -i_{S_2} + \frac{1}{r_S} v_3^{k+1} - \frac{1}{r_S} v_4^{k+1} = 0, \\ i_{L_1} - i_{S_1} - i_{S_2} = 0. \end{cases} \quad (3.1)$$

A partir de (3.1), faz-se

$$\mathbf{x}^{k+1} := \begin{bmatrix} i_{L_1}^{k+1} & i_{S_1}^{k+1} & i_{S_2}^{k+1} & v_1^{k+1} & v_2^{k+1} & v_3^{k+1} & v_4^{k+1} & v_5^{k+1} & v_6^{k+1} \end{bmatrix}^T, \quad (3.2)$$

$$\mathbf{j}_S^{k+1} := \begin{bmatrix} j_{S_1}^{k+1} & j_{S_2}^{k+1} \end{bmatrix}^T \quad (3.3)$$

e

$$\mathbf{u}^{k+1} := \left[j_{L_1}^{k+1} \quad j_{C_1}^{k+1} \quad v_{CC}^{k+1} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \right]^T, \quad (3.4)$$

em que

$$\mathbf{b}^{k+1} = \begin{bmatrix} \mathbf{j}_S^{k+1} \\ \mathbf{u}^{k+1} \end{bmatrix}. \quad (3.5)$$

Dessa forma, o conjunto de equações (3.1) pode ser organizado conforme a representação matricial (2.6), possibilitando a solução via (2.7).

Para a realização da simulação computacional no domínio do tempo, é necessário observar que

$$G_{L_1} = \frac{h}{L_1}, \quad (3.6)$$

$$G_{C_1} = \frac{C_1}{h}, \quad (3.7)$$

com

$$j_{L_1}^{k+1} = -i_{L_1}^k \quad (3.8)$$

e

$$j_{C_1}^{k+1} = G_{C_1} v_5^k. \quad (3.9)$$

Além disso, a dinâmica de operação das chaves nos modos de condução e de bloqueio é descrita conforme

$$j_{S_1}^{k+1} = \begin{cases} -i_{S_1}^k, & \text{se } S_1^{k+1} = 1, \\ G_{S_1} v_6^k, & \text{se } S_1^{k+1} = 0, \end{cases} \quad (3.10)$$

e

$$j_{S_2}^{k+1} = \begin{cases} -i_{S_2}^k, & \text{se } S_2^{k+1} = 1, \\ G_{S_2} (v_4^k - v_5^k), & \text{se } S_2^{k+1} = 0. \end{cases} \quad (3.11)$$

Por meio da Tabela 2.1, as lógicas de comutação das chaves são dadas por

$$S_1^{k+1} = d^{k+1} + (v_6^k \leq 0), \quad (3.12)$$

para o IGBT, e

$$S_2^{k+1} = [(v_4^k - v_5^k) \geq 0], \quad (3.13)$$

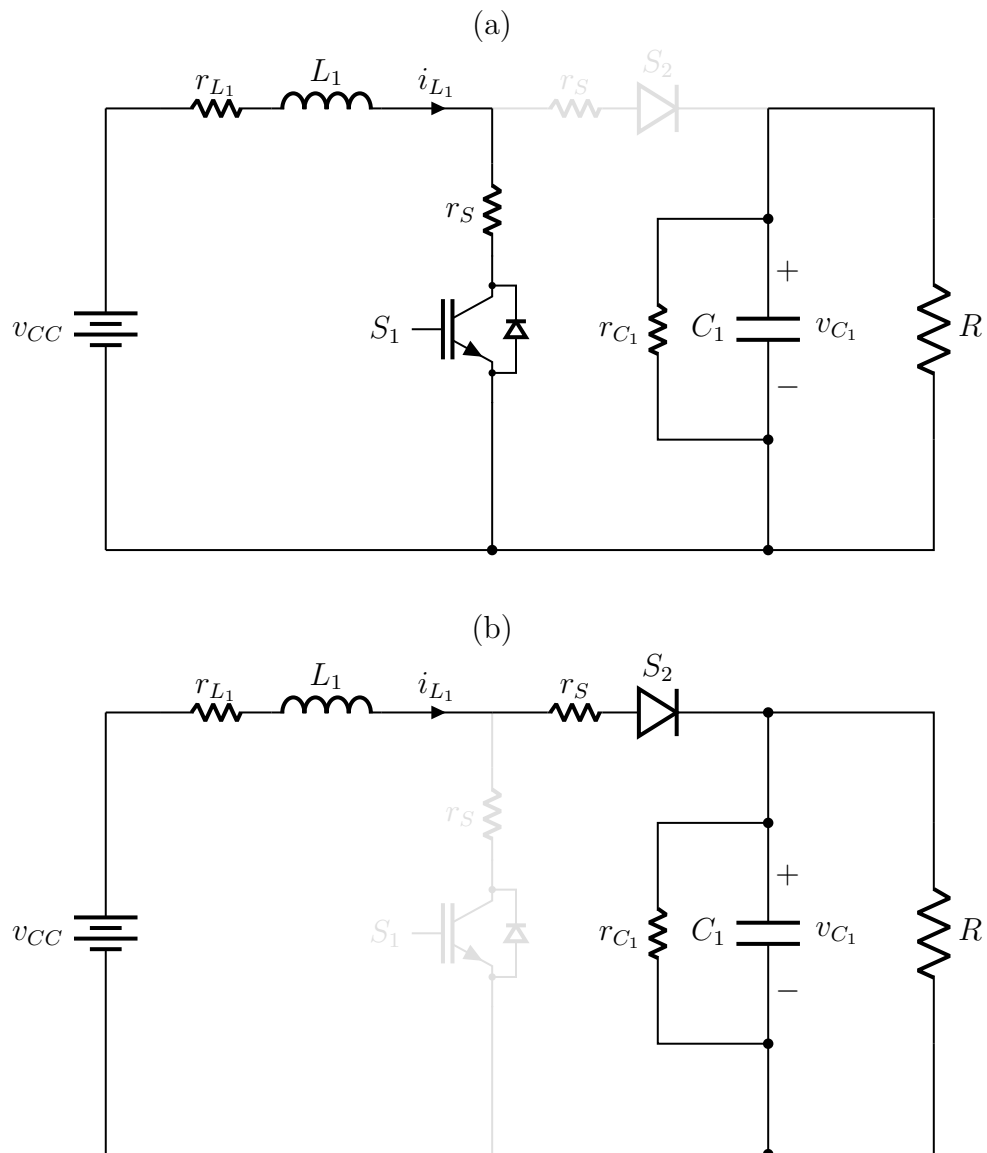
para o diodo.

A partir de todas essas descrições matemáticas, o procedimento de simulação computacional do conversor *boost* por meio da técnica ADC pode ser implementado a partir das etapas descritas no fluxograma da Figura 2.2, lembrando que a inversa da matriz \mathbf{H} pode ser pré-calculada.

3.2.2 Modelagem pelo Método dos Subcircuitos

No conversor *boost*, existem $2^2 = 4$ possíveis subcircuitos, visto que o mesmo apresenta duas chaves em sua topologia. Considerando apenas o modo de condução contínua, o conversor *boost* pode ser representado por meio da consideração $S_1 = \bar{S}_2 = S$, de maneira que apenas dois subintervalos de chaveamento são factíveis, $S = 1$ e $S = 0$, em que os respectivos subcircuitos são os mostrados por meio da Figura 3.3.

Figura 3.3 – Subcircuitos referentes aos subintervalos de chaveamento factíveis do conversor *boost*, (a) $S = 1$ e (b) $S = 0$.



Fonte: Próprio autor.

Como a aplicação do método dos subcircuitos não envolve a adição de elementos parasitas durante o processo de modelagem, o modelo dinâmico do conversor *boost* pode ser obtido a partir de apenas duas variáveis de estado, que são a corrente do indutor, i_{L_1} , e a tensão sobre o capacitor, v_{C_1} .

Aplicando as leis de Kirchhoff nos subcircuitos da Figura 3.3, são encontrados os conjuntos de equações de estado

$$\begin{cases} \frac{d}{dt}i_{L_1} = -\frac{1}{L_1}(r_{L_1} + r_S)i_{L_1} + \frac{1}{L_1}v_{CC}, \\ \frac{d}{dt}v_{C_1} = -\frac{1}{C_1}\left(\frac{1}{r_{C_1}} + \frac{1}{R}\right)v_{C_1}, \end{cases} \quad (3.14)$$

para o subintervalo de chaveamento $S = 1$, e

$$\begin{cases} \frac{d}{dt}i_{L_1} = -\frac{1}{L_1}(r_{L_1} + r_S)i_{L_1} - \frac{1}{L_1}v_{C_1} + \frac{1}{L_1}v_{CC}, \\ \frac{d}{dt}v_{C_1} = \frac{1}{C_1}i_{L_1} - \frac{1}{C_1}\left(\frac{1}{r_{C_1}} + \frac{1}{R}\right)v_{C_1}, \end{cases} \quad (3.15)$$

para o subintervalo de chaveamento $S = 0$.

Além disso,

$$\mathbf{x} := [i_{L_1} \quad v_{C_1}]^T \quad (3.16)$$

e

$$u := v_{CC}, \quad (3.17)$$

ou seja, ambos os conjuntos de equações (3.14) e (3.15) podem ser representados no formato (2.8), o que fornece as matrizes $\{\mathbf{A}_1, \mathbf{A}_2\} \in \mathbb{R}^{2 \times 2}$ e $\{\mathbf{B}_1, \mathbf{B}_2\} \in \mathbb{R}^{2 \times 1}$.

Seguindo o procedimento de simulação definido por meio das etapas do fluxograma da Figura 2.4, a implementação da simulação computacional no domínio do tempo pode ser alcançada com base na integração numérica de

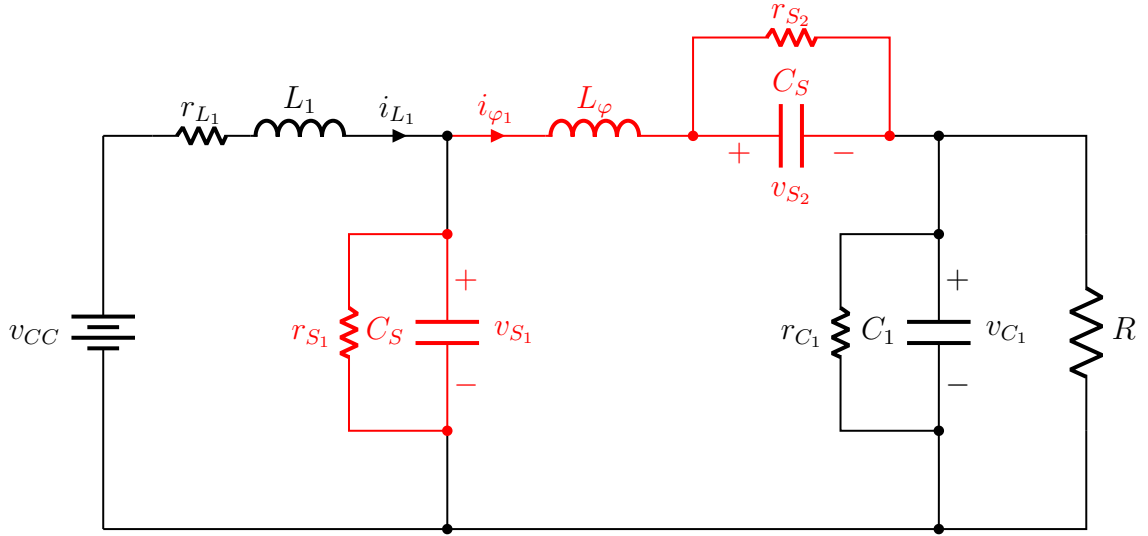
$$\dot{\mathbf{x}}^{k+1} = \begin{cases} \mathbf{A}_1 \mathbf{x}^k + \mathbf{B}_1 u^{k+1}, & \text{se } S^{k+1} = 1, \\ \mathbf{A}_2 \mathbf{x}^k + \mathbf{B}_2 u^{k+1}, & \text{se } S^{k+1} = 0, \end{cases} \quad (3.18)$$

em que $S^{k+1} = d^{k+1}$.

3.2.3 Modelagem pelo Método do Modelo Equivalente de Chave

Outra possibilidade de modelar o conversor *boost* é a partir da substituição de suas chaves de acordo com o modelo equivalente da Figura 2.5, para obter o circuito elétrico da Figura 3.4. Note que, com o intuito de facilitar a obtenção do modelo dinâmico associado, foi inserido um indutor virtual L_φ em série com o modelo equivalente do diodo, procedimento este em conformidade com a descrição apresentada na Figura 2.6.

Figura 3.4 – Circuito elétrico do conversor *boost* com a substituição das chaves para a aplicação da técnica do modelo equivalente de chave.



Fonte: Próprio autor.

Diferentemente do método dos subcircuitos, o modelo dinâmico do conversor *boost* por meio da aplicação do método do modelo equivalente de chave é descrito por meio de um único conjunto de equações. Contudo, devido a inserção dos elementos parasitas, um indutor L_{φ} e dois capacitores C_S , é necessário o cômputo de cinco variáveis de estado, as correntes i_{L_1} e i_{φ_1} , e as tensões v_{C_1} , v_{S_1} e v_{S_2} , para que o comportamento dinâmico do conversor possa ser completamente representado.

Aplicando as leis de Kirchhoff no circuito elétrico da Figura 3.4, é encontrado o conjunto de equações de estado

$$\begin{cases} \frac{d}{dt} i_{L_1} = -\frac{r_{L_1}}{L_1} i_{L_1} - \frac{1}{L_1} v_{S_1} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{\varphi_1} = -\frac{1}{L_{\varphi}} v_{C_1} + \frac{1}{L_{\varphi}} v_{S_1} - \frac{1}{L_{\varphi}} v_{S_2}, \\ \frac{d}{dt} v_{C_1} = \frac{1}{C_1} i_{\varphi_1} - \frac{1}{C_1} \left(\frac{1}{r_{C_1}} + \frac{1}{R} \right) v_{C_1}, \\ \frac{d}{dt} v_{S_1} = \frac{1}{C_S} i_{L_1} - \frac{1}{C_S} i_{\varphi_1} - \frac{1}{r_{S_1} C_S} v_{S_1}, \\ \frac{d}{dt} v_{S_2} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{r_{S_2} C_S} v_{S_2}. \end{cases} \quad (3.19)$$

Definindo

$$\mathbf{x} := \left[i_{L_1} \quad i_{\varphi_1} \quad v_{C_1} \quad v_{S_1} \quad v_{S_2} \right]^T \quad (3.20)$$

e

$$u := v_{CC}, \quad (3.21)$$

o conjunto de equações (3.19) pode ser representado no formato (2.8), o que fornece as matrizes $\mathbf{A} \in \mathbb{R}^{5 \times 5}$ e $\mathbf{B} \in \mathbb{R}^{5 \times 1}$, com \mathbf{A} variante no tempo devido às resistências das chaves r_{S_1} e r_{S_2} , que são alternadas entre $r_{\text{ON}} = r_S$ e $r_{\text{OFF}} = r_\infty$.

O procedimento de simulação computacional no domínio do tempo é implementado conforme as etapas do fluxograma mostrado na Figura 2.7.

Por meio da Tabela 2.2, as lógicas de comutação das chaves são dadas por

$$S_1^{k+1} = d^{k+1} + (v_{S_1}^k \leq 0), \quad (3.22)$$

para o IGBT, e

$$S_2^{k+1} = (v_{S_2}^k \geq 0), \quad (3.23)$$

para o diodo.

A partir disso, a resposta temporal de todas as variáveis de estado do modelo dinâmico é obtida por meio da integração numérica de

$$\dot{\mathbf{x}}^{k+1} = \mathbf{A}^{k+1} \mathbf{x}^k + \mathbf{B}^{k+1} u, \quad (3.24)$$

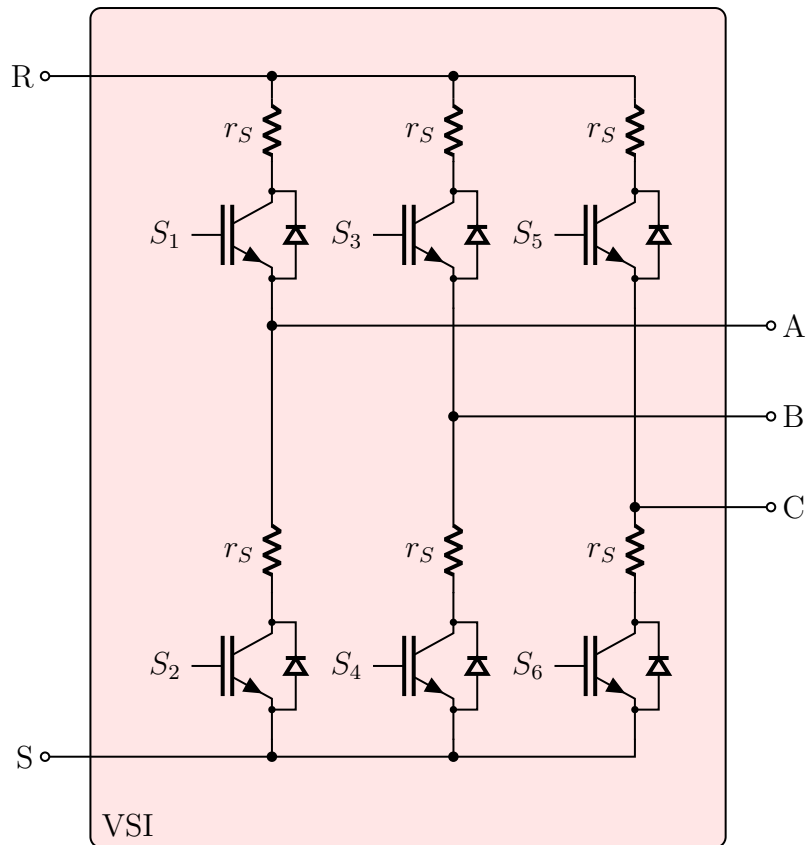
em que \mathbf{A}^{k+1} é a matriz \mathbf{A} encontrada após as atualizações das resistências r_{S_1} e r_{S_2} , modificadas pelas lógicas de comutação S_1^{k+1} e S_2^{k+1} , respectivamente, e $\mathbf{B}^{k+1} = \mathbf{B}^k = \mathbf{B}$ é uma matriz invariante no tempo.

3.3 Conversores CC–CA

Por serem capazes de realizar a transferência de potência de uma fonte CC para uma carga CA, os conversores CC–CA são dispositivos eletrônicos chaveados que desempenham uma tarefa fundamental em acionamentos elétricos e no processamento de energia, como, por exemplo, na aplicação de controle em máquinas elétricas e na manutenção de fontes de energia ininterruptas (UPSs). No contexto de fontes alternativas de energia, conversores CC–CA implementam uma interface controlável entre o elo CC e a rede de distribuição, o que permite a integração entre microrredes CC e o sistema de distribuição de energia elétrica (Rashid; Kumar; Kulkarni, 2014).

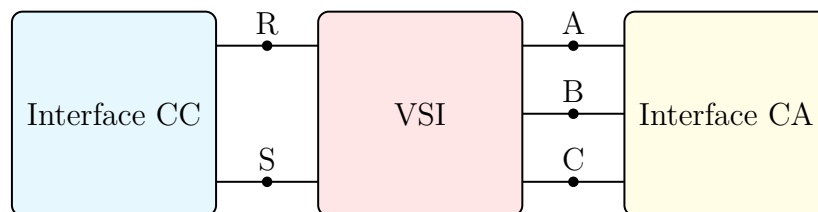
Uma das estruturas mais convencionais de conversor CC–CA é o VSI trifásico com seis chaves semicondutoras, em que a configuração com IGBTs e perdas resistivas é apresentada por meio da Figura 3.5. Conforme mostrado na Figura 3.6, no ponto de acoplamento R-S do VSI, é conectada a interface CC, que pode ser tanto uma fonte primária, ou um elo CC, como também um conversor CC–CC empregado para um pré-estágio de processamento de energia. No ponto de acoplamento A-B-C, o VSI é conectado a uma interface CA que, na grande maioria das aplicações, se resume em uma composição de diferentes topologias e configurações de filtros, voltados a melhorar a qualidade da energia na rede (He et al., 2014; Ruan et al., 2018).

Figura 3.5 – Estrutura básica do conversor VSI trifásico.



Fonte: Próprio autor.

Figura 3.6 – Configuração esquemática genérica para montagem de conversores CC-CA.



Fonte: Próprio autor.

Na sequência, é realizada a obtenção dos modelos dinâmicos de algumas configurações de conversores CC-CA que são encontradas na literatura, a saber o VSI nos modos de operação ilhado e conectado, além do ZSI e da topologia quasi-ZSI.

Para fins de redução de redundância nos modelos dinâmicos e de esforço computacional, visando a implementação do procedimento de simulação para aplicações de tempo real em HIL, serão consideradas as versões monofásicas em ponte completa das configurações supracitadas. Além disso, a técnica ADC não será abordada, devido ao volume considerável de equações nodais e de variáveis que são introduzidas.

3.3.1 Conversor VSI com filtro LC no Modo Ilhado

A interface CA composta por um filtro LC no modo ilhado é mostrada por meio da Figura 3.7. Na Figura 3.8, são apresentadas as configurações monofásica em ponte completa e trifásica do conversor VSI a partir da conexão com uma fonte de tensão primária, v_{CC} , e com o filtro LC no modo ilhado da Figura 3.7.

Figura 3.7 – Interface CA composta por um filtro LC alimentando uma carga R no modo ilhado.

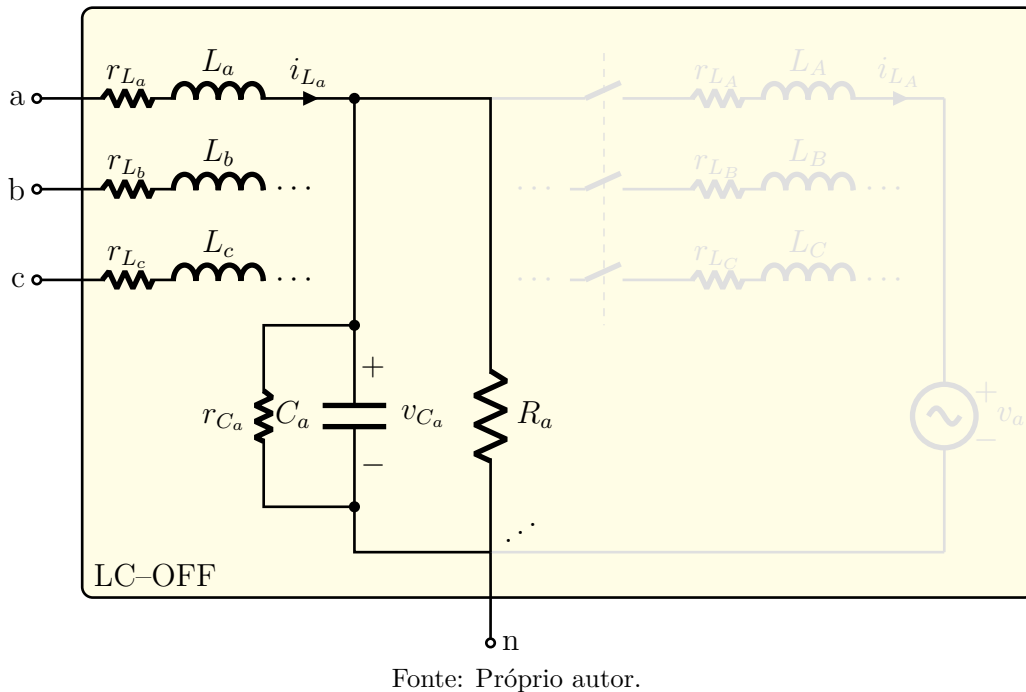
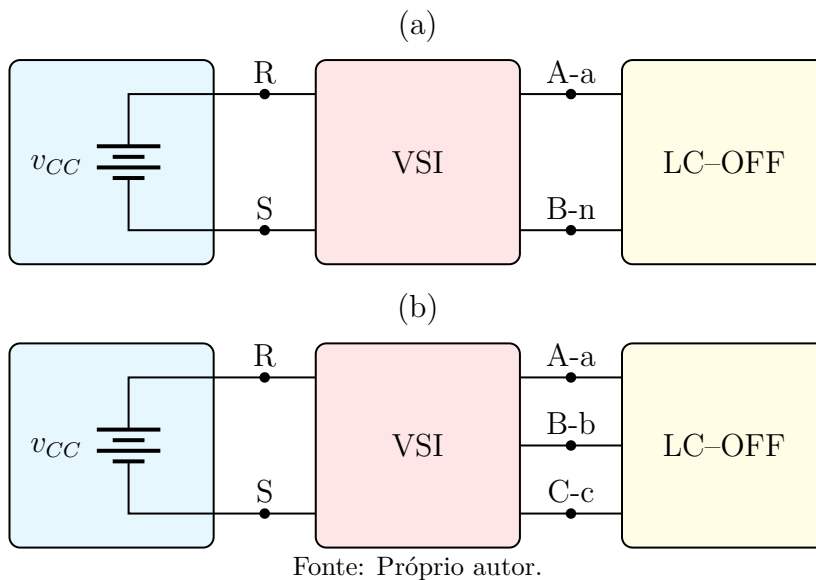


Figura 3.8 – Estrutura do conversor VSI com filtro LC no modo ilhado. Configurações (a) monofásica em ponte completa e (b) trifásica.

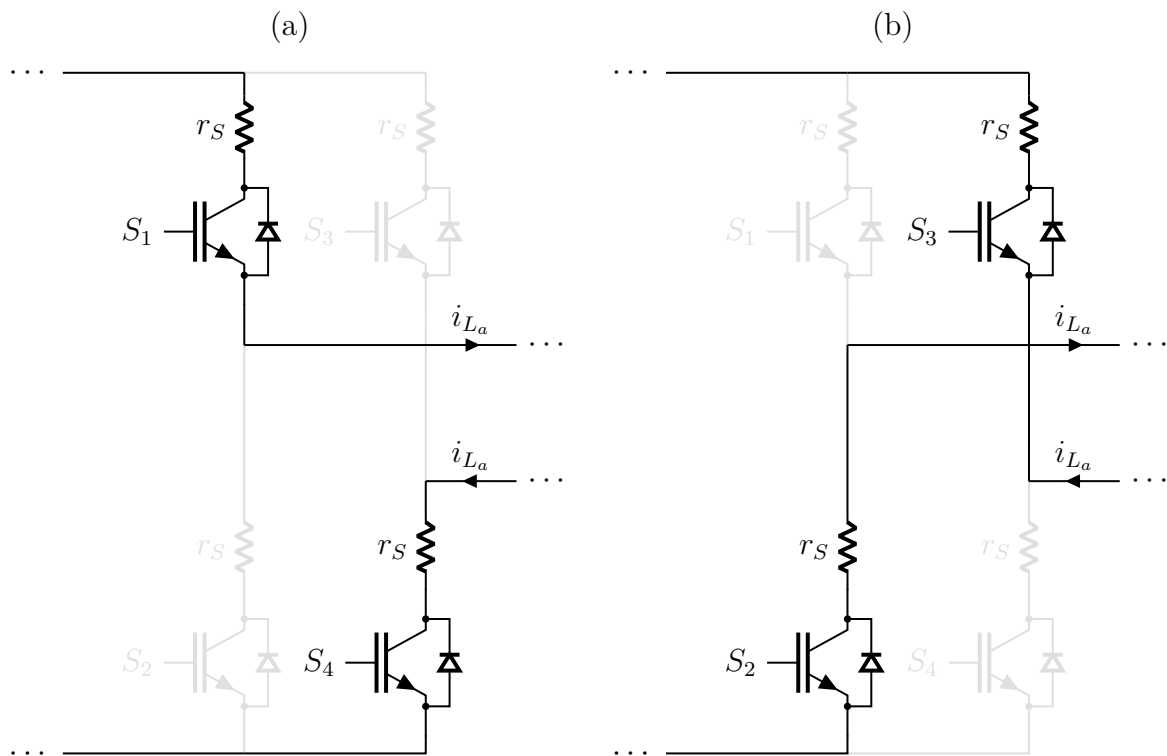


3.3.1.1 Modelagem pelo Método dos Subcircuitos

O VSI monofásico em ponte completa, por si só, apresenta $2^4 = 16$ possíveis subcircuitos, visto que o mesmo contém quatro chaves em sua topologia. No entanto, a partir da configuração mostrada na Figura 3.8(a), é possível observar que, se ambas as chaves de um mesmo braço do VSI são fechadas, a fonte de tensão v_{CC} entra em curto-circuito, o que não é factível. Além disso, não é uma estratégia de acionamento razoável manter as chaves de um mesmo braço do VSI abertas de maneira simultânea.

A partir dessas observações, conclui-se que a estrutura mostrada na Figura 3.8(a) é acionada impondo $S_1 = S_4 = \bar{S}_2 = \bar{S}_3 = S$, de tal forma que apenas os subintervalos de chaveamento $S = 1$ e $S = 0$ são factíveis. Os respectivos subcircuitos são os mostrados por meio da Figura 3.9.

Figura 3.9 – Subcircuitos referentes aos subintervalos de chaveamento factíveis do conversor VSI na configuração monofásica em ponte completa. (a) $S = 1$ e (b) $S = 0$.



Fonte: Próprio autor.

Procedendo para a aplicação das leis de Kirchhoff nos subcircuitos da Figura 3.8(a), são obtidos os conjuntos de equações de estado

$$\begin{cases} \frac{d}{dt} i_{L_a} = -\frac{1}{L_a} (2r_S + r_{L_a}) i_{L_a} - \frac{1}{L_a} v_{C_a} + \frac{1}{L_a} v_{CC}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{cases} \quad (3.25)$$

para o subintervalo de chaveamento $S = 1$, e

$$\begin{cases} \frac{d}{dt} i_{L_a} = -\frac{1}{L_a} (2r_S + r_{L_a}) i_{L_a} - \frac{1}{L_a} v_{C_a} - \frac{1}{L_a} v_{CC}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{cases} \quad (3.26)$$

para o subintervalo de chaveamento $S = 0$.

Com base nas hipóteses anteriores,

$$\mathbf{x} := \begin{bmatrix} i_{L_a} & v_{C_a} \end{bmatrix}^T \quad (3.27)$$

e

$$u := v_{CC}, \quad (3.28)$$

os conjuntos de equações (3.25) e (3.26) podem ser representados no formato (2.8), fornecendo as matrizes $\{\mathbf{A}_1, \mathbf{A}_2\} \in \mathbb{R}^{2 \times 2}$ e $\{\mathbf{B}_1, \mathbf{B}_2\} \in \mathbb{R}^{2 \times 1}$.

Por meio do procedimento de simulação definido pelas etapas do fluxograma da Figura 2.4, a implementação da simulação computacional no domínio do tempo requer a integração numérica de

$$\dot{\mathbf{x}}^{k+1} = \begin{cases} \mathbf{A}_1 \mathbf{x}^k + \mathbf{B}_1 u^{k+1}, & \text{se } S^{k+1} = 1, \\ \mathbf{A}_2 \mathbf{x}^k + \mathbf{B}_2 u^{k+1}, & \text{se } S^{k+1} = 0, \end{cases} \quad (3.29)$$

em que $S^{k+1} = d^{k+1}$, com $d = d_1 = d_4 = \bar{d}_2 = \bar{d}_3$.

3.3.1.2 Modelagem pelo Método do Modelo Equivalente de Chave

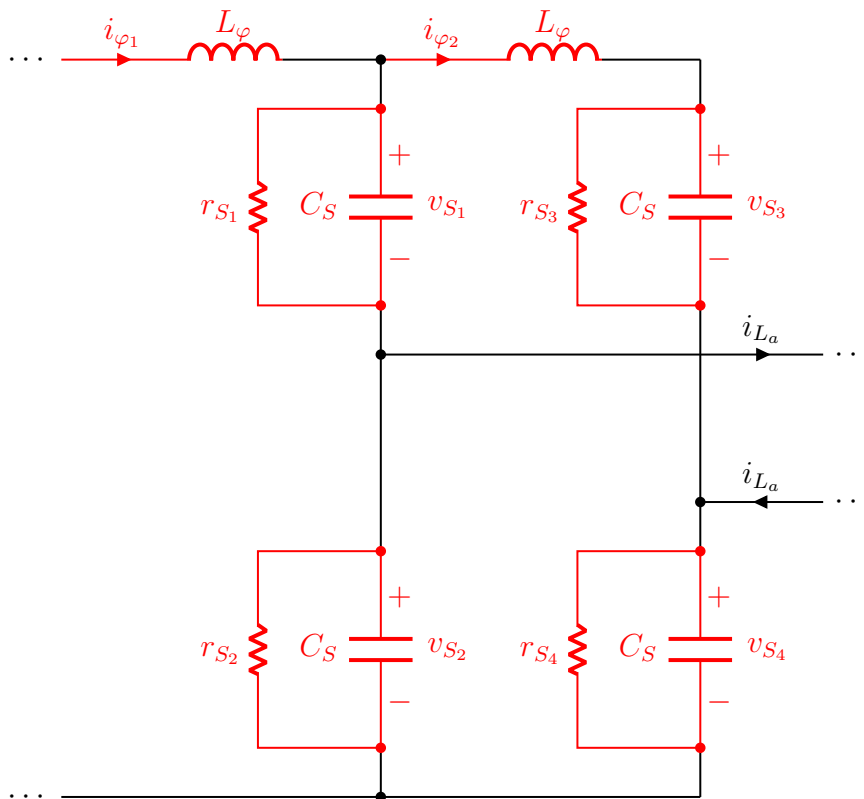
Substituindo as chaves do conversor VSI monofásico em ponte completa de acordo com o modelo equivalente da Figura 2.5, é encontrado o circuito elétrico da Figura 3.10. Note que, com o intuito de facilitar a obtenção do modelo dinâmico associado, foi adotada a inclusão de dois indutores virtuais L_φ nos ramos adjacentes às chaves S_1 e S_3 , de acordo com o procedimento apresentado na Figura 2.6.

Aplicando as leis de Kirchhoff no circuito elétrico resultante da configuração mostrada na Figura 3.8(a), porém considerando o VSI monofásico em ponte completa a partir do modelo equivalente mostrado na Figura 3.10, é possível obter o conjunto de equações de

estado dado por

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_a} = -\frac{r_{L_a}}{L_a} i_{L_a} - \frac{1}{L_a} v_{C_a} + \frac{1}{L_a} v_{S_2} - \frac{1}{L_a} v_{S_4}, \\ \frac{d}{dt} i_{\varphi_1} = -\frac{1}{L_\varphi} v_{S_1} - \frac{1}{L_\varphi} v_{S_2} + \frac{1}{L_\varphi} v_{CC}, \\ \frac{d}{dt} i_{\varphi_2} = \frac{1}{L_\varphi} v_{S_1} + \frac{1}{L_\varphi} v_{S_2} - \frac{1}{L_\varphi} v_{S_3} - \frac{1}{L_\varphi} v_{S_4}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \\ \frac{d}{dt} v_{S_1} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_1} C_S} v_{S_1}, \\ \frac{d}{dt} v_{S_2} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_2} C_S} v_{S_2}, \\ \frac{d}{dt} v_{S_3} = \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_3} C_S} v_{S_3}, \\ \frac{d}{dt} v_{S_4} = \frac{1}{C_S} i_{\varphi_2} + \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_4} C_S} v_{S_4}. \end{array} \right. \quad (3.30)$$

Figura 3.10 – Circuito elétrico do conversor VSI na configuração monofásica em ponte completa com a substituição das chaves para a aplicação da técnica do modelo equivalente de chave.



Fonte: Próprio autor.

Definindo

$$\mathbf{x} := \left[\dot{i}_{L_a} \quad i_{\varphi_1} \quad i_{\varphi_2} \quad v_{C_a} \quad v_{S_1} \quad v_{S_2} \quad v_{S_3} \quad v_{S_4} \right]^T \quad (3.31)$$

e

$$u := v_{CC}, \quad (3.32)$$

o conjunto de equações (3.30) pode ser representado no formato (2.8), o que fornece as matrizes $\mathbf{A} \in \mathbb{R}^{8 \times 8}$ e $\mathbf{B} \in \mathbb{R}^{8 \times 1}$, com \mathbf{A} variante no tempo devido às resistências das chaves r_{S_1} , r_{S_2} , r_{S_3} e r_{S_4} que alternam entre $r_{\text{ON}} = r_S$ e $r_{\text{OFF}} = r_\infty$.

O procedimento de simulação computacional no domínio do tempo é implementado conforme as etapas do fluxograma mostrado na Figura 2.7.

Conforme mostra a Tabela 2.2, as lógicas de comutação dos IGBTs são dadas por

$$S_1^{k+1} = d_1^{k+1} + (v_{S_1}^k \leq 0), \quad (3.33)$$

$$S_2^{k+1} = d_2^{k+1} + (v_{S_2}^k \leq 0), \quad (3.34)$$

$$S_3^{k+1} = d_3^{k+1} + (v_{S_3}^k \leq 0) \quad (3.35)$$

e

$$S_4^{k+1} = d_4^{k+1} + (v_{S_4}^k \leq 0). \quad (3.36)$$

Por fim, a resposta temporal de todas as variáveis de estado do modelo dinâmico é encontrada realizando a integração numérica de

$$\dot{\mathbf{x}}^{k+1} = \mathbf{A}^{k+1} \mathbf{x}^k + \mathbf{B}^{k+1} u, \quad (3.37)$$

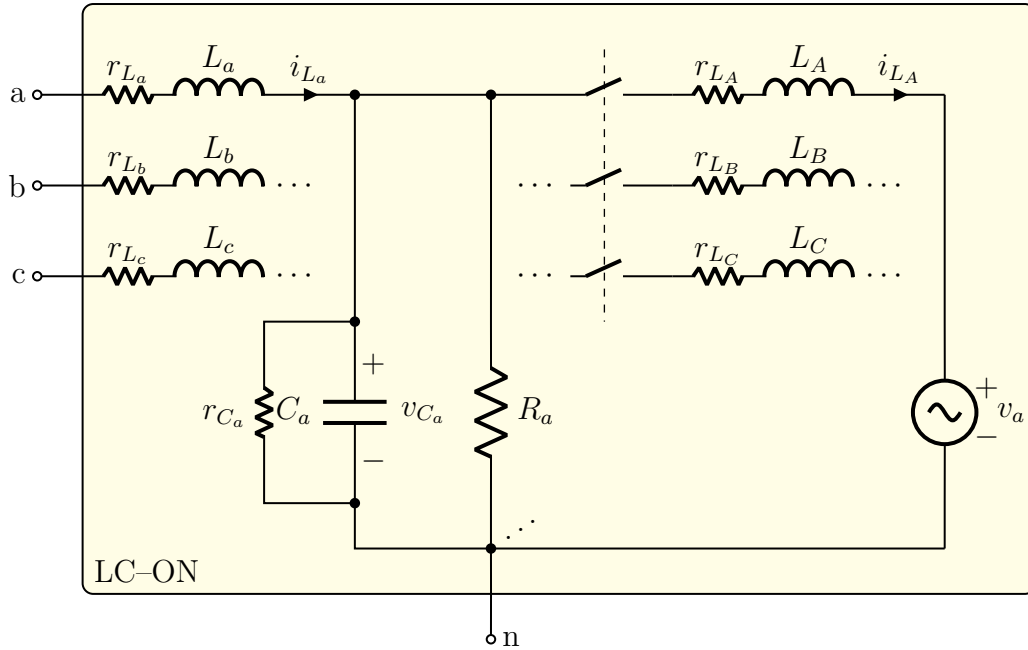
em que \mathbf{A}^{k+1} é o resultado da matriz \mathbf{A} após as atualizações das resistências r_{S_1} , r_{S_2} , r_{S_3} e r_{S_4} , produzidas pelas lógicas de comutação S_1^{k+1} , S_2^{k+1} , S_3^{k+1} e S_4^{k+1} , respectivamente, com $\mathbf{B}^{k+1} = \mathbf{B}^k = \mathbf{B}$ sendo uma matriz invariante no tempo.

3.3.2 Conversor VSI com filtro LC no Modo Conectado

A interface CA composta por um filtro LC no modo conectado é mostrada por meio da Figura 3.11. Na Figura 3.12, são ilustradas as configurações monofásica em ponte completa e trifásica do conversor VSI a partir da conexão com uma fonte de tensão primária, v_{CC} , e com o filtro LC no modo conectado da Figura 3.11.

Comparando os filtros LC nos modos ilhado e conectado, figuras 3.7 e 3.11, respectivamente, é verificado que, em termos de modelagem, a diferença reside nos estados adicionais inseridos pelas indutâncias da rede. Dessa forma, as considerações feitas para o VSI, na aplicação dos métodos dos subcircuitos e do modelo equivalente de chave para o modo ilhado, figuras 3.9 e 3.10, respectivamente, se repetem no modo conectado.

Figura 3.11 – Interface CA composta por um filtro LC alimentando uma carga R no modo conectado.



Fonte: Próprio autor.

3.3.2.1 Modelagem pelo Método dos Subcircuitos

Aplicando as leis de Kirchhoff nos subcircuitos factíveis da configuração mostrada na Figura 3.12(a), são encontrados os conjuntos de equações de estado

$$\begin{cases} \frac{d}{dt}i_{L_a} = -\frac{1}{L_a}(2r_s + r_{L_a})i_{L_a} - \frac{1}{L_a}v_{C_a} + \frac{1}{L_a}v_{CC}, \\ \frac{d}{dt}i_{L_A} = -\frac{r_{L_A}}{L_A}i_{L_A} + \frac{1}{L_A}v_{C_a} - \frac{1}{L_A}v_a, \\ \frac{d}{dt}v_{C_a} = \frac{1}{C_a}i_{L_a} - \frac{1}{C_a}i_{L_A} - \frac{1}{C_a}\left(\frac{1}{r_{C_a}} + \frac{1}{R_a}\right)v_{C_a}, \end{cases} \quad (3.38)$$

para o subintervalo de chaveamento $S = 1$, e

$$\begin{cases} \frac{d}{dt}i_{L_a} = -\frac{1}{L_a}(2r_s + r_{L_a})i_{L_a} - \frac{1}{L_a}v_{C_a} - \frac{1}{L_a}v_{CC}, \\ \frac{d}{dt}i_{L_A} = -\frac{r_{L_A}}{L_A}i_{L_A} + \frac{1}{L_A}v_{C_a} - \frac{1}{L_A}v_a, \\ \frac{d}{dt}v_{C_a} = \frac{1}{C_a}i_{L_a} - \frac{1}{C_a}i_{L_A} - \frac{1}{C_a}\left(\frac{1}{r_{C_a}} + \frac{1}{R_a}\right)v_{C_a}, \end{cases} \quad (3.39)$$

para o subintervalo de chaveamento $S = 0$.

Com base nas definições

$$\mathbf{x} := \begin{bmatrix} i_{L_a} & v_{C_a} \end{bmatrix}^T \quad (3.40)$$

e

$$\mathbf{u} := \begin{bmatrix} v_{CC} & v_a \end{bmatrix}^T, \quad (3.41)$$

os conjuntos de equações (3.38) e (3.39) podem ser representados no formato (2.8), fornecendo as matrizes $\{\mathbf{A}_1, \mathbf{A}_2\} \in \mathbb{R}^{3 \times 3}$ e $\{\mathbf{B}_1, \mathbf{B}_2\} \in \mathbb{R}^{3 \times 2}$. A simulação computacional no domínio do tempo é então implementada conforme mostrado na Seção 3.3.1.1.

3.3.2.2 Modelagem pelo Método do Modelo Equivalente de Chave

Aplicando as leis de Kirchhoff no circuito elétrico resultante da configuração mostrada na Figura 3.12(a), quando é considerado o modelo equivalente do VSI monofásico em ponte completa da Figura 3.10, é encontrado o conjunto de equações de estado dado por

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_a} = -\frac{r_{L_a}}{L_a} i_{L_a} - \frac{1}{L_a} v_{C_a} + \frac{1}{L_a} v_{S_2} - \frac{1}{L_a} v_{S_4}, \\ \frac{d}{dt} i_{L_A} = -\frac{r_{L_A}}{L_A} i_{L_A} + \frac{1}{L_A} v_{C_a} - \frac{1}{L_A} v_a, \\ \frac{d}{dt} i_{\varphi_1} = -\frac{1}{L_\varphi} v_{S_1} - \frac{1}{L_\varphi} v_{S_2} + \frac{1}{L_\varphi} v_{CC}, \\ \frac{d}{dt} i_{\varphi_2} = \frac{1}{L_\varphi} v_{S_1} + \frac{1}{L_\varphi} v_{S_2} - \frac{1}{L_\varphi} v_{S_3} - \frac{1}{L_\varphi} v_{S_4}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} i_{L_A} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \\ \frac{d}{dt} v_{S_1} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_1} C_S} v_{S_1}, \\ \frac{d}{dt} v_{S_2} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_2} C_S} v_{S_2}, \\ \frac{d}{dt} v_{S_3} = \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_3} C_S} v_{S_3}, \\ \frac{d}{dt} v_{S_4} = \frac{1}{C_S} i_{\varphi_2} + \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_4} C_S} v_{S_4}. \end{array} \right. \quad (3.42)$$

Definindo

$$\mathbf{x} := \left[i_{L_a} \quad i_{L_A} \quad i_{\varphi_1} \quad i_{\varphi_2} \quad v_{C_a} \quad v_{S_1} \quad v_{S_2} \quad v_{S_3} \quad v_{S_4} \right]^T \quad (3.43)$$

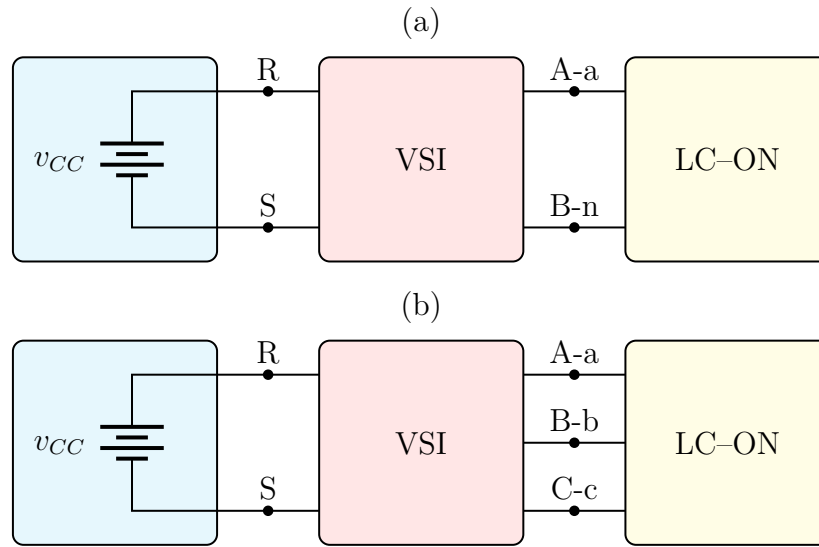
e

$$\mathbf{u} := \left[v_{CC} \quad v_a \right]^T, \quad (3.44)$$

o conjunto de equações (3.30) pode ser representado no formato (2.8), o que fornece as matrizes $\mathbf{A} \in \mathbb{R}^{8 \times 8}$ e $\mathbf{B} \in \mathbb{R}^{8 \times 1}$, com \mathbf{A} variante no tempo devido às resistências das chaves r_{S_1} , r_{S_2} , r_{S_3} e r_{S_4} , que alternam entre $r_{ON} = r_S$ e $r_{OFF} = r_\infty$.

A implementação da simulação computacional no domínio do tempo é calculada conforme as mesmas considerações da Seção 3.3.1.2.

Figura 3.12 – Estrutura do conversor VSI com filtro LC no modo conectado. Configurações (a) monofásica em ponte completa e (b) trifásica.



Fonte: Próprio autor.

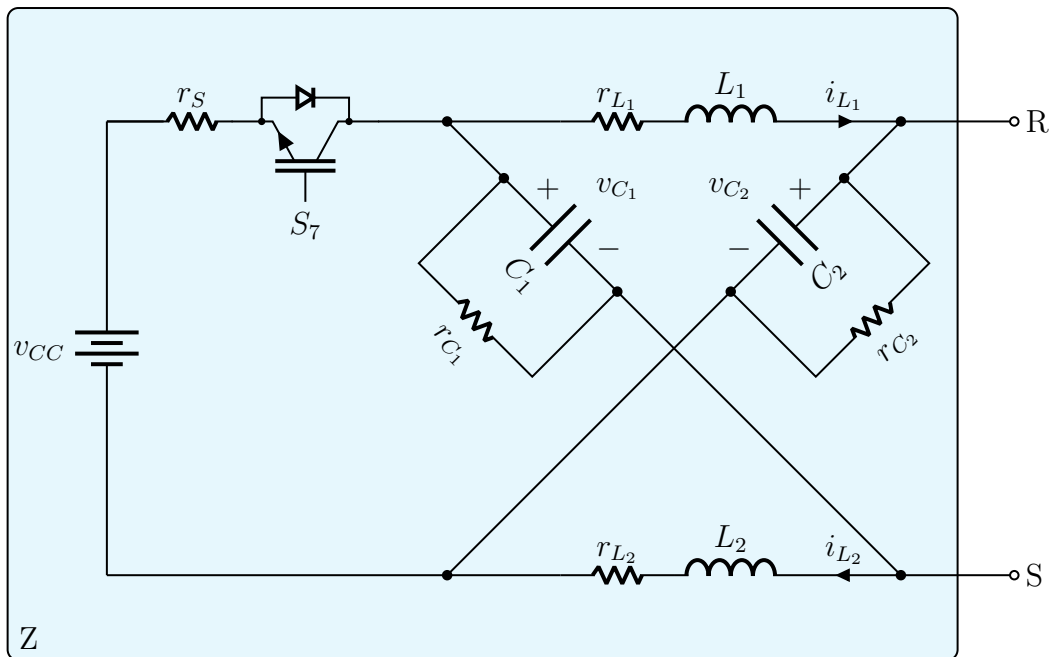
3.3.3 Conversor ZSI

O método tradicional para elevar a tensão CC de fontes primárias de energia, antes da conexão com uma interface CA, é por meio da utilização de um conversor elevador de tensão que apresente em sua saída um capacitor tanque, tal como o conversor *boost* considerado na Seção 3.2. Uma outra alternativa possível é por meio da inserção de uma malha de impedância entre a fonte primária e o ponto de acoplamento R-S, de forma a viabilizar a aplicação de curtos-circuitos consecutivos nos braços do VSI, proporcionando um alto ganho de tensão da fonte primária até os terminais de saída da malha de impedância. Em 2003, Peng propôs a utilização da malha de impedância mostrada na Figura 3.13, que ficou amplamente conhecida na literatura como malha Z. Por esse motivo, o conversor CC-CA criado a partir do acoplamento da malha Z com a entrada do VSI é comumente referenciado como ZSI.

O princípio básico de acionamento do ZSI é similar ao implementado para o VSI. Entretanto, para que a característica de elevação de tensão seja alcançada, é necessário considerar, na estratégia de chaveamento, a definição de intervalos consecutivos de tempo, conhecidos como instantes de *shoot-through*, em que todas as chaves controladas do VSI são fechadas simultaneamente por meio da atuação de um sinal lógico de comando d_{st} . Note que tal operação só é possível devido à impedância adicionada pela malha Z entre o VSI e a fonte primária.

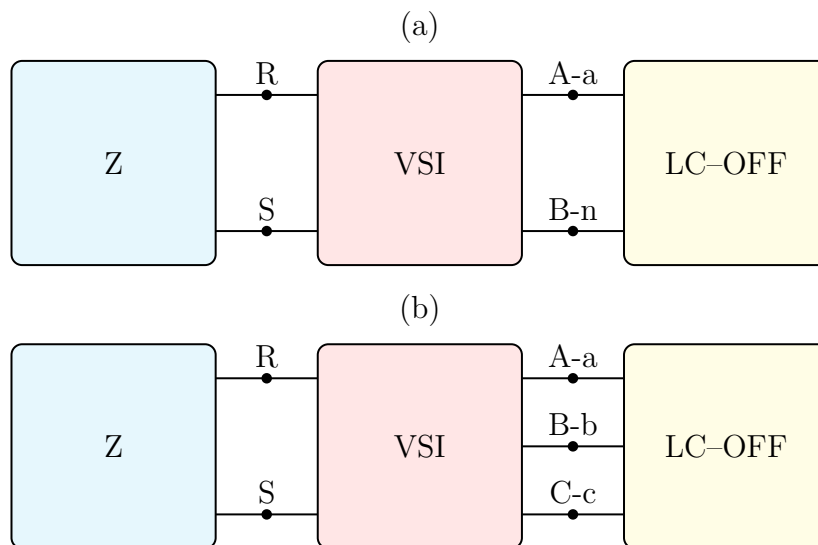
Na Figura 3.14, é mostrado o ZSI em suas versões monofásica em ponte completa e trifásica, ambas conectadas à interface CA da Figura 3.7. Para a aplicação das técnicas de modelagem em espaço de estados, é considerada a versão monofásica em ponte completa.

Figura 3.13 – Interface CC do conversor ZSI.



Fonte: Próprio autor.

Figura 3.14 – Estrutura do conversor ZSI no modo ilhado. Configurações (a) monofásica em ponte completa e (b) trifásica.



Fonte: Próprio autor.

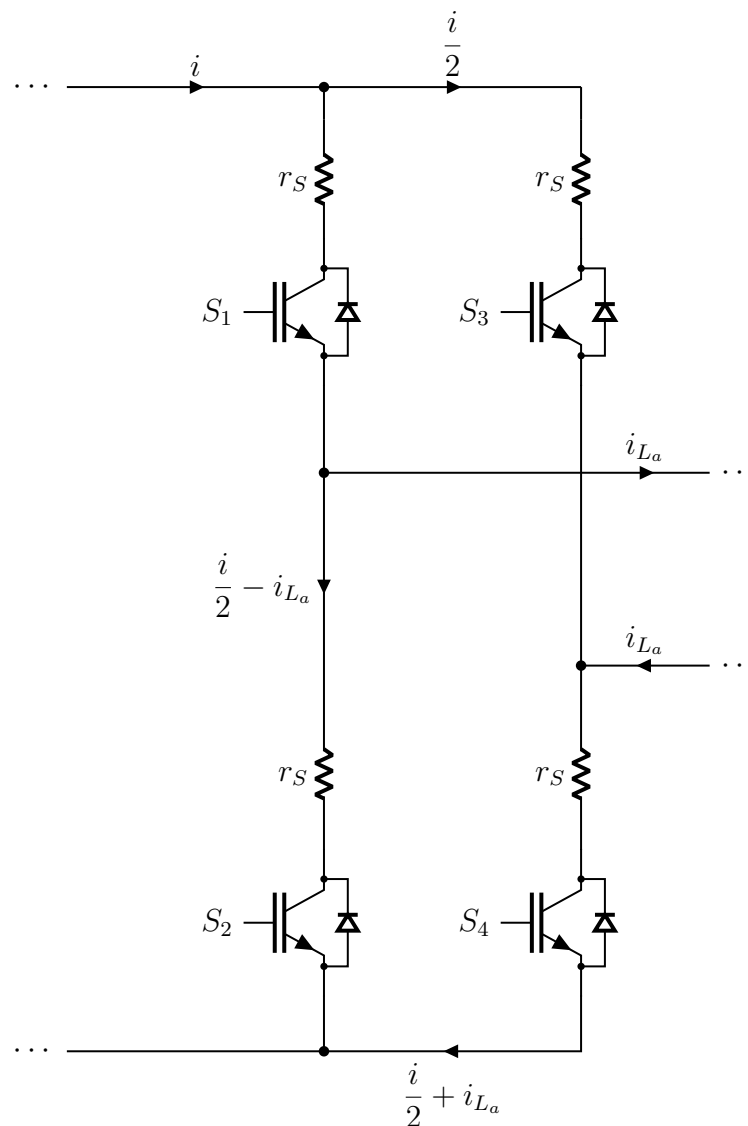
3.3.3.1 Modelagem pelo Método dos Subcircuitos

Fora do subintervalo de chaveamento que caracteriza os instantes de *shoot-through* no ZSI, ou seja, subintervalo de chaveamento $S_{st} = 0$, a chave controlada S_7 da malha Z encontra-se no modo de condução, a partir da aplicação do sinal lógico de comando $\bar{d}_{st} = 1$. Enquanto isso, o comportamento do VSI é similar ao mostrado na Figura 3.9. No subintervalo de chaveamento $S_{st} = 1$, S_7 entra no modo de bloqueio com a aplicação de $\bar{d}_{st} = 0$, e todas as chaves do VSI são fechadas simultaneamente, conforme mostrado na Figura 3.15 para a versão monofásica em ponte completa.

A partir desse critério de acionamento, é possível notar que todo o comportamento dinâmico do ZSI, pelo menos em sua configuração monofásica em ponte completa, pode ser descrito por meio de apenas três subcircuitos. Note que, no total, é possível obter $2^5 = 32$ subcircuitos com a combinação entre os estados das cinco chaves existentes.

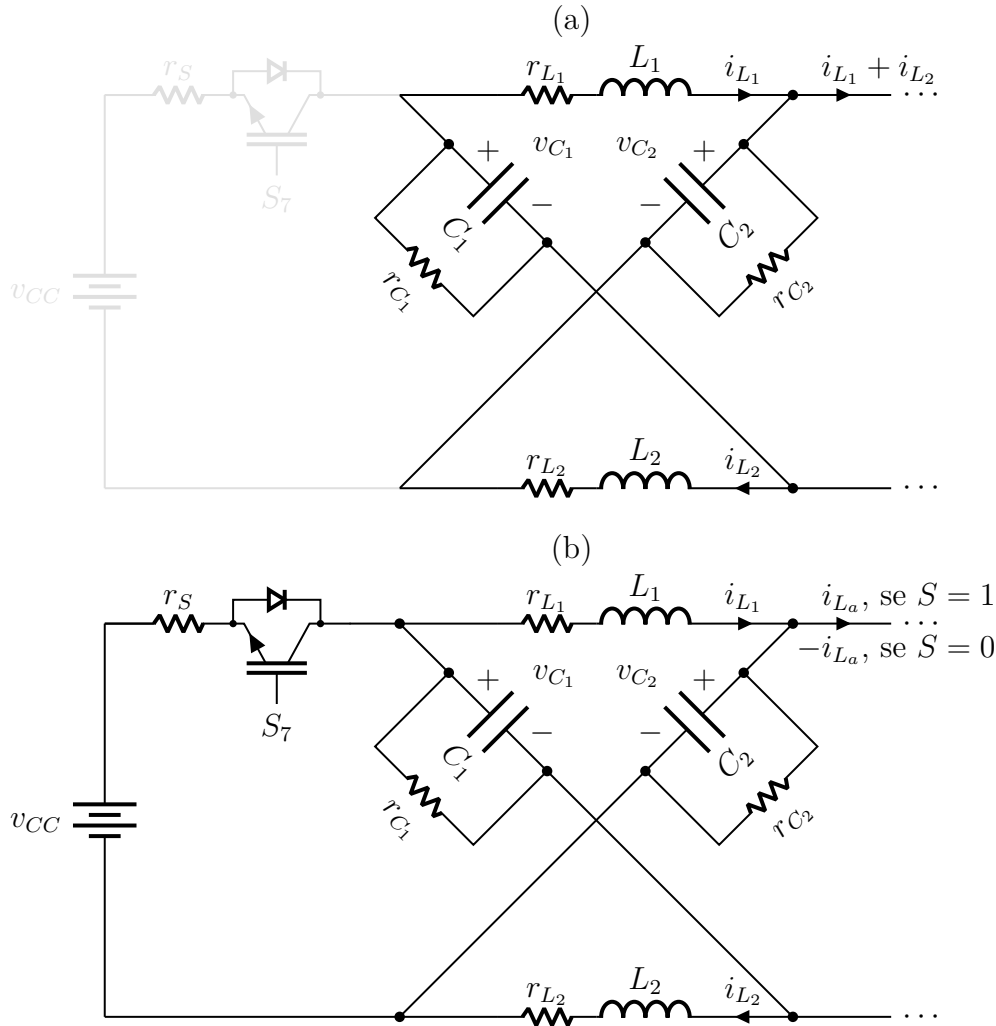
O primeiro subintervalo de chaveamento factível é, então, durante a operação em *shoot-through*, $S_{st} = 1$. Os demais são $S = 1$ e $S = 0$, porém fora do *shoot-through*, em que $S_{st} = 0$. A Figura 3.16(a) mostra o subcircuito da malha Z durante o *shoot-through*, obtido no subintervalo de chaveamento $S_{st} = 1$, enquanto que a Figura 3.16(b) mostra os respectivos subcircuitos da malha Z referentes aos outros dois subintervalos de chaveamento factíveis, $S = 1$ e $S = 0$, com $S_{st} = 0$.

Figura 3.15 – Subcircuito do VSI monofásico em ponte completa durante a operação em *shoot-through*, subintervalo de chaveamento $S_{st} = 1$.



Fonte: Próprio autor.

Figura 3.16 – Subcircuitos referentes aos subintervalos de chaveamento factíveis da malha Z. (a) $S_{st} = 1$, e (b) $S = 1$ e $S = 0$, com $S_{st} = 0$.



Fonte: Próprio autor.

Aplicando as leis de Kirchhoff nos subcircuitos factíveis da configuração mostrada na Figura 3.14(a), são encontrados os conjuntos de equações de estado

$$\begin{cases}
 \frac{d}{dt} i_{L_1} = -\frac{1}{L_1} (r_S + r_{L_1}) i_{L_1} - \frac{r_S}{L_1} i_{L_2} + \frac{r_S}{L_1} i_{L_a} + \frac{1}{L_1} v_{C_1}, \\
 \frac{d}{dt} i_{L_2} = -\frac{r_S}{L_2} i_{L_1} - \frac{1}{L_2} (r_S + r_{L_2}) i_{L_2} + \frac{r_S}{L_2} i_{L_a} + \frac{1}{L_2} v_{C_2}, \\
 \frac{d}{dt} i_{L_a} = -\frac{1}{L_a} (2r_S + r_{L_a}) i_{L_a} - \frac{1}{L_a} v_{C_a}, \\
 \frac{d}{dt} v_{C_1} = -\frac{1}{C_1} i_{L_1} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\
 \frac{d}{dt} v_{C_2} = -\frac{1}{C_2} i_{L_2} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\
 \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a},
 \end{cases} \quad (3.45)$$

para o subintervalo de chaveamento $S_{st} = 1$,

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{1}{L_1} (r_S + r_{L_1}) i_{L_1} - \frac{r_S}{L_1} i_{L_2} + \frac{r_S}{L_1} i_{L_a} - \frac{1}{L_1} v_{C_2} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_S}{L_2} i_{L_1} - \frac{1}{L_2} (r_S + r_{L_2}) i_{L_2} + \frac{r_S}{L_2} i_{L_a} - \frac{1}{L_2} v_{C_1} + \frac{1}{L_2} v_{CC}, \\ \frac{d}{dt} i_{L_a} = \frac{r_S}{L_a} i_{L_1} + \frac{r_S}{L_a} i_{L_2} - \frac{1}{L_a} (3r_S + r_{L_a}) i_{L_a} \cdots \\ \qquad \qquad \qquad \cdots + \frac{1}{L_a} v_{C_1} + \frac{1}{L_a} v_{C_2} - \frac{1}{L_a} v_{C_a} - \frac{1}{L_a} v_{CC}, \\ \frac{d}{dt} v_{C_1} = \frac{1}{C_1} i_{L_2} - \frac{1}{C_1} i_{L_a} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = \frac{1}{C_2} i_{L_1} - \frac{1}{C_2} i_{L_a} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{array} \right. \quad (3.46)$$

para o subintervalo de chaveamento $S = 1$, com $S_{st} = 0$, e

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{1}{L_1} (r_S + r_{L_1}) i_{L_1} - \frac{r_S}{L_1} i_{L_2} - \frac{r_S}{L_1} i_{L_a} - \frac{1}{L_1} v_{C_2} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_S}{L_2} i_{L_1} - \frac{1}{L_2} (r_S + r_{L_2}) i_{L_2} - \frac{r_S}{L_2} i_{L_a} - \frac{1}{L_2} v_{C_1} + \frac{1}{L_2} v_{CC}, \\ \frac{d}{dt} i_{L_a} = -\frac{r_S}{L_a} i_{L_1} - \frac{r_S}{L_a} i_{L_2} - \frac{1}{L_a} (3r_S + r_{L_a}) i_{L_a} \cdots \\ \qquad \qquad \qquad \cdots - \frac{1}{L_a} v_{C_1} - \frac{1}{L_a} v_{C_2} - \frac{1}{L_a} v_{C_a} + \frac{1}{L_a} v_{CC}, \\ \frac{d}{dt} v_{C_1} = \frac{1}{C_1} i_{L_2} + \frac{1}{C_1} i_{L_a} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = \frac{1}{C_2} i_{L_1} + \frac{1}{C_2} i_{L_a} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{array} \right. \quad (3.47)$$

para o subintervalo de chaveamento $S = 0$, com $S_{st} = 0$.

Com base nas definições

$$\mathbf{x} := [i_{L_1} \quad i_{L_2} \quad i_{L_a} \quad v_{C_1} \quad v_{C_2} \quad v_{C_a}]^T \quad (3.48)$$

e

$$u := v_{CC}, \quad (3.49)$$

os conjuntos de equações (3.45), (3.46) e (3.47) podem ser representados no formato (2.8), fornecendo as matrizes $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3\} \in \mathbb{R}^{6 \times 6}$ e $\{\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3\} \in \mathbb{R}^{6 \times 1}$.

Por meio do procedimento de simulação definido pelas etapas do fluxograma da Figura 2.4, a implementação da simulação computacional no domínio do tempo requer a integração numérica de

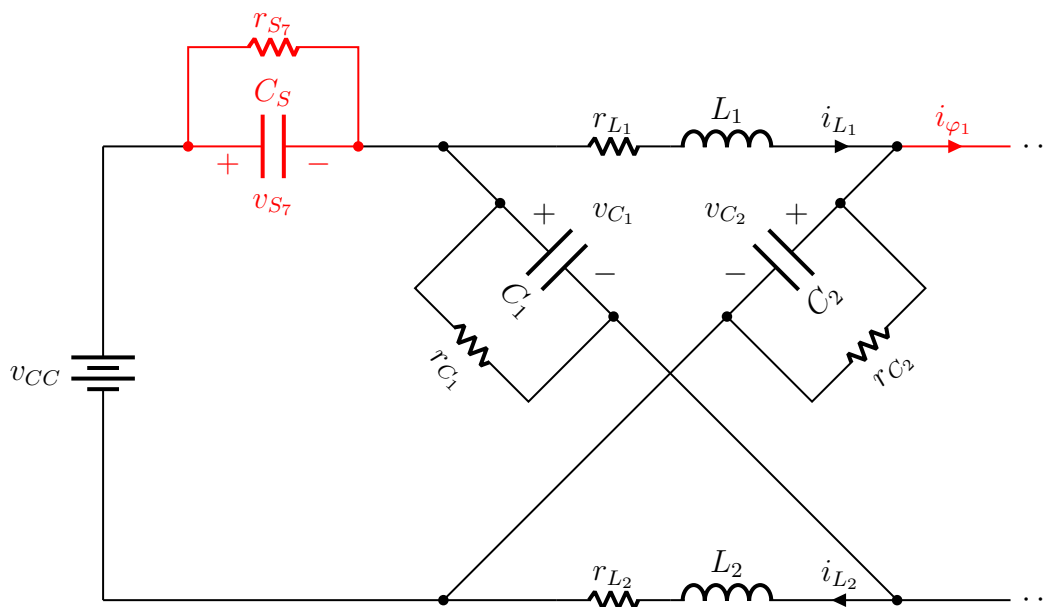
$$\dot{\mathbf{x}}^{k+1} = \begin{cases} \mathbf{A}_1 \mathbf{x}^k + \mathbf{B}_1 u^{k+1}, & \text{se } S_{st}^{k+1} = 1, \\ \mathbf{A}_2 \mathbf{x}^k + \mathbf{B}_2 u^{k+1}, & \text{se } (S^{k+1} = 1) (S_{st}^{k+1} = 0), \\ \mathbf{A}_3 \mathbf{x}^k + \mathbf{B}_3 u^{k+1}, & \text{se } (S^{k+1} = 0) (S_{st}^{k+1} = 0), \end{cases} \quad (3.50)$$

em que $S_{st}^{k+1} = d_{st}$, e $S^{k+1} = d^{k+1}$, com $d = d_1 = d_4 = \bar{d}_2 = \bar{d}_3$.

3.3.3.2 Modelagem pelo Método do Modelo Equivalente de Chave

Substituindo S_7 de acordo com o procedimento da Figura 2.5, é obtido o circuito elétrico da malha Z por modelo equivalente de chave, que pode ser visto na Figura 3.17. O circuito elétrico referente ao lado do VSI monofásico em ponte completa é descrito na Figura 3.10.

Figura 3.17 – Circuito elétrico da malha Z com a substituição da chave para a aplicação da técnica do modelo equivalente de chave.



Fonte: Próprio autor.

Aplicando as leis de Kirchhoff no circuito elétrico resultante da configuração mostrada na Figura 3.14(a), considerando os modelos equivalentes da malha Z, Figura 3.17, e do VSI monofásico em ponte completa, Figura 3.10, é encontrado o conjunto de equações de

estado calculado conforme

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{r_{L_1}}{L_1} i_{L_1} - \frac{1}{L_1} v_{C_2} - \frac{1}{L_1} v_{S_5} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_{L_2}}{L_2} i_{L_2} - \frac{1}{L_2} v_{C_1} - \frac{1}{L_2} v_{S_5} + \frac{1}{L_2} v_{CC}, \\ \frac{d}{dt} i_{L_a} = -\frac{r_{L_a}}{L_a} i_{L_a} - \frac{1}{L_a} v_{C_a} + \frac{1}{L_a} v_{S_2} - \frac{1}{L_a} v_{S_4}, \\ \frac{d}{dt} i_{\varphi_1} = \frac{1}{L_\varphi} v_{C_1} + \frac{1}{L_\varphi} v_{C_2} - \frac{1}{L_\varphi} v_{S_1} - \frac{1}{L_\varphi} v_{S_2} + \frac{1}{L_\varphi} v_{S_5}, \\ \frac{d}{dt} i_{\varphi_2} = \frac{1}{L_\varphi} v_{S_1} + \frac{1}{L_\varphi} v_{S_2} - \frac{1}{L_\varphi} v_{S_3} - \frac{1}{L_\varphi} v_{S_4}, \\ \frac{d}{dt} v_{C_1} = \frac{1}{C_1} i_{L_2} - \frac{1}{C_1} i_{\varphi_1} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = \frac{1}{C_2} i_{L_1} - \frac{1}{C_2} i_{\varphi_1} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \\ \frac{d}{dt} v_{S_1} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_1} C_S} v_{S_1}, \\ \frac{d}{dt} v_{S_2} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_2} C_S} v_{S_2}, \\ \frac{d}{dt} v_{S_3} = \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_3} C_S} v_{S_3}, \\ \frac{d}{dt} v_{S_4} = \frac{1}{C_S} i_{\varphi_2} + \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_4} C_S} v_{S_4}, \\ \frac{d}{dt} v_{S_7} = \frac{1}{C_S} i_{L_1} + \frac{1}{C_S} i_{L_2} - \frac{1}{C_S} i_{\varphi_1} - \frac{1}{r_{S_5} C_S} v_{S_5}. \end{array} \right. \quad (3.51)$$

Definindo

$$\mathbf{x} := \begin{bmatrix} i_{L_1} & i_{L_2} & i_{L_a} & i_{\varphi_1} & i_{\varphi_2} & \cdots & \cdots & v_{C_1} & v_{C_2} & v_{C_a} & v_{S_1} & v_{S_2} & v_{S_3} & v_{S_4} & v_{S_7} \end{bmatrix}^T \quad (3.52)$$

e

$$\mathbf{u} := v_{CC}, \quad (3.53)$$

o conjunto de equações (3.51) pode ser representado no formato (2.8), o que fornece as matrizes $\mathbf{A} \in \mathbb{R}^{13 \times 13}$ e $\mathbf{B} \in \mathbb{R}^{13 \times 1}$, com \mathbf{A} variante no tempo devido às resistências das chaves r_{S_1} , r_{S_2} , r_{S_3} , r_{S_4} e r_{S_7} , que se alternam entre $r_{\text{ON}} = r_S$ e $r_{\text{OFF}} = r_\infty$.

O procedimento de simulação computacional do domínio do tempo é implementado conforme as etapas do fluxograma mostrado na Figura 2.7.

Conforme mostra a Tabela 2.2, as lógicas de comutação dos IGBTs são dadas por

$$S_1^{k+1} = (d_1^{k+1} + d_{st}^{k+1}) + (v_{S_1}^k \leq 0), \quad (3.54)$$

$$S_2^{k+1} = (d_2^{k+1} + d_{st}^{k+1}) + (v_{S_2}^k \leq 0), \quad (3.55)$$

$$S_3^{k+1} = (d_3^{k+1} + d_{st}^{k+1}) + (v_{S_3}^k \leq 0), \quad (3.56)$$

$$S_4^{k+1} = (d_4^{k+1} + d_{st}^{k+1}) + (v_{S_4}^k \leq 0) \quad (3.57)$$

e

$$S_7^{k+1} = \bar{d}_{st}^{k+1} + (v_{S_4}^k \leq 0). \quad (3.58)$$

Por fim, a resposta temporal de todas as variáveis de estado do modelo dinâmico é encontrada realizando a integração numérica de

$$\dot{\mathbf{x}}^{k+1} = \mathbf{A}^{k+1} \mathbf{x}^k + \mathbf{B}^{k+1} u, \quad (3.59)$$

em que \mathbf{A}^{k+1} é o resultado da matriz \mathbf{A} após as atualizações das resistências r_{S_1} , r_{S_2} , r_{S_3} , r_{S_4} e r_{S_7} , causadas pelos resultados das lógicas de comutação S_1^{k+1} , S_2^{k+1} , S_3^{k+1} , S_4^{k+1} e S_7^{k+1} , respectivamente, e $\mathbf{B}^{k+1} = \mathbf{B}^k = \mathbf{B}$ é uma matriz invariante no tempo.

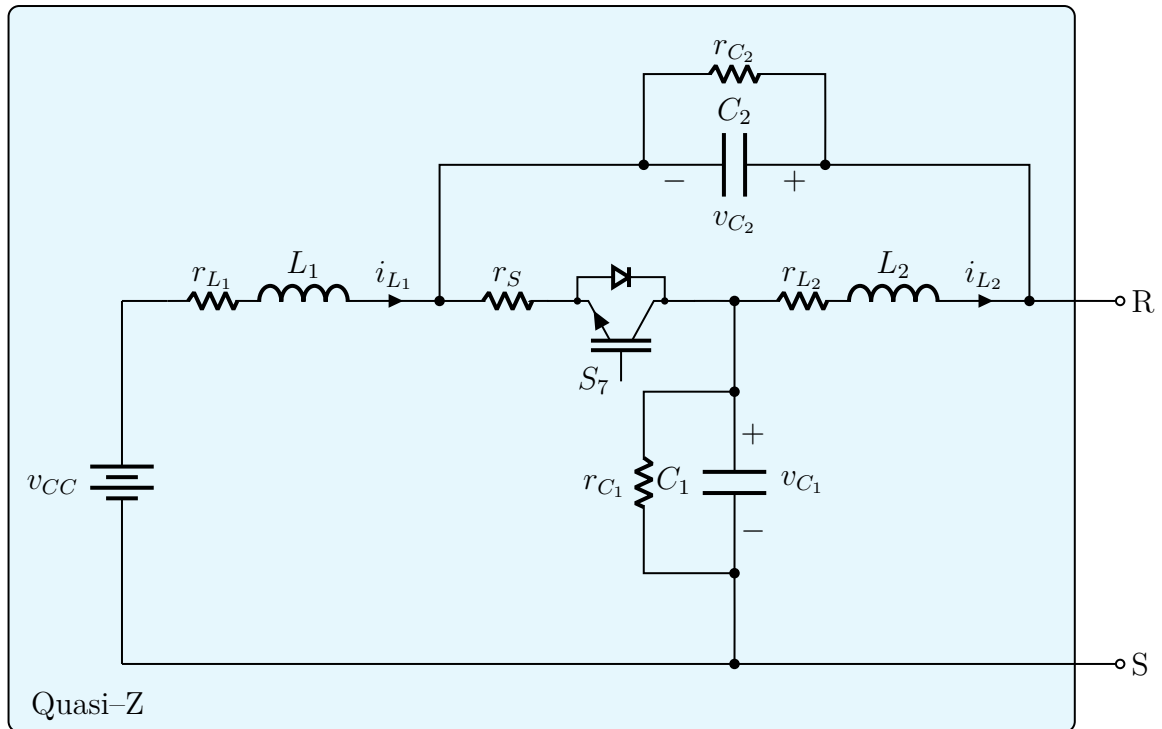
3.3.4 Conversor Quasi-ZSI

O conversor ZSI apresenta uma característica negativa muito severa em termos de qualidade de energia. A partir do acionamento condicionado à chave da malha Z, a corrente elétrica que flui pela fonte primária se torna descontínua, o que implica em um nível de ondulação elevado ($\approx 100\%$) e compromete a vida útil de baterias e de painéis fotovoltaicos. Além disso, em aspectos práticos, o conversor ZSI não pode ser inicializado sem a pré-carga dos elementos passivos da malha Z. Durante o transitório de inicialização dos capacitores C_1 e C_2 , a corrente elétrica drenada na fonte primária alcança níveis capazes de danificar a chave da malha Z quase que instantaneamente.

Visando contornar esses problemas, em 2011, Qian, Peng e Cha adotaram a reformulação da malha Z mostrada na Figura 3.18, que foi denominada como malha Quasi-Z. Em comparação com o conversor ZSI, o Quasi-ZSI apresenta os mesmos princípios de funcionamento e de acionamento, porém com o adicional de apresentar um nível de ondulação na corrente drenada da fonte primária inferior ao obtido no ZSI.

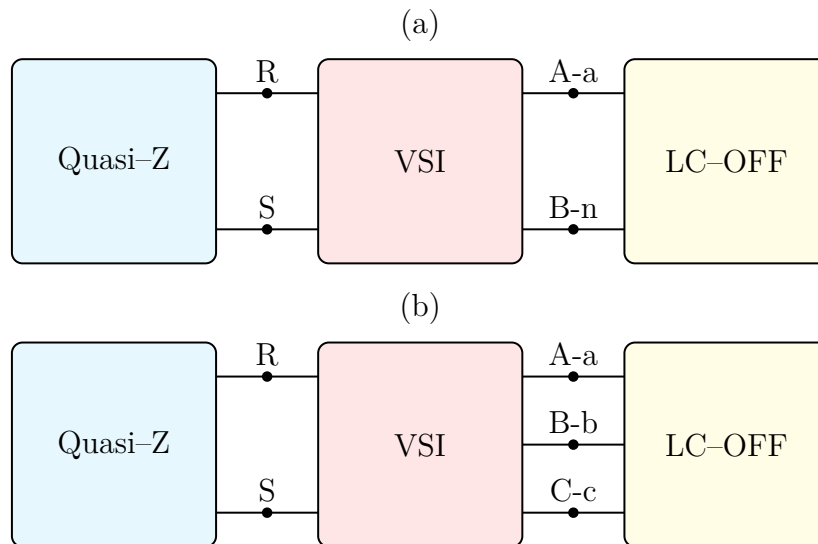
Na Figura 3.14, é ilustrado o Quasi-ZSI, em suas versões monofásica em ponte completa e trifásica, ambas conectadas à interface CA da Figura 3.7. Para a aplicação das técnicas de modelagem em espaço de estados, é considerada a versão monofásica em ponte completa.

Figura 3.18 – Interface CC do conversor Quasi-ZSI.



Fonte: Próprio autor.

Figura 3.19 – Estrutura do conversor Quasi-ZSI no modo ilhado. Configurações (a) monofásica em ponte completa e (b) trifásica.



Fonte: Próprio autor.

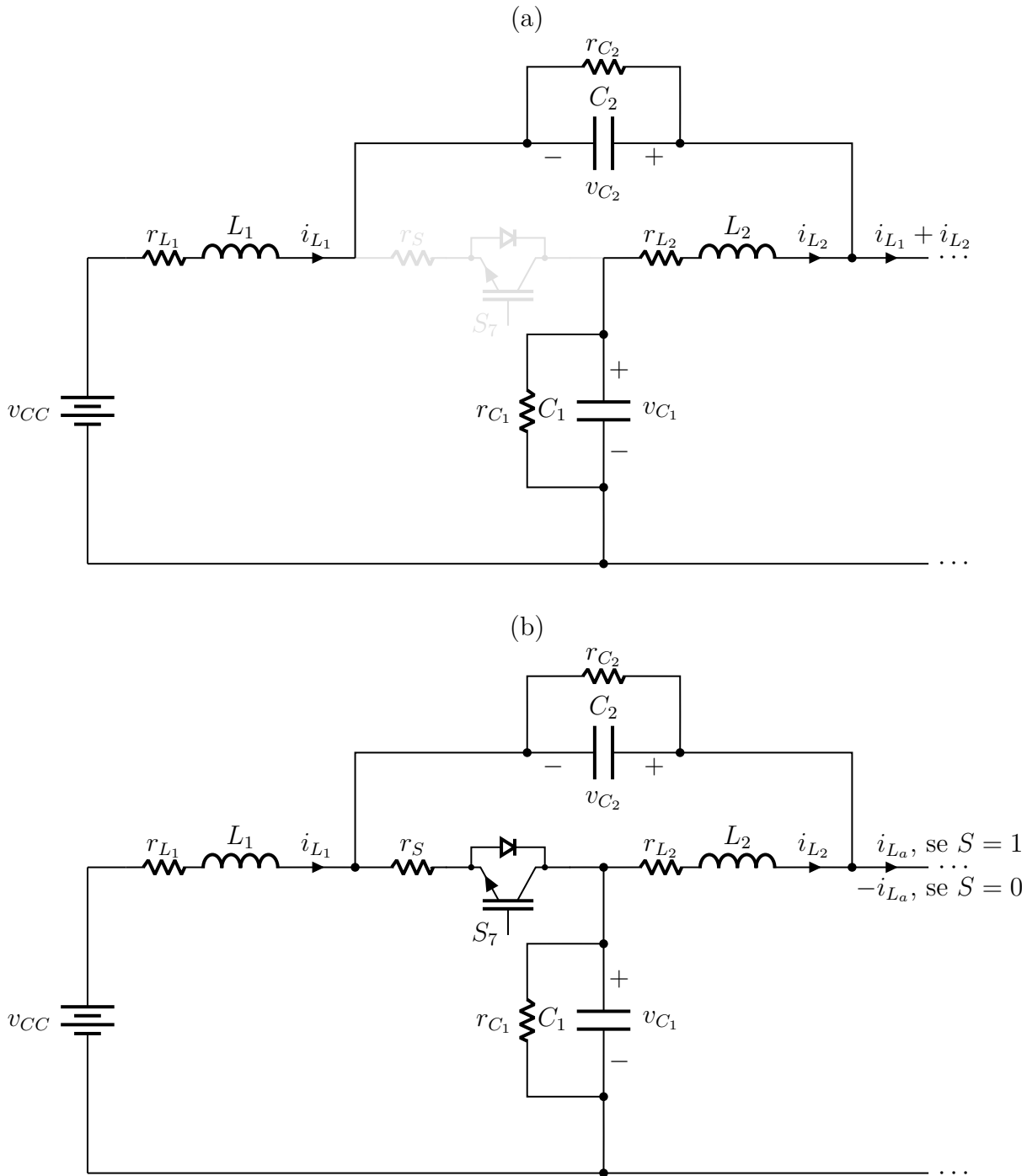
3.3.4.1 Modelagem pelo Método dos Subcircuitos

Como o Quasi-ZSI incorpora o mesmo comportamento do ZSI, é importante ressaltar que, para a aplicação do método dos subcircuitos, as considerações para a estratégia de acionamento do VSI descritas para o ZSI, na Seção 3.3.3.1, se repetem para o Quasi-ZSI.

A mudança ocorre na configuração dos subcircuitos referentes à malha de impedância. Sendo assim, a Figura 3.20 ilustra os respectivos subcircuitos da malha Quasi-Z, que são encontrados durante a operação em *shoot-through*, quando o subintervalo de chaveamento

é $S_{st} = 1$, e fora do *shoot-through*, subintervalos de chaveamento $S = 1$ e $S = 0$, enquanto $S_{st} = 0$.

Figura 3.20 – Subcircuitos referentes aos subintervalos de chaveamento factíveis da malha Quasi-Z. (a) $S_{st} = 1$, e (b) $S = 1$ e $S = 0$, com $S_{st} = 0$.



Fonte: Próprio autor.

Aplicando as leis de Kirchhoff nos subcircuitos factíveis da configuração mostrada na

Figura 3.19(a), são encontrados os conjuntos de equações de estado

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{1}{L_1} (r_S + r_{L_1}) i_{L_1} - \frac{r_S}{L_1} i_{L_2} + \frac{r_S}{L_1} i_{L_a} + \frac{1}{L_1} v_{C_2} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_S}{L_2} i_{L_1} - \frac{1}{L_2} (r_S + r_{L_2}) i_{L_2} + \frac{r_S}{L_2} i_{L_a} + \frac{1}{L_2} v_{C_1}, \\ \frac{d}{dt} i_{L_a} = -\frac{1}{L_a} (2r_S + r_{L_a}) i_{L_a} - \frac{1}{L_a} v_{C_a}, \\ \frac{d}{dt} v_{C_1} = -\frac{1}{C_1} i_{L_2} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = -\frac{1}{C_2} i_{L_1} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{array} \right. \quad (3.60)$$

para o subintervalo de chaveamento $S_{st} = 1$,

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{1}{L_1} (r_S + r_{L_1}) i_{L_1} - \frac{r_S}{L_1} i_{L_2} + \frac{r_S}{L_1} i_{L_a} - \frac{1}{L_1} v_{C_1} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_S}{L_2} i_{L_1} - \frac{1}{L_2} (r_S + r_{L_2}) i_{L_2} + \frac{r_S}{L_2} i_{L_a} - \frac{1}{L_2} v_{C_2}, \\ \frac{d}{dt} i_{L_a} = \frac{r_S}{L_a} i_{L_1} + \frac{r_S}{L_a} i_{L_2} - \frac{1}{L_a} (3r_S + r_{L_a}) i_{L_a} + \frac{1}{L_a} v_{C_1} + \frac{1}{L_a} v_{C_2} - \frac{1}{L_a} v_{C_a}, \\ \frac{d}{dt} v_{C_1} = \frac{1}{C_1} i_{L_1} - \frac{1}{C_1} i_{L_a} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = \frac{1}{C_2} i_{L_2} - \frac{1}{C_2} i_{L_a} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{array} \right. \quad (3.61)$$

para o subintervalo de chaveamento $S = 1$, com $S_{st} = 0$, e

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{1}{L_1} (r_S + r_{L_1}) i_{L_1} - \frac{r_S}{L_1} i_{L_2} - \frac{r_S}{L_1} i_{L_a} - \frac{1}{L_1} v_{C_1} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_S}{L_2} i_{L_1} - \frac{1}{L_2} (r_S + r_{L_2}) i_{L_2} - \frac{r_S}{L_2} i_{L_a} - \frac{1}{L_2} v_{C_2}, \\ \frac{d}{dt} i_{L_a} = -\frac{r_S}{L_a} i_{L_1} - \frac{r_S}{L_a} i_{L_2} - \frac{1}{L_a} (3r_S + r_{L_a}) i_{L_a} - \frac{1}{L_a} v_{C_1} - \frac{1}{L_a} v_{C_2} - \frac{1}{L_a} v_{C_a}, \\ \frac{d}{dt} v_{C_1} = \frac{1}{C_1} i_{L_1} + \frac{1}{C_1} i_{L_a} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = \frac{1}{C_2} i_{L_2} + \frac{1}{C_2} i_{L_a} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \end{array} \right. \quad (3.62)$$

para o subintervalo de chaveamento $S = 0$, com $S_{st} = 0$.

Com base nas definições

$$\mathbf{x} := [i_{L_1} \quad i_{L_2} \quad i_{L_a} \quad v_{C_1} \quad v_{C_2} \quad v_{C_a}]^T \quad (3.63)$$

e

$$u := v_{CC}, \quad (3.64)$$

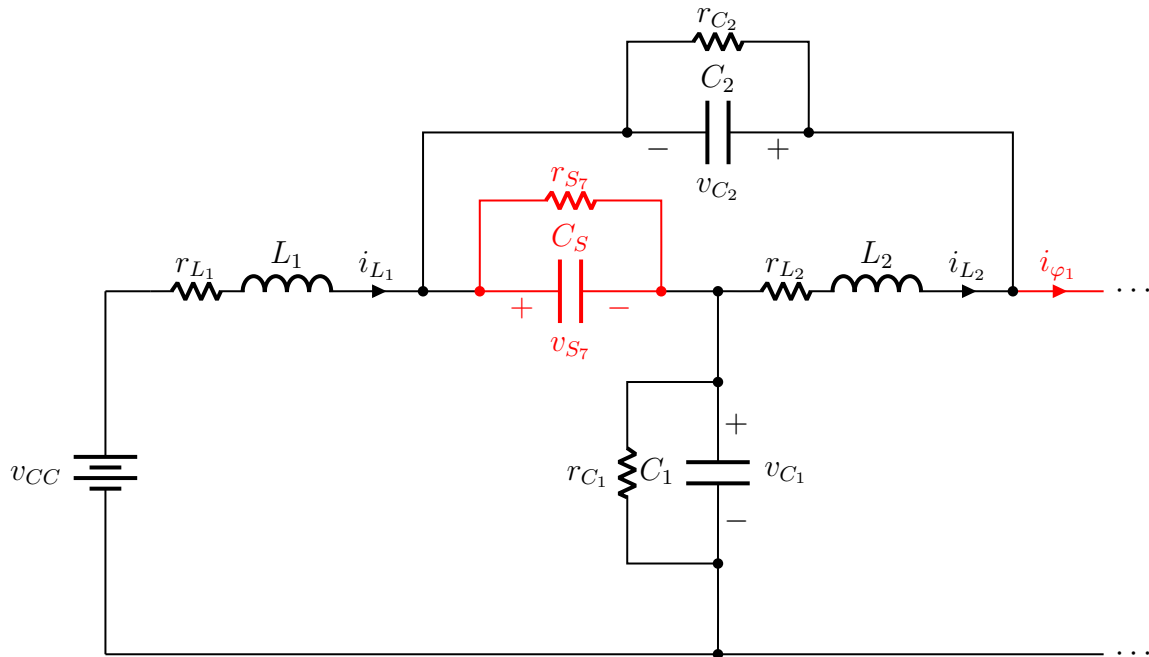
os conjuntos de equações (3.60), (3.61) e (3.62) podem ser representados no formato (2.8), fornecendo as matrizes $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3\} \in \mathbb{R}^{6 \times 6}$ e $\{\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3\} \in \mathbb{R}^{6 \times 1}$.

A implementação do procedimento de simulação computacional no domínio do tempo pode ser realizada com base nos mesmos critérios da Seção 3.3.3.1.

3.3.4.2 Modelagem pelo Método do Modelo Equivalente de Chave

Substituindo S_7 de acordo com o procedimento da Figura 2.5, é obtido o circuito elétrico da malha Quasi-Z por modelo equivalente de chave, que pode ser visto na Figura 3.21. O circuito elétrico referente ao lado do VSI monofásico em ponte completa é como mostra a Figura 3.10.

Figura 3.21 – Circuito elétrico da malha Quasi-Z com a substituição da chave para a aplicação da técnica do modelo equivalente de chave.



Fonte: Próprio autor.

Por meio da aplicação das leis de Kirchhoff no circuito elétrico resultante da configuração mostrada na Figura 3.19(a), quando são considerados os modelos equivalentes da malha Quasi-Z, Figura 3.21, e do VSI monofásico em ponte completa, Figura 3.10, é encontrado

o conjunto de equações de estado dado por

$$\left\{ \begin{array}{l} \frac{d}{dt} i_{L_1} = -\frac{r_{L_1}}{L_1} i_{L_1} - \frac{1}{L_1} v_{C_2} - \frac{1}{L_1} v_{S_5} + \frac{1}{L_1} v_{CC}, \\ \frac{d}{dt} i_{L_2} = -\frac{r_{L_2}}{L_2} i_{L_2} + \frac{1}{L_2} v_{C_1} - \frac{1}{L_2} v_{S_5}, \\ \frac{d}{dt} i_{L_a} = -\frac{r_{L_a}}{L_a} i_{L_a} - \frac{1}{L_a} v_{C_a} + \frac{1}{L_a} v_{S_2} - \frac{1}{L_a} v_{S_4}, \\ \frac{d}{dt} i_{\varphi_1} = -\frac{1}{L_\varphi} v_{C_1} + \frac{1}{L_\varphi} v_{C_2} - \frac{1}{L_\varphi} v_{S_1} - \frac{1}{L_\varphi} v_{S_2} + \frac{1}{L_\varphi} v_{S_5}, \\ \frac{d}{dt} i_{\varphi_2} = \frac{1}{L_\varphi} v_{S_1} + \frac{1}{L_\varphi} v_{S_2} - \frac{1}{L_\varphi} v_{S_3} - \frac{1}{L_\varphi} v_{S_4}, \\ \frac{d}{dt} v_{C_1} = -\frac{1}{C_1} i_{L_2} + \frac{1}{C_1} i_{\varphi_1} - \frac{1}{r_{C_1} C_1} v_{C_1}, \\ \frac{d}{dt} v_{C_2} = \frac{1}{C_2} i_{L_1} - \frac{1}{C_2} i_{\varphi_1} - \frac{1}{r_{C_2} C_2} v_{C_2}, \\ \frac{d}{dt} v_{C_a} = \frac{1}{C_a} i_{L_a} - \frac{1}{C_a} \left(\frac{1}{r_{C_a}} + \frac{1}{R_a} \right) v_{C_a}, \\ \frac{d}{dt} v_{S_1} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_1} C_S} v_{S_1}, \\ \frac{d}{dt} v_{S_2} = \frac{1}{C_S} i_{\varphi_1} - \frac{1}{C_S} i_{\varphi_2} - \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_2} C_S} v_{S_2}, \\ \frac{d}{dt} v_{S_3} = \frac{1}{C_S} i_{\varphi_2} - \frac{1}{r_{S_3} C_S} v_{S_3}, \\ \frac{d}{dt} v_{S_4} = \frac{1}{C_S} i_{\varphi_2} + \frac{1}{C_S} i_{L_a} - \frac{1}{r_{S_4} C_S} v_{S_4}, \\ \frac{d}{dt} v_{S_7} = \frac{1}{C_S} i_{L_1} + \frac{1}{C_S} i_{L_2} - \frac{1}{C_S} i_{\varphi_1} - \frac{1}{r_{S_5} C_S} v_{S_5}, \end{array} \right. \quad (3.65)$$

Definindo

$$\mathbf{x} := \begin{bmatrix} i_{L_1} & i_{L_2} & i_{L_a} & i_{\varphi_1} & i_{\varphi_2} & \cdots \\ \cdots & v_{C_1} & v_{C_2} & v_{C_a} & v_{S_1} & v_{S_2} & v_{S_3} & v_{S_4} & v_{S_7} \end{bmatrix}^T \quad (3.66)$$

e

$$\mathbf{u} := v_{CC}, \quad (3.67)$$

o conjunto de equações (3.65) pode ser representado no formato (2.8), o que fornece as matrizes $\mathbf{A} \in \mathbb{R}^{13 \times 13}$ e $\mathbf{B} \in \mathbb{R}^{13 \times 1}$, com \mathbf{A} variante no tempo devido às resistências das chaves r_{S_1} , r_{S_2} , r_{S_3} , r_{S_4} e r_{S_7} que alternam entre $r_{ON} = r_S$ e $r_{OFF} = r_\infty$.

A implementação do procedimento de simulação computacional no domínio do tempo pode ser realizada com base nos mesmos critérios da Seção 3.3.3.2.

3.4 Considerações Finais

Neste capítulo, as técnicas de modelagem discutidas no Capítulo 2, Seções 2.2 e 2.3, foram utilizadas com a finalidade de proporcionar modelos dinâmicos de conversores de potência adequados para aplicações de tempo real em HIL. Nesse sentido, foram obtidos modelos dinâmicos por meio das técnicas de modelagem via ADC e espaço de estado, métodos dos subcircuitos e do modelo equivalente de chave.

O primeiro conversor considerado foi o conversor CC–CC *boost*. Em seguida, foram consideradas quatro configurações diferentes de conversor CC–CA, as versões monofásica em ponte completa do VSI, em seus modos ilhado e conectado, do ZSI e do Quasi–ZSI. Nos conversores CC–CA, foram utilizados os modelos dinâmicos apenas a partir da técnica de modelagem em espaço de estados.

Para cada modelo dinâmico levantado, o respectivo procedimento de implementação da simulação computacional no domínio do tempo foi apresentado.

Resultados Simulados

4.1 Introdução

Quando o modelo dinâmico do conversor de potência a ser analisado está disponível, o respectivo procedimento de simulação em tempo real com HIL pode ser implementado em um *hardware*. Com base na seleção do método numérico para a solução computacional, a implementação é feita de forma específica para cada *hardware* de processamento selecionado.

Em FPGAs, a implementação em tempo real se dá por meio da elaboração de um processador específico da aplicação (ASP), que é a descrição computacional que define como o modelo dinâmico será processado (Ould-Bachir et al., 2012). Para alcançar alto desempenho de processamento, o ASP precisa explorar o paralelismo promovido pelos múltiplos blocos de processadores digitais de sinais (DSPs) disponíveis na arquitetura dos FPGAs. Dessa forma, quando o ASP é otimizado para usar os recursos de paralelismo dos FPGAs, é possível alcançar tempos de amostragem menores do que 100 ns a partir de frequências de *clock* de 200 MHz, conforme se vê em Blanchette, Ould-Bachir e David (2012) e em Hadizadeh et al. (2019).

Em MCUs de alto desempenho, em contrapartida, a implementação em tempo real não requer a elaboração de um ASP. O tempo de amostragem precisa ser estabelecido a partir da configuração de uma rotina de interrupção em que o tempo de interrupção programado define o tempo de amostragem do modelo dinâmico. Dentro da rotina de interrupção, todo o processamento do modelo dinâmico é realizado de forma sequencial, o que impossibilita atingir tempos de amostragem tão pequenos quanto os que são reportados nas implementações em FPGAs. Em Castro et al. (2019), por exemplo, é mostrada a implementação em tempo real de baixo custo de um conversor *boost*, que utiliza um tempo de amostragem de 5 μ s a partir de um MCU de alto desempenho com frequência de *clock* de 180 MHz.

Diante do exposto, este capítulo tem como finalidade mostrar, antes que a implementação em *hardware* seja devidamente tratada, o desempenho desenvolvido pelas técnicas de modelagem de circuitos elétricos chaveados, apresentadas no Capítulo 2, durante a

simulação no domínio do tempo dos conversores de potência introduzidos no Capítulo 3. Como a proposta do trabalho é abordar a utilização dos MCUs de alto desempenho, os resultados simulados serão apresentados com base em um tempo de amostragem factível em *hardwares* de processamento disponíveis no mercado de embarcados atual.

4.2 Considerações Preliminares

Com o intuito de manter a coerência com a implementação experimental do procedimento de simulação em tempo real, na sequência, algumas considerações são feitas para a seleção do *hardware* de processamento, da técnica de modelagem e do método numérico.

4.2.1 Seleção do *Hardware* de Processamento

Levando em consideração a aplicação de MCUs de alto desempenho, é importante utilizar um tempo de amostragem coerente com a capacidade de processamento do *hardware* a ser utilizado. Conforme mencionado anteriormente, em Castro et al. (2019), o tempo de amostragem utilizado foi de 5 μ s para um MCU com frequência de *clock* de 180 MHz. Nesse caso, o MCU abordado é o STM32F429 da STMicroelectronics[®], que usa processador Arm[®] Cortex[®]-M4 com arquitetura de 32 bits. No entanto, vale ressaltar que esse tempo de amostragem poderia ter sido reduzido para 2 μ s, ou ainda para 1 μ s, caso fossem explorados os recursos disponibilizados pela biblioteca CMSIS-DSP da Arm[®], que implementa diversas funções matemáticas que usam recursos de DSP inclusos no pacote de instruções dos processadores Arm[®] Cortex[®]-M.

Atualmente, o processador Arm[®] mais otimizado é o Cortex[®]-M7, e a implementação de maior desempenho desse processador é encontrada no MCU STM32H723, também da STMicroelectronics[®], que, por sua vez, possui a capacidade de alcançar uma frequência de *clock* de até 550 MHz. Por esse motivo, os MCUs de alto desempenho inclusos na família STM32H723, em que os processadores são baseados na tecnologia Arm[®] Cortex[®]-M7, são alternativas atrativas para a implementação da simulação em tempo real dos conversores de potência vistos no Capítulo 3.

4.2.2 Seleção da Técnica de Modelagem para a Implementação em Tempo Real

Considerando um tempo de amostragem dentro do intervalo $500 \text{ ns} \leq h \leq 1 \mu\text{s}$, que é uma escolha razoável para MCUs da família STM32H723, é possível avaliar as respostas dos modelos dinâmicos levantados de acordo com essa faixa de valores. Uma boa estratégia é adotar apenas os limites inferior e superior, permitindo observar a qualidade na solução numérica no melhor e no pior caso, respectivamente.

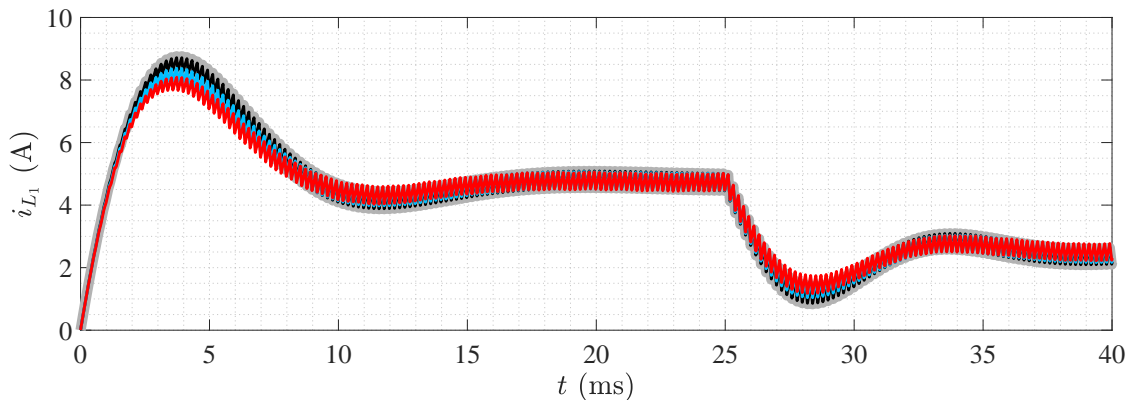
O desempenho desenvolvido pelo processamento numérico dos modelos dinâmicos pode ser avaliado de acordo com o procedimento de simulação computacional da Figura 2.9,

que permite validar as respostas dos modelos dinâmicos levantados, que são computadas a cada passo de simulação, com as respostas dos circuitos elétricos de referência.

Após realizadas diversas simulações por meio do *software* Simulink[®], da MathWorks[®], em que foram testados todos os modelos dinâmicos do Capítulo 3, foi visto que todas as técnicas de modelagem apresentadas no Capítulo 2 são suficientes para representar a dinâmica de operação dos conversores de potência considerados. Entretanto, ficou evidente que a técnica ADC e o método de modelagem por modelo equivalente de chave são inviáveis para a implementação prática em MCUs de alto desempenho, visto que o erro relativo com tais métodos, para $500 \text{ ns} \leq h \leq 1 \text{ }\mu\text{s}$, atinge valores elevados ($e_r[\bullet] > 10 \%$).

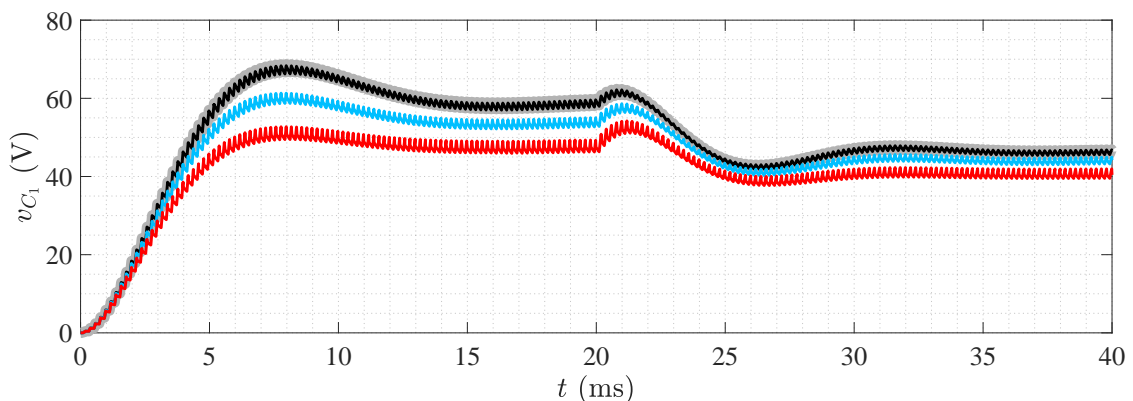
A partir da Figura 4.1, por exemplo, é visto que a resposta da corrente i_{L_1} desenvolvida pelo modelo ADC do conversor *boost* apresenta boa representação dinâmica para $h = 100 \text{ ns}$. Contudo, para $h = 500 \text{ ns}$ e $h = 1 \text{ }\mu\text{s}$, é verificado um erro considerável em relação às respostas de referência. A Figura 4.2 mostra o mesmo efeito, porém com a resposta da tensão v_{C_1} desenvolvida pelo modelo do conversor *boost* pelo método do modelo equivalente de chave com integração numérica via BDF2.

Figura 4.1 – Resposta de i_{L_1} obtida do modelo ADC do conversor *boost* para diferentes tempos de amostragem (— referência, — 100 ns, — 500 ns e — 1 μs).



Fonte: Próprio autor.

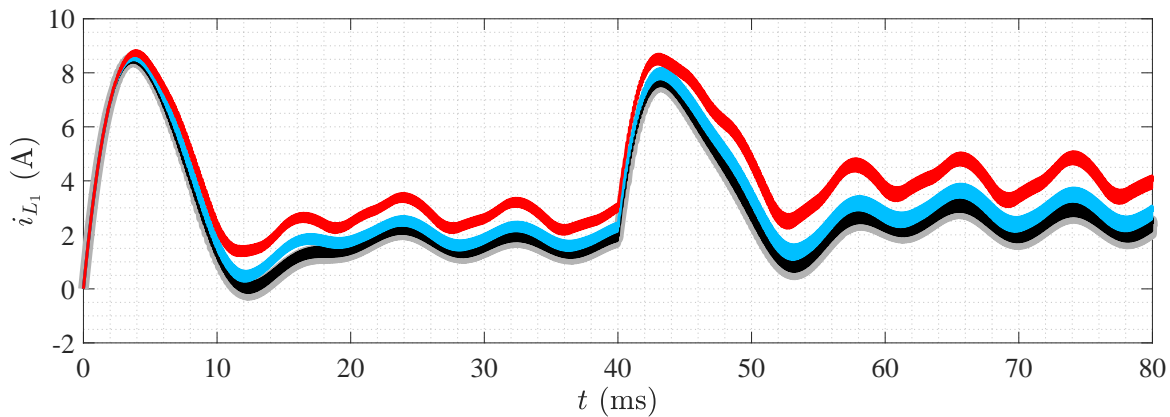
Figura 4.2 – Resposta de v_{C_1} obtida do modelo do conversor *boost* pelo método do modelo equivalente de chave com integração numérica via BDF2 para diferentes tempos de amostragem (— referência, — 100 ns, — 500 ns e — 1 μs).



Fonte: Próprio autor.

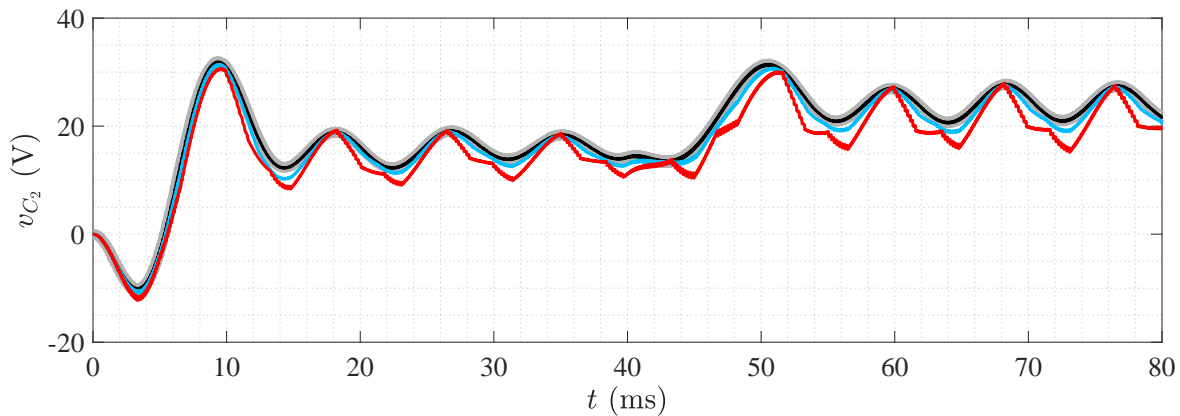
Nas Figuras 4.3 e 4.4, as respostas de i_{L_1} do ZSI e de v_{C_2} do Quasi-ZSI, desenvolvidas pelos seus respectivos modelos pelo método do modelo equivalente de chave com integração numérica via BDF2, são mostradas para $h = 100$ ns, $h = 500$ ns e $h = 1$ μ s, em que o mesmo efeito de baixa resolução numérica dentro do intervalo 500 ns $\leq h \leq 1$ μ s pode ser notado.

Figura 4.3 – Resposta de i_{L_1} obtida do modelo do ZSI monofásico em ponte completa pelo método do modelo equivalente de chave com integração numérica via BDF2 para diferentes tempos de amostragem (— referência, — 100 ns, — 500 ns e — 1 μ s).



Fonte: Próprio autor.

Figura 4.4 – Resposta de v_{C_2} obtida do modelo do Quasi-ZSI monofásico em ponte completa pelo método do modelo equivalente de chave com integração numérica via BDF2 para diferentes tempos de amostragem (— referência, — 100 ns, — 500 ns e — 1 μ s).



Fonte: Próprio autor.

Conforme pode ser constatado, o modelo ADC do conversor *boost* e os modelos monofásicos em ponte completa do ZSI e do Quasi-ZSI obtidos pelo método do modelo equivalente de chave apresentam respostas adequadas apenas quando o processamento numérico é feito de modo que $h \leq 100$ ns. Esse intervalo de tempo de amostragem é razoável para implementações em FPGAs, porém é irrealizável para MCUs de alto desempenho que, apesar de disponibilizarem elevado *clock* de processamento, resolvem os modelos dinâmicos de forma sequencial.

Vale ressaltar que, nos ensaios conduzidos com os modelos referentes aos modos ilhado e conectado do VSI monofásico em ponte completa, obtidos por meio do método do modelo equivalente de chave, foi verificado que não ocorre baixa resolução numérica dentro do intervalo $500 \text{ ns} \leq h \leq 1 \text{ }\mu\text{s}$. Isto se repete para todos os modelos obtidos segundo o método dos subcircuitos, que, inclusive, podem ser processados numericamente com tempos de amostragem superiores a $1 \text{ }\mu\text{s}$ e, ainda, manter baixo erro relativo nos resultados da integração numérica.

Como o método dos subcircuitos é uma alternativa menos problemática do ponto de vista numérico, sendo, portanto, o mais adequado para a utilização de MCUs de alto desempenho, a implementação experimental da simulação em tempo real deverá ser feita com base nesse método.

4.2.3 Seleção do Método Numérico

Nos ensaios realizados com os modelos dinâmicos obtidos pelo método do modelo equivalente de chave, foram testadas as integrações numéricas via BDF2 e TR. Em alguns casos, o método TR apresentou uma precisão numérica superior àquela encontrada por BDF2. Em outros, o método TR apresentou um desempenho muito inferior, em que foi observada a ocorrência do fenômeno de oscilações numéricas reportado em Najm (2010). Por esse motivo, o método numérico utilizado na obtenção dos gráficos mostrados nas Figuras 4.2, 4.3 e 4.4 é o BDF2.

Uma vantagem adicional em considerar o método dos subcircuitos é que, por ele não apresentar problemas de mal condicionamento, é possível realizar a integração numérica via FE, que é um método que não apresenta convergência quando utilizado para resolver modelos dinâmicos obtidos pelo método do modelo equivalente de chave.

De acordo com o que foi visto na Seção 2.4, o FE, por ser um algoritmo explícito, não requer inversão de matriz, proporcionando o menor esforço computacional possível. Em contrapartida, o método BDF2, que é um algoritmo implícito de segunda ordem, apesar de fornecer melhor precisão nos resultados da integração numérica, e sempre apresentar convergência para sistemas mal condicionados, envolve um esforço computacional muito maior. Sendo assim, na implementação prática da simulação em tempo real, em que será considerada a utilização dos modelos dinâmicos obtidos por meio do método dos subcircuitos, a integração numérica deverá ser feita via FE.

4.3 Validação de Modelos Dinâmicos

Com base nas considerações feitas na Seção 4.2, esta seção mostra com detalhes os resultados da validação de todos os modelos dinâmicos levantados no Capítulo 3 por meio do método dos subcircuitos. Dessa forma, o comportamento dinâmico obtido em

cada variável de estado é evidenciado. Para proporcionar uma métrica de avaliação de desempenho para a análise do método de integração numérica utilizado, foram encontrados os erros relativos para as variáveis de estado analisadas, cujos cálculos são realizados conforme descrito na Seção 2.5.

Os parâmetros utilizados em cada simulação são apresentados de forma individual para cada conversor de potência abordado. Tais parâmetros são os mesmos utilizados durante os ensaios mencionados na Seção 4.2.

Vale destacar que os resultados são mostrados apenas para o tempo de amostragem de 1 μ s. Por se tratar do limite superior do intervalo $500 \text{ ns} \leq h \leq 1 \mu\text{s}$, é possível verificar que $\forall h < 1 \mu\text{s}$ os resultados apresentarão melhor precisão numérica. Por esse motivo, é suficiente considerar a qualidade das respostas apenas para 1 μ s, uma vez que, caso seja possível diminuir h na implementação da simulação em tempo real, os respectivos erros relativos serão menores do que os que serão discutidos na sequência. No entanto, fica estabelecido um alvo de 1 μ s para a implementação do procedimento de simulação em HIL que, inclusive, é o tempo de amostragem utilizado nos HILs comerciais.

4.3.1 Conversor *Boost*

Os parâmetros escolhidos para a implementação do procedimento de simulação para o teste do modelo dinâmico do conversor *boost*, dado por (3.14) e (3.15), são mostrados por meio da Tabela 4.1. A simulação computacional no domínio do tempo do modelo dinâmico é executada conforme descrito na Seção 3.2.2, e a sua programação é feita por meio de um bloco Matlab-Fcn, com discretização em 1 μ s, em que as entradas são dadas pelos valores da fonte de tensão v_{CC} e do sinal lógico de comando d , e as saídas são as variáveis de estado.

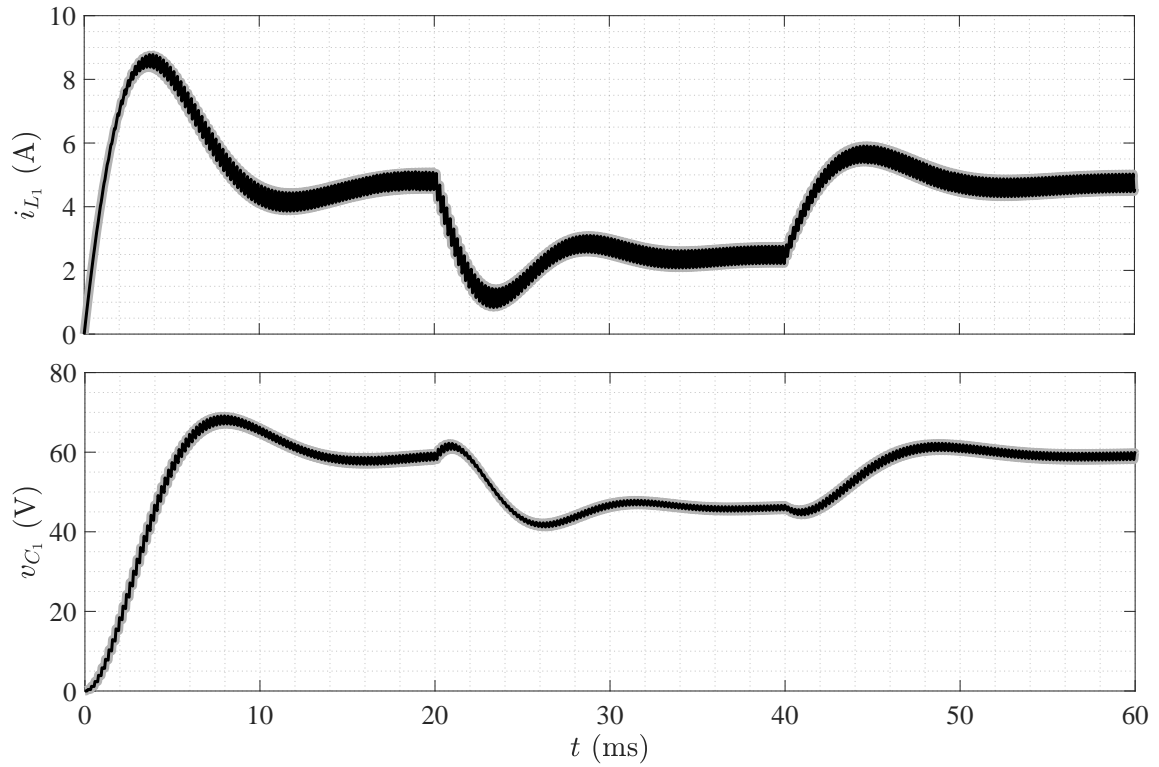
Tabela 4.1 – Parâmetros utilizados na simulação do conversor *boost*.

Parâmetro	Descrição	Valor
r_S	Perdas de condução das chaves	100 m Ω
r_{L_1}	Perda série do indutor L_1	1 Ω
r_{C_1}	Perda paralelo do capacitor C_1	100 k Ω
L_1	Indutância do indutor L_1	4 mH
C_1	Capacitância do capacitor C_1	100 μ F
R	Carga resistiva	50 Ω
f_s	Frequência de chaveamento	5 kHz
T_{sim}	Tempo de simulação	60 ms
h	Tempo de amostragem	1 μ s
f_h	Frequência de amostragem	1 MHz

Fonte: Próprio autor.

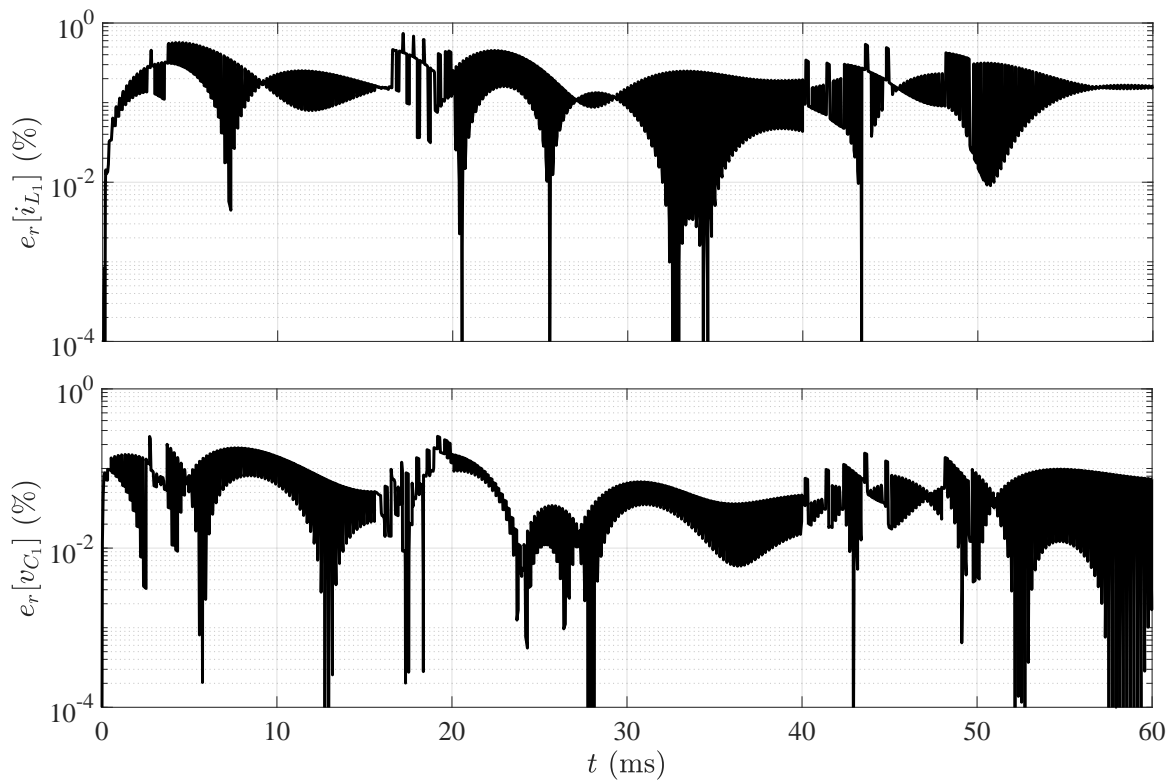
Os resultados encontrados para as variáveis de estado são mostrados por meio da Figura 4.5. Na Figura 4.6, são apresentados os respectivos erros relativos com base nos resultados do circuito elétrico de referência.

Figura 4.5 – Respostas de i_{L_1} e v_{C_1} (— referência e — modelo dinâmico) obtidas na simulação do conversor *boost*.



Fonte: Próprio autor.

Figura 4.6 – Erros relativos entre as respostas i_{L_1} e v_{C_1} do modelo dinâmico do conversor *boost* e do circuito elétrico de referência.



Fonte: Próprio autor.

O sinal lógico de comando foi produzido por meio da estratégia de acionamento por modulação PWM na frequência f_s , em que as razões cíclicas foram

$$D = \begin{cases} 75,0\%, & \text{em } t < 20 \text{ ms,} \\ 62,5\%, & \text{em } 20 \text{ ms} \leq t < 40 \text{ ms,} \\ 75,0\%, & \text{em } t \geq 40 \text{ ms,} \end{cases}$$

enquanto $v_{CC} = 20 \text{ V}$ foi mantido durante todo o tempo de simulação.

Conforme se observa, os resultados encontrados para i_{L_1} e v_{C_1} mostram que o modelo dinâmico do conversor *boost*, obtido pelo método dos subcircuitos, apresenta um comportamento dinâmico coerente em comparação com o desenvolvido pelo circuito elétrico de referência. Por meio da Figura 4.6, é visto que os erros relativos para a integração numérica via FE, com $h = 1 \mu\text{s}$, são menores do que 5%.

4.3.2 Conversor VSI no Modo Ilhado

Os parâmetros escolhidos para a implementação do procedimento de simulação para o teste do modelo dinâmico do VSI monofásico em ponte completa no modo ilhado, dado por (3.25) e (3.26), são mostrados por meio da Tabela 4.2. A simulação computacional no domínio do tempo do modelo dinâmico é executada conforme descrito na Seção 3.3.1.1, e a sua programação é feita por meio de um bloco Matlab-Fcn, com discretização em $1 \mu\text{s}$, em que as entradas são dadas pelos valores da fonte de tensão v_{CC} e do sinal lógico de comando d , e as saídas são as variáveis de estado.

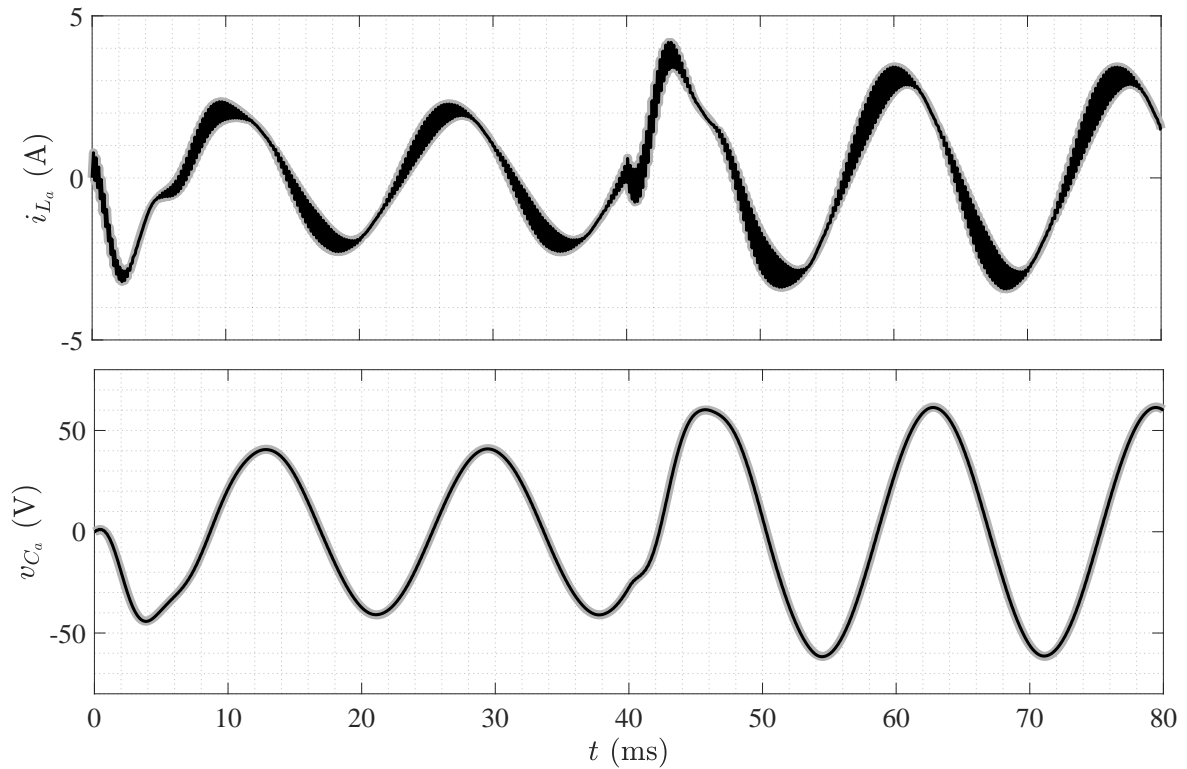
Tabela 4.2 – Parâmetros utilizados na simulação do VSI monofásico em ponte completa no modo ilhado.

Parâmetro	Descrição	Valor
r_S	Perdas de condução das chaves	100 m Ω
r_{L_a}	Perda série do indutor L_a	1 Ω
r_{C_a}	Perda paralelo do capacitor C_a	100 k Ω
L_a	Indutância do indutor L_a	5 mH
C_a	Capacitância do capacitor C_a	100 μF
R_a	Carga resistiva	30 Ω
f_s	Frequência de chaveamento	5 kHz
T_{sim}	Tempo de simulação	80 ms
h	Tempo de amostragem	1 μs
f_h	Frequência de amostragem	1 MHz

Fonte: Próprio autor.

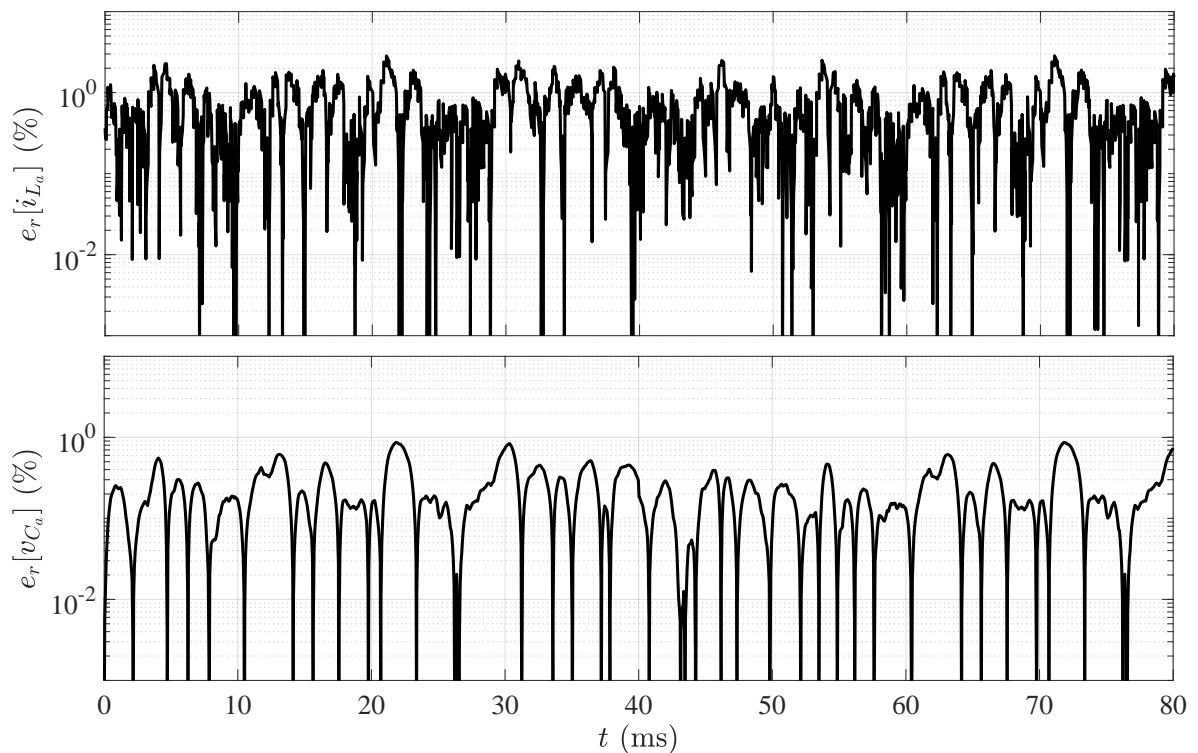
Os resultados encontrados para as variáveis de estado são mostrados por meio da Figura 4.7. Na Figura 4.8, são apresentados os respectivos erros relativos com base nos resultados calculados a partir do circuito elétrico de referência.

Figura 4.7 – Respostas de i_{L_a} e v_{C_a} (— referência e — modelo dinâmico) obtidas na simulação do VSI monofásico em ponte completa no modo ilhado.



Fonte: Próprio autor.

Figura 4.8 – Erros relativos entre as respostas i_{L_a} e v_{C_a} do modelo dinâmico do VSI monofásico em ponte completa no modo ilhado e do circuito elétrico de referência.



Fonte: Próprio autor.

O sinal lógico de comando foi produzido por meio da estratégia de acionamento a partir de um sinal PWM, criado por meio da comparação entre dois sinais unitários, uma triangular de frequência f_s e uma senoide de 60 Hz.

Com o intuito de obter uma variação dinâmica durante a simulação, foi adotado como tensão de entrada

$$v_{CC} = \begin{cases} 40 \text{ V,} & \text{em } t < 40 \text{ ms,} \\ 60 \text{ V,} & \text{em } t \geq 40 \text{ ms.} \end{cases}$$

Os resultados encontrados para i_{L_a} e v_{C_a} mostram que o modelo dinâmico do VSI monofásico em ponte completa no modo ilhado, obtido pelo método dos subcircuitos, apresenta um comportamento dinâmico coerente em comparação com o desenvolvido pelo circuito elétrico de referência. Por meio da Figura 4.8, é visto que os erros relativos para a integração numérica via FE, com $h = 1 \mu\text{s}$, são menores do que 5%.

4.3.3 Conversor VSI no Modo Conectado

Os parâmetros escolhidos para a implementação do procedimento de simulação para o teste do modelo dinâmico do VSI monofásico em ponte completa no modo conectado, dado por (3.38) e (3.39), são mostrados por meio da Tabela 4.3. A simulação computacional no domínio do tempo do modelo dinâmico é executada conforme descrito na Seção 3.3.1.1, e a sua programação é feita de forma similar ao modo ilhado, conforme mencionado na Seção 4.3.2.

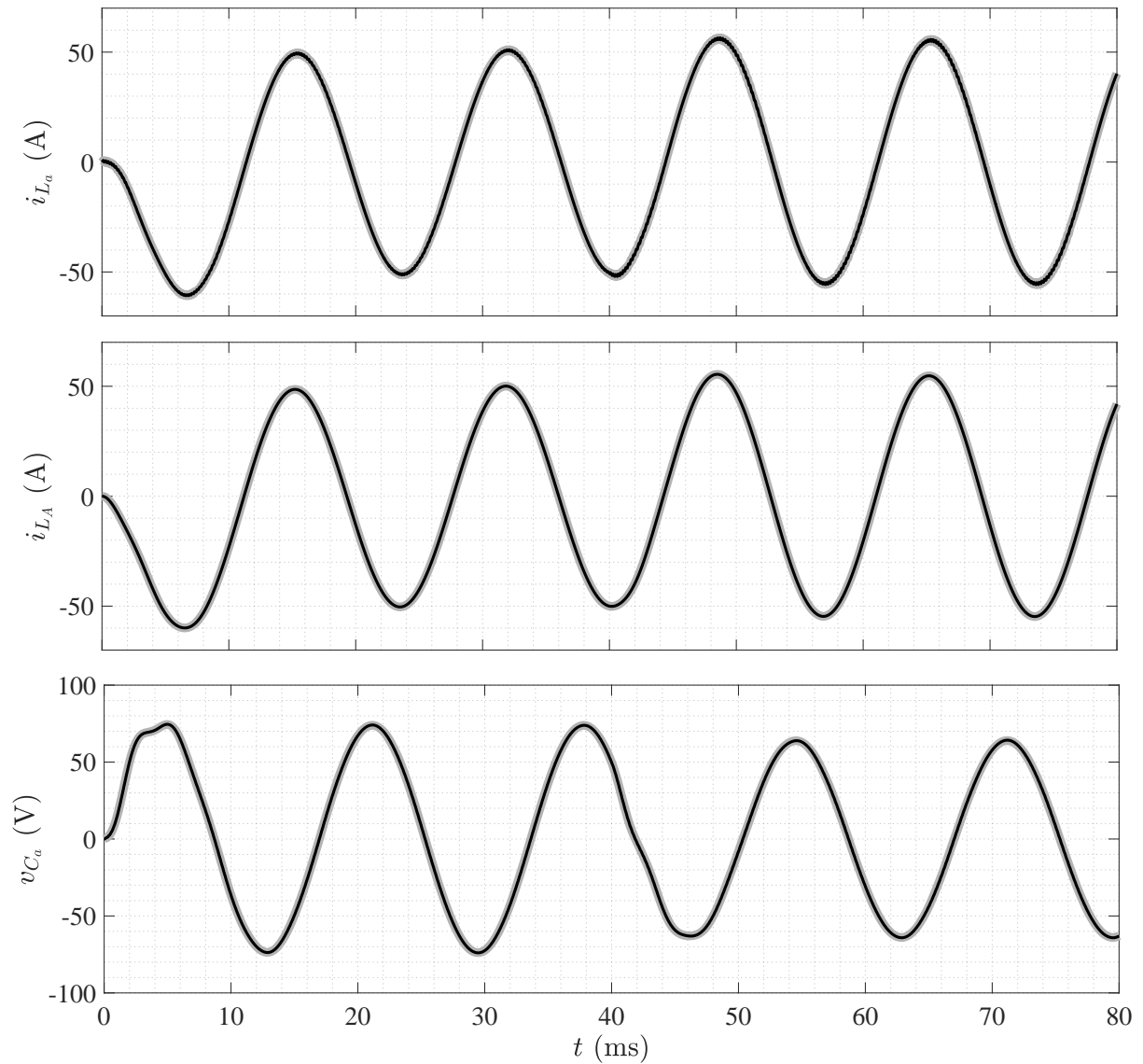
Tabela 4.3 – Parâmetros utilizados na simulação do VSI monofásico em ponte completa no modo conectado.

Parâmetro	Descrição	Valor
r_S	Perdas de condução das chaves	100 m Ω
r_{L_a}	Perda série do indutor L_a	1 Ω
r_{L_A}	Perda série do indutor L_A	1 Ω
r_{C_a}	Perda paralelo do capacitor C_a	100 k Ω
L_a	Indutância do indutor L_a	5 mH
L_A	Indutância do indutor L_A	5 mH
C_a	Capacitância do capacitor C_a	100 μF
R_a	Carga resistiva	30 Ω
f_s	Frequência de chaveamento	5 kHz
T_{sim}	Tempo de simulação	80 ms
h	Tempo de amostragem	1 μs
f_h	Frequência de amostragem	1 MHz

Fonte: Próprio autor.

Os resultados encontrados para as variáveis de estado são mostrados por meio da Figura 4.9. Na Figura 4.10, são apresentados os respectivos erros relativos com base nos resultados do circuito elétrico de referência.

Figura 4.9 – Respostas de i_{L_a} , i_{L_A} e v_{C_a} (— referência e — modelo dinâmico) obtidas na simulação do VSI monofásico em ponte completa no modo conectado.



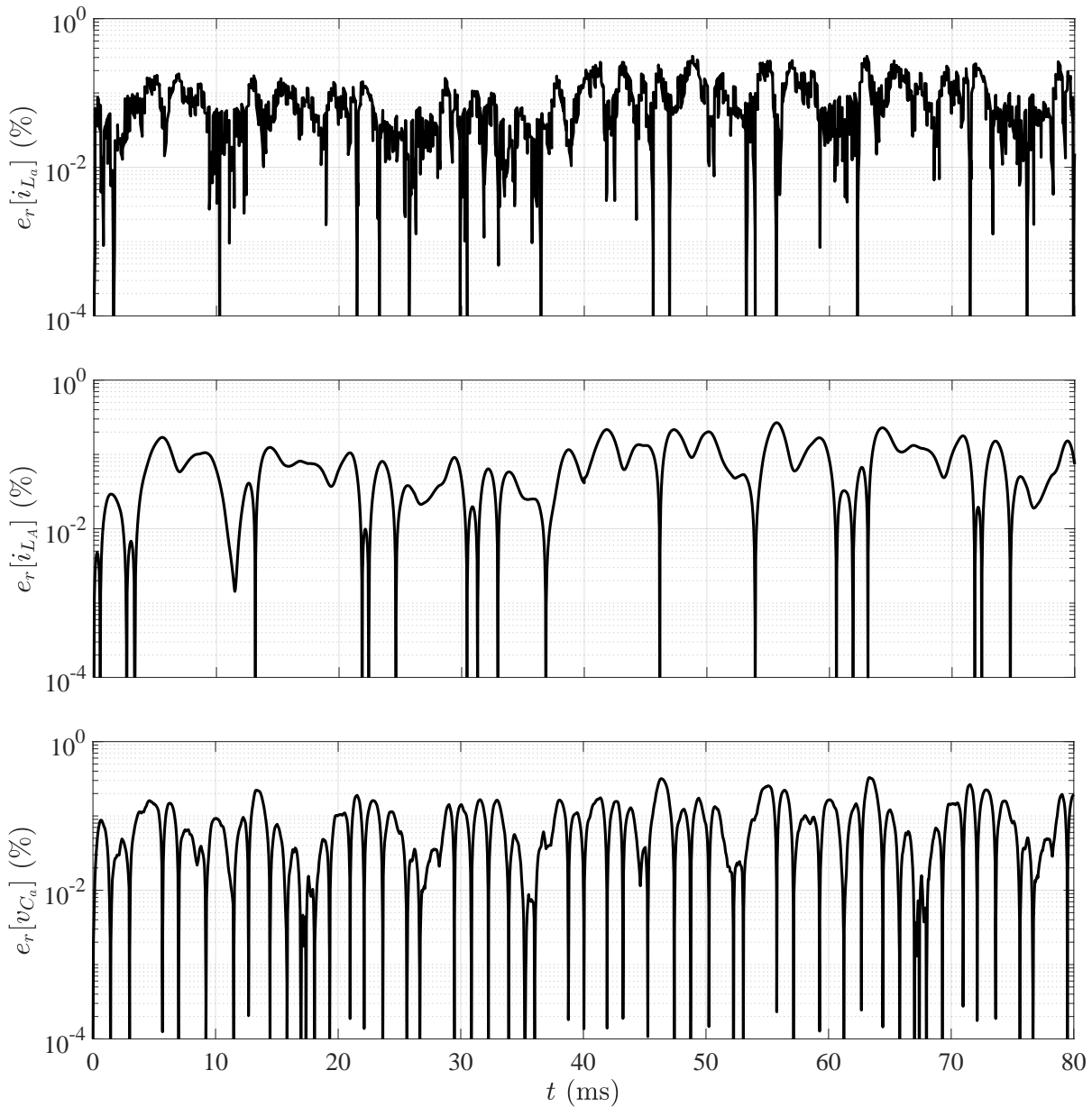
Fonte: Próprio autor.

O sinal lógico de comando foi produzido de acordo com a mesma estratégia de acionamento adotada na Seção 4.3.2 para o modo ilhado. Da mesma forma, no teste foi utilizada a mesma variação dinâmica da tensão de entrada v_{CC} .

Conforme pode ser observado, a partir da análise dos resultados encontrados para i_{L_a} , i_{L_A} e v_{C_a} , é possível ver que o modelo dinâmico do VSI monofásico em ponte completa no modo conectado, obtido pelo método dos subcircuitos, assim como o modo ilhado, apresenta um comportamento dinâmico coerente em comparação com o desenvolvido pelo circuito elétrico de referência.

A partir da Figura 4.10, é possível perceber que os erros relativos para a integração numérica via FE, com $h = 1 \mu\text{s}$, são menores do que 5%.

Figura 4.10 – Erros relativos entre as respostas i_{L_a} , i_{L_A} e v_{C_a} do modelo dinâmico do VSI monofásico em ponte completa no modo conectado e do circuito elétrico de referência.



Fonte: Próprio autor.

4.3.4 Conversor ZSI

Por meio da Tabela 4.4, são mostrados os parâmetros que foram escolhidos para a implementação do procedimento de simulação para o teste do modelo dinâmico do ZSI monofásico em ponte completa, de acordo com (3.45), (3.46) e (3.47). A simulação computacional no domínio do tempo do modelo dinâmico é executada em conformidade com a descrição fornecida na Seção 3.3.3.1, e a sua programação é feita a partir de um bloco Matlab-Fcn, com discretização em $1 \mu\text{s}$, em que as entradas são dadas pelos valores da fonte de tensão v_{CC} e dos sinais lógicos de comando d e d_{st} . De acordo com a descrição apresentada na Seção 3.3.3, vale lembrar que d é utilizado para a estratégia de acionamento

do VSI monofásico em ponte completa, enquanto que d_{st} ativa os instantes de *shoot-through*.

Tabela 4.4 – Parâmetros utilizados na simulação do ZSI monofásico em ponte completa.

Parâmetro	Descrição	Valor
r_S	Perdas de condução das chaves	100 m Ω
r_{L_1}	Perda série do indutor L_1	1 Ω
r_{L_2}	Perda série do indutor L_2	1 Ω
r_{L_a}	Perda série do indutor L_a	1 Ω
r_{C_1}	Perda paralelo do capacitor C_1	100 k Ω
r_{C_2}	Perda paralelo do capacitor C_2	100 k Ω
r_{C_a}	Perda paralelo do capacitor C_a	100 k Ω
L_1	Indutância do indutor L_1	4 mH
L_2	Indutância do indutor L_2	4 mH
L_a	Indutância do indutor L_a	2 mH
C_1	Capacitância do capacitor C_1	500 μ F
C_2	Capacitância do capacitor C_2	500 μ F
C_a	Capacitância do capacitor C_a	10 μ F
R_a	Carga resistiva	30 Ω
f_s	Frequência de chaveamento	5 kHz
T_{sim}	Tempo de simulação	60 ms
h	Tempo de amostragem	1 μ s
f_h	Frequência de amostragem	1 MHz

Fonte: Próprio autor.

Os resultados encontrados para as variáveis de estado da interface CC são mostrados por meio da Figura 4.11. Na Figura 4.12, são ilustrados os respectivos erros relativos com base nos resultados do circuito elétrico de referência.

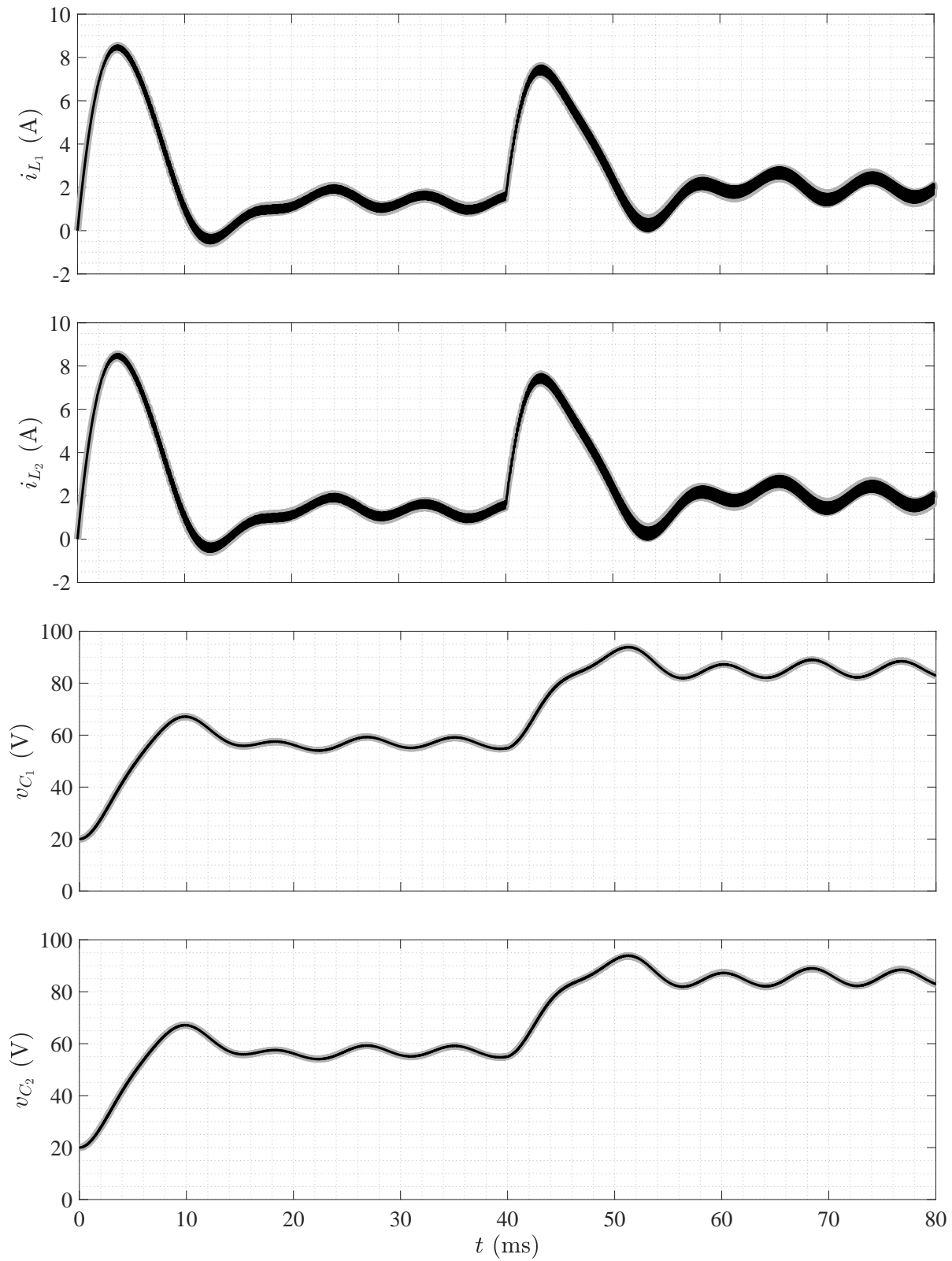
A estratégia de acionamento do ZSI foi feita com base em uma técnica que leva o nome de *maximum boost*, em que o VSI é acionado por meio de um sinal lógico de comando formado pela soma lógica (operação OR) de d com d_{st} .

No *maximum boost*, o sinal lógico de comando d é produzido a partir de um sinal PWM, criado por meio da comparação entre uma triangular de amplitude unitária com frequência f_s e uma senoide de 60 Hz, o que é similar ao que foi feito para as simulações das seções 4.3.2 e 4.3.3. A diferença agora é que a amplitude da senoide é variável, podendo ser menor ou igual a da triangular.

O sinal lógico de comando d_{st} é então criado com base na definição de duas bandas simétricas, uma superior e outra inferior, em que os seus valores modulam a amplitude da senoide. Ao alterar essas bandas, a faixa de tempo que configura o *shoot-through* pode ser incrementada ou decrementada, permitindo elevar ou reduzir, respectivamente, a elevação de tensão no pré-estágio CC da malha de impedância. Mais detalhes podem ser encontrados em Peng, Shen e Qian (2005).

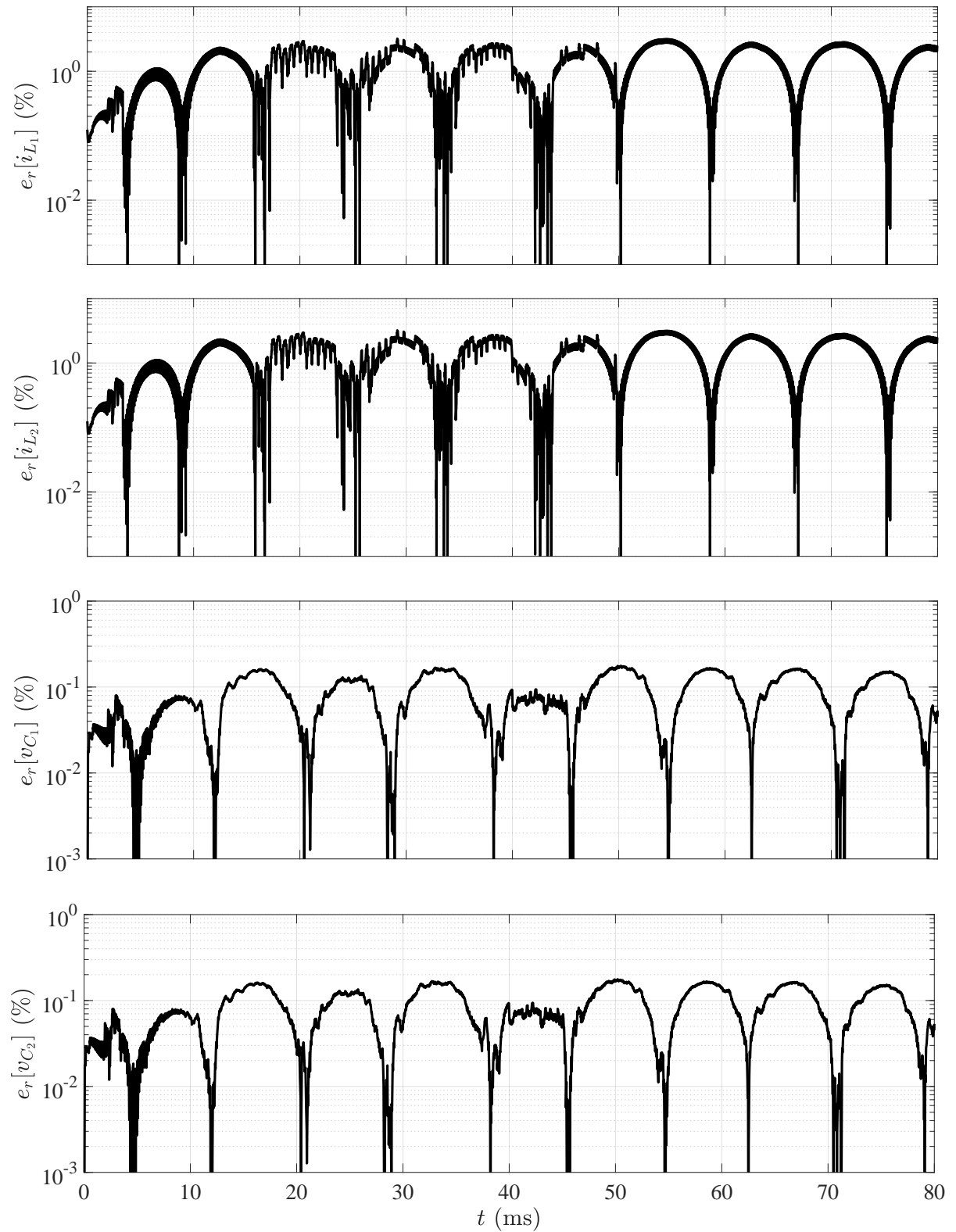
Em termos da fonte de tensão v_{CC} , foi adotada a mesma variação dinâmica mostrada na Seção 4.3.2.

Figura 4.11 – Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} (— referência e — modelo dinâmico) obtidas na simulação do ZSI monofásico em ponte completa.



Fonte: Próprio autor.

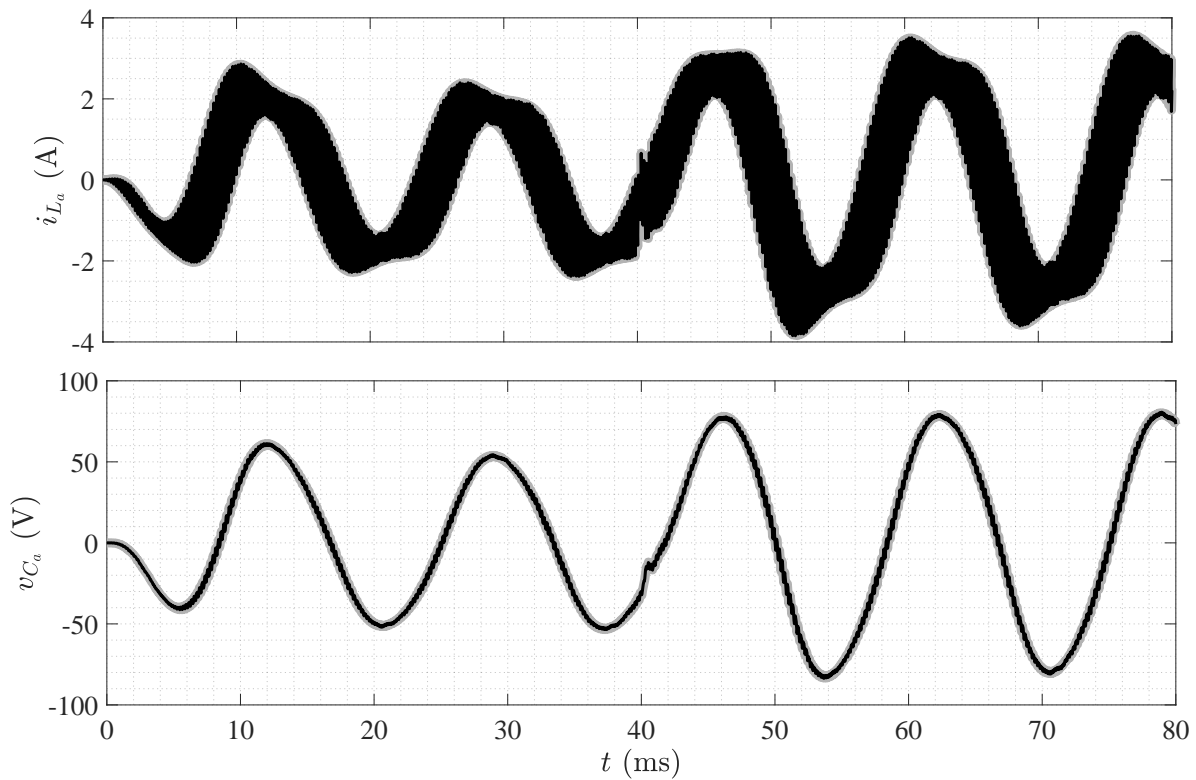
Figura 4.12 – Erros relativos entre as respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} do modelo dinâmico do ZSI monofásico em ponte completa e do circuito elétrico de referência.



Fonte: Próprio autor.

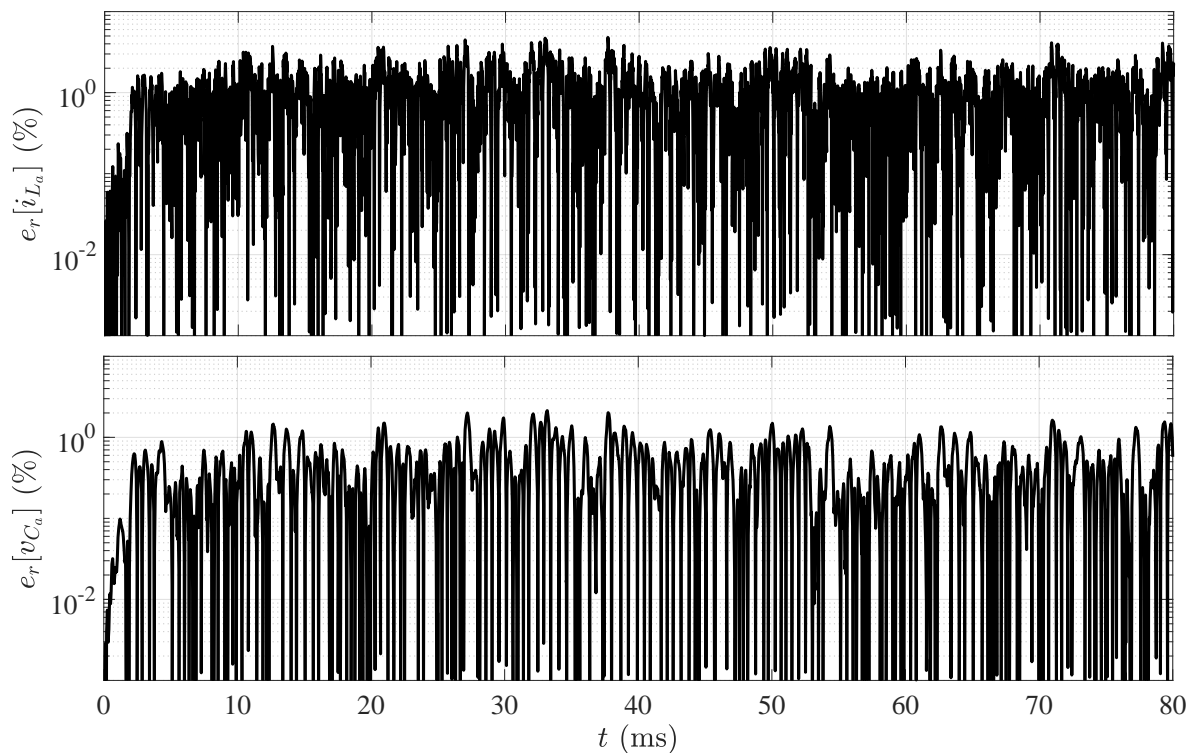
Os resultados encontrados para as variáveis de estado da interface CA são mostrados por meio da Figura 4.13. Na Figura 4.14, são apresentados os respectivos erros relativos com base nos resultados do circuito elétrico de referência.

Figura 4.13 – Respostas de i_{L_a} e v_{C_a} (— referência e — modelo dinâmico) obtidas na simulação do ZSI monofásico em ponte completa.



Fonte: Próprio autor.

Figura 4.14 – Erros relativos entre as respostas de i_{L_a} e v_{C_a} do modelo dinâmico do ZSI monofásico em ponte completa e do circuito elétrico de referência.



Fonte: Próprio autor.

Conforme pode ser observado, a partir da análise dos resultados encontrados para i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} , referentes à interface CC, e para i_{L_a} e v_{C_a} , referentes à interface CA, é visto que o modelo dinâmico do ZSI monofásico em ponte completa, obtido pelo método dos subcircuitos, apresenta um comportamento dinâmico coerente em comparação com o desenvolvido pelo circuito elétrico de referência.

Com base nos resultados das figuras 4.12 e 4.14, é possível se perceber que os erros relativos para a integração numérica via FE, com $h = 1 \mu\text{s}$, são razoáveis ($< 5\%$).

4.3.5 Conversor Quasi-ZSI

Por meio da Tabela 4.5, são mostrados os parâmetros que foram escolhidos para a implementação do procedimento de simulação para o teste do modelo dinâmico do Quasi-ZSI monofásico em ponte completa, dado por (3.60), (3.61) e (3.62). A simulação computacional no domínio do tempo do modelo dinâmico é executada em conformidade com a descrição fornecida na Seção 3.3.3.1, e a sua programação é feita de forma similar ao que foi feito para o ZSI, conforme descrito na Seção 4.3.4.

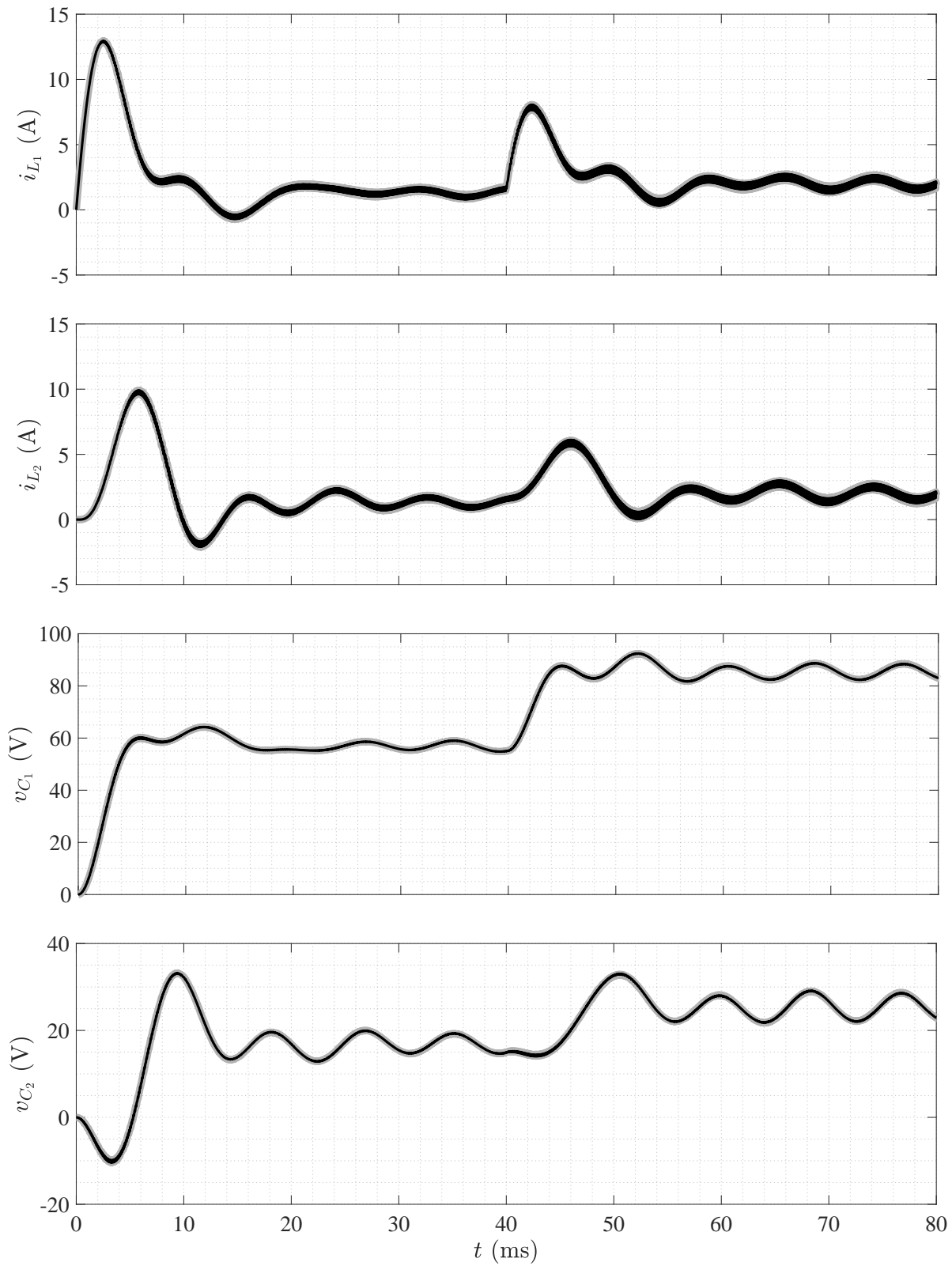
Tabela 4.5 – Parâmetros utilizados na simulação do Quasi-ZSI monofásico em ponte completa.

Parâmetro	Descrição	Valor
r_S	Perdas de condução das chaves	100 m Ω
r_{L_1}	Perda série do indutor L_1	1 Ω
r_{L_2}	Perda série do indutor L_2	1 Ω
r_{L_a}	Perda série do indutor L_a	1 Ω
r_{C_1}	Perda paralelo do capacitor C_1	100 k Ω
r_{C_2}	Perda paralelo do capacitor C_2	100 k Ω
r_{C_a}	Perda paralelo do capacitor C_a	100 k Ω
L_1	Indutância do indutor L_1	4 mH
L_2	Indutância do indutor L_2	4 mH
L_a	Indutância do indutor L_a	2 mH
C_1	Capacitância do capacitor C_1	500 μF
C_2	Capacitância do capacitor C_2	500 μF
C_a	Capacitância do capacitor C_a	10 μF
R_a	Carga resistiva	30 Ω
f_s	Frequência de chaveamento	5 kHz
T_{sim}	Tempo de simulação	60 ms
h	Tempo de amostragem	1 μs
f_h	Frequência de amostragem	1 MHz

Fonte: Próprio autor.

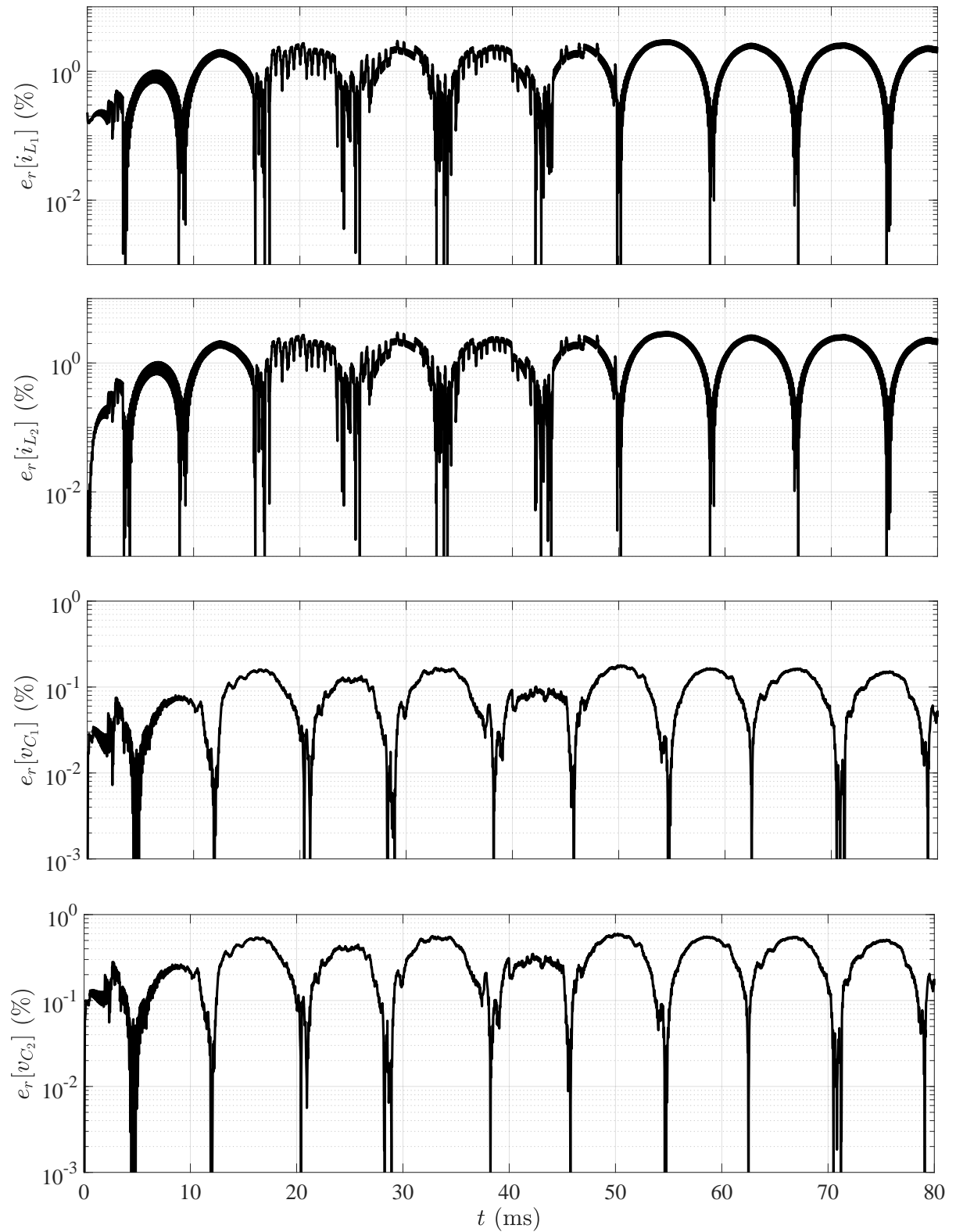
Os resultados encontrados para as variáveis de estado da interface CC são mostrados na Figura 4.15. Na Figura 4.16, são apresentados os respectivos erros relativos com base nos resultados do circuito elétrico de referência.

Figura 4.15 – Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} (— referência e — modelo dinâmico) obtidas na simulação do Quasi-ZSI monofásico em ponte completa.



Fonte: Próprio autor.

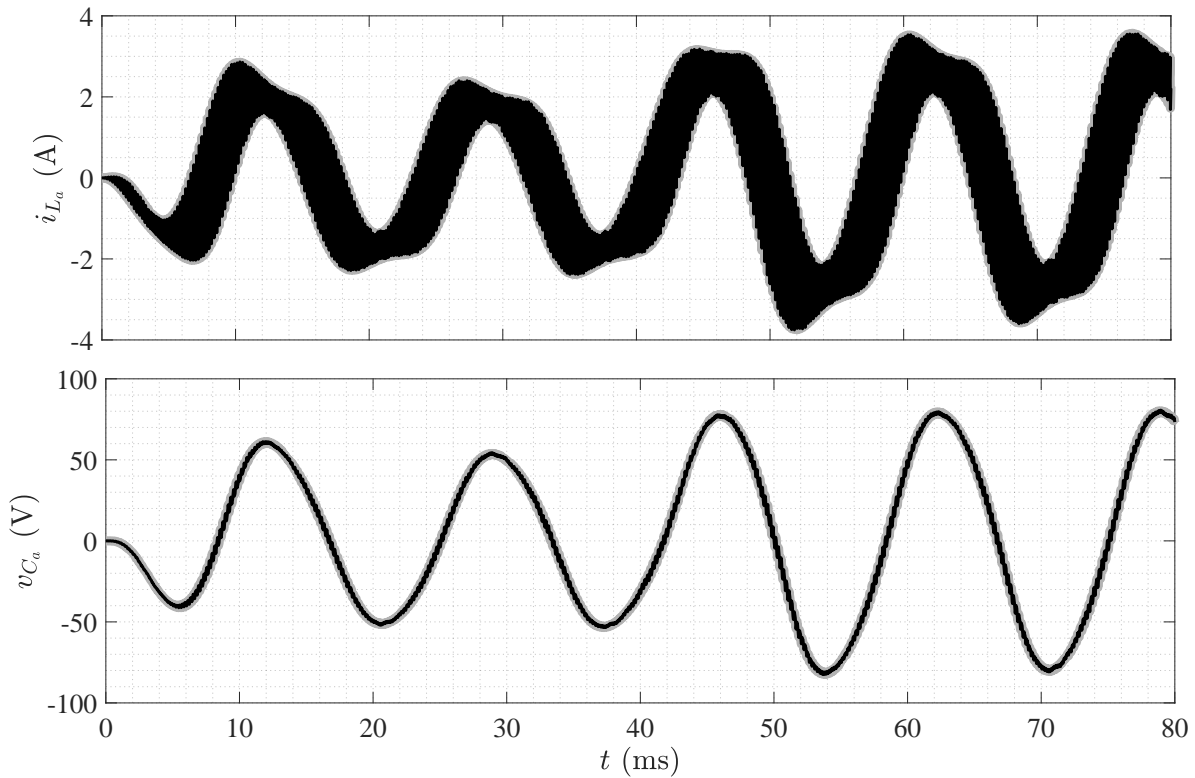
Figura 4.16 – Erros relativos entre as respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} do modelo dinâmico do Quasi-ZSI monofásico em ponte completa e do circuito elétrico de referência.



Fonte: Próprio autor.

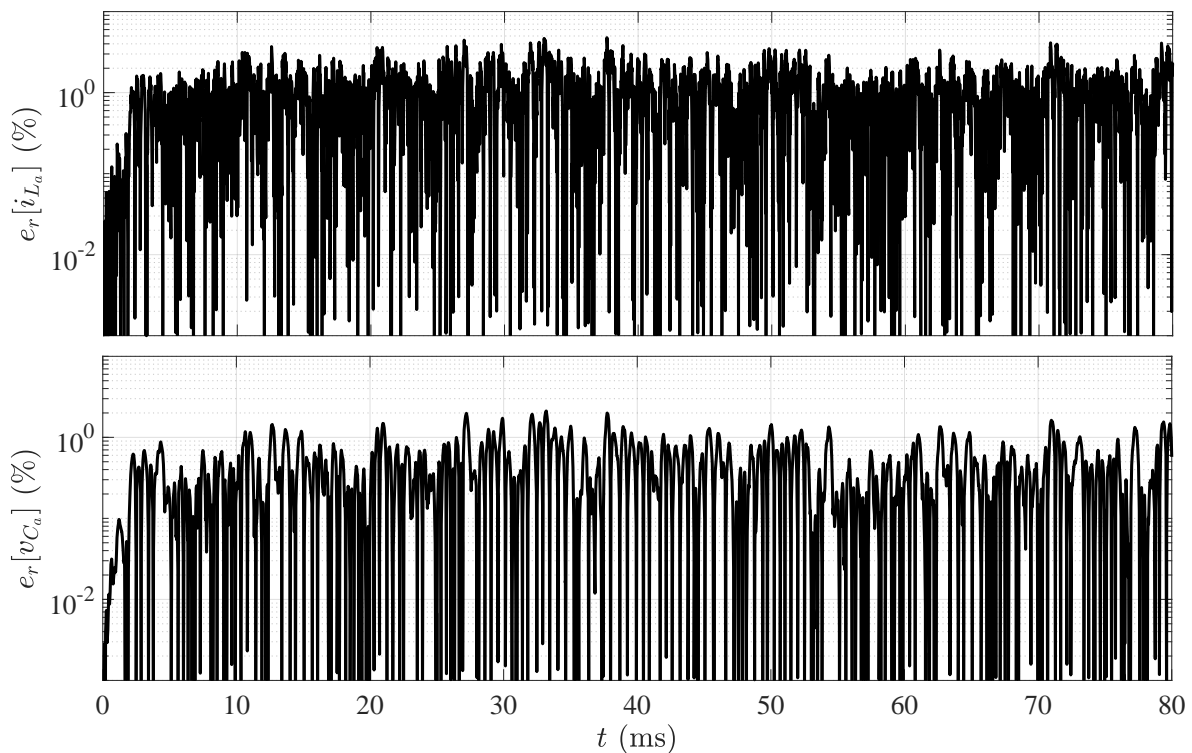
Os resultados encontrados para as variáveis de estado da interface CA são mostrados por meio da Figura 4.17. Na Figura 4.18, são apresentados os respectivos erros relativos com base nos resultados do circuito elétrico de referência.

Figura 4.17 – Respostas de i_{L_a} e v_{C_a} (— referência e — modelo dinâmico) obtidas na simulação do Quasi-ZSI monofásico em ponte completa.



Fonte: Próprio autor.

Figura 4.18 – Erros relativos entre as respostas de i_{L_a} e v_{C_a} do modelo dinâmico do Quasi-ZSI monofásico em ponte completa e do circuito elétrico de referência.



Fonte: Próprio autor.

A estratégia de acionamento foi feita com base na técnica *maximum boost*, descrita na Seção 4.3.4. Para a fonte de tensão v_{CC} , foi adotada a mesma variação dinâmica da Seção 4.3.2.

Conforme pode ser observado, a partir da análise dos resultados encontrados para i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} , referentes à interface CC, e para i_{L_a} e v_{C_a} , referentes à interface CA, é visto que o modelo dinâmico do Quasi-ZSI monofásico em ponte completa, obtido pelo método dos subcircuitos, apresenta um comportamento dinâmico coerente em comparação com o desenvolvido pelo circuito elétrico de referência.

Com base nos resultados das figuras 4.16 e 4.18, é possível perceber que os erros relativos para a integração numérica via FE, com $h = 1 \mu\text{s}$, são razoáveis ($< 5\%$).

4.4 Considerações Finais

Neste capítulo, foi mostrado que, devido à necessidade de se empregar tempos de amostragem menores que 100 ns para manter uma precisão numérica adequada nos resultados ($e_r[\bullet] < 10\%$), tanto a técnica ADC quanto o método do modelo equivalente de chave não são adequados para implementações em MCUs de alto desempenho, tais como os *hardwares* de processamento da família STM32H723. O método dos subcircuitos, por outro lado, entrega resultados satisfatórios a um tempo de amostragem de 1 μs .

Conforme foi discutido, o método dos subcircuitos não apresenta problemas de mal condicionamento. Por esse motivo, baixo custo computacional pode ser alcançado, visto que a integração numérica via FE, que não requer inversão de matriz a cada passo de simulação, pode ser utilizada.

A partir dessas considerações, todos os resultados simulados, referentes à implementação em tempo real dos conversores de potência introduzidos no Capítulo 3, foram apresentados. Para isso, o procedimento de simulação da Figura 2.9 foi realizado para cada conversor, o que permitiu comparar as respostas dos modelos dinâmicos com aquelas dos respectivos circuitos elétricos de referência em termos dos comportamentos dinâmicos das variáveis de estado e dos erros relativos.

Simulação em *Hardware-in-the-Loop*

5.1 Introdução

Com a definição do *hardware* de processamento a ser utilizado na implementação do procedimento de simulação em HIL dos conversores de potência considerados, se torna necessário escolher o tempo de amostragem para a configuração da rotina de interrupção que será utilizada durante o processamento numérico dos modelos dinâmicos levantados. Dessa forma, durante a simulação em HIL de um determinado conversor de potência, a cada evento de interrupção que é disparado pelo *hardware* de processamento, é realizado o cálculo do seu modelo dinâmico a partir da leitura dos parâmetros utilizados como entrada, tais como os sinais digitais de PWM. Na sequência, ocorre a integração numérica desse modelo dinâmico, o que permite atualizar as saídas para visualização em tempo real do comportamento dinâmico das variáveis de estado.

No Capítulo 4, o alvo estabelecido para o tempo de amostragem é de 1 μs , ou seja, a cada 1 μs um evento de interrupção precisa ser disparado no *hardware* de processamento para que o cômputo do respectivo modelo dinâmico da iteração atual inicie. Ainda nesse capítulo, os modelos dinâmicos escolhidos para a implementação em HIL dos conversores de potência considerados são obtidos pelo método dos subcircuitos e, pelo fato da inexistência de problemas de mal condicionamento dos modelos dinâmicos desse método, a integração numérica é via FE, que é um algoritmo explícito e, por isso, não requer inversão de matriz.

Por se tratar da implementação do processador Arm[®] Cortex[®]-M7 que obteve o maior desempenho dentre os MCUs encontrados no mercado de embarcados, um determinado *part number* dos MCUs de alto desempenho inclusos na família STM32H723 precisa ser escolhido para viabilizar a implementação em HIL dos conversores de potência considerados no Capítulo 3. Por questões de maior disponibilidade nos grandes distribuidores do mercado, o *part number* de MCU a ser utilizado é o STM32H723ZGT6U que, inclusive, também pode ser encontrado embarcado no *kit* de desenvolvimento NUCLEO-H723ZG da STMicroelectronics[®] (ST, 2020).

Neste capítulo, a implementação em *hardware* para o procedimento de simulação em

HIL é tratada. Em um primeiro momento, será apresentada a estratégia de compilação utilizada para minimizar o esforço computacional do processamento em tempo real dos modelos dinâmicos com integração numérica via FE. Em seguida, os recursos disponíveis na biblioteca CMSIS-DSP da Arm[®], que são cruciais para se extrair o máximo desempenho do processador Cortex[®]-M7, serão mostrados. Por fim, as simulações em HIL dos conversores de potência do Capítulo 3 serão realizadas sob as mesmas condições das simulações do Capítulo 4 e os resultados encontrados serão apresentados.

5.2 Compilação dos Modelos Dinâmicos

Conforme visto anteriormente na Seção 2.4, a implementação do método de integração numérica via FE ocorre quando se resolve a equação explícita mostrada em (2.11). Observando os termos

$$\left(\mathbf{I} + h \mathbf{A}^k\right) \quad \text{e} \quad h \mathbf{B}^k \mathbf{u}^k \quad (5.1)$$

em (2.11), é possível notar que, pelo método dos subcircuitos da Seção 2.3, o primeiro termo fornece uma matriz constante do tipo \mathbf{M}_Q para cada um dos Q subcircuitos existentes, ao passo que o segundo termo pode ser representado por meio de uma matriz do tipo \mathbf{N}_Q^k variante no tempo devido as atualizações das entradas em \mathbf{u}^k .

Na matriz \mathbf{N}_Q , é importante ressaltar que não é necessário recalcular todos os elementos a cada iteração, porém apenas aqueles associados a entradas que foram de fato alteradas. Caso não tenha ocorrido alterações nas entradas, isto é, $\mathbf{u}^k = \mathbf{u}^{k-1}$, é possível reduzir o custo computacional ao se adotar $\mathbf{N}_Q^k = \mathbf{N}_Q^{k-1}$.

A partir dessas observações, pode ser implementada uma rotina *offline* para compilação dos modelos dinâmicos obtidos no Capítulo 3 para o método dos subcircuitos. Com o resultado da execução dessa rotina, todas as matrizes \mathbf{M}_Q , para cada conversor de potência considerado, são obtidas. Com isso, essas matrizes são carregadas no *software* desenvolvido para o HIL de forma pré-calculada. Dessa maneira, a cada evento de interrupção, ou a cada amostragem do modelo dinâmico, as matrizes correspondentes são acessadas.

Note que, por meio dessa etapa de compilação, o processamento dos modelos dinâmicos dos conversores de potência é reduzido a uma operação de multiplicação matricial,

$$\mathbf{M}_X^{k+1} = \mathbf{M}_Q^k \mathbf{x}^k \quad (5.2)$$

e uma operação de soma matricial

$$\mathbf{x}^{k+1} = \mathbf{M}_X^{k+1} + \mathbf{N}_Q^k \quad (5.3)$$

em que os valores numéricos carregados em \mathbf{x}^{k+1} são os resultados da integração numérica via FE, ou seja, as variáveis de estado cujos comportamentos dinâmicos devem ser observados.

5.3 Recursos Arm[®] CMSIS-DSP

O *Common Microcontroller Software Interface Standard* (CMSIS) é um fornecedor independente de camadas de abstração de *hardware* para microcontroladores que utilizam processadores Arm[®] Cortex[®]-M e Cortex[®]-A. Essas camadas são implementadas por meio de interfaces de *software* que, por sua vez, utilizam de forma otimizada todo o pacote de instruções desenvolvidos pela Arm[®] para esses processadores. Por esse motivo, os recursos disponibilizados pelo CMSIS permitem alcançar o máximo desempenho dos processadores com tecnologia Cortex[®]-M e Cortex[®]-A, tanto em termos de velocidade de processamento quanto em simplificação na integração com periféricos de microcontroladores (ARM, 2020).

As interfaces de *software* implementadas pelo CMSIS podem ser acessadas por meio de um conjunto extenso de bibliotecas que atendem os mais diversos tipos de aplicação. Para cumprir com os propósitos definidos neste trabalho, se destaca a biblioteca CMSIS-DSP, que reúne um conjunto de funções matemáticas especiais desenvolvidas exclusivamente para otimizar o uso de microcontroladores com processadores Arm[®] Cortex[®]-M e Cortex[®]-A para o processamento digital de sinais com números em ambos os formatos de ponto fixo (q7, q15, q31 e q63) e de ponto flutuante (f32 e f64) com padrão IEEE 754.

Para processadores a partir do Arm[®] Cortex[®]-M4, os recursos CMSIS-DSP executam instruções matemáticas em *hardware* por meio de uma *floating-point unit* (FPU) opcional. Em versões anteriores, essas instruções matemáticas são implementadas por *software* e, por isso, requerem mais ciclos de *clock* para realizar operações de multiplicação e acumulação. Conforme é observado, a FPU é uma unidade de processamento opcional, isto é, não nativa. Por esse motivo, se torna necessário habilitá-la para que o pacote de instruções DSP estejam disponíveis e ela possa ser devidamente acessada (ARM, 2018). Vale ressaltar que, com a utilização do CMSIS, a habilitação da FPU ocorre por consequência.

De forma geral, a biblioteca CMSIS-DSP é dividida em um número de funções específicas que se enquadra em cada uma das categorias listadas a seguir (ARM, 2020).

- Funções matemáticas básicas.
- Funções matemáticas rápidas.
- Funções matemáticas complexas.
- Filtros digitais.
- Funções matriciais.
- Transformadas.
- Funções para controle de máquinas elétricas.
- Funções estatísticas.

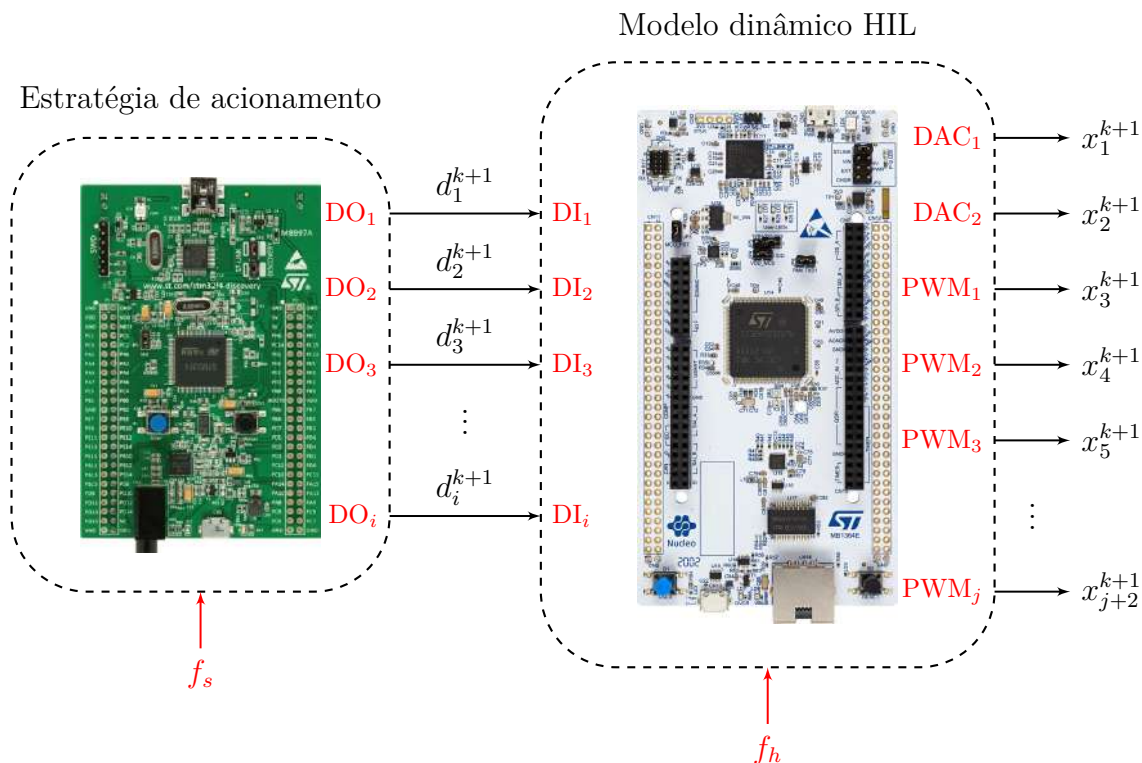
- Funções de suporte.
- Funções de interpolação.
- Funções para máquinas de suporte vetorial.
- Funções de distância.

Para o desenvolvimento do *software* para a implementação do procedimento de simulação em HIL dos conversores de potência do Capítulo 3, o interesse na biblioteca CMSIS-DSP concentra-se na utilização das funções matriciais, uma vez que a estratégia de processamento dos modelos dinâmicos tratada neste trabalho é puramente matricial.

5.4 Simulação em Tempo Real

O diagrama mostrando como deve ser feita a implementação experimental do procedimento de simulação em HIL pode ser visto na Figura 5.1. Como se pode observar, um MCU é utilizado separadamente para se programar a estratégia de acionamento dos conversores de potência a serem simulados e, com isso, gerar os $i \in \mathbb{N}$ sinais lógicos de comando \mathbf{d}^{k+1} , com frequência de chaveamento f_s . Para esse propósito, está sendo utilizado o *kit* de desenvolvimento STM32F407G-DISC1 da STMicroelectronics®, disponibilizado para o MCU STM32F407VGT6U de 168 MHz, com processador Arm® Cortex®-M4 de 32 *bits*.

Figura 5.1 – Diagrama para implementação experimental do procedimento de simulação em HIL.



Fonte: Próprio autor.

Conforme mencionado anteriormente, os modelos dinâmicos a serem simulados em HIL são processados em tempo real dentro de uma rotina de interrupção, com duração de 1 μs , programada no MCU STM32H723ZGT6U, que é embarcado no *kit* de desenvolvimento NUCLEO-H723ZG da STMicroelectronics®.

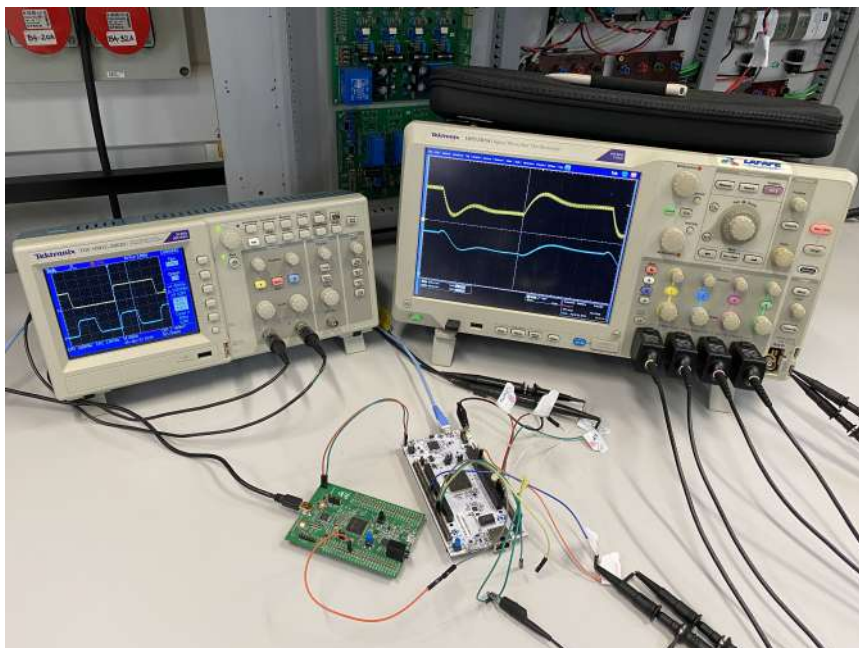
Os sinais lógicos de comando \mathbf{d}^{k+1} são gerados por meio de *digital outputs* (DOs) no STM32F407VGT6U. Logo no início de cada evento de interrupção que é disparado, *digital inputs* (DIs) no STM32H723ZGT6U são utilizados para realizar a leitura desses mesmos sinais que, neste MCU, são tratados como entradas digitais. Após o processamento do modelo dinâmico em questão, as $(j + 2) \in \mathbb{N}$ saídas \mathbf{x}^{k+1} são atualizadas por meio de *digital-to-analog converters* (DACs) ou, ainda, por meio de saídas de PWM com frequência maior que 200 kHz, em cascata com filtros *RC* com frequência de corte entre 100 kHz e 150 kHz (filtros *RC* omitidos na Figura 5.1), o que torna possível o monitoramento em tempo real das formas de onda das saídas \mathbf{x}^{k+1} em um osciloscópio digital.

Na sequência, os resultados obtidos na simulação em HIL dos conversores de potência do Capítulo 3, com base na abordagem tratada nesta dissertação, são apresentados. Esses resultados podem ser comparados com aqueles mostrados no Capítulo 4 que, porventura, foram reproduzidos a partir de *software* de simulação computacional específico.

5.4.1 Conversor *Boost*

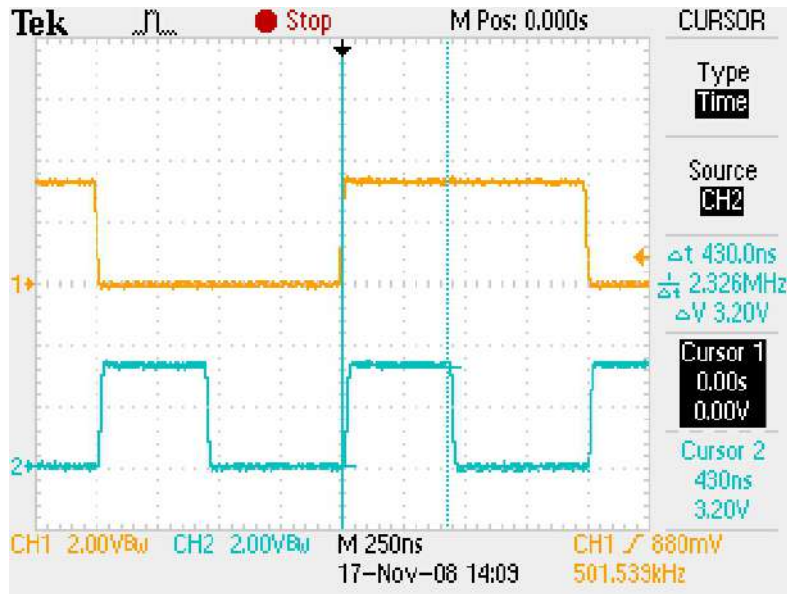
O *setup* de testes que foi utilizado para a simulação em HIL do conversor *boost* pode ser visto na Figura 5.2. Na Figura 5.3, é apresentada a imagem de osciloscópio mostrando o tempo de processamento do modelo dinâmico em comparação com o tempo de interrupção.

Figura 5.2 – *Setup* de testes para a simulação em HIL do conversor *boost*.



Fonte: Próprio autor.

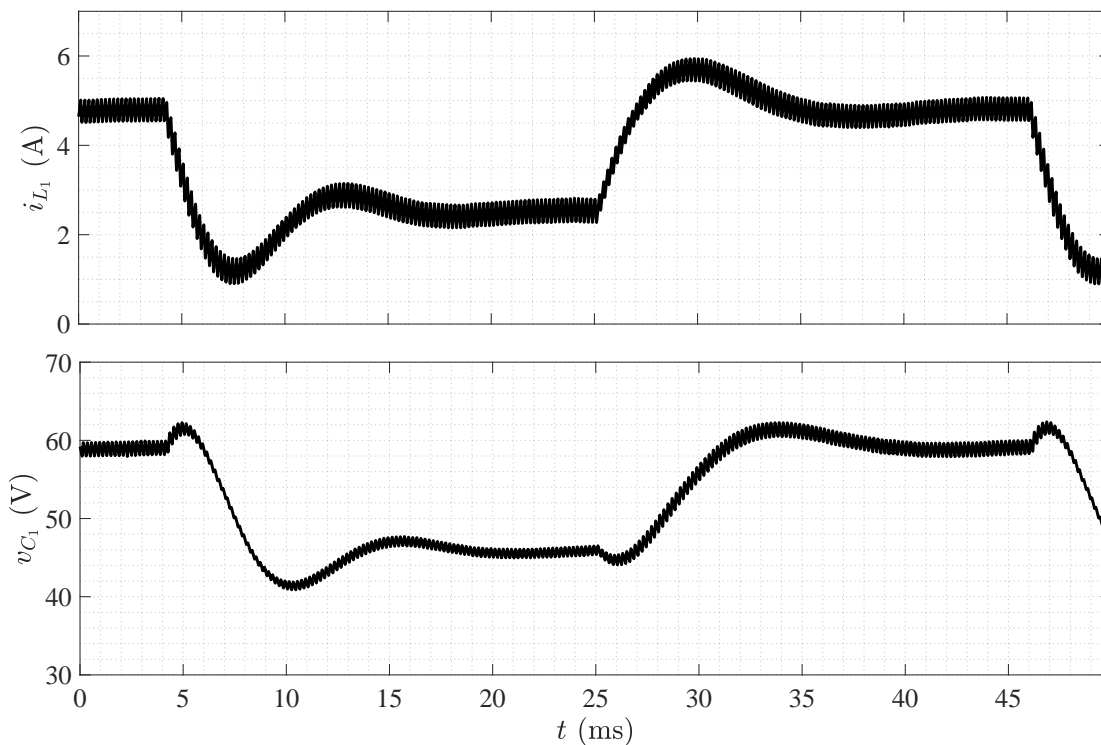
Figura 5.3 – Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor *boost* () em comparação com o tempo de interrupção ().



Fonte: Próprio autor.

Pela Figura 5.3, se observa que o processamento do modelo dinâmico do conversor *boost* consome, aproximadamente, 430 ns. As formas de onda extraídas do osciloscópio e, posteriormente, processadas no MATLAB, são mostradas na Figura 5.4.

Figura 5.4 – Respostas de i_{L_1} e v_{C_1} obtidas na simulação em HIL do conversor *boost*.



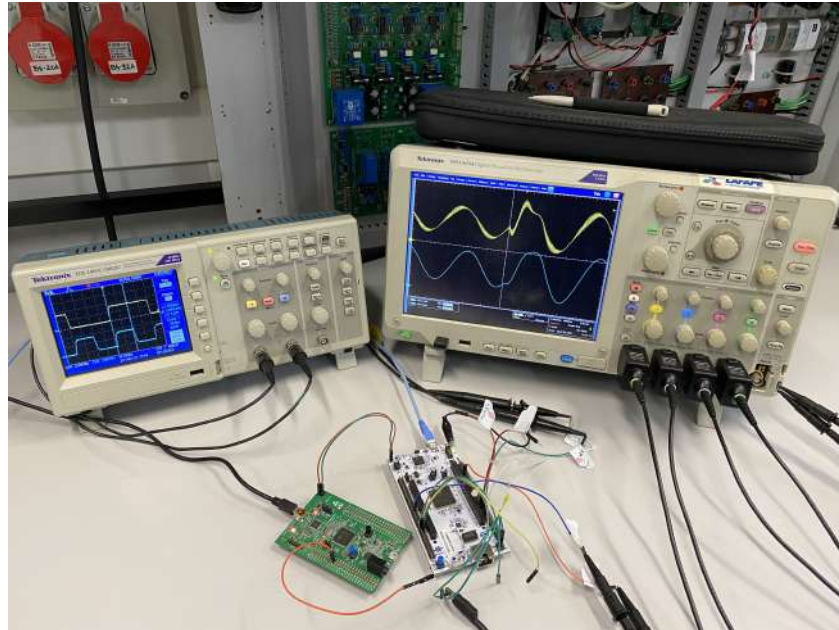
Fonte: Próprio autor.

Comparando as formas de onda da Figura 5.4 com aquelas da Figura 4.5, é possível concluir que os resultados são coerentes com o esperado.

5.4.2 Conversor VSI no Modo Ilhado

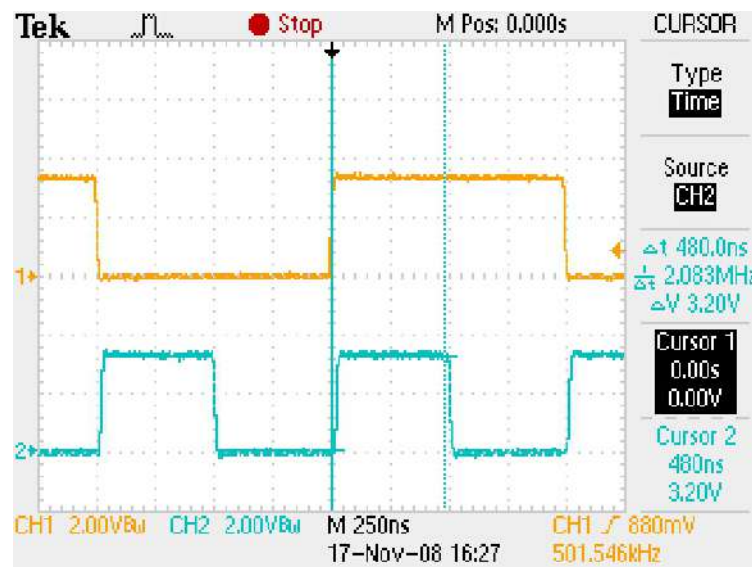
O *setup* de testes que foi utilizado para a simulação em HIL do conversor VSI no modo ilhado pode ser visto por meio da Figura 5.5. Na Figura 5.6, é mostrada a imagem de osciloscópio com o tempo de processamento do modelo dinâmico em comparação com o tempo de interrupção.

Figura 5.5 – *Setup* de testes para a simulação em HIL do conversor VSI no modo ilhado.



Fonte: Próprio autor.

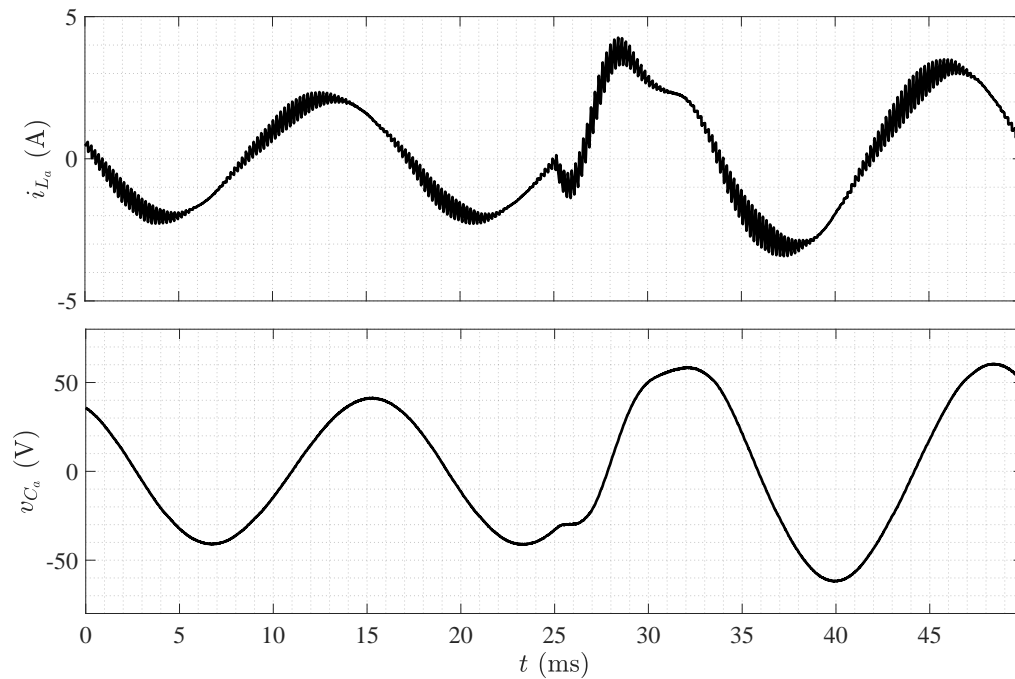
Figura 5.6 – Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor VSI no modo ilhado (—) em comparação com o tempo de interrupção (—).



Fonte: Próprio autor.

Pela Figura 5.6, é visto que o processamento do modelo dinâmico do conversor VSI no modo ilhado consome, aproximadamente, 480 ns. As formas de onda extraídas do osciloscópio e, posteriormente, processadas no MATLAB, são mostradas na Figura 5.7.

Figura 5.7 – Respostas de i_{L_a} e v_{C_a} obtidas na simulação em HIL do VSI monofásico em ponte completa no modo ilhado.



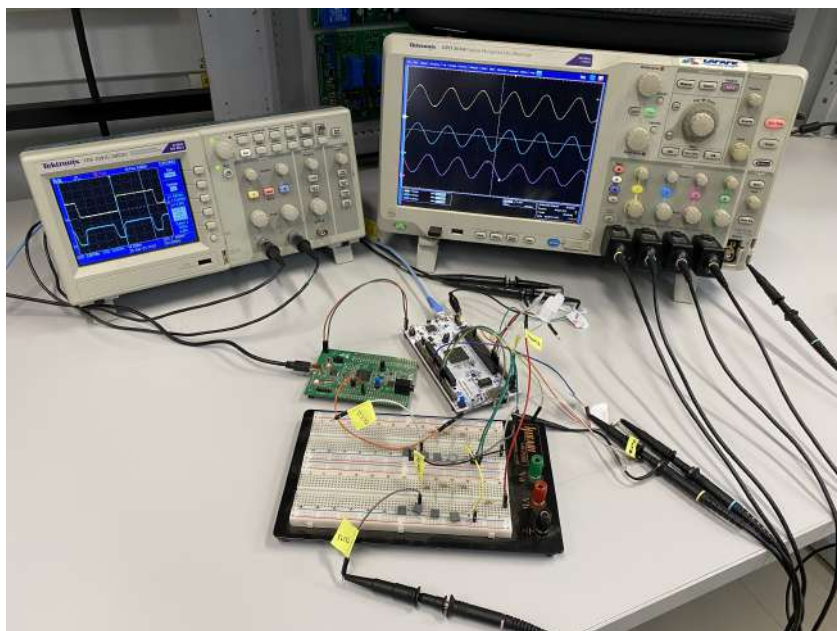
Fonte: Próprio autor.

Comparando as formas de onda da Figura 5.7 com aquelas da Figura 4.7, é possível concluir que os resultados são coerentes com o esperado.

5.4.3 Conversor VSI no Modo Conectado

O *setup* de testes que foi utilizado para a simulação em HIL do conversor VSI no modo conectado pode ser visto por meio da Figura 5.8.

Figura 5.8 – *Setup* de testes para a simulação em HIL do conversor VSI no modo conectado.



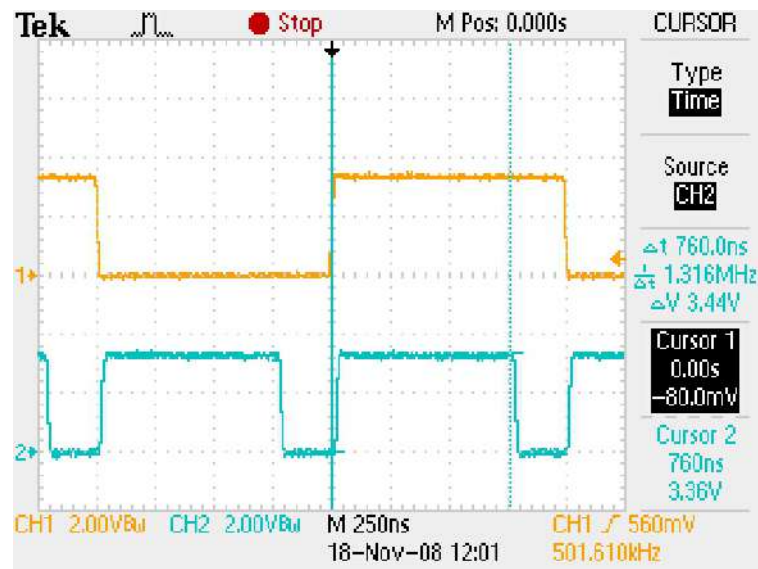
Fonte: Próprio autor.

Na Figura 5.9, é possível observar a imagem de osciloscópio que mostra o tempo de processamento do modelo dinâmico do conversor VSI no modo conectado. Conforme pode ser visto, o tempo de processamento registrado foi de, aproximadamente, 760 ns em comparação com o tempo de interrupção de 1 μ s.

Ao se comparar o tempo de processamento do conversor VSI no modo conectado com os demais conversores simulados em HIL mostrados até o momento, é observado que houve um aumento significativo no tempo de processamento do modelo dinâmico, apesar de o mesmo ser de baixa complexidade. Isto ocorreu devido à necessidade de se gerar a tensão da rede, v_a , que é expressa por meio de uma função seno, em tempo real juntamente com o modelo dinâmico processado. Sendo assim, a cada evento de interrupção, o valor v_a^{k+1} é calculado, e o resultado encontrado é utilizado na atualização da respectiva matriz \mathbf{N}_Q^k .

Vale ressaltar que, de acordo com medições feitas durante a realização do experimento, o tempo gasto pelo STM32H723ZGT6U para se resolver a função seno, dentro da rotina de interrupção, foi de, aproximadamente, 150 ns. A função seno foi calculada com a utilização de um recurso de aceleração trigonométrica implementado na biblioteca CMSIS-DSP, disponível dentro da categoria de funções matemáticas rápidas.

Figura 5.9 – Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor VSI no modo conectado (—) em comparação com o tempo de interrupção (—).

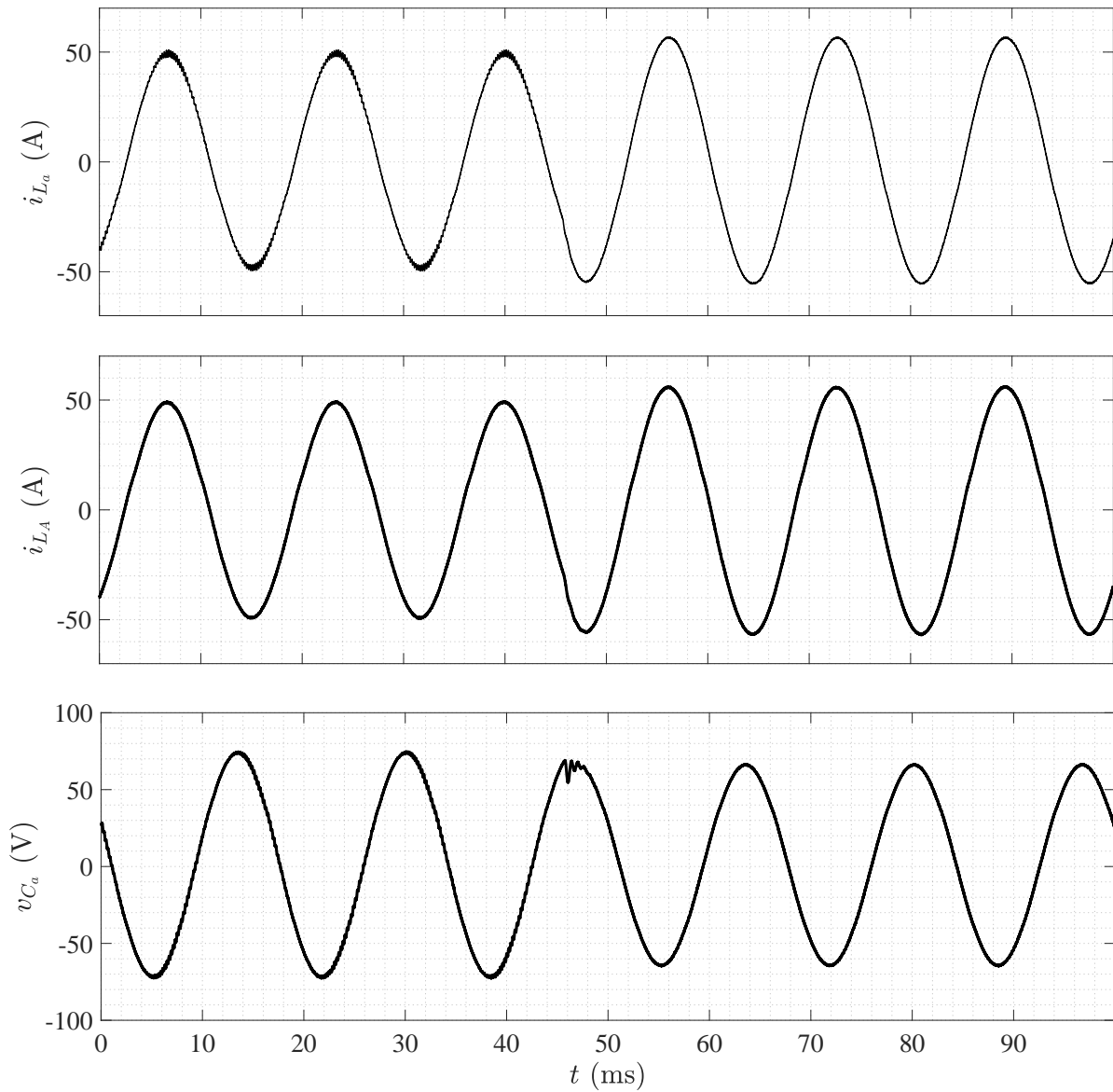


Fonte: Próprio autor.

As formas de onda que foram extraídas do osciloscópio e, posteriormente, processadas no MATLAB, podem ser vistas por meio da Figura 5.10.

Mais uma vez é possível verificar a coerência do comportamento dinâmico das formas de onda obtidas durante o procedimento de simulação em HIL que foi implementado. Dessa forma, ao comparar as formas de onda da Figura 5.10 com aquelas que são mostradas por meio da Figura 4.9, se conclui que os resultados estão de acordo com o esperado.

Figura 5.10 – Respostas de i_{L_a} , i_{L_A} e v_{C_a} obtidas na simulação em HIL do VSI monofásico em ponte completa no modo conectado.

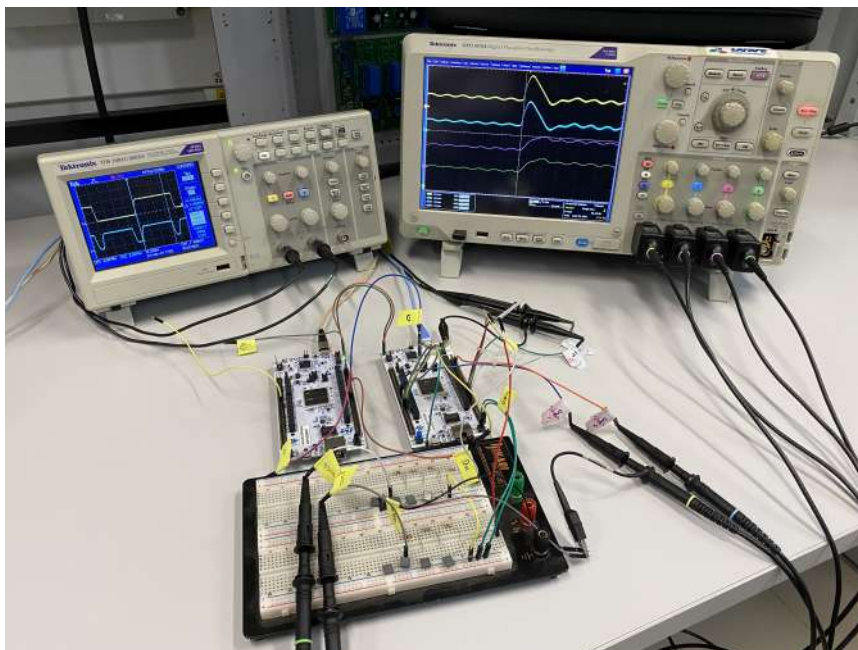


Fonte: Próprio autor.

5.4.4 Conversor ZSI

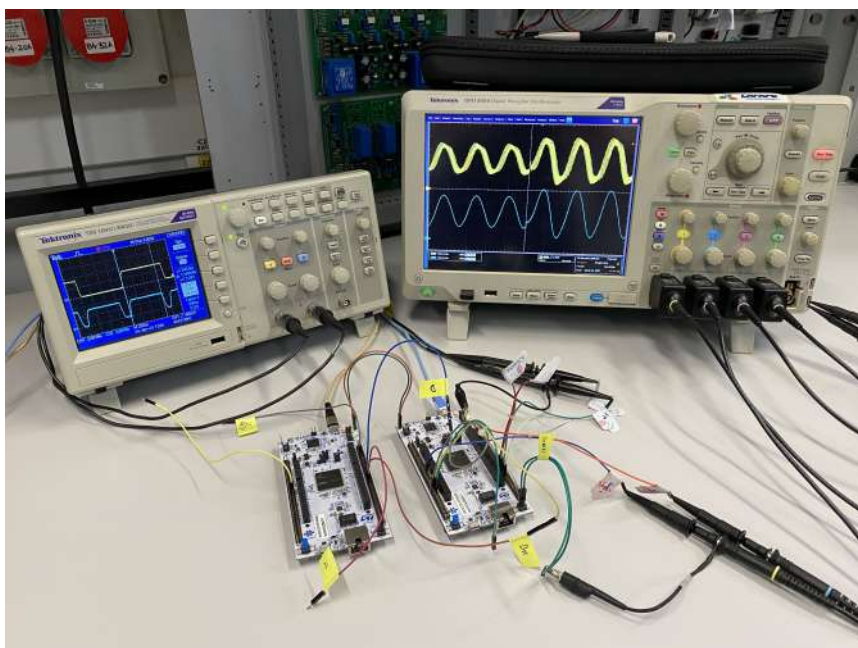
O *setup* de testes que foi utilizado para a simulação em HIL do lado CC do conversor ZSI pode ser visto por meio da Figura 5.11, em que pode ser observada a utilização de dois filtros RC , mencionados anteriormente para saídas em PWM, que foram utilizados para medição de v_{C_1} e v_{C_2} . Para o lado CA, o *setup* de testes é mostrado por meio da Figura 5.12.

Figura 5.11 – *Setup* de testes para a simulação em HIL do lado CC do conversor ZSI.



Fonte: Próprio autor.

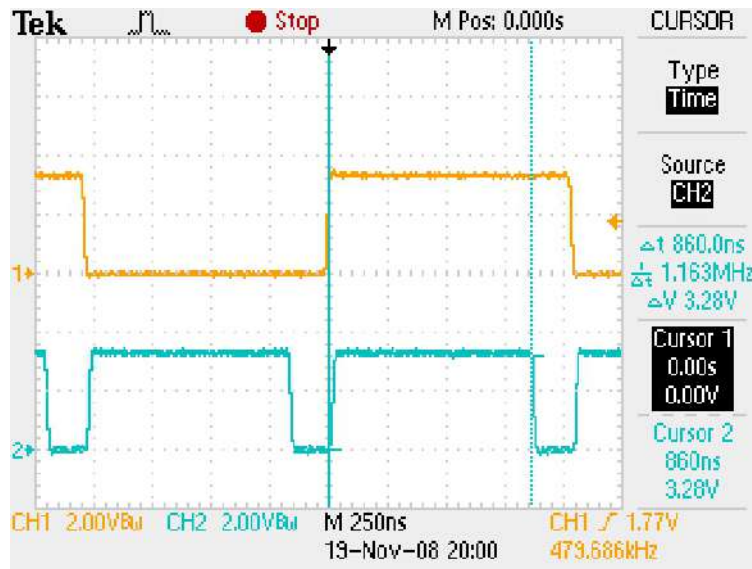
Figura 5.12 – *Setup* de testes para a simulação em HIL do lado CA do conversor ZSI.



Fonte: Próprio autor.

Na Figura 5.13, é apresentada a imagem de osciloscópio mostrando o tempo de processamento do modelo dinâmico, em que o valor registrado foi de, aproximadamente, 860 ns, em comparação com o tempo de interrupção.

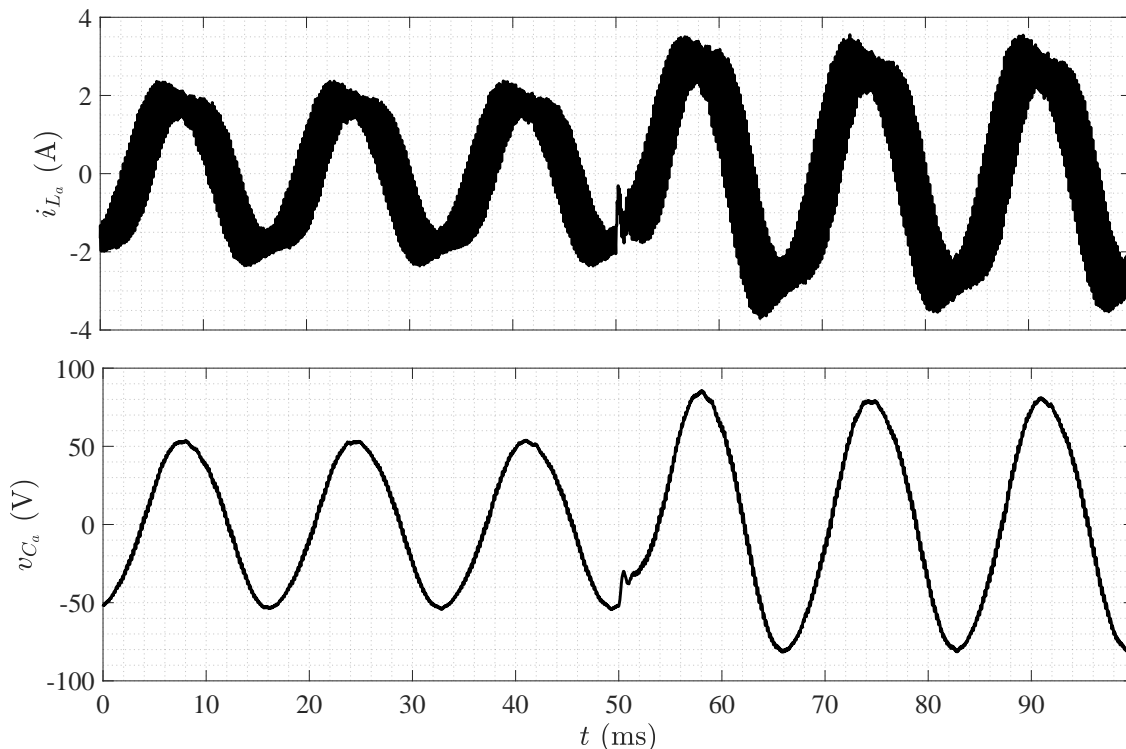
Figura 5.13 – Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor ZSI (—) em comparação com o tempo de interrupção (—).



Fonte: Próprio autor.

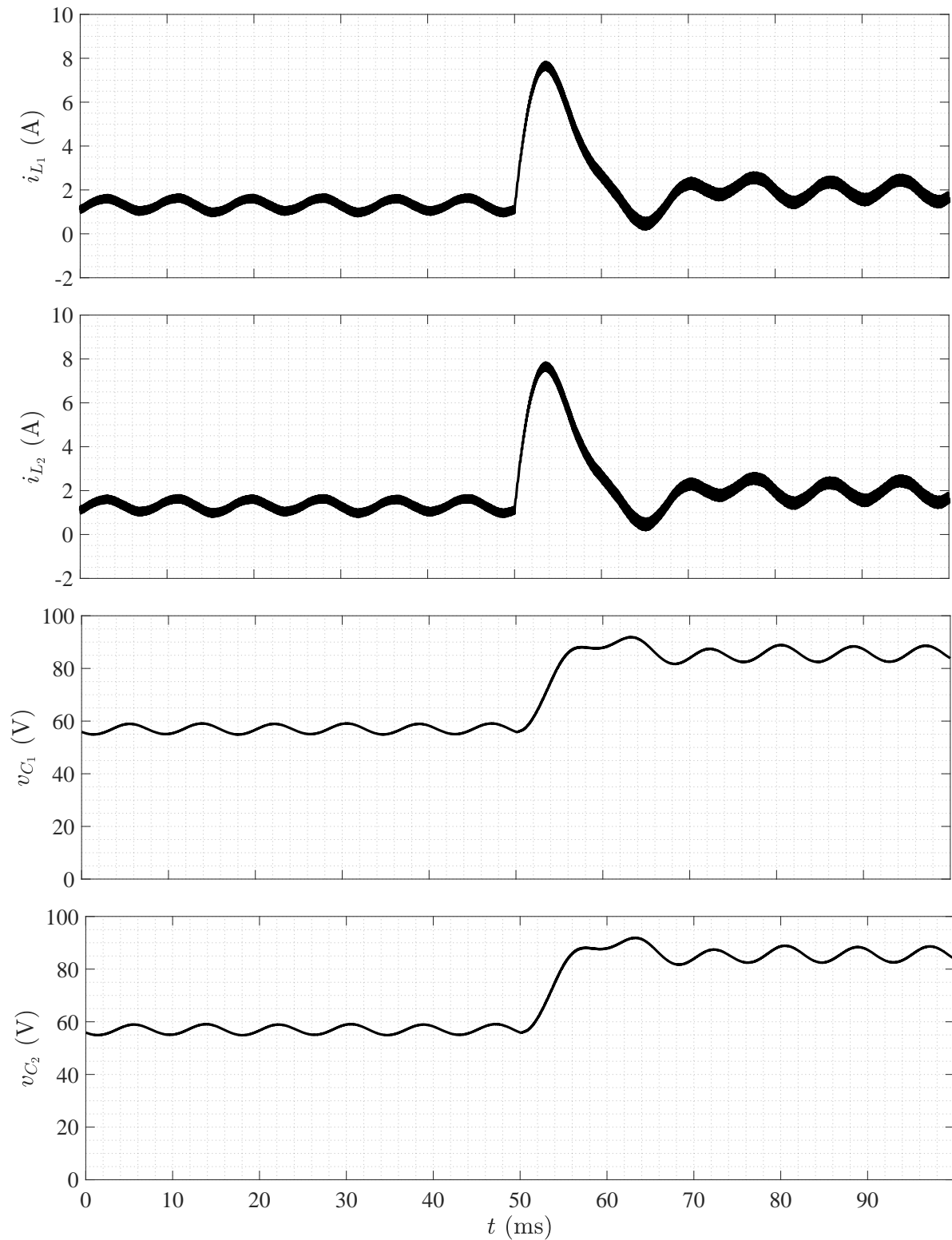
A Figura 5.14 mostra os resultados encontrados para o lado CA, ao passo que a Figura 5.15 mostra os resultados referentes ao lado CC.

Figura 5.14 – Respostas de i_{L_a} e v_{C_a} obtidas na simulação em HIL do ZSI monofásico em ponte completa.



Fonte: Próprio autor.

Figura 5.15 – Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} obtidas na simulação em HIL do ZSI monofásico em ponte completa.



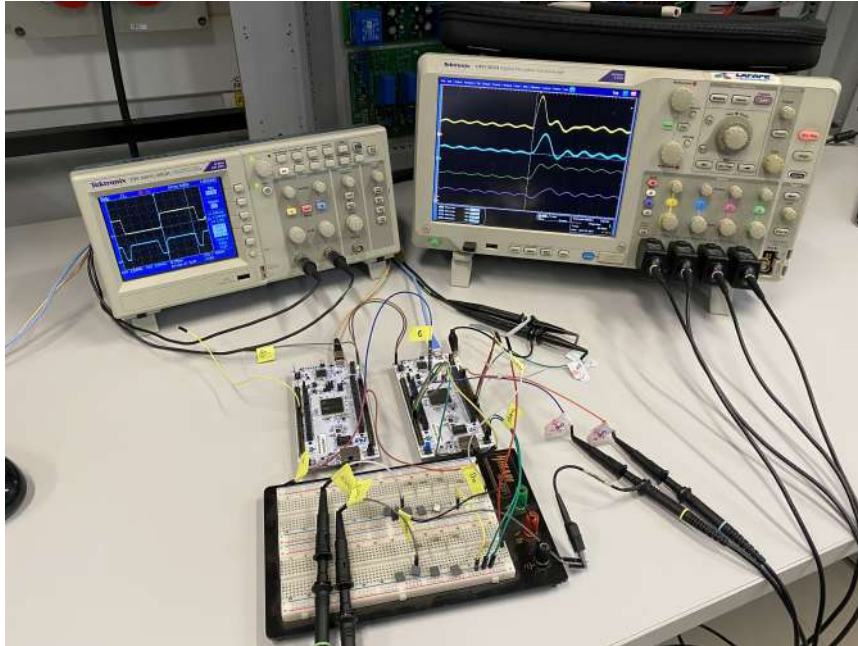
Fonte: Próprio autor.

Ao se comparar os resultados mostrados nas Figuras 5.14 e 5.15 com aqueles apresentados nas Figuras 4.13 e 4.11, respectivamente, se pode verificar que as respostas dinâmicas encontradas na simulação em HIL do conversor ZSI estão em conformidade com o esperado.

5.4.5 Conversor Quasi-ZSI

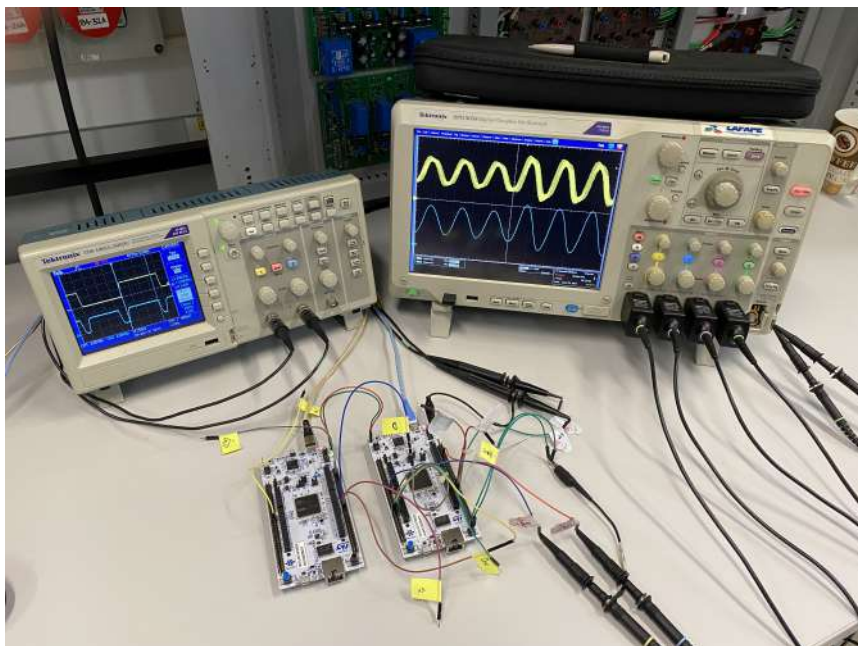
O *setup* de testes que foi utilizado para a simulação em HIL do lado CC do conversor Quasi-ZSI pode ser visto por meio da Figura 5.16, em que, novamente, é possível observar a mesma aplicação dos filtros RC utilizados para medição de v_{C_1} e v_{C_2} no conversor ZSI. Para o lado CA, o *setup* de testes é mostrado por meio da Figura 5.17.

Figura 5.16 – *Setup* de testes para a simulação em HIL do lado CC do conversor Quasi-ZSI.



Fonte: Próprio autor.

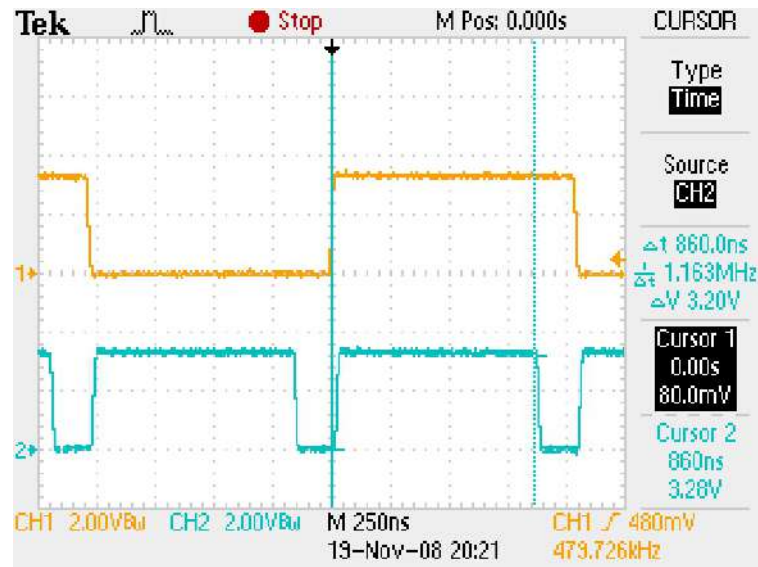
Figura 5.17 – *Setup* de testes para a simulação em HIL do lado CA do conversor Quasi-ZSI.



Fonte: Próprio autor.

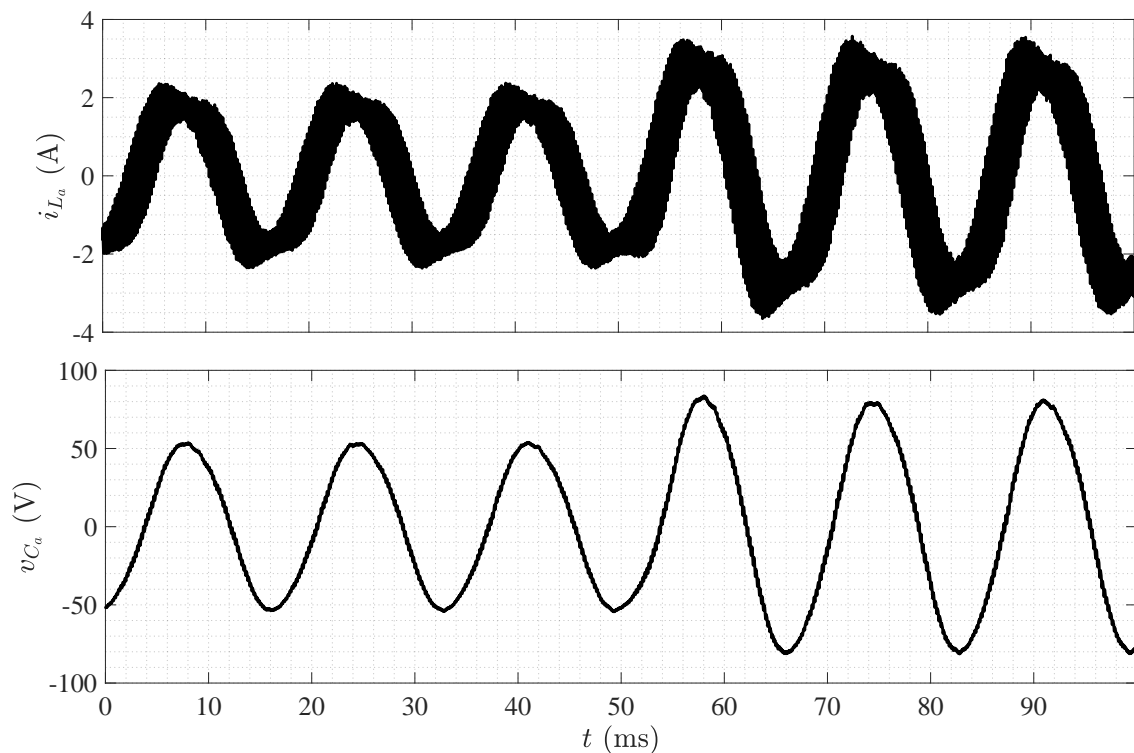
Na Figura 5.18, é mostrada a imagem de osciloscópio com o tempo de processamento de, aproximadamente, 860 ns. Note que este valor é o mesmo obtido durante o processamento do ZSI. Isto ocorre devido ao fato de ambos os modelos dinâmicos possuírem a mesma complexidade computacional, uma vez que são processados o mesmo número de variáveis de estado e de entradas.

Figura 5.18 – Imagem de osciloscópio com o tempo de processamento do modelo dinâmico do conversor Quasi-ZSI (—) em comparação com o tempo de interrupção (—).



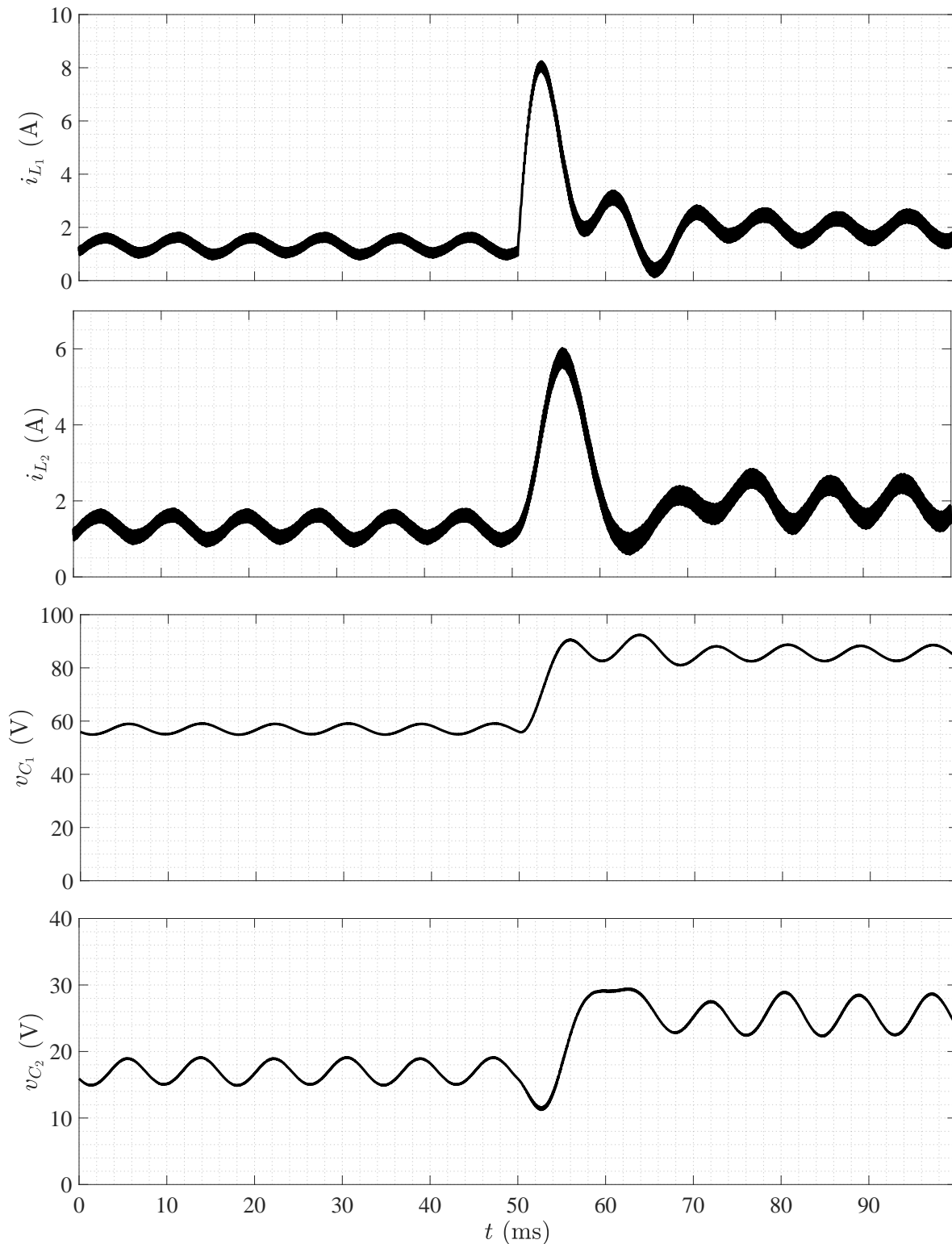
Fonte: Próprio autor.

Figura 5.19 – Respostas de i_{L_a} e v_{C_a} obtidas na simulação em HIL do Quasi-ZSI monofásico em ponte completa.



Fonte: Próprio autor.

Figura 5.20 – Respostas de i_{L_1} , i_{L_2} , v_{C_1} e v_{C_2} obtidas na simulação em HIL do Quasi-ZSI monofásico em ponte completa.



Fonte: Próprio autor.

Ao se comparar os resultados mostrados nas Figuras 5.19 e 5.20 com aqueles apresentados nas Figuras 4.17 e 4.15, respectivamente, é verificado que as respostas dinâmicas encontradas na simulação em HIL do conversor Quasi-ZSI estão em conformidade com o esperado.

5.5 Considerações Finais

Neste capítulo, os recursos utilizados para se viabilizar a implementação experimental do procedimento de simulação em HIL dos conversores de potência do Capítulo 3 foram devidamente discutidos. A estratégia utilizada para a compilação dos modelos dinâmicos, desenvolvida para o método dos subcircuitos com integração numérica via FE, foi apresentada, além de terem sido mostrados todos os recursos disponibilizados pela biblioteca CMSIS-DSP da Arm®.

Por fim, os conversores de potência discutidos no Capítulo 3 foram simulados em HIL com as mesmas condições e parâmetros utilizados nas simulações com *software* computacional do Capítulo 4. De acordo com o que foi observado, é possível afirmar que os resultados encontrados são satisfatórios e, por isso, estão em conformidade com o esperado. O comportamento dinâmico obtido em cada uma das variáveis de estado dos conversores de potência testados em HIL é equivalente a sua contraparte simulada em *software* computacional.

Considerações Finais

6.1 Conclusões

Nesta dissertação, foi abordada a modelagem e a implementação de conversores de potência para aplicações de tempo real em HIL com base na utilização de MCUs de alto desempenho. Dessa forma, buscou-se contextualizar as principais técnicas de modelagem de conversores de potência que podem ser utilizadas na implementação em tempo real, e os métodos numéricos que são viáveis para o processamento em tempo real de modelos dinâmicos.

Duas técnicas de modelagem de circuitos elétricos chaveados foram discutidas ao longo do desenvolvimento do trabalho. Na primeira, chamada de técnica ADC, foi vista a introdução de uma metodologia de discretização de elementos armazenadores de energia e de chaves. Enquanto que, na segunda, que é por espaço de estado, foi mostrado como representar circuitos elétricos chaveados a partir das variáveis de estado dos elementos armazenadores de energia. Na técnica por espaço de estados, dois métodos foram vistos, que são o método dos subcircuitos, em que é obtida uma representação em espaço de estados para cada subintervalo de chaveamento factível, e o método do modelo equivalente de chave, que visa representar cada chave por meio de uma resistência variável em paralelo com um capacitor de baixa capacitância.

Em termos da implementação de procedimentos de simulação computacional no domínio do tempo, foi visto que soluções numéricas precisam ser consideradas para a realização do processamento dos modelos dinâmicos. Nesse sentido, enquanto a solução numérica aplicada à técnica ADC se resume em resolver uma única multiplicação de matrizes, a partir de uma matriz inversa pré-calculada, os métodos dos subcircuitos e do modelo equivalente de chave, por serem baseados na representação em espaço de estados, requerem a aplicação de métodos de integração numérica específicos para o devido processamento dos modelos dinâmicos. Os métodos de integração numérica considerados foram o FE, o BDF1, o BDF2 e o TR.

Para os cenários de teste, foram considerados o conversor CC–CC *boost*, e quatro

configurações diferentes de conversor CC–CA, as versões monofásicas em ponte completa do VSI, nos modos ilhado e conectado, do ZSI e do Quasi–ZSI. Devido à necessidade de se empregar tempos de amostragem menores que 100 ns para manter precisão numérica nos resultados, tanto a técnica ADC quanto o método do modelo equivalente de chave não são adequados para implementações em MCUs de alto desempenho. Nesse aspecto, o método dos subcircuitos se mostrou adequado, pois fornece resultados com precisão numérica satisfatória com o tempo de amostragem de 1 μ s.

Conforme foi discutido, uma vantagem adicional do método dos subcircuitos é que o mesmo não apresenta problemas de mal condicionamento. Sendo assim, baixo custo computacional pode ser alcançado, visto que a integração numérica via FE, que não requer inversão de matriz a cada passo de simulação, pode ser utilizada.

Os resultados simulados foram apresentados, em que apenas as respostas obtidas por meio do método dos subcircuitos foram mostradas. Para cada conversor de potência considerado, o respectivo procedimento de simulação computacional no domínio do tempo foi elaborado com base no processamento numérico do modelo dinâmico em 1 μ s. De acordo com os gráficos dos erros relativos encontrados, em que as comparações foram feitas a partir das respostas de circuitos elétricos de referência, foi constatado que todos os modelos dinâmicos obtidos por meio do método dos subcircuitos podem ser utilizados para a implementação do procedimento de simulação em tempo real em *hardware*.

Por fim, a simulação em HIL para cada um dos conversores de potência considerados como cenários de teste foi implementada por meio do MCU de alto desempenho STM32H723ZGT6U da STMicroelectronics®, que utiliza processador Arm® Cortex®-M7 de 32 *bits* em 550 MHz. Os resultados finais encontrados com as simulações em HIL foram obtidos a partir das mesmas condições definidas para as simulações realizadas em *software* computacional. Conforme foi observado, o comportamento dinâmico obtido em cada uma das variáveis de estado dos conversores de potência testados em HIL é equivalente a sua contraparte simulada em *software* computacional. Por esse motivo, é possível afirmar que o procedimento de simulação em HIL desenvolvido obteve um desempenho satisfatório, o que cumpre com parte dos objetivos estabelecidos no Capítulo 1 para a execução deste trabalho.

6.2 Publicações

Os trabalhos concluídos em que o autor detém participação são listados na sequência.

6.2.1 Trabalhos Completos Publicados em Anais de Eventos

1. Castro, D. S.; Magossi, R. Q.; Bastos, R. F.; Machado, R. Q.; Oliveira, V. A. Low-cost Hardware-in-the-Loop implementation of a boost converter. In: **2019 European Control Conference (ECC)**. Naples: IEEE, 2019.

6.2.2 Trabalhos Completos Publicados em Periódicos

1. Magossi, R. F. Q.; Castro, D. S.; Oliveira, A. L. R.; Machado, R. Q. A comprehensive non-ideal steady-state analysis of a threefold operation mode interleaved-based DC–DC converter. **IEEE Access**, v. 8, p. 144167–144183, aug. 2020.
2. Agnoletto, E. J.; Castro, D. S.; Neves, R. V. A.; Machado, R. Q.; Oliveira, V. A. An optimal energy management technique using the ϵ -constraint method for grid-tied and stand-alone battery-based microgrids. **IEEE Access**, v. 7, p. 165928–165942, nov. 2019.

Referências Bibliográficas

- ARM. **ARMv7 Architecture Reference Manual**. Cambridge, UK, 2018. Rev. E.d.
- _____. **Software Interface Standard for Arm Cortex-based Microcontrollers**. Cambridge, UK, 2020. Disponível em: <www.keil.com/pack/doc/CMSIS/General/html/index.html>. Acesso em: 27/06/2021.
- Blanchette, H. F.; Ould-Bachir, T.; David, J. P. A state-space modeling approach for the FPGA-based real-time simulation of high switching frequency power converters. **IEEE Transactions on Industrial Electronics**, v. 59, n. 12, p. 4555–4567, dez. 2012.
- Castro, D. S.; Magossi, R. Q.; Bastos, R. F.; Machado, R. Q.; Oliveira, V. A. Low-cost hardware-in-the-loop implementation of a boost converter. In: **2019 European Control Conference (ECC)**. Naples: IEEE, 2019.
- Cellier, F. E.; Kofman, E. **Continuous System Simulation**. 1st. ed. New York: Springer, 2006.
- Chua, L. O.; Lin, P.-M. **Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques**. 1st. ed. New York: Prentice Hall, 1975.
- Cornea, O.; Andreescu, G.; Muntean, N.; Hulea, D. Bidirectional power flow control in a DC microgrid through a switched-capacitor cell hybrid DC–DC converter. **IEEE Transactions on Industrial Electronics**, v. 64, n. 4, p. 3012–3022, abr. 2017.
- Dufour, C.; Mahseredjian, J.; Belanger, J. A combined state-space nodal method for the simulation of power system transients. **IEEE Transactions on Power Delivery**, v. 26, n. 2, p. 928–935, abr. 2011.
- Erickson, R. W.; Maksimović, D. **Fundamentals of Power Electronics**. 2nd. ed. New York: Kluwer Publishers, 2001.
- Forouzesh, M.; Siwakoti, Y. P.; Gorji, S. A.; Blaabjerg, F.; Lehman, B. Step-up DC–DC converters: A comprehensive review of voltage-boosting techniques, topologies, and applications. **IEEE Transactions on Power Electronics**, v. 32, n. 12, p. 9143–9178, dez. 2017.
- Fuzato, G. H. F.; Aguiar, C. R.; Ottoboni, K. d. A.; Bastos, R. F.; Machado, R. Q. Voltage gain analysis of the interleaved boost with voltage multiplier converter used as electronic interface for fuel cells systems. **IET Power Electronics**, v. 9, n. 9, p. 1842–1851, 2016.

- Gregoire, L.; Al-Haddad, K.; Nanjundaiah, G. Hardware-in-the-loop (HIL) to reduce the development cost of power electronic converters. In: **India International Conference on Power Electronics 2010 (IICPE2010)**. New Delhi: IEEE, 2011. p. 1–6.
- Hadizadeh, A.; Hashemi, M.; Labbaf, M.; Parniani, M. A matrix-inversion technique for FPGA-based real-time EMT simulation of power converters. **IEEE Transactions on Industrial Electronics**, v. 66, n. 2, p. 1224–1234, fev. 2019.
- Hager, W. W. Updating the inverse of a matrix. **SIAM Review**, v. 31, n. 2, p. 221–239, 1989.
- Hart, D. W. **Power Electronics**. 1st. ed. New York: McGraw-Hill, 2011.
- He, J.; Li, Y. W.; Blaabjerg, F.; Wang, X. Active harmonic filtering using current-controlled, grid-connected DG units with closed-loop power control. **IEEE Transactions on Power Electronics**, v. 29, n. 2, p. 642–653, fev. 2014.
- Herrera, L.; Li, C.; Yao, X.; Wang, J. FPGA-based detailed real-time simulation of power converters and electric machines for EV HIL applications. **IEEE Transactions on Industry Applications**, v. 51, n. 2, p. 1702–1712, mar. 2015.
- Huang, Z.; Dinavahi, V. A fast and stable method for modeling generalized nonlinearities in power electronic circuit simulation and its real-time implementation. **IEEE Transactions on Power Electronics**, v. 34, n. 4, p. 3124–3138, abr. 2019.
- Hui, S. Y. R.; Christopoulos, C. A discrete approach to the modeling of power electronic switching networks. **IEEE Transactions on Power Electronics**, v. 5, n. 4, p. 398–403, out. 1990.
- Hui, S. Y. R.; Morrall, S. Generalised associated discrete circuit model for switching devices. **IEE Proceedings - Science, Measurement and Technology**, v. 141, n. 1, p. 57–64, jan. 1994.
- Li, J.; Liu, J. A negative-output high quadratic conversion ratio DC–DC converter with dual working modes. **IEEE Transactions on Power Electronics**, v. 34, n. 6, p. 5563–5578, jun. 2019.
- Li, X.; Guo, L.; Li, Y.; Hong, C.; Zhang, Y.; Guo, Z.; Huang, D.; Wang, C. Flexible interlinking and coordinated power control of multiple DC microgrids clusters. **IEEE Transactions on Sustainable Energy**, v. 9, n. 2, p. 904–915, abr. 2018.
- Liu, C.; Ma, R.; Bai, H.; Gechter, F.; Gao, F. A parallel solver to the three-level VSC modeling for HIL application. In: **2018 IEEE Transportation Electrification Conference and Expo (ITEC)**. Long Beach: IEEE, 2018. p. 108–113.
- Matar, M.; Iravani, R. FPGA implementation of the power electronic converter model for real-time simulation of electromagnetic transients. **IEEE Transactions on Power Delivery**, v. 25, n. 2, p. 852–860, abr. 2010.
- Najm, F. N. **Circuit Simulation**. 1st. ed. Hoboken: John Wiley & Sons, 2010.
- Ould-Bachir, T.; Blanchette, H. F.; Al-Haddad, K. A network tearing technique for FPGA-based real-time simulation of power converters. **IEEE Transactions on Industrial Electronics**, v. 62, n. 6, p. 3409–3418, jun. 2015.

- Ould-Bachir, T.; Dufour, C.; Bélanger, J.; Mahseredjian, J.; David, J. Effective floating-point calculation engines intended for the FPGA-based HIL simulation. In: **2012 IEEE International Symposium on Industrial Electronics**. Hangzhou: IEEE, 2012. p. 1363–1368.
- Pejović, P.; Maksimović, D. A method for fast time-domain simulation of networks with switches. **IEEE Transactions on Power Electronics**, v. 9, n. 4, p. 449–456, jul. 1994.
- Peng, F. Z. Z-source inverter. **IEEE Transactions on Industry Applications**, v. 39, n. 2, p. 504–510, mar. 2003.
- Peng, F. Z.; Shen, M.; Qian, Z. Maximum boost control of the Z-source inverter. **IEEE Transactions on Power Electronics**, v. 20, n. 4, p. 833–838, jul. 2005.
- Pong, M. H.; Jackson, R. D. Computer-aided design of power electronic circuits. **IEE Proceedings B – Electric Power Applications**, v. 132, n. 6, p. 301–306, nov. 1985.
- Qazi, A.; Hussain, F.; Rahim, N. A.; Hardaker, G.; Alghazzawi, D.; Shaban, K.; Haruna, K. Towards sustainable energy: A systematic review of renewable energy sources, technologies, and public opinions. **IEEE Access**, v. 7, p. 63837–63851, 2019.
- Qian, W.; Peng, F. Z.; Cha, H. Trans-Z-source inverters. **IEEE Transactions on Power Electronics**, v. 26, n. 12, p. 3453–3463, dez. 2011.
- Rashid, M. H.; Kumar, N.; Kulkarni, A. R. **Power Electronics – Devices, Circuits and Applications**. 4th. ed. Essex: Pearson Education, 2014.
- Ruan, X.; Wang, X.; Pan, D.; Yang, D.; Li, W.; Bao, C. **Control Techniques for LCL-Type Grid-Connected Inverters**. 1st. ed. Singapore: Springer, 2018.
- Song, Y.; Chen, L.; Chen, Y.; Huang, S. A general parameter configuration algorithm for associate discrete circuit switch model. In: **2014 International Conference on Power System Technology**. Chengdu: IEEE, 2014. p. 956–961.
- ST. **High-performance and DSP with DP-FPU, 32-bit ARM Cortex-M7 550 MHz MCU**. Geneva, CH, 2020. Rev. 2.
- Teodorescu, R.; Liserre, M.; Rodríguez, P. **Grid Converters for Photovoltaic and Wind Power Systems**. 1st. ed. West Sussex: John Wiley & Sons, 2011.
- Wang, K.; Xu, J.; Li, G.; Tai, N.; Tong, A.; Hou, J. A generalized associated discrete circuit model of power converters in real-time simulation. **IEEE Transactions on Power Electronics**, v. 34, n. 3, p. 2220–2233, mar. 2019.
- WHITEMAN, A.; ESPARRAGO, J.; RUEDA, S.; ELSAYED, S.; ARKHIPOVA, I. **Renewable Energy Statistics 2019**. New York, 2019. Disponível em: <www.irena.org/Publications>. Acesso em: 03/07/2019.