
Contribuição ao uso de métodos formais no estudo de circuitos de eletrônica de potência

Márcia Verônica Costa Miranda

Tese de Doutorado submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Processamento da Informação

Antonio Marcus Nogueira Lima, Dr.
Orientador

Campina Grande, Paraíba, Brasil
©Márcia Verônica Costa Miranda, Outubro de 2003



M672c Miranda, Marcia Veronica Costa
Contribuicao ao uso de metodos formais no estudo de circuitos de eletronica de potencia / Marcia Veronica Costa Miranda. - Campina Grande, 2003.
178 f.

Tese (Doutorado em Engenharia Eletrica) - Universidade Federal de Campina Grande, Centro de Ciencias e Tecnologia.

1. Sistemas Híbridos 2. Circuito de Eletronica de Potencia 3. Metodos Formais 4. Automatos Híbridos 5. Verificacao Formal 6. Reprojeto do Controlador 7. Tese I. Lima, Antonio Marcus Nogueira II. Universidade Federal de Campina Grande - Campina Grande (PB) III. Título

CDU 621.3.049(043)


CONTRIBUIÇÃO AO USO DE MÉTODOS FORMAIS NO ESTUDO DE
CIRCUITOS DE ELETRÔNICA DE POTÊNCIA

MÁRCIA VERÔNICA COSTA MIRANDA


Tese Aprovada em 24.10.2003




PROF. ANTONIO MARCUS NOGUEIRA LIMA, Dr., UFCG
Orientador



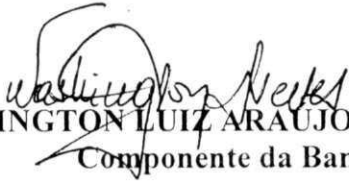
PROF. JOSÉ EDUARDO RIBEIRO CURY, Ph.D., UFSC
Componente da Banca



PROF. FÁBIO GONÇALVES JOTA, Ph.D., UFMG
Componente da Banca



PROF. EDISON ROBERTO CABRAL DA SILVA, Dr.Ing., UFCG
Componente da Banca



PROF. WASHINGTON LUIZ ARAUJO NEVES, Ph.D., UFCG
Componente da Banca

CAMPINA GRANDE – PB
OUTUBRO - 2003

Dedicatória

Ao meu amado esposo Érico Miranda, fonte constante de amor, companherismo e compreensão, sem ele nada disso teria sido possível.

A meus filhos Caio, André e Gabriela, motivos de grande orgulho e alegria.

*“Mas é claro que o sol
Vai voltar amanhã
Mais uma vez, eu sei
Escuridão já vi pior
De endoidecer gente sã
Espera que o sol já vem
Se você quiser alguém em quem confiar
Confie em si mesmo
Quem acredita sempre alcança
Nunca deixe que lhe digam
Que não vale a pena acreditar no sonho que se tem
Ou que seus planos nunca vão dar certo
Ou que você nunca vai ser alguém ”*

Renato Russo, cantor e compositor brasileiro (1960-1996).

Agradecimentos

A Deus, por guiar a minha vida e sempre iluminar o meu caminho.

Ao meu marido, Érico Miranda, por não me deixar fraquejar, pela presença constante, companheirismo, carinho, amor e incentivo em todos os momentos de minha vida, alegres ou difíceis. Partilhamos juntos mais uma conquista.

Aos meus filhos Caio, André e Gabriela, pelo carinho, alegria, inspiração e por suportarem, pacientemente, minha ausência durante estes anos.

Aos meus pais, D. Zefinha e Seu Paizinho, por valorizarem o conhecimento como a maior fonte de riqueza do ser humano e pelo apoio e incentivo nas horas de aflições.

Ao meu irmão, Jefferson, pela torcida e incentivo.

Ao Professor Antonio Marcus, por estender-me a mão em momentos difíceis e pela orientação, sem a qual este trabalho não seria realizado.

Aos componentes da banca, pelas contribuições, sugestões e por se disporem a participar da avaliação desta tese.

Aos funcionários da COPELE, Ângela, Pedro e Eleonora, pelo apoio recebido.

Aos professores e funcionários do Departamento de Ciências Fundamentais e Sociais, do Centro de Ciências Agrárias, UFPB, pelo companheirismo.

Aos amigos que adquiri ao longo destes anos, por tornarem o ambiente de trabalho alegre, pelas conversas, pelo incentivo recebido. Em especial, aos amigos Alexandre Cunha e Gustavo Arruda pela ajuda e apoio dado nestes últimos anos.

Ao PICDT, por financiar este trabalho.

Enfim, a todos aqueles que, direta ou indiretamente, me ajudaram para a realização deste trabalho.

A vocês, agradeço.

Resumo

Esta tese propõe uma alternativa às técnicas de simulação, normalmente, utilizadas para a análise de circuitos de eletrônica de potência (comumente constituídos de processadores de potência e um controlador, chamado de conversor), as quais não garantem a correção global dos resultados obtidos para avaliar, por exemplo, o efeito de intervalos de condições iniciais e variação paramétrica nos componentes do circuito. Esta alternativa compreende a utilização de métodos formais para obter o modelo de circuitos de potência e, a partir deste modelo, empregar técnicas para estudo, verificação do comportamento destes sistemas e também para o reprojeto de um novo controlador.

Sabendo-se que um circuito de potência exibe, em seu comportamento, dinâmicas contínuas e discretas, típicas de sistemas híbridos, esta tese introduz um procedimento sistemático para obter o modelo matemático que captura o comportamento híbrido destes sistemas, na forma de autômatos híbridos. O modelo de autômato híbrido obtido é baseado na representação da estrutura do conversor resultante de um método de montagem de blocos de elementos que representam os componentes deste conversor. O procedimento de modelagem permite tratar refinamentos de dispositivos semi-condutores do circuito e circuitos com incertezas nos valores do sinal de entrada, considerando tanto entrada constante como entrada variante no tempo.

Uma vez obtido o autômato híbrido, representando o modelo formal do circuito de eletrônica de potência, foram implementados métodos de verificação de propriedades através da análise de alcançabilidade sobre o próprio espaço de estado híbrido do sistema. Os procedimentos de verificação, implementados nesta tese, possibilitam verificar propriedades, tais como: a compatibilidade dos valores de corrente e tensão nos componentes do circuito com os limites estabelecidos; a ocorrência de faltas ocasionadas quando os valores de corrente ou tensão nos dispositivos semi-condutores ultrapassam os limites de ruptura; o nível de tensão sobre o qual uma chave comutaria; a compatibilidade dos valores de tensão na carga com um dado valor de referência de projeto; a conformidade dos valores de tensão e corrente em regime permanente como especificados, dentre outros.

Foram implementados procedimentos para automaticamente reprojeter um novo controlador para satisfazer a uma dada especificação. O objetivo do reprojeto do controlador é garantir que todas as configurações alcançáveis do sistema levem a um conjunto de configurações seguras, como especificadas em seu projeto. Através desta abordagem de controle de sistemas híbridos, abre-se espaço para tratar uma ampla gama de problemas comuns na automação industrial, particularmente, circuitos de eletrônica de potência.

Abstract

This thesis presents an alternative method to the simulation process of power electronic circuits, composed by power processors and a controller, named conversor. Computer simulations are normally used to analyze the behavior of power electronic circuits, and do not guarantee a reliable adjustment to the results attained while evaluating, for example, the effect of either any set of initial conditions or parametric deviation in the circuit components.

The alternative technique proposed in this work comprehends the use of formal methods to obtain a model for power circuits, to analyze and verify the behavior of these systems, and to redesign a new controller from a set of desired specifications as well.

Power electronic circuits exhibit a continuous and discrete dynamics, which is a typical feature of hybrid systems. This work introduces a systematic procedure to derive a mathematical model that captures the hybrid behavior of these systems as hybrid automata. The model of the hybrid automaton obtained is based on the representation of the structure of a converter, as a result of an assembly method of blocks that represent the components of such converter.

The modeling procedure allows dealing with the improvement of semiconductor circuit devices as well as the existence of uncertainties in the input signal, for both constant and time variant input.

Once obtained the hybrid automaton, which represents the formal model of the power circuit, methods are developed to verify its properties by using the reachability analysis on the hybrid system state-space itself.

The methods of verification make possible the analysis of properties such as the compatibility of current and voltage values in the circuit components under specified limits, the fault occurrences when the current and voltage values exceeds the rupture limits, the voltage level necessary to turn on a semiconductor device, the compliance of the steady-state current and voltage values as desired, etc.

Finally, methods to automatically redesign a controller were implemented. The controller goal is to guarantee that every reachable state of the system will converge to a set of reliable configurations, as specified in the design. By using this approach it is possible to get around several problems that may occur in the industrial automation, and in particular, concerning the power electronic circuits.

Conteúdo

1	Introdução	1
1.1	Sistemas Híbridos e Circuitos de Eletrônica de Potência	3
1.2	Objetivos do Trabalho e Motivação	7
1.3	Organização do Trabalho	9
2	Formalismos Matemáticos para Sistemas Híbridos	10
2.1	Sistemas Híbridos	10
2.2	Estudos em sistemas híbridos	12
2.3	Autômatos Híbridos	15
2.3.1	Definição Formal	16
2.3.2	Composição	17
2.3.3	Exemplo: Aquecedor e Monitor	19
2.3.4	Configurações e Trajetórias	22
2.4	Cálculo dos estados alcançáveis	23
2.4.1	Alcançabilidade de autômatos híbridos	23
2.5	Método de Aproximação	31
2.6	Extensão para entrada incerta	37
2.6.1	Algoritmo de alcançabilidade com entrada incerta	38
2.7	Verificação de propriedades	42
2.8	Conclusões	44
3	Modelagem de circuitos de eletrônica de potência	46
3.1	Circuitos Eletrônicos de Potência	46
3.1.1	Dispositivos de Eletrônica de Potência	48
3.1.2	Dispositivos com disparo e bloqueio espontâneos - Diodo de Potência	49
3.1.3	Dispositivos com disparo controlado e bloqueio espontâneo	50
3.1.4	Dispositivos com disparo e bloqueio controlados	51

3.1.5	Modelos de autômatos híbridos	54
3.2	Técnica para representação da estrutura de conversores	58
3.2.1	Elementos pré-caracterizados	60
3.2.2	Procedimento para representação da estrutura do conversor	62
3.2.3	Linguagem para os elementos pré-caracterizados	64
3.2.4	Exemplos de aplicação do procedimento de montagem	65
3.3	Procedimento de modelagem, utilizando autômatos híbridos	70
3.4	Aplicação - Conversor Buck	75
3.4.1	Etapas de Modelagem	77
3.4.2	Autômatos Híbridos Resultantes	80
3.4.3	Resultados da Simulação	81
3.4.4	Simulação de circuitos de eletrônica de potência com outras ferramentas	83
3.4.5	Variação paramétrica em componentes do circuito	87
3.4.6	Circuito conversor com histerese na corrente da carga	93
3.4.7	Processo de refinamento - exemplo com diodo	96
3.4.8	Entrada variante no tempo - Aplicação	97
3.5	Conclusões	103
4	Verificação de propriedades de circuitos de eletrônica de potência	105
4.1	Verificação Formal	105
4.2	Métodos para Verificação Formal	108
4.2.1	Verificação Automática	108
4.2.2	Sistemas de Prova Dedutiva	110
4.2.3	Verificação automática - lógica ACTL	111
4.3	Análise de Alcançabilidade	113
4.3.1	Verificação de propriedades - Problema e Algoritmos	115
4.3.2	Algoritmo de Verificação para Trás	125
4.4	Estudo de Caso	125
4.4.1	Espaço alcançável	126
4.4.2	Verificação de propriedades - conversor Buck	132
4.5	Conclusões	138
5	Reprojeto de controladores de sistemas híbridos	140
5.1	Introdução	140
5.1.1	Comportamento Restritivo	143
5.2	Problema de Reprojeto	144

5.2.1	Conjunto Invariante Máximo	144
5.3	Procedimentos para reprojeto do controlador	149
5.3.1	Algoritmo para cálculo do conjunto invariante máximo	150
5.4	Controlador Chaveado	150
5.5	Estudo de Caso	151
5.5.1	Conversor com controle por histerese da tensão da carga	151
5.5.2	Conversor com controle por histerese da corrente da carga	152
5.6	Conclusões	152
6	Conclusões e propostas para futuros trabalhos	155
6.1	Contribuições	155
6.1.1	Modelagem	156
6.1.2	Verificação de Propriedades	156
6.1.3	Reprojeto de Controladores	158
6.2	Futuros Trabalhos	158
A	Glossário	160
	Bibliografia	168

Lista de Símbolos e Abreviaturas

Símbolo	Descrição
\mathbb{N}	Números Naturais
\mathbb{R}	Números Reais
\mathcal{T}	Domínio do tempo
$L(c)$	Linguagem formal que representa o circuito c
Sistemas Contínuos	
ξ	Solução de $\dot{x} = f(x, u)$, representando a trajetória
$\xi_{x, \mu}$	Trajétória começando do ponto x sob entrada μ
$\delta(F)$	Conjunto alcançável de F
$\delta_r(F)$	Sucessores do conjunto F no tempo r
$\delta_{[0, r]}(F)$	Sucessores do conjunto F no intervalo de tempo $[0, r]$
$\Delta(F)$	Conjunto alcançável para trás do conjunto F
Autômato Híbrido	
δ_c	Operador Sucessor contínuo
δ_d	Operador Sucessor discreto
$\delta_{qq'}$	Operador Sucessor discreto de q para q'
π	Operador predecessor uma-etapa
π_q^∞	Operador predecessor ilimitado do estado discreto q
\mathcal{U}_q	Operador Until no estado discreto q
μ^*	Entrada incerta do sistema híbrido
Método de Aproximação	
$\text{bloat}(C, \mathcal{V})$	Expande o poliedro C uma quantidade \mathcal{V}
$\text{conv}(F)$	Casca convexa do conjunto F
$\text{grid}_o(C)$	Super-aproximação do poliedro convexo C
$\text{grid}_u(C)$	Sub-aproximação do poliedro convexo C

Lista de Tabelas

3.1	Tabela com elementos pré-caracterizados	59
3.2	Linguagem representativa das famílias de elementos	64
3.3	Modos de Operação do Conversor	78
3.4	Modos de Operação do Controlador	79
3.5	Descrição das locações do autômato híbrido do conversor Buck	81
3.6	Descrição das locações do autômato híbrido do conversor com variação em L	90
3.7	Descrição das locações do autômato híbrido do conversor com variação em L	93
3.8	Descrição das locações do autômato híbrido do conversor com histerese na corrente da carga	95
3.9	Descrição das locações do autômato híbrido do conversor com histerese na corrente da carga	97
3.10	Descrição das locações do autômato híbrido do conversor Buck	102
3.11	Modos de Operação do Controlador	102
4.1	Tabela com operadores ACTL	112
4.2	Resultado da verificação de propriedades no sistema conversor	138

Lista de Figuras

2.1	Representação gráfica de um autômato híbrido	17
2.2	Autômato simples de um aquecedor	19
2.3	Autômato Monitor	20
2.4	Autômato resultante da composição entre o autômatos do aquecedor e monitor .	21
2.5	Sucessores contínuos de F	25
2.6	Sucessores discretos de F pela transição de q para q'	25
2.7	Predecessores contínuos e discretos de F	31
2.8	(a) Uma grade G_β (b) Poliedros ortogonais de grade	33
2.9	Aproximação do conjunto F	34
2.10	Ilustração da técnica de aproximação de Asarin-Dang	36
2.11	Conjunto alcançável de um sistema dinâmico com entrada.	38
2.12	Trajetórias a partir de γ com entrada incerta.	40
2.13	Super-aproximação do conjunto alcançável $\delta_\epsilon(F)$	42
3.1	Diagrama de Bloco de um sistema eletrônico de potência	47
3.2	Diodo: (a) Símbolo, (b) Característica idealizada (c) Característica I-V	50
3.3	SCR: (a) Símbolo, (b) Característica idealizada (c) Característica I-V	50
3.4	(a) SCR e (b) Triac	51
3.5	Interruptor genérico controlável	51
3.6	GTO: (a) Símbolo, (b) Características idealizadas (c) Característica I-V.	52
3.7	BJT: (a) Símbolo, (b) Características idealizadas (c) Característica I-V.	52
3.8	MOSFET: (a) Símbolo, (b) Características idealizadas (c) Característica I-V.	53
3.9	IGBT: (a) Símbolo, (b) Características I-V idealizadas.	54
3.10	Modelo do autômato híbrido representando um diodo de potência.	56
3.11	Modelo do autômato híbrido representando um tiristor.	56
3.12	Modelo do autômato híbrido representando um GTO.	57
3.13	Modelo do autômato híbrido representando um BJT.	57
3.14	Modelo do autômato híbrido representando um MOSFET.	57

3.15	Modelo do autômato híbrido representando um IGBT.	57
3.16	Elementos que caracterizam transferência de energia: (a) transfere tensão e (b) transfere corrente	59
3.17	Representação gráfica do elemento <i>Fontes</i>	60
3.18	Representação gráfica do elemento <i>Ganho</i>	60
3.19	Representação gráfica dos elementos <i>Transformadores</i> e <i>Giradores</i>	61
3.20	Representação gráfica dos <i>Elementos de conexão</i>	61
3.21	Representação gráfica dos elementos <i>Integradores</i>	62
3.22	Circuito conversor Buck	66
3.23	Etapas de construção da estrutura de um circuito conversor	67
3.24	Circuito conversor Boost	68
3.25	Etapas de construção da estrutura de um circuito conversor Boost	69
3.26	Diagrama do processo de refinamento	71
3.27	Circuito conversor Buck	75
3.28	Estágio 1 - Descarga do capacitor C	76
3.29	Estágio 2 - Armazenamento de energia na indutância L	76
3.30	Estágio 3 - Transferência de energia da indutância L para o capacitor C	77
3.31	Exemplo de controle por histerese	80
3.32	Autômato híbrido para o circuito conversor Buck	81
3.33	Autômato híbrido para o controlador do conversor Buck	81
3.34	Formas de ondas de corrente (i_L) e tensão (v_C).	82
3.35	Formas de ondas de corrente (i_L) e tensão (v_C).	82
3.36	Formas de ondas de corrente i_L e tensão v_C para o estágio inicial de simulação, locação Loc_2	83
3.37	Formas de ondas de corrente i_L e tensão v_C para estágio representado pela locação Loc_3	83
3.38	Formas de ondas de tensão (v_C) e corrente (i_L) após estado representado pela locação Loc_1	84
3.39	Formas de ondas de tensão (v_C) em regime permanente	84
3.40	Formas de ondas de Corrente i_L em regime permanente	84
3.41	Formas de ondas de tensão (v_C) e corrente (i_L) obtidas pelo PSPICE.	87
3.42	Exemplo de linearização por partes do componente L de um dado circuito.	89
3.43	Autômato híbrido para o conversor com variação paramétrica em L	90
3.44	Formas de ondas de tensão (v_C) e corrente (i_L) para verificar o comportamento do conversor para variação do parâmetro L	91

3.45	Autômato híbrido para o conversor com variação paramétrica em R	93
3.46	Formas de ondas de tensão (v_C) para verificar o comportamento do conversor para $R = 5\Omega$ e para variação do valor de R	94
3.47	Autômato híbrido para o conversor com histerese na corrente da carga.	95
3.48	Autômato híbrido para o controlador.	95
3.49	Formas de ondas de tensão e corrente na carga e corrente na indutância (i_L).	96
3.50	Diodo com comportamento mais refinado.	96
3.51	Autômato do Diodo com comportamento mais refinado.	97
3.52	Autômato do conversor após refinamento do diodo.	98
3.53	Circuito conversor Boost	98
3.54	Modos de operação do conversor Boost	100
3.55	Autômato híbrido para o circuito conversor Boost	101
3.56	Autômato híbrido para o controlador do conversor Buck	102
3.57	Formas de ondas de corrente (i_L) e tensão (v_C)	103
4.1	Verificação formal no processo de desenvolvimento de um circuito de eletrônica de potência.	108
4.2	Família de lógicas computacionais.	109
4.3	Exemplo de representação de um poliedro ortogonal.	114
4.4	Ilustração do problema de verificação de propriedades, dados F e o conjunto S	115
4.5	Conjunto de estados alcançáveis do circuito controlado via histerese na tensão de saída.	127
4.6	Conjunto de estados alcançáveis do circuito com variação no parâmetro L	128
4.7	Conjunto de estados alcançáveis do circuito com variação na resistência da carga (R).	128
4.8	Conjunto de estados alcançáveis do circuito considerando entrada incerta na fonte de tensão.	129
4.9	Conjunto de estados alcançáveis do circuito com controle via histerese na corrente.	130
4.10	Conjunto de estados alcançáveis do circuito com entrada variante no tempo.	132
4.11	Verificação de propriedade de segurança - Sistema seguro	133
4.12	Verificação de propriedade de segurança - Sistema não seguro	133
4.13	Verificação de propriedade de segurança - Sistema seguro	134
4.14	Verificação de propriedade de alcançabilidade - Estado alcançável	135
4.15	Verificação de propriedade de alcançabilidade - Estado não-alcançável	135
4.16	Verificação de propriedade de alcançabilidade - Estado não-alcançável	136
4.17	Verificação de propriedade de invariância - Propriedade não satisfeita	137

4.18	Verificação de propriedade de invariância - Propriedade satisfeita	137
4.19	Verificação de propriedade de invariância - Propriedade satisfeita	137
5.1	Diagrama representando a utilização de autômatos híbridos para análise e controle de sistemas híbridos	141
5.2	Estrutura de controle para sistemas chaveados	142
5.3	Diagrama representando procedimento de reprojeção do supervisor	143
5.4	Cálculo de S'_q através da caracterização dos operadores π_q^∞ e \mathcal{U}_q	146
5.5	Cálculo de S'_1 de S_1 e S_2	147
5.6	(a) Operador π para transições controláveis (T_c) (b) Operador π para transições controláveis (T_c) e não-controláveis (T_u)	148
5.7	Conjunto P^* da locação 2.	151
5.8	Conjunto P^* da locação 3.	151
5.9	Controlador obtido após execução do procedimento de reprojeção	152
5.10	Conjunto P^* da locação 2.	153
5.11	Conjunto P^* da locação 3.	153
5.12	Controlador obtido após execução do procedimento de reprojeção	154

Capítulo 1

Introdução

A eletrônica de potência é o ramo da eletrônica cujos dispositivos e circuitos trabalham com potências de unidades de watt a dezenas de megawatts. As áreas de aplicações dos dispositivos têm crescido de forma significativa, principalmente, devido à elevada demanda de fontes de alimentação de alto rendimento, à disseminação do uso de conversores no acionamento de máquinas elétricas e na busca de melhoria da qualidade de energia elétrica, entre outras.

A crescente utilização da eletrônica de potência nas mais diversas áreas da tecnologia moderna faz com que sejam também crescentes as exigências de qualidade e confiabilidade dos circuitos eletrônicos utilizados nestas aplicações. Além disto, o custo associado à implantação de uma linha de produção de um novo circuito de eletrônica de potência exige que a equipe de projeto realize exaustivos ensaios, simulações e testes operacionais, sob um amplo conjunto de condições de operação, de modo a verificar se o comportamento do mesmo atende às especificações de projeto e também avaliar aspectos relacionados à durabilidade e à confiabilidade do circuito.

Os circuitos de eletrônica de potência, aqui tratados, englobam o conversor e os circuitos de controle a ele associados para garantir o funcionamento conforme requerido pelo projetista. O método mais utilizado para validar o projeto de um circuito é testar sua operação, a partir da simulação digital de um modelo que mais aproximadamente o descreva. A simulação deve considerar as condições de operação esperadas para o circuito, de modo a ser genérica o suficiente para descrever seu comportamento global. Embora a simulação digital seja uma ferramenta excelente para avaliar um comportamento desejado, e adequada para detectar alguns erros de projeto, certamente, não é uma técnica que garanta total avaliação da operação de um circuito complexo, uma vez que é difícil considerar todos os cenários possíveis de serem testados.

Simulações numéricas são utilizadas para obter uma aproximação do comportamento do sistema para um determinado estado. Para simular um circuito, são necessários a especificação de sua estrutura, bem como as suas condições iniciais de operação. Todavia, na prática, as condições

iniciais não são exatamente conhecidas, mas sabe-se somente que seus valores pertencem a um determinado intervalo. Conseqüentemente, ao invés de uma única trajetória, como as observadas nos resultados das simulações, é necessário considerar um número infinito de trajetórias. Neste caso, é praticamente impossível simular conjuntos de trajetórias, pois seria necessário executar um número elevado de simulações.

Um outro aspecto que poderia ser citado, em relação à dificuldade do uso de técnicas de simulação digital, refere-se ao momento em que se deseja analisar circuitos com incerteza na fonte (corrente ou tensão) de entrada. A incerteza de entrada torna difícil a análise do circuito, por ser preciso caracterizar todos os possíveis comportamentos resultantes de um intervalo de valores de entrada do sistema. É difícil simular um sistema com todas as entradas, obtendo conjuntos de trajetórias, pois seria necessário simulações exaustivas para isto. Nestes casos, a abordagem de simulação não é suficiente para o estudo de circuitos de eletrônica de potência, uma vez que as soluções obtidas não provêm uma garantia absoluta de que o sistema satisfaz aos requisitos determinados pelo projetista.

Como alternativa às técnicas de simulação, utilizam-se métodos formais matemáticas para desenvolvimento de sistemas e, neste trabalho, os sistemas que estamos estudando são os circuitos de eletrônica de potência. Técnicas ou métodos formais foram utilizados, inicialmente, para definição de requisitos e especificação de sistemas. Entretanto, as vantagens e os benefícios relacionados com a utilização de procedimentos e modelos matemáticos são evidentes (relevantes) e foram estendidos para serem utilizados na verificação, análise e síntese de controladores de sistemas complexos. Por *métodos formais* entende-se o uso de notações e métodos matemáticos na especificação de sistemas e suas propriedades de maneira precisa e sem ambigüidades (CLARKE; WING, 1996; SAIEDIAN, 1996). Métodos formais são formas rigorosas utilizadas para verificação e análise de modelos matemáticos de circuitos, provendo uma descrição formal do seu comportamento através de um procedimento sistemático que determina se todos os comportamentos analisados satisfazem à especificação do projetista e, portanto, estão corretos.

As principais vantagens da utilização de métodos matemáticos no projeto de sistemas são a determinação e, conseqüentemente, eliminação de ambigüidades, inconsistências e erros residuais de especificação de projeto, evitando, desta forma, a necessidade de retomar atividades relativas a fases anteriores no ciclo de desenvolvimento do projeto. Este argumento enfatiza o caráter econômico associado à concepção, ao desenvolvimento e à utilização de métodos matemáticos que provavelmente é determinante na disseminação destes métodos e sua adoção pela indústria. É conhecido o fato de que uma parcela bastante significativa do custo de desenvolvimento de sistemas é relativa às atividades de teste, simulação e manutenção, o que vem a corroborar com a utilização de métodos rigorosos de desenvolvimento de sistemas.

Uma outra vantagem que podemos destacar é a possibilidade de especificação de métodos algorítmicos de análise e verificação de comportamentos destes sistemas. Uma vez que a semântica de uma especificação formal é única, pode-se submeter um conjunto de expressões, em um dado formalismo, a processos sistemáticos de verificação de consistência ou completude sob certos aspectos. Por exemplo, na verificação formal de circuitos de eletrônica de potência, alguns aspectos possíveis de serem considerados são: a verificação do nível de tensão sobre o qual uma chave comutaria, a compatibilidade dos valores de tensão na carga com um dado valor de referência de projeto, a conformidade dos valores de tensão e corrente em regime permanente como especificados, dentre outros.

A necessidade de projetos corretos em aplicações críticas em segurança, aliada com o custo dos produtos implementados posteriormente, são os dois principais fatores determinantes no uso de métodos formais no projeto de circuitos de eletrônica de potência. Além disso, tem-se observado um aumento na complexidade de projetos de circuitos de eletrônica de potência, implicando numa necessidade maior de garantias na avaliação da confiabilidade destes projetos. A utilização de técnicas de simulação, como ferramenta única para validação de projetos de circuitos, não garantirá a sua correção total e confiabilidade, uma vez que simulações só verificarão uma amostra dos possíveis comportamentos destes sistemas.

Além de prover uma redução no tempo de simulação e teste, técnicas de verificação formal possibilitam aumentar o nível de confiança no projeto de sistemas, constituindo-se em requisito de extrema importância quando da construção de sistemas críticos, como circuitos de eletrônica de potência, nos quais falhas podem levar a perdas significativas de recursos materiais ou até mesmo vidas humanas.

1.1 Sistemas Híbridos e Circuitos de Eletrônica de Potência

O enfoque desta tese consiste no estudo de circuitos de eletrônica de potência através de técnicas matemáticas como uma alternativa às simulações digitais, que são as técnicas numéricas de análise comumente utilizadas nesta área.

Circuitos de eletrônica de potência são redes compostas de componentes eletrônicos e dispositivos semicondutores e formam uma parte central de muitos sistemas de controle, regulando o fluxo de potência entre uma fonte e uma carga.

Os dispositivos semicondutores de potência são componentes essenciais em um conversor estático e suas características interferem nas características finais do equipamento. De modo

geral, esses dispositivos operam como interruptores e apresentam um estado de condução (impedância baixa) e outro de bloqueio (impedância elevada). A mudança de estado é denominada de comutação (disparo e bloqueio) ou de chaveamento (SILVA, 1999). A ocorrência de uma mudança de estado nestes dispositivos semicondutores fará com que a rede, que representa o circuito, tenha configuração variável e o circuito de eletrônica de potência poderá ser classificado como um sistema híbrido.

Um *sistema híbrido* é um sistema dinâmico cujo comportamento é descrito através da dinâmica contínua com a intervenção de eventos discretos (LEMMON; MARKOVSKY, 1999). No caso de circuitos de eletrônica de potência, os eventos discretos correspondem, por exemplo, à abertura e ao fechamento de chaves e as dinâmicas contínuas correspondem às equações diferenciais, representando cada uma das configurações do circuito, conforme os estados das chaves.

O principal problema no comportamento de circuitos de eletrônica de potência pode ser caracterizado pelas interações complexas entre as dinâmicas contínuas e os eventos discretos que os constituem. Dispositivos, tais como geradores e transformadores, possuem comportamento contínuo que é representado por equações diferenciais. Outros dispositivos exibem comportamento discreto, tais como chaves e relés. As dinâmicas, neste caso, são descritas por regras lógicas que dependem das entradas das dinâmicas contínuas.

Os primeiros modelos desenvolvidos para estudar os sistemas híbridos tratavam, separadamente, as dinâmicas dirigidas a eventos e as dinâmicas contínuas. As técnicas disponíveis para a modelagem, análise e síntese de controladores, bem como as ferramentas associadas a estas técnicas não eram capazes de tratar conjuntamente tais dinâmicas.

A formalização de sistemas híbridos apresenta certa dificuldade, na medida em que envolve processos contínuos e discretos. Assim, para abordar estes sistemas, um modelo deve incluir aspectos de cada um destes "mundos", representados por modelos matemáticos distintos: *equações diferenciais*, *equações de estado* e *Bond Graph* (KARNOPP; ROSENBERG, 1975; KARNOPP; MARGOLIS; ROSENBERG, 1990) são comumente utilizados para tratar dos aspectos contínuos, enquanto que *máquinas de estado finito*, *Grafset* ou *redes de Petri* são utilizadas para representar os aspectos discretos. Assim, sistemas de natureza híbrida são tratados como entidades puramente contínuas ou puramente discretas, de acordo com o objetivo a ser alcançado (NERODE; KOHN, 1993).

A descrição de sistemas híbridos, considerando apenas seus comportamentos discretos ou contínuos, possui vantagens e desvantagens. Uma das vantagens é que este tipo de abordagem pode prover modelos mais simplificados do sistema, facilitando o desenvolvimento de métodos de análise e projeto do controlador. Uma desvantagem que é o fato de omitir aspectos do comportamento do sistema, tanto para abstração para um modelo discreto como para um modelo

contínuo, que possam comprometer uma análise mais completa ou a construção de um sistema de controle mais preciso. Entretanto, na maioria dos casos não triviais, estas duas dinâmicas interagem de forma significativa, de tal forma que não podem ser separadas efetivamente por nenhum tipo de abstração e precisam ser analisadas simultaneamente. Assim, para abordar de maneira completa o comportamento de um sistema, em geral, torna-se necessário modelar e analisar conjuntamente tais dinâmicas, bem como suas interações (MALER, 2001). Isto é realizado através da utilização do modelo formal na representação de sistemas híbridos.

Conforme mencionado anteriormente, circuitos de eletrônica de potência são considerados sistemas híbridos por possuírem, em seu comportamento, características discretas e contínuas interagindo entre si. As mudanças nestes sistemas acontecem em resposta a eventos que ocorrem assincronamente (e freqüentemente não-deterministicamente) e dinâmicas que representam a evolução causal, como descrito por equações diferenciais (COFER, 1997). A natureza híbrida destes sistemas requer novas formas de análise, pois mudanças discretas não são adequadamente manipuladas através de algoritmos que tratam de dinâmicas contínuas, e abstrair comportamento de sistemas para modelos discretos pode resultar em perda de informação crítica para controle e estudo de suas propriedades (NARASIMHAN et al., 2000).

A análise e o projeto de sistemas híbridos necessitam de uma abordagem matemática apropriada. Esta abordagem deve ser simples e eficiente para lidar com descrições gerenciáveis, que possam ser analisadas com técnicas matemáticas disponíveis, e deve ser precisa para descrever os aspectos essenciais do comportamento dinâmico de um sistema (ANTSACLIS; STIVER; LEMMON, 1993; KOUTSOUKOS; ANTSACLIS, 1999).

O desenvolvimento de técnicas para obtenção de modelos que possibilitem o estudo de sistemas híbridos traz uma nova dimensão à especificação comportamental destes sistemas, possibilitando o estudo automático de suas propriedades e leis de controle. Várias abordagens têm sido propostas para modelagem, análise e síntese do controlador de sistemas híbridos. Estas abordagens se baseiam nos mais diversos modelos matemáticos, tais como Bond graphs chaveados, redes de Petri de alto nível e autômatos híbridos.

O uso de técnicas matemáticas no estudo de circuitos de eletrônica de potência possibilita executar métodos de análise de propriedades, validação do sistema e reprojeto do controlador com o objetivo de obter uma análise mais compreensiva do sistema, reduzindo os custos de implementação e manutenção, e auxiliar os projetistas destes sistemas durante as fases de especificação, desenvolvimento e projeto.

A utilização de métodos formais no contexto de eletrônica de potência, especialmente, no tratamento formal de análise de propriedades e reprojeto de controladores, ainda não foi efetivamente explorado, constituindo-se, assim, um desafio científico e inovador para esta área.

Alguns trabalhos já foram feitos neste sentido, dos quais podemos citar o trabalho de Mos-terman e Biswas (1996, 1995) que utilizam o formalismo de Bond Graph associado a autômatos finitos para modelar circuitos e o trabalho de Barnes e Lefley (1999), Lefley e Smith (2000) que trata da modelagem de controladores adaptativos, utilizando conceitos de máquinas de estado finito. Entretanto, em todos estes trabalhos não há um tratamento de análise ou validação formal do comportamento dos sistemas que modelam, restringindo-se apenas a resultados de simulação, para uma determinada condição inicial, dos modelos matemáticos por eles obtidos.

A utilização de métodos formais para verificação de propriedades já vem sendo feita em sistemas de software de forma bastante sedimentada, utilizando modelos matemáticos como autômatos finitos, redes de Petri, lógica modal, etc. Assim, como a utilização de métodos formais é considerada uma ferramenta poderosa e bastante comum na área da Ciência da Computação, este trabalho pretende utilizá-la para o estudo de circuitos de eletrônica de potência.

A forma de utilização de técnicas matemáticas para circuitos de potência prevê, inicialmente, a caracterização destes sistemas como sistemas híbridos, a identificação de suas dinâmicas contínuas e os fenômenos discretos responsáveis pelas mudanças de configuração dos circuitos. Posteriormente, como sistemas híbridos, define-se o modelo matemático que será utilizado para representar estes sistemas e os procedimentos de verificação de propriedades e reprojeto do controlador. Neste trabalho, o modelo formal de representação de sistemas híbridos utilizado são os autômatos híbridos.

Autômatos Híbridos são máquinas de estado finito generalizadas, cujos estados estão associados com um conjunto de equações diferenciais utilizadas para descrever a especificação dinâmica da evolução do sistema naquela configuração particular e cujos arcos modelam uma mudança abrupta no comportamento dinâmico do sistema, ocasionando a transição de uma configuração para outra (MIRANDA; LIMA, 2001). Entre todos os modelos disponíveis, o uso de autômato híbrido parece ser o mais adequado para representar o comportamento de circuitos de eletrônica de potência por descrever sistemas híbridos através de uma representação clara e simples, exprimindo num único formalismo as dinâmicas contínuas e discretas inerentes aos sistemas híbridos. Constitui-se uma representação adequada para exprimir as características e o comportamento do tipo de sistema que se pretende abordar, além da possibilidade de implementar, sem um tratamento de maior complexidade, os procedimentos de verificação e reprojeto do controlador que serão aqui tratados.

1.2 Objetivos do Trabalho e Motivação

O objetivo deste trabalho consiste na utilização de métodos formais no estudo de circuitos de eletrônica de potência. Métodos formais, aplicado à eletrônica de potência, possibilita responder perguntas do tipo:

- Será que existe alguma condição de operação que põe em risco algum componente do circuito ?
- O valor da tensão de saída obedece a um determinado limiar especificado ?
- Existe alguma condição inicial que comprometa o comportamento do circuito ?
- Caso haja alguma falta de energia, durante a operação do circuito, existe alguma condição de operação em que o circuito pode entrar em uma situação de falta ou erro ?
- Existe alguma condição em que a tensão ou corrente, sobre um determinado componente, pode ultrapassar os limites de projeto estabelecidos?

Para isto, é introduzido um procedimento sistemático de modelagem para obter o autômato híbrido que representa o comportamento destes sistemas.

Uma vez obtido o autômato híbrido, representando o modelo formal do circuito de eletrônica de potência, serão aplicados métodos de verificação de propriedades através da realização da análise de alcançabilidade sobre o próprio espaço de estado híbrido do sistema. Sistemas híbridos possuem espaço de estados infinito, assim sendo, é necessária a obtenção de uma representação simbólica do seu espaço de estados para que procedimentos de verificação automática e reprojeção do controlador sejam computáveis e decidíveis. Este trabalho baseia-se na abordagem de aproximação dos estados alcançáveis, representando o comportamento do sistema e na teoria para reprojeção do controlador desenvolvida por Asarin et al. (2001), Asarin, Dang e Maler (2000).

Os motivos que levaram a escolha desta abordagem foram determinados, essencialmente, pela aplicação tratada neste trabalho. Esta abordagem possibilita a modelagem de dinâmicas contínuas consideradas não-triviais, incluindo representação de entradas e perturbações incertas, em comparação com os demais modelos existentes para estudo de sistemas híbridos. Possibilita modelar sistemas de dimensões elevadas, enquanto que as demais abordagens têm sérios problemas para modelar sistemas com dimensões maiores que 3 e tratam, no máximo, de sistemas com entradas constantes (CHUTINAN; KROGH, 1998), dinâmicas lineares (ALUR; HENZINGER; HO, 1996; HO, 1995), sendo poucos os casos em que são tratados sistemas não-lineares (CHUTINAN; KROGH, 2000a, 2000b). Além disto, o método sobre o qual este trabalho se baseia para obter a aproximação do espaço de estado provê a garantia de término, ou seja, é decidível e

computável, e a propagação do erro, durante o cálculo da aproximação, é considerado mínimo entre os métodos relacionados.

A verificação de propriedades e o procedimento de reprojeto do controlador, tratados nesta tese, são implementados através de técnicas de análise de alcançabilidade, ou seja, calculando todos os estados que podem ser alcançados por qualquer trajetória do sistema, dado um intervalo de condições iniciais. Esta pesquisa é realizada no próprio espaço de estados do sistema sem que seja necessário obter um sistema de transição finito e, assim, não havendo o risco de perdas de informações importantes do sistema.

A partir do modelo do sistema, este trabalho utiliza um procedimento para automaticamente reprojeter o controlador que chaveia o sistema entre modos de operação, com suas diferentes dinâmicas, para satisfazer a uma dada especificação. O objetivo do controlador é garantir que todas as configurações alcançáveis do sistema levem a um conjunto de configurações seguras, tendo-se especificado o conjunto de configurações que são consideradas seguras para o controlador.

O problema de reprojeto do controlador, tratado neste trabalho, tem o objetivo de, a partir do modelo do autômato híbrido completo do sistema, encontrar o maior subsistema que satisfaça uma determinada propriedade de segurança especificada. Neste caso, o sistema deve evitar atingir um conjunto de estados proibidos, ou seja, o controlador deve garantir que a planta não entrará em uma região de operação não-segura (estado proibido). Através desta abordagem de controle de sistemas híbridos, pode-se tratar uma ampla gama de problemas comuns na automação industrial, particularmente, circuitos de eletrônica de potência.

É importante ressaltar que a abordagem de controle considerada, neste trabalho, não tem o intuito de tratar problemas de controle contínuo, tais como problemas de estabilidade. Assim, qualquer ação de controle contínuo deve ser considerada como parte da planta a ser controlada.

A análise de propriedades é utilizada com o propósito de aumentar a confiabilidade do sistema. Um sistema é confiável, se é capaz de executar as funções requeridas pelas condições iniciais de operação. Uma análise de comportamento (sejam faltas ou outras propriedades quaisquer) de circuitos geralmente é realizada em eletrônica de potência através de simulações exaustivas de casos para posterior análise ou estudo destes casos. Nosso trabalho objetiva realizar estes estudos de forma mais estruturada e sistemática através de métodos formais.

A motivação por trás deste objetivo é explorar o uso de ferramentas formais no contexto de eletrônica de potência para obter um modelo de circuito de eletrônica de potência mais confiável, de modo a auxiliar os projetistas, durante o desenvolvimento de projetos de circuitos, e prover um menor custo de implantação e manutenção industrial para estes equipamentos.

1.3 Organização do Trabalho

A presente tese está estruturada em seis capítulos. O presente capítulo tem como objetivo prover uma visão geral ao leitor sobre sistemas híbridos, a utilização de métodos formais no estudo deste tipo de sistemas e apresentar os objetivos e motivação para o desenvolvimento deste trabalho.

Os demais capítulos estão assim organizados:

Capítulo 2 : É feita uma descrição sucinta sobre os formalismos desenvolvidos para estudo de sistemas híbridos e a apresentação formal de autômato híbrido, que será o modelo matemático utilizado no desenvolvimento dos métodos utilizados neste trabalho. Além disto, será apresentada a abordagem para aproximação do espaço alcançável, sobre o qual esta tese se baseia, para utilização dos procedimentos de verificação de propriedades e reprojeto do controlador.

Capítulo 3 : É apresentado um resumo sobre circuitos de eletrônica de potência, com seus componentes principais e leis de funcionamento. É apresentada uma técnica para a obtenção da representação da estrutura do circuito, na forma de blocos de montagem, que representam os componentes do circuito, e sua respectiva descrição através de linguagem formal. Além disto, é introduzido um procedimento sistemático de modelagem para a obtenção de modelos de autômatos híbridos que representem o comportamento de circuitos de potência.

Capítulo 4 : São apresentados os procedimentos de verificação de propriedades para sistemas híbridos. As propriedades tratadas nesta tese são chamadas de propriedades de invariância. Provar tais propriedades pode ser feito utilizando técnicas de análise de alcançabilidade, através de métodos algorítmicos de verificação para sistemas híbridos.

Capítulo 5 : É apresentado o método para a realização do reprojeto do controlador para sistemas híbridos. Nosso trabalho baseia-se na abordagem direta para sistematicamente sintetizar um controlador para circuitos de eletrônica de potência que satisfaça propriedades de segurança.

Capítulo 6 : Finalmente, são apresentadas as contribuições desta tese e as sugestões para futuros trabalhos.

Glossário : São apresentadas as definições dos principais termos utilizados neste trabalho.

Capítulo 2

Formalismos Matemáticos para Sistemas Híbridos

Para analisar propriedades de sistemas híbridos, é necessária a utilização de um modelo que seja rico o bastante para representar ambas dinâmicas contínuas e discretas inerentes a estes sistemas. Dinâmicas contínuas são tradicionalmente representadas por meio de equações diferenciais e dinâmicas discretas são normalmente modeladas e analisadas utilizando autômatos. Existem diversos formalismos matemáticos para representar sistemas híbridos, revelando a diversidade de pesquisas desenvolvidas nesta área. Este capítulo apresenta um resumo sobre alguns destes formalismos, apresentando com maiores detalhes o modelo matemático de autômatos híbridos, uma vez que será o formalismo de modelagem utilizado para o desenvolvimento desta tese. Autômatos híbridos combinam estes dois modelos matemático e provêm um formalismo efetivo que abordam estas dinâmicas, em que o comportamento contínuo é capturado por equações diferenciais associados com os estados discretos; e o comportamento discreto é capturado pelas transições entre os estados.

2.1 Sistemas Híbridos

Um sistema híbrido (PETTERSSON, 1999; MANNA; PNUELI, 1993) é um sistema reativo cujo comportamento resulta da interação entre as dinâmicas contínuas e discretas que o compõem. As dinâmicas contínuas e discretas interagem entre si e reagem a eventos, quando o estado contínuo alcança regiões ou conjunto de regiões no seu espaço de estado (BRANICKY, 1996a).

Constitui-se uma tarefa bastante desafiadora modelar sistemas híbridos, descrevendo exatamente seu comportamento, pois envolvem modelos matemáticos diferentes. A partir do modelo do sistema, é possível fazer a análise de suas propriedades e desenvolver estratégias de controle

adequadas às especificações das dinâmicas acima referidas.

Os primeiros modelos desenvolvidos para representar os processos físicos tratavam, separadamente, os sub-sistemas contínuos e discretos. Embora estas abordagens implicassem numa maior simplicidade de construção dos modelos, tinham a desvantagem de não representar adequadamente a natureza híbrida inerente ao processo físico, uma vez que não proviam um modelo global que o representasse. É importante a utilização de um modelo que represente os diferentes fenômenos físicos, contínuos e discretos, de forma integrada, representando o seu comportamento de forma mais completa.

O interesse no estudo de sistemas híbridos se intensificou nas últimas décadas. Atualmente, a modelagem, projeto e implementação de tais sistemas são áreas ativas de pesquisa na Ciência da Computação e engenharia de controle (GROSSMAN et al., 1993). Exemplos de sistemas híbridos são comumente encontrados em aplicações em sistemas flexíveis de manufatura (PE-PYNE; CASSANDRAS, 2000), controle de processos químicos (TITTUS; LENNARTSON, 1998; LENNARTSON; TITTUS, 1994; ENGELL et al., 2000), sistemas de distribuição de energia elétrica (KLOAS; FRIESEN; SIMONS, 1995), redes de comunicação de computadores, dentre outros (STIVER; ANTSAKLIS; LEMMON, 1996).

Sistemas dinâmicos híbridos merecem ser investigados tanto por razões teóricas como práticas. Existe uma grande variedade de aplicações industriais, nas quais as propriedades dinâmicas do sistema dependem da interação entre as variáveis discretas e contínuas. Observa-se um crescente interesse nas pesquisas sobre sistemas híbridos também por causa da modernização de processos industriais. Em ambientes industriais, é necessário que a produção ocorra com maior segurança e qualidade, bem como com um menor custo. Para tanto, existem ambientes de supervisão e controle que coletam informações do chão de fábrica, permitindo que os usuários ou operadores visualizem as ocorrências de cada processo em um menor espaço de tempo possível. Isto faz com que o usuário possa analisar os acontecimentos, alterar parâmetros de controle, corrigir erros, aperfeiçoando o processo de tomada de decisões na empresa. Estes ambientes de automação são responsáveis pelo controle supervisorio dos diversos sistemas de produção existentes e possibilitam que vários setores da empresa monitorem e controlem com precisão todos os aspectos das operações de seus processos.

O controle supervisorio e o monitoramento de um processo industrial são baseados na detecção de eventos que ocasionam mudanças de estado do sistema. É natural que estes sistemas tenham uma representação formal para que sejam melhor analisados, operados e controlados. A obtenção do modelo de um sistema híbrido auxilia no processo de tomada de decisões, monitoramento e ajuste do comportamento dinâmico do sistema. Assim, o estudo de sistemas híbridos é também importante para aplicações industriais, porque seu modelo exprime dinâmicas

importantes do sistema real, uma vez que expressa os diversos fenômenos que nele ocorrem.

Uma vez justificada a relevância da pesquisa de sistemas híbridos em aplicações industriais, focalizamos a atenção de nosso trabalho para aplicações de circuitos de eletrônica de potência. Circuitos de eletrônica de potência são sistemas dinâmicos compostos por componentes eletrônicos e dispositivos semicondutores. O comportamento destes sistemas é descrito através da dinâmica contínua com a intervenção de eventos discretos, representados pela abertura e fechamento das chaves que o compõem. O projeto de circuitos de eletrônica de potência com o auxílio do computador deve ser baseado em uma representação adequada, utilizando um modelo de descrição apropriado para que se evidencie a natureza híbrida de suas dinâmicas. O comportamento dinâmico de um sistema, tanto quanto a descrição dos controladores, podem ser formulados através de sistemas contínuos enquanto operações de chaveamento ou lógicas são representadas por sistemas discretos. Assim, é necessário encontrar uma especificação formal com representação rica o bastante, do ponto de vista da engenharia, para estudar este tipo de sistema (MALER, 2001).

A seguir, são apresentados os modelos matemáticos para sistemas híbridos mais relevantes na literatura para, enfim, detalharmos a teoria dos autômatos híbridos que será a ferramenta formal de base do nosso trabalho.

2.2 Estudos em sistemas híbridos

A primeira referência direta a sistemas híbridos conhecida é o trabalho de Witsenhausen (1966) que formulou um problema de controle ótimo para uma classe de sistemas dinâmicos de tempo contínuo. Este trabalho teve prosseguimento por Pavlidis (1967), que estudou estabilidade de sistemas com impulsos, utilizando funções de Lyapunov. Outros trabalhos sobre sistemas híbridos se seguiram: Johnson (1981), Wimpey (1982) focalizaram seus estudos em controladores de estado finito; Ezzin e Haddad (1989) examinaram estabilidade, controlabilidade e observabilidade de uma classe restrita de sistemas híbridos, os sistemas lineares chaveados.

Motivado por um interesse em sistemas com histerese, Tavernini (1987) introduziu um modelo de sistema híbrido baseado em autômatos diferenciais e apresentou soluções para problemas com condições iniciais e suas aproximações numéricas. Back e Myers (1993) se basearam no modelo de Tavernini e introduziram um formalismo que modela chaveamentos e saltos autônomos e é adequado para simulação numérica. Nerode e Kohn (1993) introduziram uma abordagem da teoria de autômatos para sistemas compostos de autômatos finitos e equações diferenciais que interagem entre si; Antsaklis, Stiver e Lemmon (1993) introduziram uma abordagem de sistemas dinâmicos a eventos discretos; Brockett (1993) combinou equações diferenciais e fenômenos

discretos para descrever sistemas de movimento; Alur et al. (1993) introduziram os autômatos híbridos, uma extensão de autômatos temporizados (ALUR; DILL, 1994a), para representação de sistemas híbridos e Benveniste (1996) propôs uma abordagem para modelagem de sistemas híbridos com ênfase na compossibilidade dos seus componentes. Branicky (1996a) introduziu um modelo de sistemas unificados, capturando todos os fenômenos discretos que surgem em sistemas híbridos.

A partir do final dos anos 80, houve uma renovação em relação ao interesse pelos estudos sobre sistemas híbridos junto à comunidade de sistemas de controle. Isto deveu-se ao desenvolvimento de pesquisas sobre controle de sistemas a eventos discretos, ocorrido no final da década de 80, ao surgimento de controle adaptativo, nos anos 70 e 80 e ao renovado interesse em formulações de controle ótimo em sistemas de controle digital (ANTSAKLIS, 2000).

Sistema híbrido constitui um tópico interdisciplinar que se encontra na junção de Ciência da Computação e Teoria de Controle, representando uma área desafiadora de pesquisa que compreende uma variedade de problemas que podem ser abordados em diferentes níveis de detalhes e sofisticação. As abordagens para sistemas híbridos diferem em relação à ênfase ou à complexidade das dinâmicas contínuas e discretas que essas abordagens tratam e se enfatizam resultados de análise ou síntese, ou somente análise ou somente simulação.

De um lado, existem abordagens para sistemas híbridos que representam extensões da teoria clássica de sistemas de tempo contínuo, descritos por equações diferenciais, para incluir em seus modelos variáveis de tempo discreto ou variáveis que exibem saltos, ou ainda, procuram estender resultados para sistemas chaveados. Estas abordagens são eficientes para tratar dinâmicas contínuas complexas e sua ênfase tem sido o estudo de estabilidade de sistemas com descontinuidades. Na Teoria de Controle, existem vários trabalhos nesta área, incluindo sistemas de estrutura variável (BRANICKY, 1994; LAFFERRIERE, 1994), sistemas lineares de saltos (JI; CHIZECK, 1990), sistemas com efeito de impulso (BAINOV; SIMEONOV, 1989) e processos determinísticos por etapa (DAVIS, 1993).

Por outro lado, existem abordagens para sistemas híbridos baseados em modelos e métodos provenientes da Ciência da Computação. O advento de máquinas digitais fez com que os sistemas híbridos se tornassem muito comuns, pois sempre que um dispositivo digital interage com o mundo contínuo, o comportamento envolve fenômenos híbridos que precisam ser analisados e compreendidos. Estas abordagens representam extensões de metodologias de verificação de sistemas discretos para sistemas híbridos e são eficientes para tratar dinâmicas discretas descritas por autômatos finitos e enfatizar resultados de verificação e de simulação. Na Ciência da Computação, surgiram sucessivos trabalhos enfocando a verificação formal (BENVENISTE; BERRY, 1991; JOSEPH, 1989; MANNA; PNUELI, 1992) e para sistemas com variáveis discretas

e contínuas (ANTSAKLIS; STIVER, 1995; GROSSMAN et al., 1993; MALER; MANNA; PNUELI, 1992; ASARIN ODED MALER; SIFAKIS, 1998).

Existem, ainda, muitas outras metodologias adicionais que combinam conceitos de sistemas de controle contínuo com conceitos da teoria de controle supervisorio de sistemas a eventos discretos, objetivando obter resultados de análise e síntese para sistemas híbridos (CURY; KROGH, 1999; CURY; KROGH; NIINOMI, 1995; RAISCH; O'YOUNG, 1998).

Podem ser encontrados na literatura trabalhos com interesse no cálculo analógico (BRANICKY, 1996b; SIEGELMAN; SONTAG, 1991), análise de sistemas chaveados (BRANICKY, 1994; LAFFERRIERE, 1994; PELETIES; DECARLO, 1991), sistemas hierárquicos (CAINES; WEI, 1995) e sistemas contínuos controlados por eventos discretos (KULKARNI; RAMADGE, 1995; MORSE, 1993, 1994), dentre outros.

Foram propostos vários formalismos matemáticos para representação de sistemas híbridos. Dentre eles, incluem-se as redes de Petri temporizadas (HE; LEMMON, 1998), redes de Petri diferenciais (DEMONGODIN; KOUSSOULAS, 1998), redes de Petri híbridas (BAIL; ALLA; DAVID, 1991; DAVID; ALLA, 1994), redes de Petri híbridas de alto nível (GIUA; USAI, 1996), entre outros modelos com representação contínua de redes de Petri. Existem, também, estudos com aplicações do formalismo de Bond Graph para sistemas híbridos (STROMBERG, 1994; KARNOPP; MARGOLIS; ROSENBERG, 1990; MOSTERMAN; BISWAS, 1997a, 1997b), além das extensões dos autômatos finitos para representação de sistemas híbridos, entre os quais podemos destacar os autômatos temporizados (ALUR; DILL, 1994a) e os autômatos híbridos (ALUR et al., 1993; HENZINGER, 1996).

O modelo matemático adotado para representação da aplicação deste trabalho foi autômato híbrido. As razões pelas quais motivaram esta escolha, em detrimento aos formalismos citados acima, foram as seguintes:

- Permite capturar, em um único modelo, as dinâmicas discretas e contínuas do sistema e a representação dos diferentes modos de operação, bem como a forma e as condições de mudanças entre estes modos. Em Bond Graph, esta unificação não é natural pois os elementos de chaveamento não representam funções relacionadas com energia, como é comum aos demais componentes deste formalismo (STROMBERG; SODERMAN, 1993) e o modelo mais eficiente para representação híbrida com Bond Graph utiliza-o associado a autômatos finitos (MOSTERMAN; BISWAS, 1997a), dificultando o processo de modelagem.
- Possibilita modelar os tipos mais genéricos de fenômenos discretos que ocorrem em sistemas híbridos, como saltos autônomos e controlados e chaveamentos autônomos e contro-

lados (BRANICKY, 1996a). Formalismos como redes de Petri híbridas e Bond Graph não provêm formas para modelar todos estes fenômenos.

- Possibilita modelar dinâmicas contínuas mais genéricas que os demais formalismo, através de equações diferenciais lineares ou não-lineares. Redes de Petri não modelam dinâmicas não-lineares.
- Permite modelagem hierárquica e composicional, ao contrário dos modelos Bond Graph.
- Ao contrário do formalismo de Bond Graph, possui estudos sobre procedimentos sistemáticos para análise formal e síntese do controlador, além de prover modelos para fins de simulação. Ainda não existem procedimentos de verificação formal e provas de decidibilidade para redes de Petri híbridas, sendo, portanto, mais utilizadas para fins de simulação.
- Considerando a aplicação deste trabalho, autômato híbrido foi a opção que mais naturalmente possibilitou exprimir o comportamento de circuitos de eletrônica de potência.

Uma vez expostos os motivos para a escolha da representação matemática que este trabalho se baseia, serão apresentados nas próximas seções a definição formal de autômatos híbridos.

2.3 Autômatos Híbridos

A aplicação base de nosso trabalho são os circuitos de eletrônica de potência. Circuitos de eletrônica de potência são considerados sistemas híbridos, uma vez que possuem comportamento contínuo que pode sofrer intervenções de eventos discretos, mudando a topologia do circuito, representados pela mudança de estado dos dispositivos semicondutores que o compõem. Como mencionado na Seção 2.2, autômato híbrido constitui-se no formalismo de representação da aplicação que representa o sistema do nosso trabalho.

Os autômatos híbridos são alguns dos poucos formalismos que, em sua representação gráfica, conseguem capturar, em um único modelo, as dinâmicas discretas e contínuas e a representação dos diferentes modos de operação de sistemas reativos.

Um autômato híbrido (ALUR; HENZINGER; HO, 1996) é uma máquina de estado finito generalizada que é equipada com variáveis contínuas. As ações discretas são representadas pelas transições entre os estados do autômato, aqui denominadas de locações de controle. As atividades contínuas são modeladas por variáveis definidas no conjunto dos números reais, cujos valores mudam continuamente durante o tempo de acordo com leis da física: para cada locação de controle, as atividades contínuas são descritas por equações diferenciais. Este modelo para sistemas híbridos é inspirado nos *Sistemas de Transição de Fase*, introduzidos nos trabalhos

de Maler, Manna e Pnueli (1992) e Nicollin, Sifakis e Yovine (1992), e pode ser visto como uma generalização de *autômato temporizado* com variáveis discretas e contínuas (ALUR; DILL, 1994b).

2.3.1 Definição Formal

Definição 1 Um Autômato Híbrido é um sistema $A = \langle X, Q, f, H, init, E, G, \Sigma, syn, R \rangle$ e consiste dos seguintes componentes (ALUR et al., 1995, 1993; HENZINGER, 1996).

- *Variáveis de Dados.* Um conjunto finito $X = \{x_1, x_2, \dots, x_n\}$ de variáveis definidas no conjunto dos números reais, onde n é considerada a dimensão de A . Denota-se por \dot{X} o conjunto $\{\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n\}$ representando a derivada primeira de X . Denota-se por X' o conjunto $\{x'_1, x'_2, \dots, x'_n\}$ representando os novos valores de X após uma mudança discreta (transição).
- *Modo de Controle ou de Operação.* Um conjunto finito Q de modos de controle, também chamado de locações.
- *Condições de Fluxo ou de Atividade Contínua.* Uma função f_q que atribui uma condição de fluxo a cada modo de controle $q \in Q$. A condição de fluxo f_q é um predicado sobre o conjunto de variáveis $X \cup \dot{X}$. Enquanto o controle do autômato híbrido A está no modo q , o vetor X evolui de tal modo que todos os pontos X e \dot{X} satisfazem à condição de atividade contínua f_q .
- *Condições Invariantes.* Uma função H que atribui uma condição invariante a cada modo de controle $q \in Q$. A condição invariante H_q é um predicado sobre as variáveis em X . Enquanto o controle do autômato híbrido H está no modo q , o vetor X deve satisfazer à condição invariante H_q .
- *Condições Iniciais.* Uma função $init$ que atribui uma condição inicial a cada modo de controle $q \in Q$. A condição inicial $init(q)$ é um predicado sobre as variáveis em X . O controle do autômato híbrido A pode iniciar no modo de controle q quando a condição inicial $init(q)$ é verdadeira.
- *Chaves de controle ou Transições.* Um multi-conjunto¹ finito E de transições. Cada chave de controle (q, q') é um arco dirigido entre um modo fonte $q \in Q$ e um modo destino $q' \in Q$ e representa uma transição entre as locações.

¹Um multi-conjunto é semelhante a um conjunto, porém a multiplicidade de elementos no conjunto é ex-

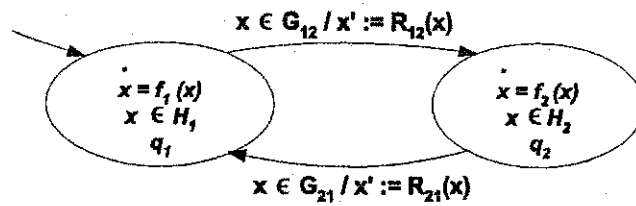


Figura 2.1: Representação gráfica de um autômato híbrido

- **Condições de Guardas.** Uma função G que atribui a cada chave de controle $e \in E$ um predicado. Cada condição de guarda $G(e)$ é um predicado sobre as variáveis em X . No autômato, as *guardas* (G) representam as condições de salto ou mudança de fase.
- **Eventos ou Mensagens de Sincronização.** Um conjunto finito Σ de eventos e uma função syn que associa um evento em Σ a cada transição $e \in E$.
- **Funções de Reset.** Uma função de mapeamento, $R(q, q')$, que define como as variáveis contínuas podem mudar quando o autômato chaveia da locação q para q' , ou seja, quais os valores que as variáveis do vetor X terão ao chavear para a locação q' .

Autômatos híbridos podem ser representados graficamente como um grafo dirigido. Esta representação gráfica está ilustrada na Figura 2.1, na qual o autômato híbrido tem $Q = \{q_1, q_2\}$ e $X = \{x\}$. Para cada vértice do autômato, existe um modo discreto, um campo vetorial ($f_1(x)$ para q_1 , $f_2(x)$ para q_2) e uma condição invariante (H_1, H_2). Cada arco está rotulado com uma transição de guarda (G_{12}, G_{21}) e uma função de reset ($R_{12}(x), R_{21}(x)$), conforme mostrado, na Figura 2.1. O estado inicial $Init(q)$ está indicado através de um arco dirigido sem um vértice fonte e apontando para um vértice, conforme ilustrado no vértice esquerdo da Figura 2.1.

2.3.2 Composição

O sistema híbrido, a ser modelado, no caso circuitos de eletrônica de potência, pode ser decomposto em componentes, representando os diferentes sub-sistemas identificados pelo projetista, e cada componente é modelado por um autômato híbrido apropriado. O comportamento do sistema como um todo é obtido da interação e integração dos autômatos independentes. As transições de estado num dos componentes são sinalizadas por eventos ou mensagens, recebidos de outros componentes ou do ambiente externo. O autômato híbrido que representa o sistema

plicitamente significativa, ao contrário do que ocorrem em conjuntos, cuja ordem e repetição de elementos é ignorada. Por exemplo, os multi-conjuntos $\{1, 2, 3\}$ e $\{2, 1, 3\}$ são equivalentes, mas $\{1, 1, 2, 3\}$ e $\{1, 2, 3\}$ são diferentes.

global é obtido através do produto dos autômatos componentes do sistema e seu sincronismo é tratado de duas maneiras (BONIFÁCIO; MOURA, 2000):

1. sincronização de rótulos das transições, forçando com que transições ocorram no mesmo evento, e
2. usando variáveis compartilhadas.

As transições no autômato produto são representadas da seguinte maneira: considere o autômato híbrido $A_1 = \langle X_1, Q_1, f_1, H_1, init_1, E_1, G_1, \Sigma_1, syn_1, R_1 \rangle$ e $A_2 = \langle X_2, Q_2, f_2, H_2, init_2, E_2, G_2, \Sigma_2, syn_2, R_2 \rangle$. O autômato produto de A_1 e A_2 é um sistema $A = \langle X_1 \cup X_2, Q_1 \times Q_2, f, init, H, E, G, \Sigma_1 \cup \Sigma_2, syn, R \rangle$, onde

- $f_{q_1, q_2} = f_{q_1} \wedge f_{q_2}$,
- $H_{q_1, q_2} = H_{q_1} \wedge H_{q_2}$,
- $init(q_1, q_2) = init(q_1) \wedge init(q_2)$,
- $e = ((q_1, q'_1), (q_2, q'_2)) \in E$ se, e somente se, uma das três condições abaixo for verificada:
 1. $q_1 = q'_1, e_2 = (q_2, q'_2) \in E_2, syn_2(e_2) \notin \Sigma_1$.
Daí, $G(e) = G_2(e_2)$ e $syn(e) = syn_2(e_2)$.
 2. $q_2 = q'_2, e_1 = (q_1, q'_1) \in E_1, syn_1(e_1) \notin \Sigma_2$.
Daí, $G(e) = G_1(e_1)$ e $syn(e) = syn_1(e_1)$.
 3. $e_1 = (q_1, q'_1) \in E_1, e_2 = (q_2, q'_2) \in E_2, syn_1(e_1) = syn_2(e_2)$.
Daí, $G(e) = G_1(e_1) \wedge G_2(e_2)$ e $syn(e) = syn_1(e_1)$.

As transições e_1 e e_2 são executadas simultaneamente no autômato produto quando $syn_1(e_1) = syn_2(e_2)$, isto é, quando a ambas está associado um mesmo evento. Por outro lado, essas transições são colocadas de forma intercalada no autômato produto se o evento $syn_1(e_1)$ não ocorre, no conjunto de eventos de A_2 , e nem $syn_2(e_2)$ ocorre, no conjunto de eventos de A_1 .

Normalmente, a utilização de uma ferramenta de software para modelagem e implementação de autômatos híbridos possibilita a construção do autômato produto automaticamente. Um exemplo de composição de autômatos será mostrado, na Seção 2.3.3.

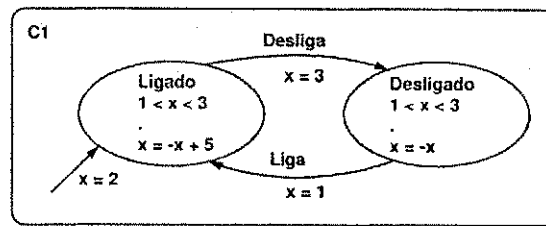


Figura 2.2: Autômato simples de um aquecedor

2.3.3 Exemplo: Aquecedor e Monitor

Um exemplo bastante simples de um sistema modelado por autômatos híbridos pode ser ilustrado por um aquecedor com seu respectivo controlador (monitor).

Neste exemplo, o número de componentes do sistema é 2, representados pelos autômatos C_1 , representando o autômato do aquecedor, ilustrado na Figura 2.2, e C_2 , representando o monitor, este último será apresentado mais adiante, na Figura 2.3. Para simplificar a explicação, detalharemos agora o autômato do componente C_1 . Uma trajetória do sistema é uma curva em \mathbb{R}^n , onde $n = 1$, uma vez que $X = \{x\}$. Cada modo de operação do modelo recebe um rótulo e descreve a dinâmica do sistema através de uma equação diferencial. No autômato híbrido da Figura 2.2, existem dois modos de operação, o modo ligado, cujo rótulo é *Ligado* e o modo desligado, cujo rótulo é *Desligado*. No modo *Ligado*, a temperatura cresce à taxa de $-x + 5$ graus por minuto e quando o sistema está no modo *Desligado*, a temperatura cai à taxa de $-x$ graus por minuto. A cada modo de operação, está também associada uma condição inicial. A condição inicial no modo *Ligado* é $x = 2$ graus. No modo *Desligado*, a condição inicial é *falso*, logo não precisa ser graficamente ilustrada, na Figura 2.2.

Para forçar uma mudança na dinâmica, atribui-se a cada modo de operação uma condição invariante. O sistema só pode se manter num certo modo de operação, enquanto sua condição invariante for satisfeita. No exemplo, a condição invariante para ambos os modos de operação é $1 < x < 3$, ou seja, a mudança de modo deve ocorrer antes de a temperatura sair do intervalo de operação de $[1, 3]$ graus. O aquecedor só é desligado quando, no modo *Ligado*, a temperatura alcança 3 graus. E o aquecedor só é ligado quando, no modo *Desligado*, a temperatura cai para 1 grau. A evolução das variáveis de estado é descrita, a partir da condição inicial, em função das equações diferenciais presentes nos modos e das transições entre estes modos.

Existem duas transições na Figura 2.2, (*Ligado*, *Desligado*) e (*Desligado*, *Ligado*). (*Ligado*, *Desligado*) dispara quando a condição dada por $x = 3$ é satisfeita e a transição (*Desligado*, *Ligado*) dispara quando a condição por $x = 1$ é satisfeita. A função *syn* associa o evento *Liga* à transição (*Desligado*, *Ligado*) e o evento *Desliga*, à transição (*Ligado*, *Desligado*). A ocorrência

destes eventos sincronizará os autômatos do sistema, C_1 e C_2 .

Resumindo, para o exemplo da Figura 2.2, os elementos que compõem o autômato do componente C_1 são:

- $n = 1$.
- $X_1 = \{x\}$.
- $Q_1 = \{Ligado, Desligado\}$.
- Condição de atividade contínua:
 - O modo de operação *Ligado*: $\dot{x} = -x + 5$,
 - O modo de operação *Desligado*: $\dot{x} = -x$.
- A condição invariante para ambos estados é $1 < x < 3$.
- A condição inicial é $x = 2$, no modo *Ligado*; no modo *Desligado*, a condição inicial é *falso*.
- São duas transições: (*Ligado, Desligado*) e (*Desligado, Ligado*).
- A transição (*Ligado, Desligado*) tem condição de guarda dada pelo predicado $x = 3 \wedge x' = x$. A transição (*desligado, ligado*) tem como condição de guarda o predicado $x = 1 \wedge x' = x$.
- $\Sigma_1 = \{liga, desliga\}$.

Para ilustrar como obter o autômato produto, considera-se que o aquecedor da Figura 2.2 é monitorado pelo autômato da Figura 2.3, cujo objetivo é verificar se a temperatura está no intervalo $[1.5, 2.5]$, nos primeiros 60 minutos de operação.

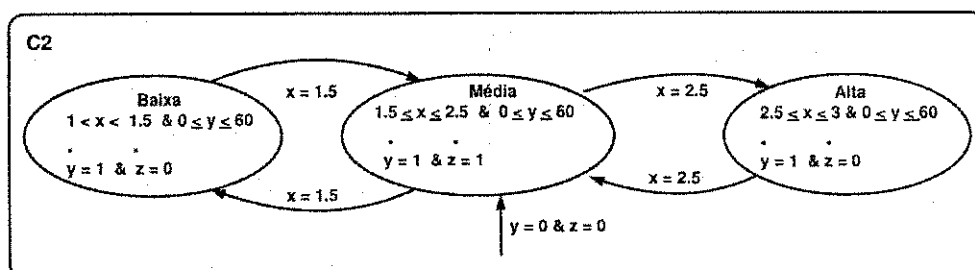


Figura 2.3: Autômato Monitor

A composição paralela do autômato monitor da Figura 2.3 e do autômato do aquecedor da Figura 2.2 está ilustrada, na Figura 2.4. O autômato resultante desta composição é obtido, segundo os critérios estabelecidos para composição de autômatos híbridos apresentados, na Seção 2.3.2.

O conjunto de variáveis do autômato composto é a união dos conjuntos de variáveis dos autômatos componentes, ou seja, $X = \{x, y, z\}$, e o grafo do autômato composto é o produto cartesiano dos grafos dos autômatos componentes componentes, ou seja, $Q = \{(Ligado, Baixa), (Ligado, Média), (Ligado, Alta), (Desligado, Baixa), (Desligado, Média), (Desligado, Alta)\}$. Cada modo de controle do autômato composto corresponde a um modo de controle do primeiro autômato componente e um modo de controle do segundo autômato componente. Cada evento de controle do autômato composto corresponde ou a eventos simultâneos de controle (com rótulos de eventos coincidentes) de ambos autômatos componentes ou a um evento de controle (com rótulo de evento não aparecendo no conjunto de evento do outro componente) de um dos autômatos componentes, verificando sempre a consistência das condições de guarda entre as locações do autômato obtido. Isto pode fazer com que algumas transições entre estas locações não sejam representadas no modelo final do autômato produto.

Os fluxos e os invariantes de cada modo de controle são representados pela conjunção dos fluxos e invariantes dos modos de controle dos autômatos C_1 e C_2 , por exemplo, a condição de permanência da locação $(Ligado, Baixa)$ do autômato composto corresponde à conjunção das condições $1 < x < 3$ do autômato C_1 e $1 < x < 1.5 \wedge 0 \leq y \leq 60$ do autômato C_2 .

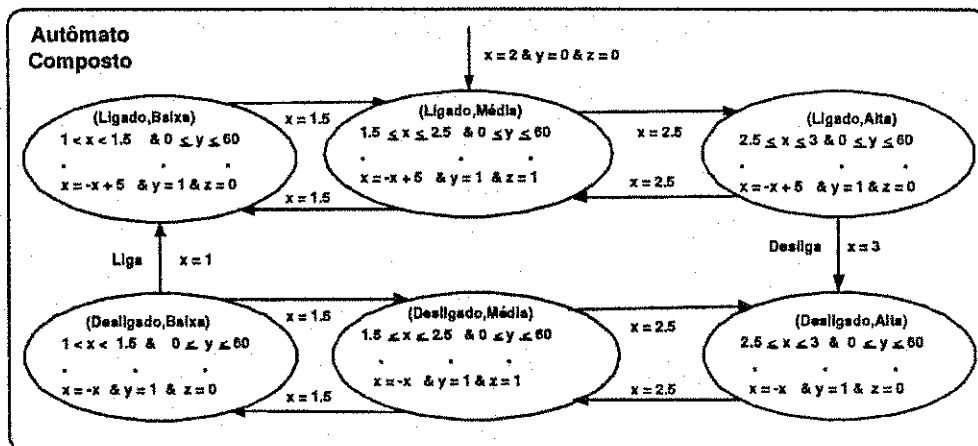


Figura 2.4: Autômato resultante da composição entre o autômatos do aquecedor e monitor

Durante o processo de modelagem, normalmente, decompõe-se o sistema em componentes e cada um deles é representado por um autômato híbrido. Procedimentos, tais como verificação de propriedades e síntese do controlador, são executados sobre o autômato resultante da composição dos autômatos componentes do sistema. Existem ferramentas desenvolvidas para o estudos

de sistemas híbridos que realizam o processo de composição em tempo real, durante o processo de análise e verificação do sistema, dentre as quais, podem-se citar HyTech (HENZINGER; HO; WONG-TOI, 1995), CheckMate (CHUTINAN; KROGH, 2000a), Matlab/Simulink/Stateflow (MATWORKS, 2002), SHIFT (DESHPANDE; GOLLU; SEMENZATO, 1995), d/dt (ASARIN; DANG; MALER, 2001), UppAal (BENGTSSON et al., 1998), etc.

2.3.4 Configurações e Trajetórias

Uma *configuração* ou *estado* de um autômato híbrido A é um par (q, a) , consistindo de um modo de operação $q \in Q$ e de um vetor $a = (a_1, \dots, a_n)$. O vetor a contém um valor $a_i \in \mathbb{R}$ para cada variável $x_i \in X$. A configuração (q, a) de A é *admissível*, se o predicado H_q é verdadeiro, quando a variável x_i é substituída pelo valor a_i , para $i = 1, \dots, n$, onde n é a dimensão do sistema. Observe, no exemplo da Figura 2.2, que a configuração $(Ligado, 1.5)$ é admissível. Já a configuração $(Ligado, 0.5)$ não é admissível. A configuração (q, a) é *inicial* se o predicado $init(q)$ for verdadeiro quando todo x_i for substituído pelo valor a_i . No mesmo exemplo, é fácil ver que a configuração $(Ligado, 2)$ é inicial.

Considere um par (v, v') formado por duas configurações admissíveis, $v = (q, a)$ e $v' = (q', a')$. O par (v, v') é uma *mudança de fase* de A , se existe uma transição $e \in E$ com origem no modo q e destino no modo q' tal que o predicado $G(e)$ seja verdadeiro, quando toda variável x_i for substituída pelo valor a_i e toda variável x'_i for substituída pelo valor a'_i . No exemplo da Figura 2.2, existem duas mudanças de fase, $((Ligado, 3), (Desligado, 3))$ e $((Ligado, 1), (Desligado, 1))$. Por outro lado, o par (v, v') é uma *atividade contínua* de A se $q = q'$, se existe um real não negativo $\tau \in \mathbb{R}$ (tempo de duração de uma atividade contínua) e se existe também uma função diferenciável $\rho : [0, \tau] \rightarrow \mathbb{R}^n$ (a curva da atividade contínua), tais que os três requisitos que se seguem sejam satisfeitos:

1. Pontos inicial e final: $\rho(0) = a$ e $\rho(\delta) = a'$.
2. Condição invariante: Para todo instante de tempo $t \in [0, \tau]$, a configuração $(v, \rho(t))$ é admissível.
3. Condição de atividade contínua: Seja $\dot{\rho} : [0, \tau] \rightarrow \mathbb{R}^n$ a derivada primeira de ρ . Para todo instante de tempo $t \in [0, \tau]$, o predicado f_q é verdadeiro quando cada variável x_i é substituída pela i -ésima coordenada do vetor $\rho(t)$, e toda variável \dot{x}_i é substituída pela i -ésima coordenada do vetor $\dot{\rho}(t)$.

No exemplo da Figura 2.2, os pares $((Desligado, 3), (Desligado, 2))$ e $((Desligado, 3), (Desligado, 2.5))$ são atividades contínuas.

Quando ocorre uma mudança de fase (v, v') , diz-se que q' é um modo de operação sucessor de q . Quando (v, v') é uma atividade contínua, a' é dito um ponto sucessor de a .

Uma *trajetória* de um autômato híbrido A é uma seqüência finita v_0, v_1, \dots, v_k , de configurações admissíveis, onde v_0 é uma configuração inicial de A e, onde todo par (v_j, v_{j+1}) de configurações consecutivas da seqüência é uma mudança de fase ou uma atividade contínua de A . Uma configuração v_n de A é *alcançável* se v_n é a última configuração de alguma trajetória de A . Uma *trajetória inicial* de A é qualquer uma de sua trajetória que inicia em uma configuração inicial. No exemplo da Figura 2.2, todas as configurações admissíveis são alcançáveis.

2.4 Cálculo dos estados alcançáveis

Para analisar formalmente sistemas híbridos, é necessário um modelo matemático que descreva sua dinâmica contínua e as transições observadas por descontinuidades possíveis de ocorrer nestes sistemas. Autômatos híbridos é o formalismo de modelagem escolhido para estudar sistemas híbridos e, uma vez que o modelo formal do sistema híbrido, representado pelo circuito de eletrônica de potência, é obtido, é possível analisar o comportamento deste sistema. O procedimento mais importante para sistemas híbridos para verificação formal e estudo de seu comportamento é a *análise de alcançabilidade*. Esta técnica consiste no cálculo de todos os estados que podem ser alcançados por quaisquer trajetórias do sistema, dado um conjunto de condições iniciais.

Calcular os estados alcançáveis para sistemas híbridos tem sido considerado um dos principais problemas em pesquisas para estes tipos de sistemas, uma vez que a obtenção do conjunto de estados alcançáveis constitui-se num dos pré-requisitos para a utilização de metodologias de verificação algorítmica de propriedades de sistemas e é realizado sobre conjuntos no espaço \mathbb{R}^n , difíceis de serem calculados exatamente.

O conjunto alcançável de um dado conjunto de estados F de um autômato híbrido pode ser definido como o conjunto de todos estados visitados pelas trajetórias, começando dos estados em F . Uma trajetória de um autômato híbrido corresponde à seqüência de trajetórias de dinâmicas contínuas e discretas.

2.4.1 Alcançabilidade de autômatos híbridos

Considere, como modelo formal de sistema híbrido, o autômato híbrido definido pela tupla $A = \langle X, Q, f, H, G, R \rangle$, conforme definido na Seção 2.3.1. Existem dois tipos de evolução de um estado (q, x) do autômato, compreendendo a evolução contínua, dada pela dinâmica

f_q na locação q , e a evolução discreta dada pela ocorrência de transições habilitadas em q . Serão adotadas as notações abaixo para descrever estas evoluções: sejam $x, x' \in X$, $q, q' \in Q$, $T \in \mathbb{R}_+$, $t \in T$ e $\alpha : T \rightarrow X$ uma solução de $\dot{x} = f_q(x)$ com a condição inicial $\alpha(0) = x$.

- $x \xrightarrow{q,t}$: representa que a dinâmica contínua f_q está habilitada em x , para o tempo $t > 0$, se $\alpha(t') \in H_q, \forall t' \in [0, t]$. Se a trajetória permanece em $F \subseteq X$, para $t > 0$, então, denota-se como $x \xrightarrow[F]{q,t}$.
- $x \xrightarrow{q,t} x'$: representa que o estado (q, x') é alcançável de (q, x) , chamado assim de q -alcançável, pela dinâmica f_q , para o tempo $t > 0$ tal que $\alpha(t) = x'$ e $\alpha(t') \in H_q$ para todo $t' \in [0, t]$. Se $\alpha[t'] \in F$, para $t' \in [0, t]$, então, denota-se como $x \xrightarrow[F]{q,t} x'$.
- $x \xrightarrow[F]{q,t} G$: representa que um conjunto $G \subseteq X$ é alcançável de (q, x) pela dinâmica f_q , para o tempo $t > 0$ tal que $\alpha(t) = x'$ e $x' \in G$. Se $\alpha(t') \in F$ e $F \subseteq X$ para todo $t' \in [0, t]$, então, denota-se $x \xrightarrow[F]{q,t} G$ (lê-se: F até G no modo q).

Para obter os estados alcançáveis ou pesquisar o espaço de estado do sistema, foram especificados operadores de alcançabilidade para o autômato híbrido a ser estudado. Estes operadores avaliam os sucessores e predecessores contínuos e discretos do espaço de estado $Q \times X$ do autômato. Para definição destes operadores, considere as seguintes notações (com $F \subseteq X$ e $x \in X$):

- O operador *sucessor contínuo* $\delta_c : 2^X \rightarrow 2^X$

Dado um conjunto de estados (q, F) , onde $q \in Q$ e $F \subseteq X$, o conjunto de sucessores contínuos de (q, F) , denotado por $\delta_c(q, F)$, é definido como

$$\delta_c(q, F) = \{(q, x') | \exists x \in F, \exists t > 0; x \xrightarrow{q,t} x'\}$$

Esta notação quer dizer que a trajetória da dinâmica f_q de x para x' está dentro de H_q . O comportamento do autômato híbrido, no estado discreto q , está restrito às condições de permanência H_q .

A notação δ_r é utilizada para representar os estados alcançáveis passado um tempo r . O conjunto alcançável de F é $\delta_{[0,\infty)}(q, F)$, denotado por $\delta(q, F)$. A Figura 2.5 ilustra este operador. A região preenchida representa $\delta_r(F)$ e o conjunto alcançável no intervalo de tempo $[0, r]$ é o "tubo" entre as duas linhas pontilhadas, a fronteira externa de F e o próprio $\delta_r(F)$.

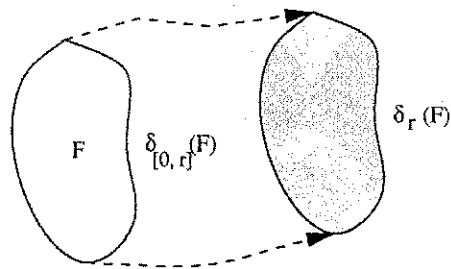


Figura 2.5: Sucessores contínuos de F

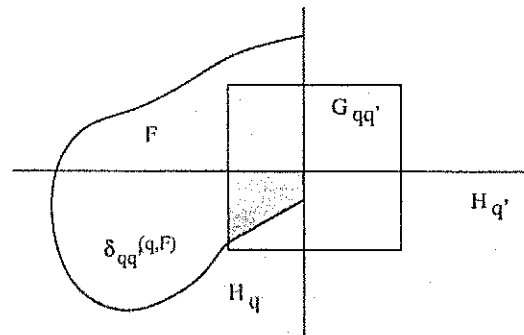


Figura 2.6: Sucessores discretos de F pela transição de q para q'

- O operador *sucessor discreto*

Dada uma transição $q, q' \in Q$ e um conjunto de estados (q, F) , onde $F \subseteq X$, o conjunto de sucessores discretos de (q, F) , denotado por $\delta_{qq'}(q, F)$, é definido como

$$\delta_{qq'}(q, F) = \{(q', x') | \exists x \in F \cap G_{qq'}, x' \in R_{qq'}(x) \cap H_{q'}\}$$

O conjunto $\delta_{qq'}(q, F)$ contém os estados (q', x') , onde x' são os pontos em $H_{q'}$, aplicando a relação reset $R_{qq'}$ para os pontos x em F que satisfazem a guarda $G_{qq'}$.

A região que representa o resultado da aplicação deste operador está ilustrada na Figura 2.6. O conjunto $H_{q'}$ é o meio-espaço abaixo da linha horizontal e a guarda $G_{qq'}$ é o retângulo. O sucessor discreto de (q, F) em relação à transição de q para q' pode ser obtido pela intersecção de F com $G_{qq'}$ e depois com $H_{q'}$, que corresponde à região hachurada, na Figura 2.6.

O conjunto de sucessores discretos de (q, F) , executando todas as transições habilitadas de q , denotado por $\delta_d(q, F)$, é definido como

$$\delta_d(q, F) = \bigcup_{q' \in Q} \delta_{qq'}(q, F)$$

Uma vez obtidos os sucessores contínuos e discretos, através dos operadores δ_c e δ_d , o operador sucessor δ para F é definido como o conjunto de estados alcançáveis de F , através das dinâmicas contínuas e transições discretas:

$$\delta(F) = \delta_d(\delta_c(F))$$

O cálculo do conjunto de estados alcançáveis, a partir de um conjunto $F \subseteq X$, $\delta(F)$, é a operação básica para qualquer procedimento de pesquisa no espaço de estado híbrido, especialmente, verificação de propriedades.

O método de alcançabilidade para frente (*forward*) consiste em obter o espaço alcançável dado um conjunto inicial F de estados e calcular seus sucessores até que o conjunto de todos estados alcançáveis seja obtido. O cálculo de $\delta(F)$, representando o conjunto alcançável pelo método para frente a partir de F , está descrito no Algoritmo 1.

Algoritmo 1 Cálculo de $\delta(F)$

$P^0 := F$

repita

$k = 0, 1, 2, \dots$

$P^{k+1} := P^k \cup \delta_{[0,r]}(P^k)$

até $P^{k+1} = P^k$

O conjunto P^k , neste caso, é o conjunto de estados alcançáveis de F durante o intervalo $[0, kr]$, para $k = 0, 1, 2, \dots$, e dada uma etapa de tempo $r > 0$. O algoritmo termina quando nenhum novo estado alcançável é encontrado.

O operador $\delta(F)$ é definido como o conjunto de estados alcançáveis de F pela dinâmica contínua e, então, pela dinâmica discreta.

Cálculo do conjunto alcançável, $\delta(F)$, com entrada constante

O cálculo de $\delta(F)$ é realizado, etapa-a-etapa, calculando os sucessores dos vértices de F . Para isto, considere que a dinâmica contínua do sistema híbrido seja $\dot{x} = Ax + Bu$, onde A é uma matriz $n \times n$, B é um vetor $n \times 1$ e u é uma entrada constante. Para simplificação, seja $b = Bu$, onde b é um vetor constante $n \times 1$. A solução analítica para a equação de estado é dada por

$$x(t) = e^{At}x_0 + e^{At} \int_0^t e^{-A\tau} b d\tau \tag{2.1}$$

Seja $\nu(t) = e^{At} \int_0^t e^{-A\tau} b d\tau$.

Assim, para uma determinada condição inicial $x(0) \in F$, o conjunto de sucessores de F é obtido pelo conjunto representado pela solução da equação diferencial que modela o comportamento do sistema para os vértices extremos de F , para uma etapa de tempo $r > 0$, escrita como:

$$\delta_t(F) = e^{A(t-r)}\delta_r(F) + \nu(t)$$

e^{At} e $\nu(t)$ podem ser calculados numericamente, utilizando integração numérica para $\nu(t)$.

Cálculo do conjunto alcançável, $\delta(F)$, com entrada variante no tempo

O cálculo dos conjuntos sucessores do sistema ($\delta(F)$), a partir da condição inicial $x_0 \in F$, para a dinâmica contínua do circuito, é realizado para sistemas sem sinal de entrada ou cujas entradas são constantes. Para calcular $\delta(F)$, considerando o sinal de entrada variante no tempo, é necessário transformar o sistema de estado original em um sistema autônomo. A obtenção do sistema autônomo, chamado de sistema de estado expandido (SILVA; MARTINS, 1996), é realizado aumentando a dimensão do sistema original. O sistema de estado expandido tem solução algébrica independente do vetor de entrada, possibilitando a utilização do algoritmo de cálculo dos estados alcançáveis (Algoritmo 1) considerando, também, entrada variante no tempo.

As equações de estado do circuito são da forma:

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t), \\ y(t) = Cx(t) + Du(t) \end{cases} \quad (2.2)$$

onde A é uma matrix $n \times n$, B é um vetor $n \times 1$ e u é o vetor de entrada variante no tempo. A solução analítica para esta equação de estado é dada por:

$$x(t) = e^{At}x_0 + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau$$

A utilização do Algoritmo 1, na obtenção do espaço de estado do sistema para fontes variáveis, é possível transformando o sistema, representado pela Eq. 2.2, em um sistema autônomo da seguinte forma (SILVA; MARTINS, 1996):

$$\begin{cases} \dot{x}_a(t) = A_a x_a(t), \\ y_a(t) = C_a x_a(t) \end{cases} \quad (2.3)$$

onde $x_a(t)$ é o vetor de estado do sistema autônomo, A_a é a sua matriz dinâmica, de dimensão $n_a \times n_a$, e C_a é a matriz de saída.

O novo sistema é obtido aumentando a dimensão do sistema original com as variáveis de estado, representando o sistema autônomo do sinal de entrada, conforme descrito abaixo:

$$\begin{cases} \dot{x}(t) = Ax(t) + BC_a x_a(t) \\ \dot{x}_a(t) = A_a x_a(t), \\ y(t) = Cx(t) + DC_a x_a(t) \end{cases} \quad (2.4)$$

O novo vetor de estado tem a seguinte forma: $x_e(t) = \begin{bmatrix} x(t) \\ x_a(t) \end{bmatrix}$.

Assim, a nova equação de estado, representando o comportamento do sistema é:

$$\begin{cases} \dot{x}_e(t) = A_e x_e(t) \\ y(t) = C_e x_e(t) \end{cases} \quad (2.5)$$

onde $A_e = \begin{bmatrix} A & BC_a \\ 0 & A_a \end{bmatrix}$, $C_e = [C \quad DC_a]$ e $x_e(0) = \begin{bmatrix} x(0) \\ x_a(0) \end{bmatrix}$.

A determinação das matrizes A_e e C_e depende, essencialmente, da natureza das fontes consideradas. Normalmente, as fontes para conversores estáticos de potência são senoidais, facilitando a determinação destas matrizes.

A solução do sistema representado pela Eq. 2.5, é:

$$x_e(t) = e^{A_e t} x_e(t_0) \quad (2.6)$$

A solução dada através da Eq. 2.6 é independente do vetor de entrada, tornando, agora, possível a utilização do Algoritmo 1 para o cálculo do espaço de estado do sistema com fonte variante no tempo. O sinal $u(t)$ considerado, neste trabalho, é aquele que pode ser representado por uma série, ou seja, é um sinal periódico, como, por exemplo, um sinal senoidal.

Considerando que $u(t)$ é representado por um sinal senoidal da forma: $u(t) = V_m \text{sen}(\omega t)$, o sistema autônomo do sinal de entrada será representado da seguinte forma:

$$\begin{cases} \dot{x}_a(t) = A_a x_a(t), \\ u(t) = C_a x_a(t) \end{cases} \quad (2.7)$$

onde:

- $A_a = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$,
- $C_a = [\omega \quad 0]$ e
- $x_a(0) = \begin{bmatrix} V_m \\ 0 \end{bmatrix}$.

O novo sistema, cuja solução é independente do vetor de entrada, é dado por:

$$\begin{cases} \dot{x}_e(t) = \begin{bmatrix} \dot{x} \\ \dot{x}_a \end{bmatrix} = \begin{bmatrix} A & BC_a \\ 0 & A_a \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix}, \\ y = [C \ 0] \begin{bmatrix} x \\ x_a \end{bmatrix} \end{cases} \quad (2.8)$$

onde:

- x é o vetor de estado do sistema original;
- x_a é o vetor de estado do sistema autônomo do sinal de entrada;
- A, B, C são matrizes do sistema original;
- $A_a = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$;
- $C_a = [\omega \ 0]$;
- $y = Cx$.

Ao sistema original, serão acrescentadas duas variáveis de estado para cada senóide representada no modelo.

Considere, agora, um exemplo de um sinal de entrada senoidal em série, da forma: $u(t) = V_1 \text{sen}(\omega_1 t) + V_2 \text{sen}(\omega_2 t)$.

O sistema autônomo do sinal de entrada, composto por uma série de senóides, será representado da seguinte forma:

$$\begin{cases} \dot{x}_a(t) = A_a x_a(t), \\ u(t) = C_a x_a(t) \end{cases} \quad (2.9)$$

onde:

$$\bullet A_a = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_2^2 & 0 \end{bmatrix},$$

$$\bullet C_a = [\omega_1 \ 0 \ \omega_2 \ 0] \text{ e}$$

$$\bullet x_a(0) = \begin{bmatrix} V_1 \\ 0 \\ V_2 \\ 0 \end{bmatrix}.$$

O novo sistema, cuja solução é independente do vetor de entrada, é dado por:

$$\begin{cases} \dot{x}_e(t) = \begin{bmatrix} \dot{x} \\ \dot{x}_a \end{bmatrix} = \begin{bmatrix} A & BC_a \\ 0 & A_a \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix}, \\ y = [C \ 0] \begin{bmatrix} x \\ x_a \end{bmatrix} \end{cases}, \quad (2.10)$$

onde:

- x é o vetor de estado do sistema original;
- x_a é o vetor de estado do sistema autônomo do sinal de entrada;
- A, B, C são matrizes do sistema original;

- $A_a = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\omega_1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega_2^2 & 0 \end{bmatrix};$

- $C_a = [\omega_1 \ 0 \ \omega_2 \ 0];$

- $x_e(0) = \begin{bmatrix} x(0) \\ x_a(0) \end{bmatrix};$

- $y = Cx.$

Ao sistema original, serão acrescentadas quatro variáveis de estado para a série de senos representada no modelo do sinal de entrada.

Cálculo dos estados predecessores

Para realizar o cálculo de estados predecessores, através do procedimento de alcançabilidade para trás (*backward*), utiliza-se o *operador predecessor*. De forma semelhante ao operador sucessor, são definidos os operadores predecessores contínuos e discretos do espaço de estado $Q \times X$ do autômato híbrido.

- O operador *predecessor contínuo* $\Delta_c : 2^X \rightarrow 2^X$

Dado um conjunto de estados (q, F) , onde $q \in Q$ e $F \subseteq X$, o conjunto de predecessores contínuos de (q, F) , denotado por $\Delta_c(q, F)$, é definido como

$$\Delta_c(q, F) = \{(q, x') | \exists x \in F, \exists t > 0; x' \xrightarrow{q,t} x\}$$

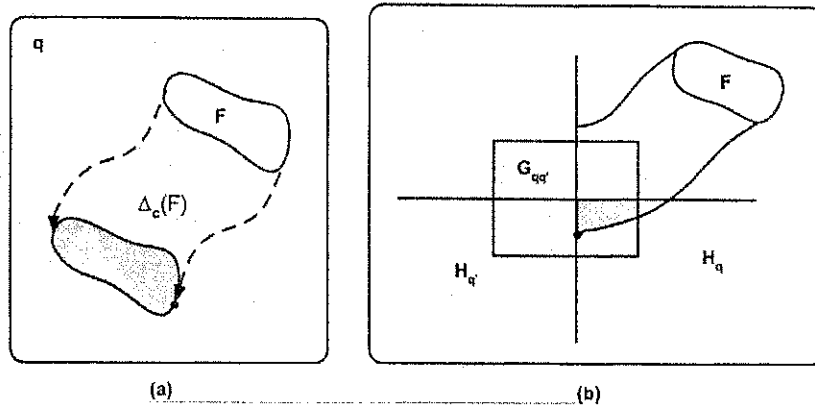


Figura 2.7: Predecessores contínuos e discretos de F

A Figura 2.7(a) ilustra este operador.

- O operador *predecessor discreto*

Dada uma transição $q, q' \in Q$ e um conjunto de estados (q, F) , onde $F \subseteq X$, o conjunto de predecessores discretos de (q, F) , denotado por $\Delta_{q'q}(q, F)$, é definido como

$$\Delta_{q'q}(q, F) = \{(q', x') | \exists x \in F \cap R_{q'q}(x') \wedge x' \in H_{q'} \cap G_{q'q}\}$$

O conjunto de predecessores discretos de (q, F) , executando todas as transições habilitadas de q , denotado por $\Delta_d(q, F)$, é definido como

$$\Delta_d(q, F) = \bigcup_{q' \in Q} \Delta_{q'q}(q, F)$$

A região hachurada da Figura 2.7(b) ilustra a aplicação deste operador.

Não há como determinar, previamente, qual a melhor opção a utilizar entre os procedimentos de cálculo de estados alcançáveis: método de alcançabilidade para frente ou para trás. A escolha, entre estes procedimentos, depende do tipo de estudo a ser realizado e do tipo de modelo ou aplicação a ser tratado(a). O método de alcançabilidade para trás é normalmente utilizado quando se deseja verificar se, a partir de uma dada condição (representada por uma região no espaço de estado), um dado conjunto de condições iniciais do sistema é alcançável.

2.5 Método de Aproximação

A caracterização dos estados alcançáveis para autômatos híbridos pode resultar em um cálculo que possivelmente pode não terminar dada a dificuldade de calcular exatamente os conjuntos

alcançáveis de sua dinâmica contínua. Assim, para aplicar algoritmos de cálculo dos estados alcançáveis, aplicando os operadores sucessores (δ) e predecessores (Δ), cuja terminação seja garantida, são necessários métodos efetivos para caracterizar conjuntos de dinâmica contínua, ou seja, métodos que possibilitem calcular, realmente, os estados alcançáveis. Estes métodos consistem na aproximação dos conjuntos alcançáveis através de uma representação simbólica de conjuntos em \mathbb{R}^n que, neste trabalho, são os objetos geométricos poliedrais.

Para que uma função seja considerada computável por uma ferramenta computacional, os elementos, em seu domínio, devem ter uma representação sintática finita e deve haver um procedimento efetivo e com término garantido que receba como entrada uma representação de um elemento do domínio e retorne como saída uma representação do valor da função aplicada a este elemento.

Nesta tese, trabalhamos com funções que operam sobre subconjunto de X tal que $X \subset \mathbb{R}^n$. Subconjuntos de números reais podem parecer objetos muito estranhos e não admitem uma representação explícita ou enumerativa. Procedimentos, como, por exemplo, para calcular os estados alcançáveis para autômatos híbridos, como o Algoritmo 1, podem ser indecidíveis, uma vez que envolvem o cálculo de funções sobre subconjuntos de números reais, tais como intersecção ($\cap : 2^X \times 2^X \rightarrow 2^X$) e verificação de equivalência ($= : 2^X \times 2^X \rightarrow \{0, 1\}$).

Para tratar os problemas de efetividade e terminação, subconjuntos de números reais podem ser expressos simbolicamente através de objetos sintáticos finitos. Esta representação simbólica pode ser feita, por exemplo, utilizando uma classe de conjuntos de poliedros. Poliedros consistem de conjuntos que podem ser representados por combinações Booleanas de desigualdades lineares. Através desta representação, a verificação de pertinência de qualquer ponto em um dado conjunto representado por um poliedro pode ser realizada, utilizando um número finito de operações lógicas e aritméticas.

A partir da utilização de uma representação simbólica de sub-conjuntos de números reais, o procedimento de alcançabilidade é alterado para calcular sub-conjuntos aproximados do espaço de estado contínuo X .

A representação sintática finita de sub-conjuntos do espaço de estado X de autômatos híbridos (sub-conjuntos em \mathbb{R}^n) consiste, neste trabalho, de conjuntos de poliedros ortogonais de grade. O poliedro ortogonal de grade é uma super(sub)-aproximação do estado alcançável calculado e contém (está contido) as trajetórias destes estados. Cada estado alcançável é aproximado para um poliedro convexo para ser, posteriormente, aproximado na forma representativa de um poliedro de grade.

Um poliedro $C \subseteq \mathbb{R}^n$ é convexo se, para qualquer par de pontos $p_1, p_2 \in C$, todo ponto z sobre o segmento de linha de p_1 a p_2 também está em C . Assim, z pode ser expresso da

seguinte forma: $z = \alpha p_1 + (1 - \alpha)p_2, 0 \leq \alpha \leq 1$.

A técnica de aproximação utilizada, nesta tese, baseia-se na técnica de cálculo de *poliedros ortogonais de grade* para representar sub-conjuntos de estados em X (ASARIN; DANG; MALER, 2000; ASARIN et al., 2001).

Os poliedros ortogonais, aqui utilizados, são subconjuntos de \mathbb{R}^n . O conjunto de grades, que constituem estes poliedros, é representado por um produto de n cubos, subconjunto de \mathbb{R}^n . Uma *grade* de um poliedro é um conjunto de pontos tal que $G_\beta = \{(z_1\beta, \dots, z_n\beta) | z_i \in \mathbb{R}^n\}, \forall \beta \geq 0, \beta \in \mathbb{R}^n$. A representação de uma grade está ilustrada, na Figura 2.8 (a).

Um *cubo* de um poliedro é um conjunto de pontos de grade $v = (v_1, \dots, v_n)$, definido como $g(v) = [v_1, v_1 + \beta] \times \dots \times [v_n, v_n + \beta]$, ou seja, é um cubo de tamanho β cujo vértice da extrema esquerda é v . Um poliedro de grade é um conjunto de cubos, como ilustrado, na Figura 2.8 (b).

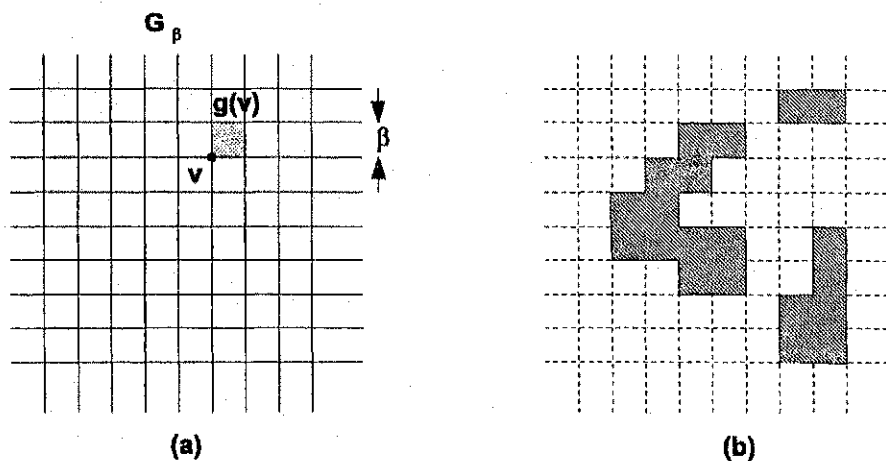
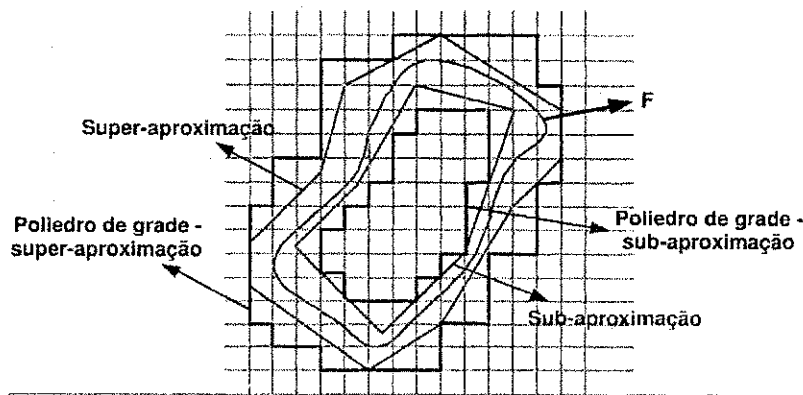


Figura 2.8: (a) Uma grade G_β (b) Poliedros ortogonais de grade

A representação de poliedros ortogonais, através de um conjunto de seus vértices, é canônica, porque todo conjunto de vértices define exatamente um poliedro convexo e todo poliedro ortogonal tem um único conjunto de vértices que o representa.

Os procedimentos de verificação de propriedades utilizam técnicas de análise de alcançabilidade, realizando uma *super-aproximação* do espaço alcançável obtido pelo operador para frente δ . Para procedimentos de reprojeto do controlador, é realizada uma *sub-aproximação* do espaço alcançável obtido pelo operador para trás, Δ . Uma super-aproximação do espaço alcançável garante que todas as trajetórias do comportamento do sistema estão contidas no espaço alcançável calculado. Uma sub-aproximação, por sua vez, garante que o comportamento do sistema controlado está restrito a um sub-conjunto máximo permitido do espaço alcançável.

A implementação do procedimento de aproximação sobre o Algoritmo 1 faz com que qualquer conjunto $F \subseteq X$ possa ser super ou sub-aproximado para um conjunto F' , consistindo de uma



Um conjunto F super e sub-aproximado por um poliedro.

Figura 2.9: Aproximação do conjunto F

união finita de poliedros ortogonais, conforme ilustrado, na Figura 2.9. Assim, todas as operações (operações Booleanas, teste de equivalência e cálculo de δ) são trocadas por suas respectivas versões aproximadas.

A utilização de poliedros de grade pelo procedimento de alcançabilidade faz com que não seja gerada uma quantidade infinita de poliedros durante a execução do procedimento. Além disto, não provê poliedros complexos para os quais não existam formas canônicas de representação. Assim sendo, o custo computacional para o teste de equivalência do algoritmo de alcançabilidade não seria muito elevado.

Os poliedros de grade são escritos como uniões de hipercubos fechados, conforme ilustrado, na Figura 2.9.

Uma vez apresentados os operadores de cálculo dos conjuntos sucessores para a dinâmica híbrida do autômato A , bem como a forma de aproximação dos conjuntos alcançáveis (poliedros de grade) para que o algoritmo de cálculo dos conjuntos alcançáveis aproximado seja computável e decidível, agora será apresentado o procedimento de alcançabilidade com aproximação.

O procedimento de aproximação consiste em aproximar todo o conjunto alcançável, passo-a-passo, a partir de um poliedro inicial, F . Inicialmente, tem-se o poliedro convexo inicial F , que pode ser representado como $F = conv(V)$, onde $conv$ é o operador casca convexa e V é um conjunto finito de vértices, como pode ser visto, na Figura 2.10(a). A partir destes vértices, calcula-se o conjunto $\delta_r(F)$, representando a aproximação do conjunto de sucessores, no tempo r , dos vértices de F (que é calculado através da integração numérica, por exemplo, para os sucessores contínuos).

O método de aproximação consiste das seguintes etapas:

1. Calcular o conjunto $C = conv(F \cup \delta_r(F))$, que é uma aproximação do poliedro alcançável

$\delta_r(F)$. Ver Figura 2.10 (b). Este conjunto não é uma super-aproximação e nem uma sub-aproximação, mas pode existir uma distância \mathcal{V} entre C e o conjunto alcançável.

2. Calcular um novo poliedro convexo C' , através da expansão das faces de C para fora do conjunto alcançável. C' , garantidamente, contém o conjunto desejado, conforme Figura 2.10 (c). O operador $bloat(C, \mathcal{V})$ expande a casca convexa de C para obtenção de um novo poliedro. O procedimento de empurrar para dentro as faces de C resultará em uma sub-aproximação.
3. Super-aproximar (ou sub-aproximar) C' pelo poliedro de grade do estado alcançável, $\delta'_r(F)$, executado pelo operador $grid_0(C)$. Ver Figura 2.10 (d).

Este método está detalhado, no Algoritmo 2.

Algoritmo 2 Algoritmo de Super-aproximação

$P^0 := \emptyset;$

$X^0 := F;$

repita

$k = 0, 1, 2 \dots$

$X^{k+1} := \delta_r(X^k);$

$C^{k+1} := conv(X^{k+1} \cup X^k);$

$C^{k+1} := bloat(C^{k+1}, \mathcal{V});$

$G^{k+1} := grid_0(C_0^{k+1});$

$P^{k+1} := P^k \cup G^{k+1};$

até $P^{k+1} = P^k$

A Figura 2.10 ilustra os passos do Algoritmo 2. O conjunto X_0 é inicializado pelo conjunto inicial F . (a) Calcula-se $X^{k+1} = \delta_r(X^k)$ representado pela linha pontilhada, que é a trajetória real começando dos vértices de X^0 , e o conjunto $\delta_r(F)$ é o tubo entre estas linhas; (b) calcula-se C^{k+1} como a casca convexa de $X^k \cup X^{k+1}$; (c) expande C^{k+1} em uma quantidade \mathcal{V} para obter o poliedro convexo C_0^{k+1} ; (d) calcula-se o poliedro ortogonal super-aproximado G^{k+1} de C_0^{k+1} . Depois da primeira iteração, obtém-se o poliedro ortogonal $G^1 \supseteq \delta_{[0,r]}(F)$. Na próxima iteração, estas 4 etapas são repetidas, começando de X^1 , para obter $G^2 \supseteq \delta_{[r,2r]}(F)$. O poliedro ortogonal $P^2 = G^1 \cup G^2$ é uma super-aproximação dos conjuntos alcançáveis depois de 2 iterações.

A representação da aproximação de conjuntos alcançáveis na forma de poliedros de grade garante a convergência do algoritmo de alcançabilidade, uma vez que poliedros de grade constituem uma classe finita de conjuntos do espaço de estados e permitem uma representação canônica relativamente eficiente para conjuntos de poliedros ortogonais. Assim, durante o cálculo dos

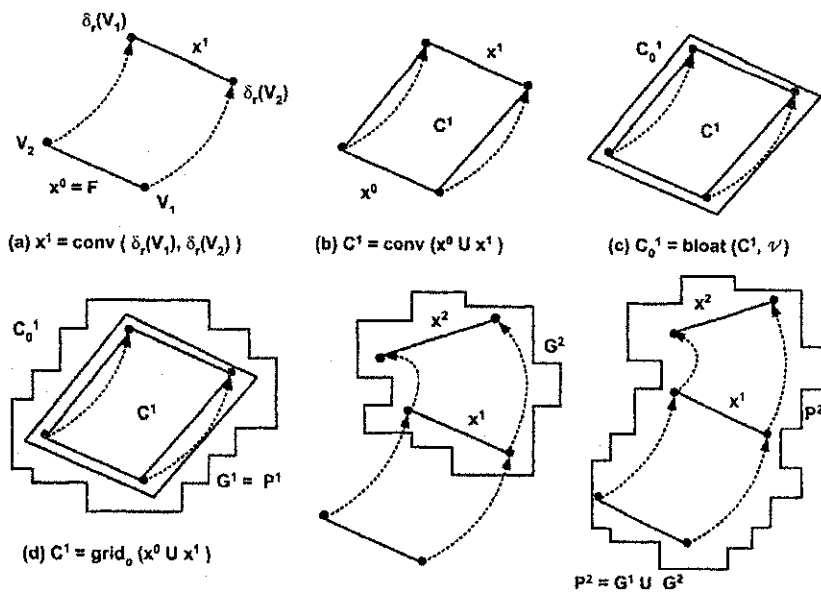


Figura 2.10: Ilustração da técnica de aproximação de Asarin-Dang

estados alcançáveis, não são calculados um número infinito de conjuntos poliedrais, uma vez que o espaço alcançável calculado é representado por um poliedro de grade único. Ao término do algoritmo, o conjunto alcançável é representado, depois de sucessivas iterações, como um único poliedro ortogonal, para que o teste de término do algoritmo seja feito de forma eficiente.

Considerações para sucessores contínuos

Para calcular o operador δ_c do Algoritmo 2 de super-aproximação de um autômato híbrido, é necessário considerar as condições de permanência, H_q , durante a evolução contínua em um estado discreto q . Para tratar as condições de permanência, é necessário remover dos conjuntos P^{k+1} os estados que não satisfazem H_q e começar a próxima iteração do conjunto resultante. No Algoritmo 2, os comandos principais para calcular o conjunto alcançável são os seguintes:

$$P^{k+1} = P^k \cup G^{k+1}$$

e

$$X^{k+1} = \delta_r(X^k)$$

onde P^k é o conjunto utilizado para acumular os estados alcançáveis e X^k representa o conjunto alcançável exato no ponto kt , utilizado como a base para o cálculo de P^{k+1} .

Para utilizar o procedimento de aproximação, considerando mudança de estados discretos em autômato híbrido, é necessário calcular a intersecção dos conjuntos P^{k+1} e X^{k+1} com H_q .

que representa a condição de permanência em cada estado q . Em outras palavras, o algoritmo é modificado para calcular o conjunto de estados sucessores contínuos, δ_c , da seguinte forma:

$$P^{k+1} = (P^k \cup G^{k+1}) \cap H_q$$

$$X^{k+1} = \delta_r(X^k) \cap H_q$$

Considerações para sucessores discretos

De forma equivalente às considerações feitas para sucessores contínuos, o operador δ_d é o conjunto de sucessores discretos de (q, F) e é escrito como

$$\delta_d(q, F) = \bigcup_{q' \in Q} \delta_{qq'}(q, F)$$

onde $\delta_{qq'}(q, F_q)$ é o conjunto de sucessores discretos de (q, F) em relação à transição de q para q' , logo $\delta_{qq'}$ é definido como

$$\delta_{qq'}(q, F) = (q', R_{qq'}(F \cap G_{qq'}) \cap H_q)$$

O poliedro ortogonal, P , é calculado de tal forma que $P \supseteq R_{qq'}(F \cap G_{qq'}) \cap H_{q'}$.

2.6 Extensão para entrada incerta

O algoritmo de alcançabilidade, visto na Seção 2.4, é utilizado para sistemas sem entradas ou com entrada constante. Este algoritmo deve ser alterado para considerar a possibilidade de modelar entradas incertas no sistema.

Considere o comportamento de um sistema, \mathcal{C} , descrito pela seguinte equação diferencial: $\dot{x} = f(x, u)$, onde $x \in X$ é o estado do sistema, $X \subseteq \mathbb{R}^n$ é o espaço contínuo do sistema e $u \in U$ é a entrada, sendo $U \subset \mathbb{R}^m$ o conjunto de entradas. Seja \mathcal{U} um conjunto de entradas admissíveis, consistindo de entradas mensuráveis da forma $\mu : T \rightarrow U$.

Uma *trajetória* de \mathcal{C} , começando de um ponto $x \in X$ sob uma dada entrada $\mu : T \rightarrow U$, é representada por $\xi_{x,\mu} : T \rightarrow X$ tal que $\xi_{x,\mu}(t)$ é a solução de $\dot{x} = f(x(t), \mu(t))$ com condição inicial $x(0) = x$.

Quando a entrada não pode ser observada, o comportamento do sistema é não-determinístico. Para uma dada condição inicial x , cada entrada μ gera uma solução diferente para a equação $\dot{x} = f(x, u)$. Assim, sob todas entradas admissíveis o sistema produz um conjunto de trajetórias.

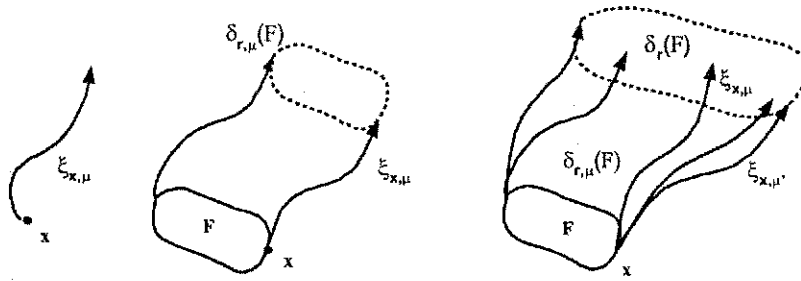


Figura 2.11: Conjunto alcançável de um sistema dinâmico com entrada.

O conjunto de estados alcançáveis de F , para $F \subseteq X$, no tempo t sob uma dada entrada $\mu \in U$, denotado por $\delta_{t,\mu}(F)$, é o conjunto de estados visitados no tempo t por todas as trajetórias começando dos pontos em F sob a entrada μ :

$$\delta_{t,\mu}(F) = \bigcup_{x \in F} \xi_{x,\mu}(t)$$

O conjunto de todos estados alcançáveis de F no tempo t , denotado por $\delta_t(F)$, é

$$\delta_t(F) = \bigcup_{u \in U} \delta_{t,\mu}(F)$$

Logo, o conjunto de todos estados alcançáveis de F durante o intervalo de tempo $[0, r]$ é

$$\delta_{[0,r]}(F) = \bigcup_{t \in [0,r]} \delta_t(F)$$

O conjunto de todos estados alcançáveis de F , depois de uma quantidade não-negativa de tempo, é $\delta_{[0,\infty]}(F)$, denotado por $\delta(F)$.

Os conjuntos $\delta_{r,\mu}(F)$, $\delta_{[0,r]}(F)$, $\delta_r(F)$ e a noção de trajetória do sistema C estão ilustrados na Figura 2.11. Nela, pode ser visto que o sistema gera trajetórias diferentes, a partir de x , sob entradas diferentes.

2.6.1 Algoritmo de alcançabilidade com entrada incerta

Considere o sistema $C = \{X, U, f\}$, onde $X \subseteq \mathbb{R}^n$ e $U \subset \mathbb{R}^n$, com dinâmicas definidas pela equação diferencial

$$\dot{x} = Ax + Bu \tag{2.11}$$

onde $x \in X$ e $u \in U$, com o conjunto de entrada U convexo.

Considere o *problema da alcançabilidade*: dado um poliedro convexo F , deseja-se calcular uma super-aproximação do conjunto alcançável de F , ou seja, $\delta(F)$.

O algoritmo de aproximação dos conjuntos alcançáveis, apresentado na Seção 2.4, é adaptado para considerar incertezas na entrada do sistema.

Considere uma face e de F cujo plano suporte é $P = \{x | \langle a, x \rangle = \langle a, y \rangle\}$, onde a é a normal para fora de e e y é um ponto arbitrário sobre a face e , chamado ponto suporte de P . Assim, o poliedro F está no espaço $H = \{x | \langle a, x \rangle \leq \langle a, y \rangle\}$.

O procedimento consiste na seguinte idéia básica: para toda face e de F , existe uma entrada $\mu^* \in \mathcal{U}$ de tal forma que, o cálculo dos sucessores do plano suporte de μ^* é suficiente para encontrar a aproximação de $\delta_t(F)$.

Para encontrar $\mu^* \in \mathcal{U}$ e, assim, calcular a aproximação de $\delta_t(F)$, é necessário executar as etapas a seguir:

1 - Calcular o vetor normal a P :

A evolução da normal a P é governada pelo sistema, cuja dinâmica é descrita pela seguinte equação diferencial:

$$\dot{x} = -A^T x. \quad (2.12)$$

Considere $\lambda_a : \mathcal{T} \rightarrow X$ a trajetória do sistema representado pela equação 2.12 com condição inicial $x(0) = a$:

$$\lambda_a(t) = e^{-A^T t} a. \quad (2.13)$$

Como pode ser observado, a trajetória λ_a da normal a e não depende da entrada.

2 - Calcular a trajetória a partir de P :

Considere que $\xi_{x,\mu}$ é a trajetória de \mathcal{C} começando do ponto x e sob a entrada μ . Assim, utilizando a solução da equação 2.11, tem-se

$$\langle \lambda_a(t), \xi_{x,\mu}(t) \rangle \leq \langle \lambda_a(t), \xi_{y,\mu}(t) \rangle. \quad (2.14)$$

A desigualdade 2.14 quer dizer que o conjunto $\delta_{t,\mu}(F)$ de estados alcançáveis de F no tempo t , sob a entrada μ , está no espaço $H_\mu(t) = \{x | \langle \lambda_a(t), \xi_{x,\mu}(t) \rangle \leq \langle \lambda_a(t), \xi_{y,\mu}(t) \rangle\}$.

Uma vez que todo ponto x no plano suporte P da face e satisfaz $\langle a, x \rangle = \langle a, y \rangle$, então, o conjunto $\delta_{t,\mu}(P)$ é o plano $P_\mu(t)$, cuja normal é $\lambda_a(t)$ e o ponto suporte $\xi_{y,\mu}(t)$:

$$\delta_{t,\mu}(P) = P_\mu(t) = \{x | \langle \lambda_a(t), x \rangle = \langle \lambda_a(t), \xi_{y,\mu}(t) \rangle\}$$

A normal $\lambda_a(t)$ ao plano $P_\mu(t)$ é independente da entrada; logo, para todo $\mu \in \mathcal{U}$ os planos $P_\mu(t)$ são paralelos uns aos outros, conforme ilustrado na Figura 2.12. Pode-se observar

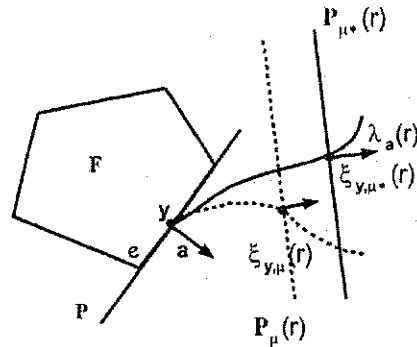


Figura 2.12: Trajetórias a partir de y com entrada incerta.

que as curvas sólidas são as trajetórias ξ_{y,μ^*} sob μ^* e as pontilhadas são as trajetórias $\xi_{y,\mu}$ sob μ . No tempo r , o plano P_{μ^*} é determinado pela normal $\lambda_a(r)$ e ponto suporte $\xi_{y,\mu^*}(r)$.

As provas e deduções para encontrar $\delta_{t,\mu}(P)$ podem ser encontradas nos artigos de Varaiya (1998) e nos trabalhos de Asarin et al. (2001), Dang (2000).

3 - Calcular μ^* :

Considere $\mu^* \in \mathcal{U}$ uma entrada que deve satisfazer a seguinte condição, para todo $t \geq 0$:

$$\langle \lambda_a(t), \xi_{y,\mu^*}(t) \rangle = \max\{\langle \lambda_a(t), \xi_{y,\mu}(t) \rangle \mid \mu \in \mathcal{U}\}.$$

Esta condição quer dizer que para todas entradas $\mu \in \mathcal{U}$, para todo $x \in F$ e para todo $t \geq 0$: $\langle \lambda_a(t), \xi_{y,\mu}(t) \rangle \leq \langle \lambda_a(t), \xi_{y,\mu^*}(t) \rangle$. Esta fórmula diz que todos os estados alcançáveis de F , no tempo t , estão dentro do espaço $H_{\mu^*}(t) = \{x \mid \langle \lambda_a(t), x \rangle \leq \langle \lambda_a(t), \xi_{y,\mu^*}(t) \rangle\}$ e o plano $P_{\mu^*}(t) = \{x \mid \langle \lambda_a(t), x \rangle = \langle \lambda_a(t), \xi_{y,\mu^*}(t) \rangle\}$ suporta o conjunto alcançável $\delta_t(F)$ em $\xi_{y,\mu^*}(t)$.

Este resultado é utilizado para calcular a aproximação de $\delta_t(F)$ e, para isto, é necessário calcular a entrada μ^* que gera o plano inicial P para $P_{\mu^*}(t)$ para todo tempo t . A função de entrada μ^* deve satisfazer a seguinte condição para todo tempo t :

$$\mu^*(t) \in \operatorname{argmax}\{\langle \lambda_a(t), u \rangle \mid u \in \mathcal{U}\}.$$

4 - Calcular a aproximação do conjunto alcançável :

Agora, as etapas acima são utilizadas para o poliedro inicial F . F pode ser representado pela intersecção de m_h espaços H_i : $F = \bigcap_{i=1}^{m_h} \{x \mid \langle a_i, x \rangle \leq \langle a_i, y_i \rangle\}$.

Seja $\mu_i^*(t) \in \operatorname{argmax}\{\langle \lambda_{a_i}(t), u \rangle | u \in U\}$, $\forall t \geq 0, i = 1, \dots, m_h$. A aproximação do conjunto alcançável a partir de F é dado por:

$$\delta_t(F) \subseteq \bigcap_{i=0}^m \{x | \langle \lambda_{a_i}(t), x \rangle \leq \langle \lambda_{a_i}(t), \xi_{y_i, \mu_i^*}(t) \rangle\}.$$

5 - Algoritmo :

Para efeito de abreviação, considere $y_i^*(t) = \xi_{y_i, \mu_i^*}(t)$ e $a(t) = \lambda_{a_i}(t)$.

Sejam $a_i(t), y_i^*(t)$ soluções para as equações diferenciais 2.15 e 2.16, $i = 0, \dots, m_h$.

$$\dot{a}_i(t) = -A^T a_i(t); a_i(0) = a_i; \quad (2.15)$$

$$\dot{y}_i^*(t) = A_{y^*}(t) + \mu_i^*(t); y_i^*(0) = y_i; \quad (2.16)$$

$$\mu_i^*(t) \in \operatorname{argmax}\{\langle a_i(t), u \rangle | u \in U\}. \quad (2.17)$$

O procedimento para calcular a aproximação do conjunto alcançável F é o seguinte:

1. Para $i = 1, \dots, m_h$;
 - (a) Calcular $a_i(t)$ resolvendo 2.15.
 - (b) Calcular $\mu_i^*(t) = \operatorname{argmax}\{\langle a_i(t), u \rangle | u \in U\}$.
 - (c) Calcular $y_i^*(t)$ resolvendo 2.16 com $\mu_i^*(t)$ obtido na etapa 1(b).
2. $\delta_t(F) = \bigcap_{i=1}^{m_h} \{x | \langle a_i(t), x \rangle \leq \langle a_i(t), y_i^*(t) \rangle\}$.

A etapa 1(a) pode ser realizada através de métodos de integração numérica. Note que, se o conjunto de entrada U é um poliedro convexo, então, a entrada $\mu_i^*(t)$ pode ser selecionada em um de seus vértices em todo ponto de tempo t .

A etapa 2 consiste na intersecção de todos espaços definidos pelos vetores normais $a_i(t)$ e os pontos $y_i^*(t)$ para obter o poliedro convexo $\delta_t(F)$, representando a aproximação dos estados alcançáveis a partir de F (ver Figura 2.13 para uma ilustração do algoritmo).

É importante notar que para sistemas lineares sem entrada (ou com entrada constante), pode-se calcular o conjunto $\delta_t(F)$ exatamente, mas para sistemas com entrada incerta, o esquema acima produz apenas uma super-aproximação deste conjunto.

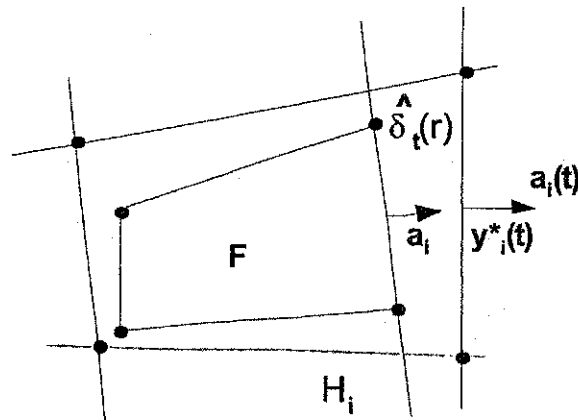


Figura 2.13: Super-aproximação do conjunto alcançável $\delta_t(F)$.

2.7 Verificação de propriedades

Verificação de propriedades é a tarefa de determinar se o sistema está construído de acordo com uma dada especificação e, para realizá-la, o projetista pode utilizar técnicas matemáticas para modelagem e análise de sistemas. Verificação de propriedades baseada em técnicas formais é utilizada na fase de projeto de circuitos de eletrônica de potência com o propósito de aumentar a confiabilidade do sistema, e um sistema é considerado confiável, se ele é capaz de executar as funções especificadas sob uma dada condição operacional.

O procedimento de verificação de propriedades mais importante para sistemas híbridos é a *análise de alcançabilidade*. Este procedimento consiste, basicamente, de técnicas de pesquisas através do cálculo dos estados alcançáveis do sistema.

As técnicas de verificação de sistemas híbridos podem ser divididas nas seguintes categorias: direta, indireta e mista. A *abordagem direta* trabalha diretamente sobre o espaço de estado contínuo do sistema. A *abordagem indireta*, por sua vez, reduz o sistema, via abstração, a um autômato de estado finito. O procedimento de abstração consiste em encontrar uma partição finita do espaço de estado tal que a alcançabilidade entre as regiões de partição é descrita pela relação de transição do autômato. Se a abstração finita pode ser construída, a verificação pode ser executada sobre o sistema abstraído com a garantia de terminação, utilizando procedimentos normalmente utilizados para sistemas de estado finito. Uma abstração finita de um sistema híbrido existe apenas para algumas de suas classes restritas (ALUR et al., 1993; ALUR; DILL, 1994b). Mesmo quando uma dada classe de sistemas admite um quociente finito, existe uma dificuldade que é calcular efetivamente o quociente, pois é necessário calcular sucessores de todas as regiões na partição, constituindo-se uma tarefa tão difícil quanto o problema de verificação.

Como uma abordagem alternativa, a *abordagem mista* é proposta em alguns trabalhos que

tentam relacionar modelos híbridos à teoria de controle supervisorio. As dinâmicas tratadas por estes trabalhos são não-triviais, assim não é construído um quociente finito exato do sistema original, porém são gerados autômatos aproximados que representam sistemas discretos finitos, cujos comportamentos incluem todos os comportamentos do sistema original (CHUTINAN; KROGH, 1999; PURI; P., 1995; STURBERG; KOWALEWSKI, 1999). A principal desvantagem desta abordagem é que, além do esforço em abstrair a dinâmica contínua, o tamanho do sistema resultante pode ser grande demais para ser utilizado por procedimentos de verificação algorítmicos.

A escolha da abordagem direta, para a análise do sistema aqui tratado, deve-se ao fato de que o estudo do sistema, através desta abordagem, é realizado diretamente no espaço de estado calculado, assemelhando-se às análises feitas nos planos de fase, comumente e tradicionalmente realizadas em estudos de circuitos de potência. Ademais, este tipo de sistema necessita de modelos matemáticos que expressem corretamente o modo e as condições sob as quais devem ocorrer as mudanças de configurações do circuito, provocadas pelas mudanças de estado dos dispositivos semicondutores, bem como devem possibilitar modelar as dinâmicas de cada modo de operação de forma não-trivial ou relativamente genéricas, tratando de entradas ou perturbações e, enfim, possibilitem tratar sistemas complexos de dimensões elevadas.

Através de um estudo realizado na literatura existente e, tendo em vista a generalidade na modelagem, algumas limitações nas ferramentas disponíveis para estudo de sistemas híbridos e a possibilidade de haver uma perda menor possível nas informações do modelo, esta tese baseia-se na abordagem direta aplicada nos procedimentos a serem utilizados para verificação e reprojeto do controlador (ASARIN et al., 2001).

O método de aproximação utilizado nesta tese e a abordagem para o reprojeto do controlador baseiam-se no método de Asarin et al. (2001) pelos seguintes motivos:

- O modelo de autômato híbrido para expressar o comportamento de circuitos de eletrônica de potência deve suportar modelagem de dinâmicas contínuas lineares, bem como lineares por partes. Assim sendo, foram preteridos modelos e métodos baseados em *autômatos temporizados* (ALUR; DILL, 1994b) por possuírem uma dinâmica bastante restrita, cuja derivada de todas as variáveis contínuas é do tipo relógio ($\dot{x} = 1$); em *autômatos lineares* (ALUR et al., 1995), cujas dinâmicas das variáveis contínuas são da forma $A \dot{x} < b$ e os problemas de verificação são semi-decidíveis; em *autômatos temporizados múltiplas taxas (multirate)* (ALUR et al., 1995) e *retangulares* (PURI; VARAIYA, 1994), cujas guardas de transições e condições de permanência são modeladas por inclusões diferenciais da forma $\dot{x} \in [a, b]$, onde os campos vetoriais são constantes em todo estado discreto. Vale ressaltar que, neste trabalho, dinâmicas não-lineares serão modeladas através de linearização por

partes.

- A representação da aproximação através de poliedros de grade tem a vantagem de garantir a terminação do procedimento de alcançabilidade, uma vez que poliedros de grade constituem uma classe finita de conjuntos do espaço de estados e permitem uma representação canônica relativamente eficiente para conjuntos de poliedros ortogonais. Com esta escolha, não há o risco de o procedimento gerar uma quantidade infinita de poliedros ortogonais, arriscando a sua não-terminação ou serem semi-decidíveis como em Wong-Toi (1997), Henzinger, Ho e Wong-Toi (1997), e diminuir o custo computacional do teste de equivalência do procedimento, uma vez que para outros tipos de poliedros, este teste seria muito mais complexo. O espaço alcançável calculado é representado por um poliedro de grade único, para que o teste de término do algoritmo seja feito de forma eficiente.
- O erro de aproximação, em cada iteração do procedimento, não é maior ou muito diferente das demais técnicas de aproximação, como por exemplo, a de Chutinan e Krogh (2000b) e elipsóide (BOTCHKAREV, 2000; B.; P., 2000; BOTCHKAREV; TRIPAKIS, 2000).
- Neste trabalho, não serão tratados sistemas de dimensões elevadas, por razões didáticas, por problemas de complexidade computacional e para possibilitar a visualização de todos os resultados dos métodos implementados. Entretanto, esta técnica consegue bons resultados para sistemas lineares de dimensões elevadas, ao contrário dos demais métodos encontrados na literatura que só trabalham com certa eficiência com sistemas com dimensão menor ou igual que 3.
- Este método contempla sistemas com dinâmica não-trivial com entradas incertas. O trabalho desenvolvido por Chutinan e Krogh (1999) trata de entradas constantes, enquanto que os trabalhos de Henzinger (2000), utilizando autômatos retangulares e lineares, e Botchkarev (2000), B. e P. (2000), introduzindo o método de elipsóides, não tratam sistemas com entradas ou perturbações.

Assim, decidiu-se por utilizar, como base para este trabalho, os métodos de Asarin, Dang e Maler (2001, 2000) por se tratar de uma abordagem que melhor se adequa aos requerimentos da aplicação base estudada.

2.8 Conclusões

Neste capítulo, foi apresentado um histórico sobre os modelos matemáticos mais importantes para estudo de sistemas híbridos. Em especial, foi apresentado com mais detalhes o formalismo de

autômatos híbridos, pois constitui-se o modelo de representação do nosso estudo sobre circuitos de eletrônica de potência. Foi necessária a inclusão de algumas definições que serão utilizadas, posteriormente, quando da apresentação dos métodos para pesquisa no espaço de estados de sistemas híbridos, que serão a base para o processo de verificação de propriedades de circuitos de eletrônica de potência, utilizando autômatos híbridos.

Foi apresentado, também, o algoritmo para calcular o conjunto de estados alcançáveis, a partir de um conjunto de condições iniciais para autômatos híbridos, e a técnica sobre a qual esta tese se baseia para calcular super-aproximações e sub-aproximações dos conjuntos alcançáveis do sistema modelado, uma vez que é necessária a representação simbólica de sub-conjuntos de números reais. Foi apresentada a técnica de aproximação de sistemas híbridos, considerando o comportamento contínuo ditado por dinâmicas lineares com entrada incerta.

Além disto, foi introduzida uma técnica para tratar sistemas com entrada variante no tempo, transformando o sistema original em um sistema autônomo. Com isso, os procedimentos de alcançabilidade, já apresentados, podem também ser utilizados para estes sistemas.

Finalmente, foram apresentadas as justificativas para a escolha da técnica de aproximação utilizada nesta tese.

No próximo capítulo, será introduzido um procedimento para obtenção do modelo do autômato híbrido que captura o comportamento de um dado circuito de potência e um exemplo de aplicação deste procedimento, considerando variação paramétrica nos seus componentes e um caso de refinamento de um dispositivo semicondutor.

Capítulo 3

Modelagem de circuitos de eletrônica de potência

Este capítulo apresenta um resumo sobre circuitos de eletrônica de potência, seus componentes ativos e passivos e seus princípios de funcionamento. É introduzido um procedimento de modelagem para obter o autômato híbrido que representa o comportamento destes circuitos. A aplicação do procedimento de modelagem proposto é exemplificado através do estudo de um conversor Buck, apresentando o modelo de autômato híbrido resultante e os resultados da simulação do comportamento do modelo obtido. Por fim, são apresentados um estudo de variações paramétricas para alguns componentes deste conversor, um exemplo do processo de refinamento de um de seus dispositivos semicondutores e um exemplo de modelo obtido para um conversor cuja fonte de entrada é variante no tempo.

3.1 Circuitos Eletrônicos de Potência

O objetivo principal dos *circuitos de eletrônica de potência* é controlar o fluxo de energia elétrica entre uma fonte e uma carga, utilizando dispositivos semicondutores de potência. A tarefa de realizar este controle de fluxo é atribuída aos conversores de potência. O conversor pode ser alimentado, a partir de uma fonte de tensão (ou corrente) alternada ou contínua, fornecendo na saída uma tensão (ou corrente) alternada ou contínua. Os tipos de controle que um conversor de potência pode executar estão limitados às características do sinal elétrico que está sendo alterado à sua estrutura, aos interruptores de potência utilizados e à técnica de comando utilizada (SILVA, 1999).

A Figura 3.1 mostra um sistema de eletrônica de potência em forma de diagrama de blocos (MOHAN; UNDELAND; ROBBINS, 1995). Normalmente, a entrada do conversor de potência é

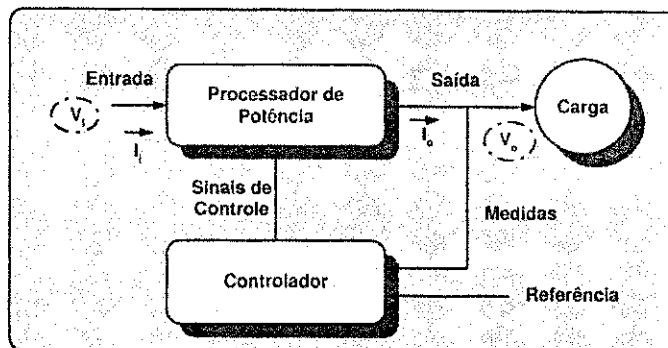


Figura 3.1: Diagrama de Bloco de um sistema eletrônico de potência

uma fonte de energia elétrica e a saída processada (tensão, corrente, frequência e o número de fases) é definida de acordo com a carga. Um controlador compara a saída do processador de potência com um valor desejado (referência), minimizando o erro, calculado a partir de medidas da saída e a referência.

O campo da eletrônica de potência tem experimentado um grande desenvolvimento devido a vários fatores. O controlador, ilustrado no diagrama da Figura 3.1, consiste de circuitos integrados ou processadores digitais de sinais, e os avanços observados na micro-eletrônica têm possibilitado o desenvolvimento de controladores com funções mais sofisticadas, ocupando uma área de silício cada vez menor, tendo, assim, maior densidade de integração. Além do mais, avanços da tecnologia de fabricação de semicondutores têm possibilitado melhorar a capacidade de processar tensão e corrente e aumentar a frequência de chaveamento das chaves semicondutoras de potência dos conversores de potência da Figura 3.1.

Existe uma demanda crescente de mercado para eletrônica de potência e uma diversidade de áreas em que estes equipamentos podem ser aplicados, tais como: conservação de energia, controle de processo e automação industrial, transporte, linhas de alta tensão, etc. A eletrônica de potência é uma área que pesquisa o processamento de energia, tendo como um de seus focos a redução do consumo, mantendo ou mesmo melhorando o desempenho dos processos e sistemas. Os aumentos constantes no custo de energia e a preocupação mundial com o meio ambiente refletem-se em pesquisas por novas técnicas de análise e ferramentas em eletrônica de potência, objetivando solucionar ou amenizar problemas de conservação de energia.

As pesquisas em eletrônica de potência, há muitos anos, vêm se pautando por critérios de qualidade dos equipamentos, dispositivos e aplicações estudadas que priorizam o aumento do rendimento elétrico (menor perda = melhor uso da energia), a minimização da "poluição" harmônica da rede elétrica e do ambiente e o aumento da confiabilidade dos sistemas eletrônicos de potência. Estima-se que a utilização da eletrônica de potência, em sistemas de energia, acarreta

uma economia de aproximadamente 35% no custo total de energia (LEE, 2001; KASSAKIAN; SCHLECHT; VERGHESE, 1992).

Apresentaremos a seguir, um resumo sobre os principais dispositivos utilizados na eletrônica de potência, com seus respectivos modos de funcionamento.

3.1.1 Dispositivos de Eletrônica de Potência

Os dispositivos semicondutores de eletrônica de potência são componentes essenciais, em um conversor, operam como interruptores e apresentam um estado de condução (baixa impedância) ou um estado de bloqueio (impedância elevada) (SILVA, 1999). A mudança de estado é chamada de comutação (disparo e bloqueio) ou de chaveamento. Os pares de terminais de potência dos interruptores eletrônicos recebem diferentes denominações, de acordo com o dispositivo considerado: anodo/catodo ou emissor/coletor ou dreno/fonte. A maioria dos dispositivos possui, além dos terminais principais, um terminal de comando, chamado de gatilho ou base, sobre o qual é possível atuar para provocar sua mudança de estado.

As atuais pesquisas relacionadas com os dispositivos semicondutores de potência são orientadas para o desenvolvimento de um interruptor eletrônico ideal. Para ser considerado um dispositivo eletrônico ideal, um interruptor deve ter as seguintes características (MOHAN; UNDELAND; ROBBINS, 1995):

- Suportar tensões direta e reversa elevadas e apresentar fluxo de corrente nula em estado de bloqueio.
- Conduzir corrente arbitrariamente elevada com queda de tensão nula em estado de condução.
- Permitir passagem instantânea do estado de bloqueio ao estado de condução, ou vice-versa, quando disparado.
- Utilizar potência desprezível para promover a comutação.

Um dispositivo real, obviamente, não possui as características ideais, elencadas acima. As principais características de funcionamento dos dispositivos reais são:

- *Tensão de bloqueio direto* - a máxima tensão positiva aplicada aos terminais do dispositivo que o mesmo suporta sem estabelecer um fluxo de corrente, quando não disparado;
- *Tensão de bloqueio reverso* - a máxima tensão negativa aplicada aos terminais do dispositivo que o mesmo suporta sem estabelecer um fluxo de corrente, no sentido reverso;

- *Máxima frequência de operação* - definida pelo tempo necessário para recombinar os portadores do dispositivo, permitindo que o mesmo readquira sua capacidade de bloqueio;
- *Potência dissipada* - existem 3 tipos de perdas nestes dispositivos: perdas no chaveamento (de disparo e de bloqueio), perdas de condução e perdas de condução reversa (devido à corrente de dispersão).

Os Principais Dispositivos

O aumento de eficiência e a diminuição dos custos têm feito com que os conversores sejam utilizados em um grande número de aplicações e em novas topologias.

Considerando os dispositivos semicondutores de potência como interruptores ideais, torna-se mais fácil estudar o funcionamento dos conversores. Com isto, os detalhes da operação do dispositivo não escondem as operações básicas do circuito e as características mais importantes do conversor podem ser mais facilmente entendidas, facilitando a escolha de quais características podem ser idealizadas.

Dispositivos semicondutores de potência podem ser classificados em 3 grupos, de acordo com seu grau de controlabilidade:

1. *Dispositivos com disparo e bloqueio espontâneos*. Conduzem e bloqueiam por condições determinadas pelo circuito.
2. *Dispositivos com disparo controlado e bloqueio espontâneo*. Conduzem através de um sinal de controle, mas bloqueiam através das condições do circuito.
3. *Dispositivos com disparo e bloqueio controlados*. Conduzem e bloqueiam através de sinais de controle.

3.1.2 Dispositivos com disparo e bloqueio espontâneos - Diodo de Potência

O diodo é um interruptor semiconductor espontâneo ou não-controlável, porque seu disparo e bloqueio são determinados pelas condições elétricas do circuito do qual faz parte, e não pela ação de um comando externo.

O diodo conduz em resposta ao comportamento de suas variáveis terminais. Se a tensão v_D for positiva, então, o diodo conduz e, quando em estado de condução, sua corrente (i_D) é positiva. O diodo bloqueia, quando a corrente que passa por ele se anula. Após a corrente se anular, uma tensão reversa deve ser aplicada em seus terminais para ajudar na recombinação dos

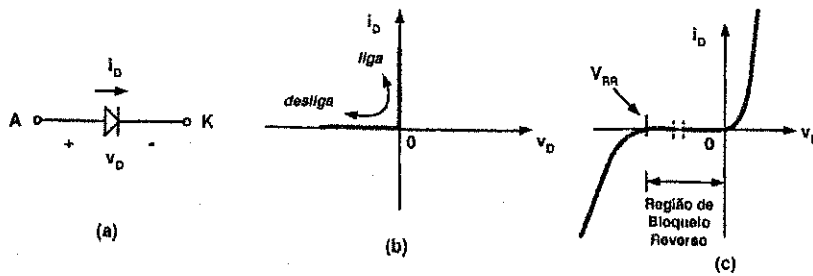


Figura 3.2: Diodo: (a) Símbolo, (b) Característica idealizada (c) Característica I-V

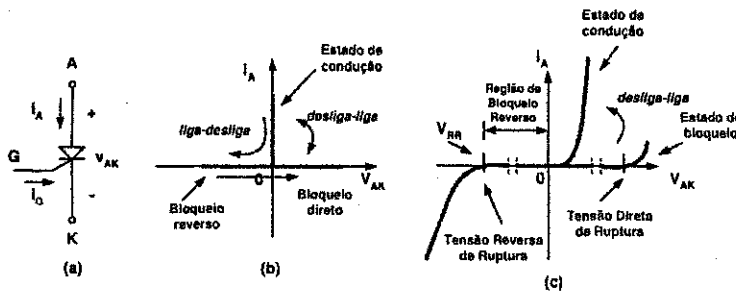


Figura 3.3: SCR: (a) Símbolo, (b) Característica idealizada (c) Característica I-V

portadores (KASSAKIAN; SCHLECHT; VERGHESE, 1992). As Figuras 3.2(a) e 3.2(b) apresentam o símbolo do circuito para o diodo e suas características *corrente* × *tensão* (I-V) idealizadas, respectivamente, e podem ser utilizadas para analisar a topologia do conversor. A Figura 3.2(c) ilustra as características *corrente* × *tensão* em estado permanente do diodo.

3.1.3 Dispositivos com disparo controlado e bloqueio espontâneo

Um tiristor é considerado, muitas vezes, um diodo semi-controlável. Seu símbolo está ilustrado na Figura 3.3(a). Em seu estado de bloqueio, o tiristor suporta uma tensão de polaridade direta e reversa, como mostrado na Figura 3.3(c), pela curva de estado de bloqueio (MOHAN; UNDELAND; ROBBINS, 1995).

O disparo do tiristor é feito injetando corrente no gatilho, quando o dispositivo está em seu estado de bloqueio direto. Quando o dispositivo começa a conduzir, a corrente do gatilho (i_G) pode ser removida e o mesmo se mantém em condução, enquanto a corrente que flui através dele (i_A) é positiva. A comutação do dispositivo ocorre sob as mesmas condições do diodo (KASSAKIAN; SCHLECHT; VERGHESE, 1992). O comportamento I-V idealizado resultante é mostrado pela curva mostrada na Figura 3.3(b).

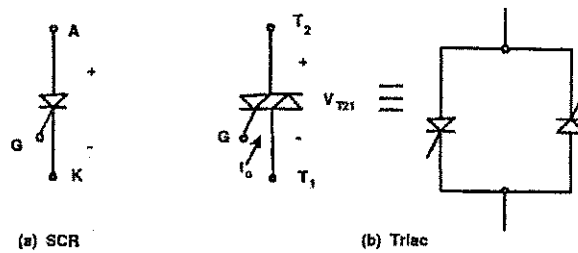


Figura 3.4: (a) SCR e (b) Triac

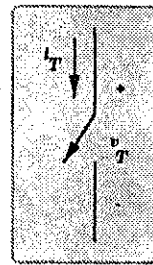


Figura 3.5: Interruptor genérico controlável

Exemplos de tiristores

O SCR (*Retificadores controlados de silício*), também chamado de tiristor convencional, é bidirecional em tensão e unidirecional em corrente. Foi o pioneiro dos dispositivos de potência e revolucionou a eletrônica de potência, nos anos 60, por ter seu disparo controlado (SILVA, 1999). Porém, depende de circuitos externos para ser bloqueado. Possui 3 terminais de acesso: o anodo (A), o catodo (K) e o gatilho (G) (Figura 3.4(a)). O SCR se comporta como um diodo em série com uma chave.

O Triac (Triode ac) é um dispositivo semicondutor desenvolvido, a partir do SCR (ou tiristor convencional), para permitir a circulação de corrente nos dois sentidos. Sua estrutura básica e seu símbolo estão indicados na Figura 3.4(b). É um dispositivo simétrico com corrente de dispersão e queda de tensão de condução direta praticamente iguais ao SCR. Além destes, o dispositivo MCT pode, também, ser considerado um exemplo de tiristor.

3.1.4 Dispositivos com disparo e bloqueio controlados

Existem vários tipos de dispositivos semicondutores de potência, que podem conduzir e bloquear através de sinais de controle aplicados ao terminal de controle do dispositivo. Estes dispositivos são representados de maneira genérica pelo símbolo do circuito mostrado na Figura 3.5. Quando o interruptor está desligado, não há fluxo de corrente e quando o interruptor está ligado, a corrente flui no sentido indicado pela seta.

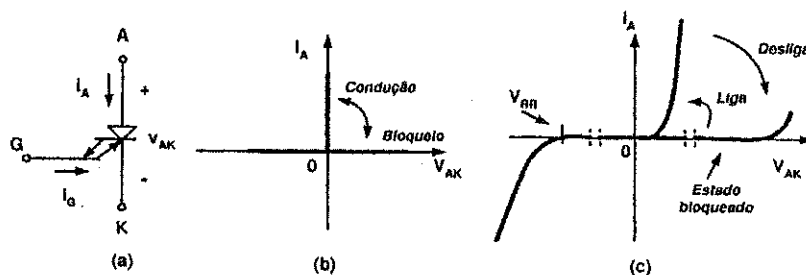


Figura 3.6: GTO: (a) Símbolo, (b) Características idealizadas (c) Característica I-V.

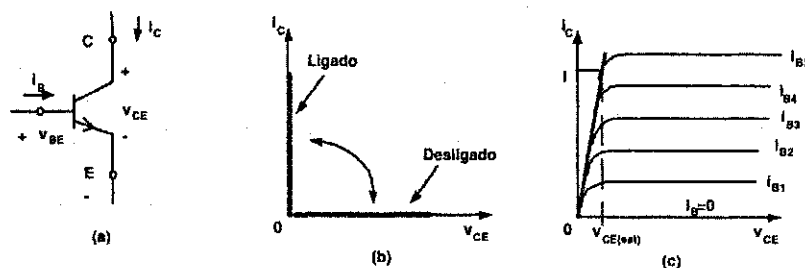


Figura 3.7: BJT: (a) Símbolo, (b) Características idealizadas (c) Característica I-V.

Agora, apresentaremos alguns interruptores controlados em disparo e bloqueio, considerando as características I-V de estado permanente. Estes dispositivos são GTOs, BJTs, MOSFETs e IGBTs.

GTO (Gate Turn-Off)

O GTO é um dispositivo que pode ser disparado através de um pulso de corrente de curta duração injetado no gatilho. Uma vez ligado, o GTO pode permanecer ligado sem a existência de qualquer corrente no gatilho, (i_G). Todavia, diferentemente do SCR, o GTO pode ser bloqueado, aplicando uma tensão negativa entre o gatilho e o catodo (v_{AK}), gerando um fluxo de corrente negativa suficientemente grande ou pelos circuitos de comutação externa. É um dispositivo unidirecional em corrente, como o SCR, mas com característica assimétrica de tensão. O símbolo do circuito para GTO é ilustrado, na Figura 3.6(a); as características idealizadas, operando como um interruptor, são mostradas, na Figura 3.6(b) e sua característica I-V é mostrada, na Figura 3.6(c) (MOHAN; UNDELAND; ROBBINS, 1995).

BJT (Bipolar Junction Transistor)

O transistor de potência - BJT - é um dispositivo, cujos estados de condução e bloqueio são determinados de forma controlada, através de um pulso de corrente. O símbolo do circuito

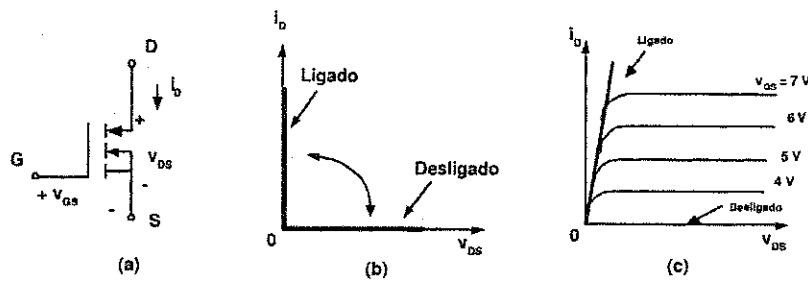


Figura 3.8: MOSFET: (a) Símbolo, (b) Características idealizadas (c) Característica I-V.

para um BJT é mostrado na Figura 3.7(a), suas características I-V idealizadas, operando como um interruptor, são mostradas na Figura 3.7(b) e as características de regime permanente são mostradas na Figura 3.7(c).

É um dispositivo unidirecional em corrente e em tensão, não suportando tensão reversa. Por esta razão, quando utilizado isoladamente, sua aplicação é limitada a chaveadores e inversores. Entretanto, associado a diodos, pode ser utilizado em circuitos de corrente alternada. O BJT necessita da presença de uma corrente contínua na base (i_B) para permanecer em estado de condução. Como ilustrado na Figura 3.7(b), uma corrente aplicada à base (i_B) muda o estado do dispositivo para condução. O bloqueio do BJT se dá através da retirada do sinal de comando na base.

MOSFET (Metal-Oxide-Semiconductor Field Effect Transistors)

O símbolo de um MOSFET é mostrado na Figura 3.8(a). É um dispositivo controlado por tensão, como indicado nas características mostradas na Figura 3.8(c). As características idealizadas da operação do dispositivo, como um interruptor, são mostradas na Figura 3.8(b). Embora a estrutura do MOSFET lembre a de um transistor bipolar de junção, a diferença aparece nas conexões feitas a essa estrutura básica. O MOSFET requer a aplicação contínua de uma tensão entre o gatilho e a fonte (v_{GS}) para se manter em condução. Se esta tensão não for aplicada, o dispositivo se comporta como dois diodos em série e permanece em estado de bloqueio. Não há fluxo de corrente no gatilho, exceto durante as transições de condução para bloqueio, ou vice-versa, quando a capacitância do gatilho está sendo carregada ou descarregada.

IGBT (Insulated Gate Bipolar Transistor)

O transistor bipolar com terminal de gatilho isolado (IGBT) foi um dos primeiros dispositivos com tecnologia híbrida bipolar-MOSFET, combinando as vantagens do MOSFET, BJT e GTO, podendo ser disparado e bloqueado pelo gatilho. Os símbolos do circuito para um IGBT são

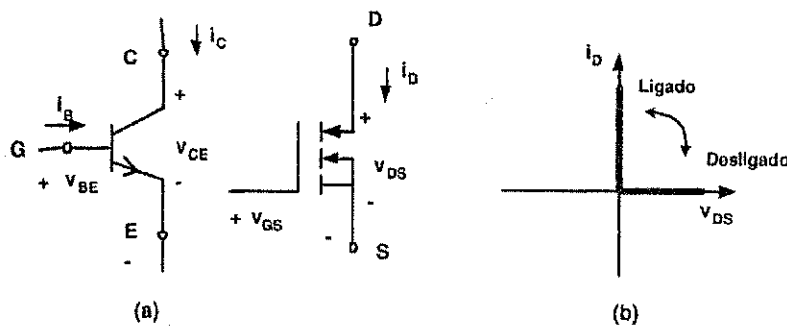


Figura 3.9: IGBT: (a) Símbolo, (b) Características I-V idealizadas.

mostrados na Figura 3.9(a) e suas características I-V idealizadas são mostradas na Figura 3.9(b).

Semelhante ao MOSFET, tem uma elevada impedância no gatilho, necessitando apenas de uma pequena quantidade de energia de comando para ser disparado o dispositivo. Como o BJT, o IGBT possui o mesmo tempo de transição para ligá-lo e tem uma pequena queda de tensão na condução, mesmo em dispositivos com tensão de bloqueio elevada. Como o GTO, o IGBT pode ser projetado para bloquear tensões negativas (MOHAN; UNDELAND; ROBBINS, 1995).

Como foi visto acima, os dispositivos semicondutores de potência estão classificados nos dispositivos com disparo espontâneo e bloqueio espontâneo (diodos), dispositivos com disparo controlado e bloqueio espontâneo (SCR e Triac) e dispositivos de disparo controlado e bloqueio controlado (GTO, MCT e transistores BJT, MOSFET, IGBT).

3.1.5 Modelos de autômatos híbridos

O comportamento de um circuito de eletrônica de potência varia conforme o estado dos dispositivos semicondutores (chaves) que o compõem. As mudanças de estados das chaves podem ocorrer devido a um sinal de comando enviado pelo controlador, representado pelos sinais de controle da Figura 3.1, gerado de acordo com uma comparação entre as medidas do processo de potência e um valor de referência especificado. Outros eventos que também podem provocar discontinuidades, no comportamento contínuo do circuito, podem ser representados por mudanças nas condições de variáveis de estado do circuito. Isto faz com que o circuito possua uma dinâmica contínua que sofre intervenções de eventos discretos, representados pelos estados das chaves. Assim, como alternativa à simulação digital, este trabalho propõe a utilização de uma ferramenta que represente o comportamento híbrido destes sistemas, considerando tanto seu comportamento contínuo quanto discreto, com o objetivo de melhorar a precisão e flexibilizar o estudo de novas topologias de conversores estáticos e de se obter uma melhor garantia de correção no comportamento destes sistemas.

É também desejável que o modelo formal trate o controlador segundo as especificações determinadas pelo projetista. Assim, se a partir do modelo de comportamento do sistema, existir um procedimento para o reprojeto do controlador, o projetista do sistema é capaz de se concentrar na concepção de novos esquemas de controle, ao invés de se preocupar com sua implementação. O projetista só tem o trabalho de verificar se o sistema controlado se comporta conforme foi especificado.

Escolhemos como ferramenta formal para modelagem e análise do comportamento global do circuito o modelo matemático denominado de autômatos híbridos. O modelo formal escolhido mostrou-se o mais adequado para representar o comportamento de circuitos de potência. Autômatos híbridos representam naturalmente o comportamento de circuitos de eletrônica de potência, espelhando, no mesmo formalismo, seu comportamento com características contínuas e discretas. Ademais, possibilita expressar o comportamento do circuito em uma dada topologia através de equações diferenciais de relativa complexidade.

Cada locação do autômato híbrido representa uma topologia (configuração¹) do circuito, com seu conjunto de equações diferenciais, exprimindo a dinâmica do circuito nesta configuração. A topologia do circuito é alterada com a ocorrência da mudança de estado dos interruptores que o compõem, caracterizando, assim, um evento discreto. Os arcos do autômato modelam estas mudanças no comportamento dinâmico do sistema que causam a transição de uma topologia para outra (MIRANDA; LIMA, 2001).

Para a obtenção do modelo matemático global, que representa o comportamento do circuito de potência, é necessária a representação em autômato híbrido dos dispositivos semicondutores que fazem parte de sua estrutura. Esta seção apresenta um conjunto de autômatos híbridos que representa estes dispositivos. Estes autômatos fazem parte do conjunto de modelos que compõem a biblioteca de autômatos híbridos (BAL), utilizada no processo de modelagem a ser apresentada, na Seção 3.3.

Os autômatos de cada dispositivo são construídos, segundo comportamento observado pela característica do dispositivo, obedecendo as seguintes regras:

1. *Locação*: Cada locação representa os estados - condução (ligado) ou bloqueio (desligado) - do dispositivo.

Uma locação adicional é acrescentada para dispositivos que apresentam em suas características um estado de ruptura reversa, como é o caso do SCR, GTO e diodo.

2. *Condição de permanência*: as invariantes de cada locação são determinadas pelas condições

¹Nesta tese, uma configuração é considerada como o modo de operação do circuito dados os estados dos dispositivos semicondutores que o compõem.

dos parâmetros do dispositivo para cada locação em particular.

3. *Transições e guardas*: as transições são construídas de acordo com a curva característica de cada dispositivo e as condições para mudança de estado são determinadas pelas variáveis que influenciam sua ocorrência.
4. *Equações dinâmicas*: as equações dinâmicas e a locação inicial são determinadas pela configuração do circuito, cujo dispositivo faz parte de sua estrutura.

Os autômatos híbridos, representando os dispositivos, apresentados na Seção anterior, modelam seus comportamentos idealizados, conforme pode ser observado pela região em destaque, e um exemplo de comportamento não ideal está representado pela condição de ruptura reversa para os dispositivos diodo, SCR e GTO.

Os autômatos híbridos para os dispositivos apresentados são os seguintes:

- Diodos de potência:

O autômato híbrido para o diodo está ilustrado, na Figura 3.10. As variáveis referenciadas no autômato são as utilizadas, na Figura 3.2, representando este dispositivo.

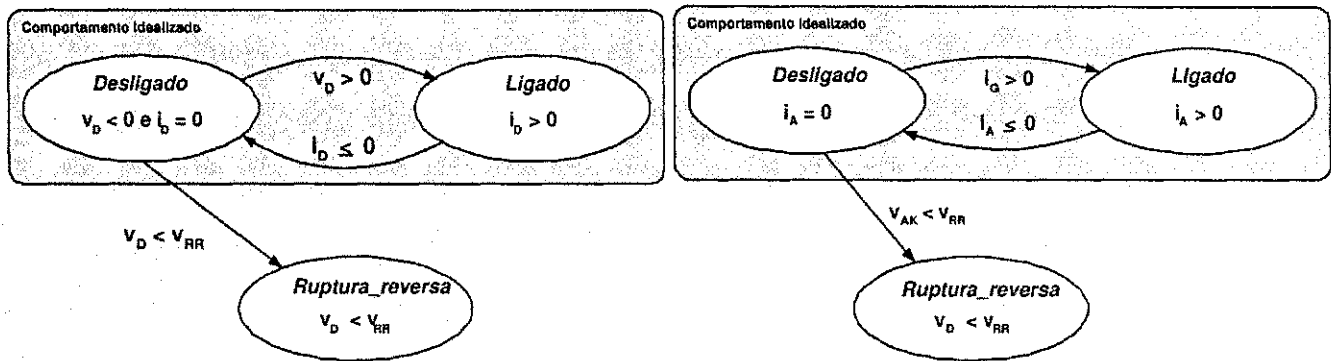


Figura 3.10: Modelo do autômato híbrido representando um diodo de potência.

Figura 3.11: Modelo do autômato híbrido representando um tiristor.

- Tiristor:

O autômato híbrido para o tiristor está ilustrado, na Figura 3.11. Este autômato utiliza as variáveis referenciadas, na Figura 3.3, representando este dispositivo.

- GTO:

O autômato híbrido para o GTO está ilustrado, na Figura 3.12. As variáveis referenciadas neste autômato são as utilizadas, na Figura 3.6, representando este dispositivo.

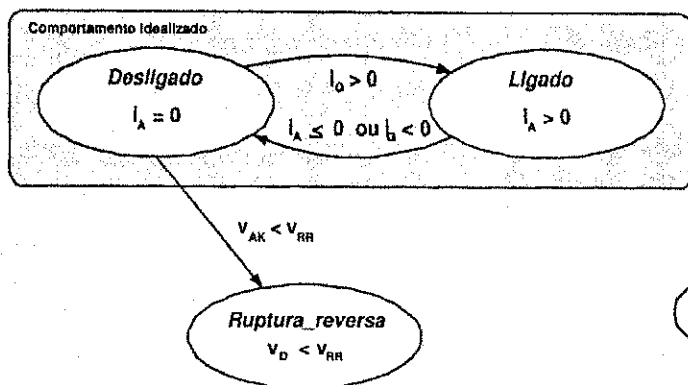


Figura 3.12: Modelo do autômato híbrido representando um GTO.

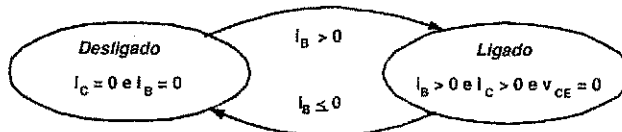


Figura 3.13: Modelo do autômato híbrido representando um BJT.

• BJT:

O autômato híbrido para o BJT está ilustrado, na Figura 3.13. Este autômato referencia as variáveis utilizadas, na Figura 3.7, representando este dispositivo.

• MOSFET:

O autômato híbrido para o MOSFET está ilustrado, na Figura 3.14. As variáveis referenciadas no autômato são as utilizadas, na Figura 3.8, representando este dispositivo.

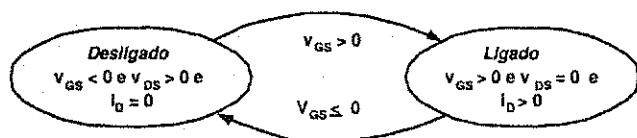


Figura 3.14: Modelo do autômato híbrido representando um MOSFET.

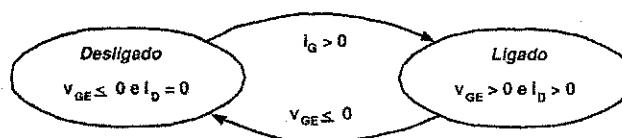


Figura 3.15: Modelo do autômato híbrido representando um IGBT.

• IGBT:

O autômato híbrido para o IGBT está ilustrado, na Figura 3.15. Este autômato referencia as mesmas variáveis da Figura 3.9, representando este dispositivo.

Neste trabalho, a utilização de técnicas formais no estudo de circuitos de potência consiste na análise de seu comportamento e no reprojeto de seu controlador. Para tanto, é necessário a obtenção do modelo matemático que capture o comportamento do circuito estudado.

É apresentado, a seguir, um procedimento que auxilia na obtenção das equações dinâmicas, variáveis de estado e configurações do circuito de forma sistemática, facilitando a construção

do seu modelo de autômato híbrido. Este procedimento constitui-se no primeiro passo para o emprego de métodos formais no estudo de circuitos de potência.

3.2 Técnica para representação da estrutura de conversores

O primeiro passo para o uso efetivo de métodos formais, no estudo de circuitos de eletrônica de potência, consiste na representação da estrutura do conversor através de blocos de elementos pré-caracterizados, representando os componentes do conversor. O produto final deste procedimento é a obtenção da topologia do conversor nos diferentes estados das chaves que o compõem, suas respectivas equações dinâmicas, o número e quais são as variáveis de estado e o número e quais são as configurações do circuito, que serão utilizadas no algoritmo para obtenção do autômato híbrido que o representa.

Para representar a estrutura de um conversor, será utilizado um procedimento sistemático baseado na técnica desenvolvida por Piquet e Demni (2000). Este procedimento consiste na combinação de elementos pré-caracterizados, representando graficamente os componentes do conversor, e é baseado num *jogo de quebra-cabeça*, no qual a montagem das peças obedece a uma especificação representada pelos contornos das mesmas.

O motivo pelo qual esta técnica é utilizada, neste trabalho, é a sua forma de simplificar e generalizar a obtenção da representação da estrutura de conversores, dada uma especificação do projetista. Além disto, o procedimento auxilia na obtenção dos elementos que compõem o modelo do autômato híbrido que representa o conversor e na eliminação automática dos estados que não podem ser fisicamente representados no modelo deste autômato.

Os exemplos de aplicação, aqui ilustrados, apesar de serem simples, demonstram a validade da técnica, bem como possibilitam identificar regras gerais que permitam o seu uso para tratar qualquer conversor.

Baseado num conjunto de elementos pré-existentes, é realizada a escolha, inicialmente, dos elementos que representam os componentes da estrutura do conversor. Os critérios de escolha baseiam-se nas propriedades dos elementos e na função que desempenham no conversor. Após a escolha dos elementos, é executado o processo de montagem destes elementos, baseado nas suas representações gráficas, para garantir a representação correta do conversor. Este método permite considerar a compatibilidade de troca de energia no conversor durante o processo de construção da sua estrutura. Os tipos de elementos componentes de um conversor que serão tratados estão descritos na Tabela 3.1.

Tabela 3.1: Tabela com elementos pré-caracterizados

Tipo	Elemento	Tipo elemento pré-caracterizado
Ativos	Chaves	Transformador (t)
Passivos	Resistência	Ganho (r)
	Indutor	Integrador (i)
	Capacitor	Integrador (c)
Fontes	Tensão	Fonte tensão (f_v)
	Corrente	Fonte corrente (f_i)

Para cada um destes componentes, é associada uma representação gráfica contendo suas características e funcionalidades. Estes componentes fazem parte de um conjunto de elementos pré-caracterizados, cuja representação gráfica das interfaces dos elementos segue o seguinte critério:

- A interface dos diferentes elementos determina a forma como é realizada a troca de energia.
- Existem duas alternativas de interface com as seguintes convenções:
 1. Transfere tensão e recebe corrente (Figura 3.16(a)).
 2. Transfere corrente e recebe tensão (Figura 3.16(b)).

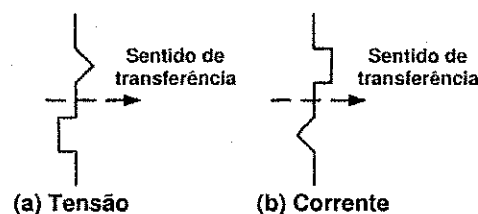


Figura 3.16: Elementos que caracterizam transferência de energia: (a) transfere tensão e (b) transfere corrente

A partir de um conjunto de elementos pré-caracterizados, para cada um deles com uma representação gráfica associada, é criada uma biblioteca de elementos que serão utilizados na obtenção da representação da estrutura de um conversor. As propriedades de cada elemento, tais como, transferência de tensão/corrente, recebimento de tensão/corrente, tipo de conversão, etc, estão expressas em sua forma gráfica.

Os elementos que fazem parte desta biblioteca e sua representação gráfica são apresentados a seguir.

3.2.1 Elementos pré-caracterizados

Fontes: São representados dois tipos de fontes:

- *Tensão:* este elemento transfere tensão sobre sua interface e recebe corrente do bloco, o qual está conectado (Figura 3.17(a)).
- *Corrente:* transfere corrente e recebe tensão (Figura 3.17(b)).

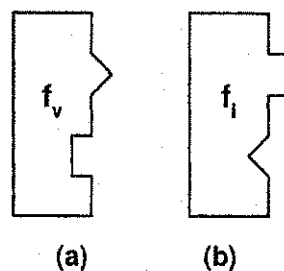


Figura 3.17: Representação gráfica do elemento *Fontes*

Ganhos: São representados pelos elementos ilustrados, na Figura 3.18. A magnitude da saída é proporcional à entrada e, comumente, representam uma *resistência*.

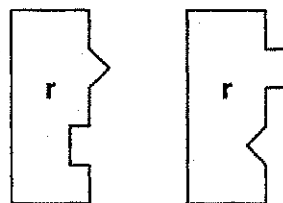


Figura 3.18: Representação gráfica do elemento *Ganho*

Transformadores e giradores: Estes elementos podem ser descritos da seguinte forma:

- *Transformadores:* São elementos que realizam conversão do tipo tensão - tensão ou corrente - corrente;
- *Giradores:* São elementos que realizam conversão do tipo tensão - corrente ou corrente - tensão;

Estes blocos preservam a potência, porém podem apresentar um ganho entre magnitudes de entrada e saída. Normalmente, estes elementos representam características de interruptores e estão ilustrados, na Figura 3.19.

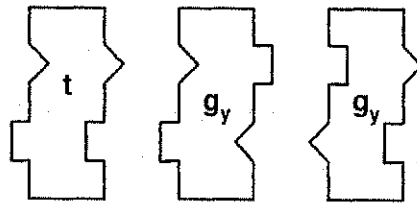


Figura 3.19: Representação gráfica dos elementos *Transformadores e Giradores*

Elementos de conexão: São representados dois tipos de elementos de conexão:

- *Transferência de tensão*(CET): uma das interfaces transfere a tensão para as outras unidades e recebe uma corrente igual a soma das correntes transferidas pelas outras interfaces do elemento. Este elemento está ilustrado, na Figura 3.20(a).
- *Transferência de corrente*(CEC): os papéis de tensão e corrente são invertidos em relação aos elementos CETs. Este elemento está ilustrado, na Figura 3.20(b).

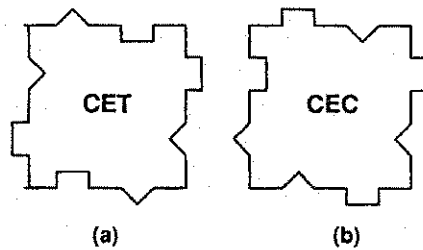


Figura 3.20: Representação gráfica dos *Elementos de conexão*

Integradores: São representados dois tipos de elementos integradores:

- *Corrente:* a tensão transferida por este elemento é a integral da corrente que este elemento recebe e, normalmente, representa um *capacitor* (ver Figura 3.21(a));
- *Tensão:* a corrente transferida por este elemento é a integral da tensão que este elemento recebe e, normalmente, representa um *indutor* (ver Figura 3.21(b)).

Conhecidos os elementos representativos dos componentes de um conversor, o próximo passo é a obtenção da representação da estrutura do conversor através de um procedimento sistemático, respeitando a compatibilidade de troca de energia.

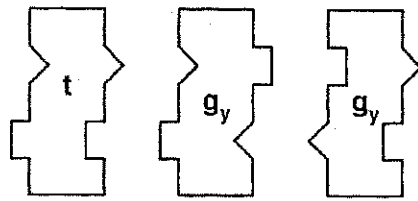


Figura 3.19: Representação gráfica dos elementos *Transformadores e Giradores*

Elementos de conexão: São representados dois tipos de elementos de conexão:

- *Transferência de tensão*(CET): uma das interfaces transfere a tensão para as outras unidades e recebe uma corrente igual a soma das correntes transferidas pelas outras interfaces do elemento. Este elemento está ilustrado, na Figura 3.20(a).
- *Transferência de corrente*(CEC): os papéis de tensão e corrente são invertidos em relação aos elementos CETs. Este elemento está ilustrado, na Figura 3.20(b).

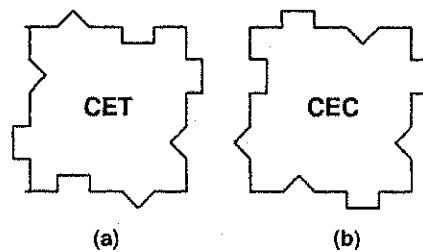


Figura 3.20: Representação gráfica dos *Elementos de conexão*

Integradores: São representados dois tipos de elementos integradores:

- *Corrente:* a tensão transferida por este elemento é a integral da corrente que este elemento recebe e, normalmente, representa um *capacitor* (ver Figura 3.21(a));
- *Tensão:* a corrente transferida por este elemento é a integral da tensão que este elemento recebe e, normalmente, representa um *indutor* (ver Figura 3.21(b)).

Conhecidos os elementos representativos dos componentes de um conversor, o próximo passo é a obtenção da representação da estrutura do conversor através de um procedimento sistemático, respeitando a compatibilidade de troca de energia.

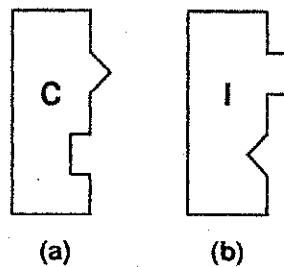


Figura 3.21: Representação gráfica dos elementos *Integradores*

3.2.2 Procedimento para representação da estrutura do conversor

A obtenção da representação da estrutura de um conversor consiste em um processo de montagem de elementos pré-caracterizados, apresentados na Seção 3.2.1, que representam os componentes do conversor. O processo de montagem destes elementos segue as regras de um jogo de *quebra-cabeça*, determinadas por suas representações gráficas.

O processo de montagem evita qualquer incoerência, uma vez que as interfaces conectadas devem ser necessariamente compatíveis. A representação gráfica dos elementos seguem critérios que são utilizados para orientar o projetista na escolha dos elementos que garantem o respeito às suas especificações.

O *procedimento de montagem da estrutura do conversor* obedece às regras abaixo:

1. Inicialmente, definir o tipo de conversão e identificar as características da fonte de energia disponível e da carga do conversor. Posteriormente, definir o tipo e as características das transformações a serem feitas para determinar a forma de energia que será adaptada obedecendo à especificação do projetista.

- (a) Identificar os elementos pré-caracterizados para fonte, conversão e carga do conversor.

Estes elementos possuem as seguintes características:

- Fonte de energia (elementos com uma interface, no lado direito)
- Carga (elementos com uma interface, no lado esquerdo)
- Elementos para a conversão a ser feita entre a fonte e a carga (elementos com 2 ou mais interfaces)

Esta etapa corresponde à definição da fonte e da carga pelo projetista, sendo possível, ainda, representar o elemento de conversão através de "macro-elementos", que contêm todas as suas características representativas.

2. Escolher os elementos, segundo as convenções determinadas pela representação gráfica de suas interfaces (conforme visto nas Figuras 3.16(a) e (b)). Estas escolhas especificam os

diferentes componentes do conversor montado, cujos perfis são conectados um ao outro, segundo suas interfaces gráficas.

3. Decompor, se necessário, os macro-elementos para especificar ou analisar seus comportamentos mais detalhadamente.

A representação final do elemento de conversão é obtida através da decomposição dos macro-elementos utilizados em sub-elementos cada vez mais precisos, aproximando-o progressivamente à realização material, enquanto respeita, em cada etapa, a coerência com os elementos já colocados.

Esta coerência pode ser descrita da seguinte forma:

- Ao identificar as características de uma interface que se deseja associar a um componente, somente os elementos possuindo uma interface compatível com este componente são candidatas a essa associação.
- As especificações e as funcionalidades representadas pelos elementos orientam o projetista em sua escolha entre os elementos candidatos.

Os níveis de profundidade no processo de decomposição e montagem dependem dos objetivos do projetista e podem ser caracterizados como *global*, se for necessário ter mais ou menos visões detalhadas do conversor, ou *local*, se for necessário enfatizar o efeito de algum elemento especial.

O procedimento de obtenção da representação da estrutura de um conversor, apresentado anteriormente, nesta Seção, garante o respeito a todas as regras representadas pelo grafismo dos elementos do *quebra-cabeça*.

A escolha de um elemento a ser montado deve ser feita, levando em consideração o objetivo geral da especificação. Seguindo a especificação, os elementos ainda não-encaixados são escolhidos, em cada etapa do procedimento, pela confrontação de regras que a traduzem e àquelas que completam os gráficos dos elementos já montados.

Após a obtenção do diagrama gráfico, através das regras do procedimento explicado anteriormente, o projetista obtém as equações do circuito para as diferentes configurações por ele assumidas, dependendo do estado dos dispositivos semicondutores modelados, utilizando as leis de nós e malhas. As equações obtidas para cada estrutura do circuito serão utilizadas para representar as dinâmicas do circuito em cada locação do autômato híbrido a ser modelado.

3.2.3 Linguagem para os elementos pré-caracterizados

A estrutura de um conversor é composta por elementos passivos (R , L e C), fontes e dispositivos semicondutores. A obtenção da representação desta estrutura pode ser feita de forma sistemática através da combinação destes elementos, de acordo com critérios pré-estabelecidos de compatibilidade e troca de energia entre eles, conforme visto na Seção 3.2.2. A representação da estrutura do conversor através de elementos pré-caracterizados pode ser formalmente descrita na forma de linguagem formal.

Uma linguagem L , definida sobre um alfabeto de símbolos Σ_L , é um conjunto de cadeias formadas por símbolos pertencentes a Σ_L . O alfabeto, Σ_L , desta linguagem é composto por símbolos que representam os elementos do *quebra-cabeça*, conforme apresentado na Tabela 3.2.

Tabela 3.2: Linguagem representativa das famílias de elementos

Família	Símbolo do alfabeto
Fonte: de tensão e corrente	f_v/F_v e f_i/F_i
Ganho	r/R
Transformador e Giradores	t/T , g_1/G_1 e g_2/G_2
Elementos de conexão	e_1/E_1 e e_2/E_2
Integradores	c/C e i/I

Os símbolos da linguagem do conversor farão parte do alfabeto

$$\Sigma_L = \{f_v, f_i, r, t, g_1, g_2, e_1, e_2, c, i, F_v, F_i, R, T, G_1, G_2, E_1, E_2, C, I\}.$$

A representação formal do conversor através de uma cadeia de caracteres da linguagem é descrita na forma de uma relação de precedência entre os símbolos do alfabeto Σ_L . A seqüência e a relação de precedência dos símbolos são determinadas pela montagem final dos elementos que compõem o conversor, descrita na Seção 3.2.2.

A estrutura do conversor pode ser representada pela cadeia de caracteres, contendo os símbolos do alfabeto Σ_L que representam os componentes do conversor.

As regras de produção para obtenção da cadeia, representando a estrutura do conversor, é definida da seguinte forma:

cadeia ::= cat(fonte, elemento _ conversão, carga)

fonte ::= f_v, F_v, f_i, F_i

elemento _ conversão ::= transformadores | giradores | elemento _ conexão | integradores | ganho | elemento _ conversão

carga ::= fonte | ganho | integradores

transformadores ::= t, T

giradores ::= g_1, G_1, g_2, G_2

ganho ::= r, R

integradores ::= c, C, i, I

elemento _conexão ::= e_1, E_1, e_2, E_2

onde $cat(c_1, \dots, c_n)$ é a operação de concatenação de símbolos e palavras sobre uma linguagem (MENEZES, 2000).

Para identificar se os componentes estão em série, é utilizado um símbolo minúsculo do alfabeto Σ_L , se o componente estiver em paralelo, ele é representado por um símbolo maiúsculo.

Os elementos transformadores e giradores representam os dispositivos semicondutores do conversor, logo estes elementos determinam o número e quais são as locações do autômato do conversor. Os elementos integradores determinam quantas e quais são as variáveis de estado do autômato híbrido.

No processo de modelagem para obter o autômato híbrido, o número de estados discretos ($NLOC$) é dado por: $NLOC = 2^D$, onde D é o número de símbolos de transformadores e giradores na cadeia de caracteres, ou seja, número de elementos discretos do autômato. O número de variáveis contínuas (CO) é determinado pela quantidade de símbolos integradores da cadeia de caracteres, representando a estrutura do circuito.

Por exemplo, um conversor, representado pela cadeia $f_v t f_i$, possui 2 modos de operação, pois em sua estrutura está o elemento t , que representa uma chave; logo, o autômato híbrido que modela o comportamento deste circuito terá duas locações.

A representação da estrutura do conversor pode ser formalmente realizada e resumida através das seguintes etapas:

1. Executar procedimento para montagem dos blocos, segundo descrito, na Seção 3.2.2.
2. Obtenção da cadeia que descreve a estrutura do conversor.
3. Executar procedimento para obtenção do autômato híbrido, conforme será descrito, no Algoritmo 3 da Seção 3.3.

3.2.4 Exemplos de aplicação do procedimento de montagem

Conversor Buck

O procedimento apresentado anteriormente será ilustrado para obtenção da estrutura do conversor da Figura 3.22.

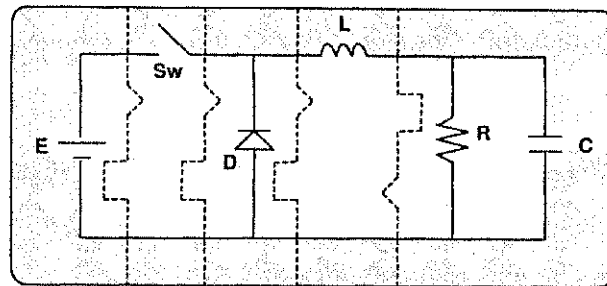


Figura 3.22: Circuito conversor Buck

Este circuito possui uma chave, Sw , associada a uma fonte de tensão (E) e uma carga, através de um elemento transformador de ganho modulado, representando uma célula de comutação.

Aplicando o conjunto de regras apresentado anteriormente, as etapas para obtenção da estrutura do circuito estão ilustradas, na Figura 3.23.

O processo da montagem dos elementos pré-caracterizados pode ser resumido nas seguintes etapas:

1. Fase 1:

- (a) Identificação da fonte de energia - fonte de tensão (f_v).
- (b) Identificação da carga.
- (c) Identificação da existência e qual tipo de elemento de conversão.

2. Fase 2:

Decomposição do elemento de conversão, inicialmente, em dois elementos de conexão: um elemento transferência de tensão (CET), ligando a fonte de tensão e o elemento de interconexão com a carga, através de uma conexão com transferência de corrente (CEC).

3. Fase 3:

Decomposição do elemento de conversão CET sob a forma de um elemento de comutação, associado a duas chaves em paralelo.

4. Fase 4:

Decomposição do elemento de conversão CEC , através da representação da indutância, composta por um elemento integrador.

5. Fase 5:

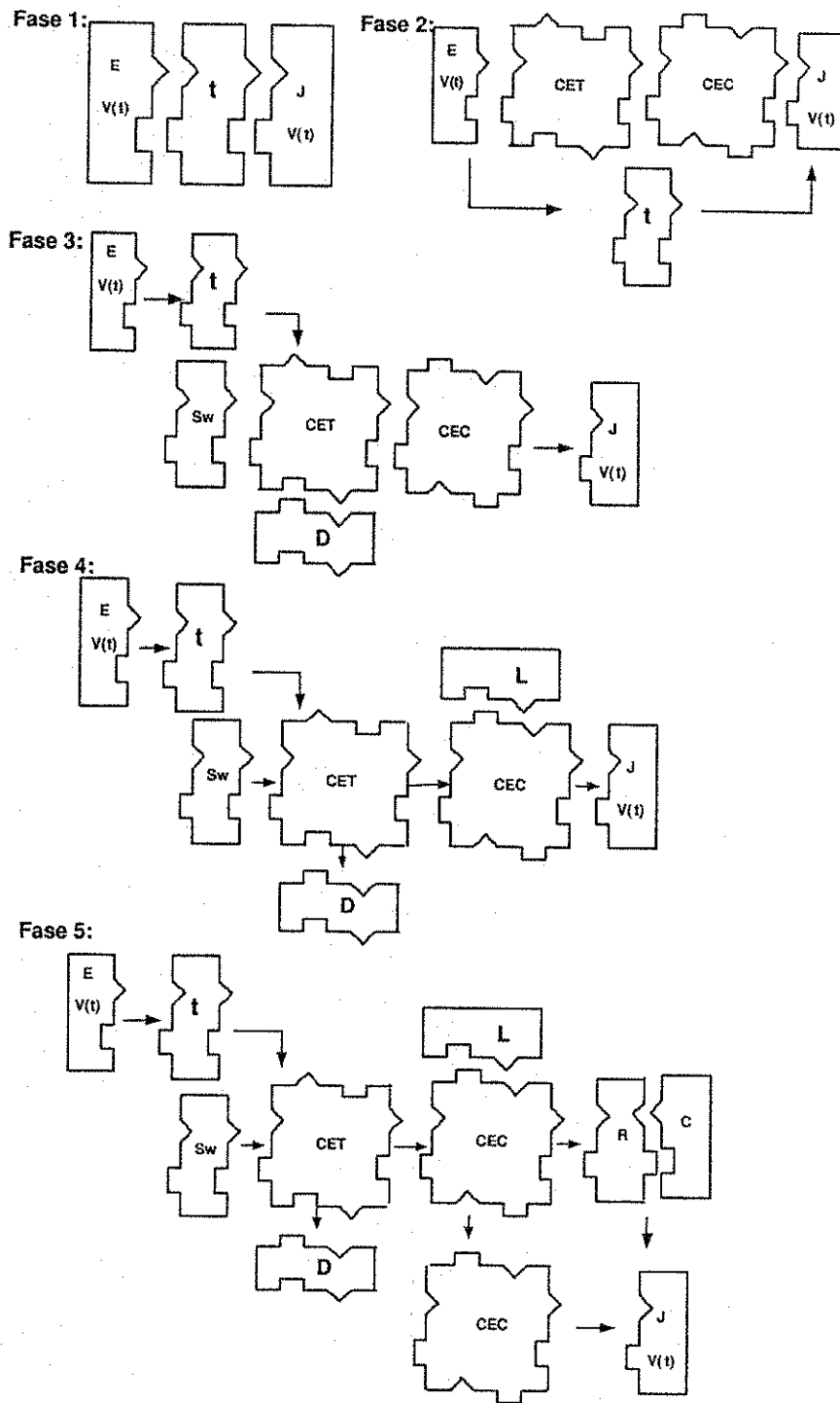


Figura 3.23: Etapas de construção da estrutura de um circuito conversor

Decomposição final do elemento de conversão CEC , através dos elementos da carga sob a forma de elementos do tipo ganho (R e C).

Uma vez obtida a estrutura através da montagem das peças, é possível representá-la através da cadeia de caracteres $f_v t T i C R$. Esta cadeia tem o seguinte significado: a fonte do conversor é do tipo tensão (f_v) em série com a chave Sw (t) e em paralelo com o diodo (T), a indutância está em série (i) com os elementos montados e, finalmente, os elementos da carga estão em paralelo com os demais (CR).

Essa cadeia obtida é utilizada no processo de modelagem para obtenção do autômato híbrido, representando o modelo matemático do conversor. Assim, são identificados os seguintes elementos:

- O número de estados discretos (NLOC) é 4, pois $NLOC = 2^D$, onde D é o número de símbolos T ou t na cadeia de caracteres, $D = 2$.
- O número de variáveis contínuas, CO é determinado pela quantidade de i ou I e C ou c da cadeia. Neste caso, o sistema é de ordem 2, com $CO = 2$.

Uma vez obtida a representação da estrutura do conversor, o projetista também obtém as equações dinâmicas para cada estado das chaves. Assim, estas equações, o número e quais os estados discretos e variáveis contínuas serão utilizados no procedimento de obtenção do autômato híbrido para circuitos de eletrônica de potência a ser apresentado na Seção 3.3.

Conversor Boost Quase-Resonante

O procedimento de obtenção da estrutura do conversor será agora aplicado para um outro exemplo, desta vez o conversor Boost Quase-Resonante ilustrado, na Figura 3.24.

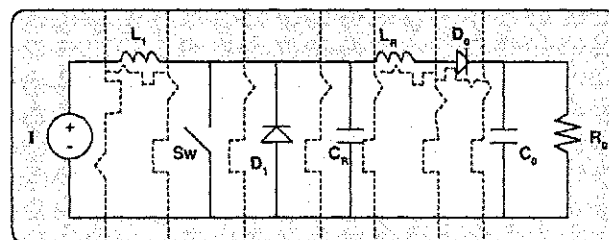


Figura 3.24: Circuito conversor Boost

Aplicando o conjunto de regras apresentado anteriormente, as etapas para obtenção da estrutura do circuito estão ilustradas, na Figura 3.25.

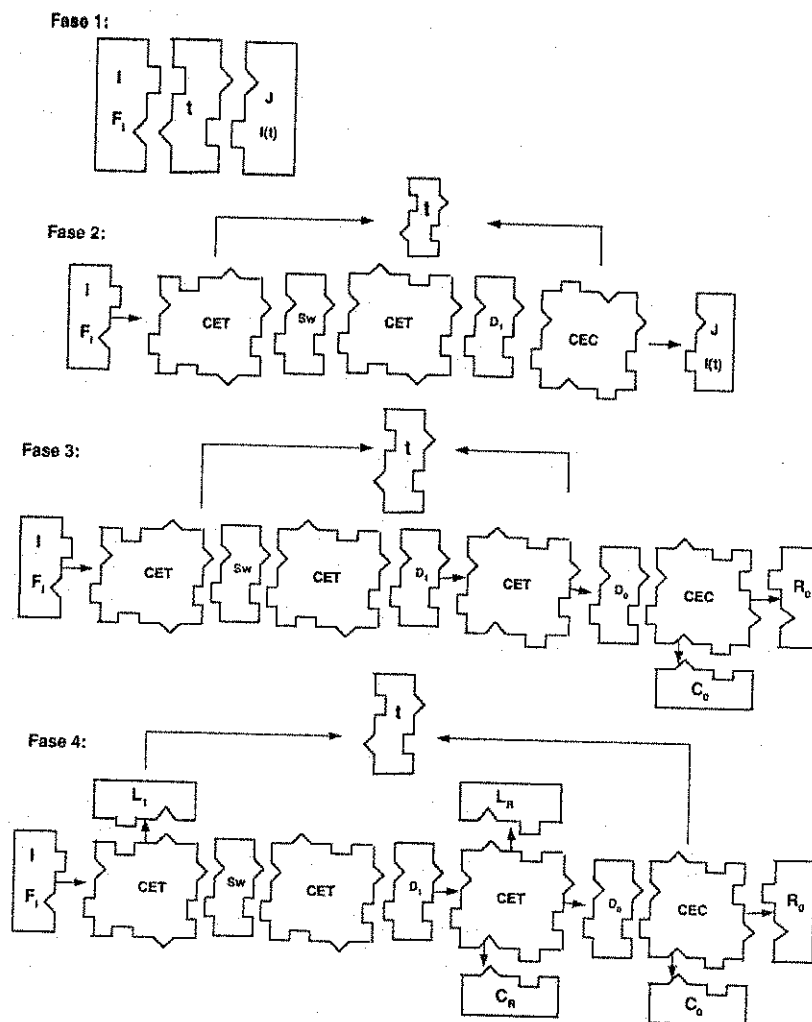


Figura 3.25: Etapas de construção da estrutura de um circuito conversor Boost

O processo da montagem dos elementos pré-caracterizados consiste das fases que seguem os critérios estabelecidos no procedimento apresentado, na Seção 3.2.2.

Após a obtenção da estrutura através da montagem das peças, a cadeia de caracteres que a representa é $f_iITTCitCR$. Os seguintes elementos são identificados:

- O número de estados discretos (NLOC) é 8, pois $NLOC = 2^D$, onde D é o número de símbolos T ou t na cadeia de caracteres, $D = 3$.
- O número de variáveis contínuas, CO , é determinado pela quantidade de i ou I e C ou c do string. Neste caso, o sistema é de ordem 4, com $CO = 4$.

Estas informações, juntamente com as equações dinâmicas obtidas através da representação da estrutura do conversor para cada estado das chaves, serão utilizados no procedimento de obtenção do autômato híbrido a ser apresentado a seguir.

3.3 Procedimento de modelagem, utilizando autômatos híbridos

Para construir os modelos de autômatos híbridos de um sistema complexo, como circuitos de eletrônica de potência, utilizamos a abordagem *top-down* (SOMMERVILLE, 2003; YOURDON, 1990; PETERS; PEDRYCZ, 2001). Inicialmente, o projetista utiliza as cadeias de caracteres e as equações dinâmicas obtidas, a partir da estrutura do conversor, e obtidos segundo a metodologia da Seção 3.2, identificando-os como os diferentes componentes do sistema que deverão ser representados por um autômato híbrido. Identifica os sub-sistemas a serem modelados e cada sub-sistema identificado será tratado como um componente do sistema global de autômatos híbridos a ser estudado. Posteriormente, especifica e modela os autômatos para cada um dos sub-sistemas, e depois o modelo global é obtido através da composição destes autômatos componentes. O processo de composição de autômatos está descrito, na Seção 2.3.2.

Este modelo pode representar, inicialmente, as características mais importantes do sistema em operação com os dispositivos semicondutores idealizados. O modelo obtido pode ser posteriormente refinado, trocando alguns modelos de dispositivos semicondutores por modelos equivalentes mais detalhados. Este processo de refinamento é realizado através da substituição dos autômatos híbridos existentes por um outro equivalente que represente o dispositivo de forma mais refinada. Este autômato equivalente é pesquisado em uma biblioteca de autômatos híbridos e, então, trocado pelo autômato original. A biblioteca de autômatos híbridos (BAH) contém um conjunto de modelos de autômatos refinados representando o comportamento detalhado dos dispositivos semicondutores. Este tipo de refinamento é empregado, geralmente, quando estuda-se

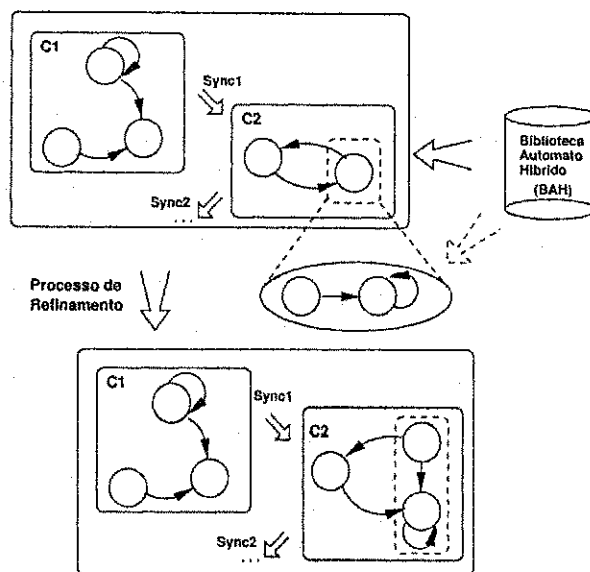


Figura 3.26: Diagrama do processo de refinamento

circuitos eletrônicos de potência com o auxílio de programas de simulação como o *PSPICE*. O processo de refinamento do modelo de autômatos híbridos está ilustrado, na Figura 3.26.

O procedimento de modelagem consiste da especificação dos componentes do circuito, através de autômatos híbridos. Isto requer a definição de todos os componentes da n -tupla $A = \langle X, Q, f, H, init, E, G, \Sigma, syn, R \rangle$. Para obter o modelo do autômato híbrido, que representa um dado circuito eletrônico de potência, as etapas descritas no Algoritmo 3 devem ser seguidas².

No Algoritmo 3, são apresentadas as etapas para a obtenção e geração dos autômatos híbridos que representam um circuito eletrônico de potência em estudo, incluindo a sincronização de eventos entre os autômatos componentes do sistema. Os dados utilizados no algoritmo são baseados na análise do sistema realizada pelo projetista, quando de sua especificação. O resultado é colocado no autômato M_g que representa o modelo global do circuito. Abaixo uma breve descrição das etapas do Algoritmo 3.

1. Inicialmente, o projetista obtém a especificação da estrutura dos sub-sistemas identificados do sistema que serão representados, segundo o procedimento da Seção 3.2. Cada cadeia de caracteres obtida ($Cadeia_i$) representa um componente C_i do modelo e o número total de cadeias obtidas é igual a N , representando o número de sub-sistemas do sistema.
2. A partir da quantidade de sub-sistemas identificados, N , o próximo passo do algoritmo

²No algoritmo, a notação $Variavel_i$ denota o conjunto de variáveis, onde i é limitado pelo contador. A notação $\leftarrow \otimes_{i=1}^N$ denota a composição dos autômatos componentes determinados por *Sync*.

Algoritmo 3 Modelagem de um Circuito Eletrônico de Potência

N : no. de sub-sistemas identificados no sistema

$Cadeia_i$: estrutura do componente do sistema

C_i : componente especificado do sistema

para $i = 1, \dots, N$ **faça**

$D \leftarrow num_T.Cadeia_i$: no. de elementos discretos da estrutura $Cadeia_i$

$CO \leftarrow num_CI.Cadeia_i$: no. de elementos contínuos da estrutura $Cadeia_i$

X_j : variáveis de estado contínuo de C_i

$NLOC \leftarrow 2^D$: no. de modos de operação de C_i

LOC_j : modos de operação de C_i

para $k = 1, \dots, NLOC$ **faça**

$F_k \leftarrow f(LOC_k, X_j)$: equação dinâmica do modo de operação k

H_k : condição de permanência no modo de operação k

fim para

TR : no. de transições de C_i

E_j : transições entre as locações de C_i

G_j : guardas das transições de C_i

$Init_i$: condição inicial de C_i

fim para

IT : no. de interações entre C_i do sistema

$Sync_j$: interação entre C_i

$Refin \leftarrow input(escolha)$

se $Refin = S$ **então**

id : componente a ser substituído

$C_{id} \leftarrow search(BAH, id)$

$Sync_j$: interação entre (C_i, C_{id})

senão

$Mg \leftarrow \otimes_{i=1}^N (Sync(C_i))$: Modelo Global

fim se

consiste em gerar um autômato híbrido para cada sub-sistema, C_i , onde $1 \leq i \leq N$.

3. O valor das variáveis D , CO , $NLOC$ de cada C_i é obtido a partir de cada $Cadeia_i$. As etapas de construção do autômato híbrido para cada C_i são:
 - (a) O número de elementos contínuos identificados é CO e cada um deles será representado por X_j , onde $1 \leq j \leq CO$.
 - (b) O número de modos de operações é dado por 2^D , equivalente ao número de locações do autômato que irá representar este componente. Alguns modos de operação, considerados fisicamente impossíveis de serem realizados, já são automaticamente descartados, utilizando o procedimento apresentado na Seção 3.2.2. A quantidade final obtida representa o número de locações do autômato ($NLOC$).
 - (c) Define-se as locações do autômato, LOC_j onde $1 \leq j \leq NLOC$.
 - (d) Para cada locação, LOC_k , definida, onde $1 \leq k \leq NLOC$:
 - i. Especificar a equação dinâmica que representa a configuração do circuito na locação LOC_k . A equação é representada pela função F_k . F_k é definida pela representação da estrutura do conversor, para cada configuração, determinada pelo procedimento da Seção 3.2.2.
 - ii. Definir as condições de permanência na locação LOC_k , H_k .
 - (e) Analisar as transições de estado do circuito a partir das configurações de cada modo de operação definida, representado pelas locações LOC_j . A quantidade de transições definida será representada por TR .
 - (f) Definir os arcos entre as locações, representadas por E_j onde $1 \leq j \leq TR$.
 - (g) Para cada arco, E_j , definir as condições de guarda (G_j , onde $1 \leq j \leq TR$) que determinam, quando satisfeitas, a mudança de modo de operação do sistema.
 - (h) Definir a locação e as condições iniciais de cada C_i .
4. Especificar os modos de sincronização entre os componentes C_i 's do sistema. O tipo de sincronização determinará a forma de composição que será realizada entre os autômatos C_i 's do sistema.
5. Verificar se é necessário realizar refinamentos em algum dispositivo semiconductor do circuito.
 - (a) Caso o refinamento seja necessário, determinar qual dispositivo (*id*) será refinado.

- (b) Realizar uma pesquisa na biblioteca de autômatos híbridos (*BAH*) e selecionar o autômato que represente o dispositivo *id*. Este autômato será chamado de C_{id} .
 - (c) Realizar a composição do autômato C_{id} com os demais autômatos componentes do sistema através do tipo de sincronização definida, na Seção 2.3.2.
6. Obtenção do *Modelo Global*, M_g , através da composição final de todos autômatos híbridos componentes C_i do sistema.

Durante o processo de modelagem, os detalhes do comportamento do circuito devem ser analisados e então expressos matematicamente, no modelo de autômato híbrido. Isto força o projetista a aprofundar seus conhecimentos sobre a operação do conversor, etapa a etapa, até que o modelo global seja obtido. A própria estrutura formal do modelo de autômato híbrido faz com que o projetista deva ter um conhecimento completo do circuito que está sendo modelado.

O modelo formal utilizado torna-se mais adequado para nossos propósitos devido aos seguintes motivos:

- É um modelo capaz de expressar as interações entre as dinâmicas discretas e contínuas e restrições de um circuito de eletrônica de potência.
- É um modelo que possibilita executar métodos de análises que são rigorosos, pois podem caracterizar todos os possíveis comportamentos do sistema e derivar seu controle.
- Um circuito de potência, como todo sistema híbrido, é complexo de ser analisado, o que torna mais do que conveniente a utilização de um procedimento automático de análise. Este modelo possibilita a utilização de uma abordagem algorítmica para verificação que analise automaticamente todos os comportamentos do sistema e decida se ele satisfaz a uma dada propriedade.

Uma vez obtido o modelo formal do circuito, agora torna-se possível realizar a análise do comportamento deste sistema, levando em consideração variações das condições iniciais e variações nos valores nominais dos componentes do circuito, ocasionadas por condições determinadas ou não pelas variáveis de estado do sistema. A verificação de propriedades destes sistemas será realizada, utilizando um procedimento de análise de alcançabilidade, ou seja, calcula-se todos os estados que podem ser alcançados por qualquer trajetória do sistema dados um conjunto de condições iniciais. Isto será feito através de um procedimento algorítmico que também será utilizada como base para o reprojeto do controlador.

Existem diversas ferramentas computacionais, comerciais ou acadêmicas, para simulação e análise de sistemas híbridos, dentre elas, podemos destacar CheckMate (CHUTINAN; KROGH,

1998), SHIFT (DESHPANDE; GOLLU; SEMENZATO, 1995), d/dt (ASARIN; DANG; MALER, 2001) e Hytech (HENZINGER; HO; WONG-TOI, 1995). Muitas delas possuem limitações quanto às dimensões do sistema a ser modelado ou à complexidade da dinâmica a ser tratada. Entretanto, este trabalho baseou-se nos métodos de aproximação implementados na ferramenta d/dt para realizar procedimentos de aproximação e reprojeto do controlador para circuitos de eletrônica de potência. Para obter maiores detalhes sobre ferramentas computacionais aplicáveis a sistemas híbridos reportar-se ao trabalho de Silva et al. (2001).

3.4 Aplicação - Conversor Buck

O uso do procedimento acima será ilustrado para o conversor Buck, visto na Seção 3.2.4 e mostrado na Figura 3.27.

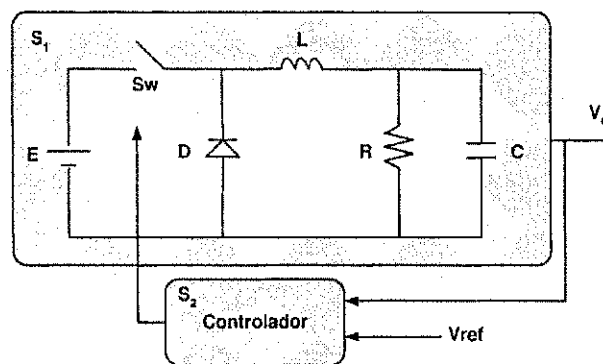


Figura 3.27: Circuito conversor Buck

Neste conversor Buck, a tensão de saída é regulada via controle por histerese, cujo funcionamento será descrito na Seção 3.4.1. A seguir, são apresentadas as condições de operação, condições iniciais e os possíveis estados de operação deste circuito para, posteriormente, explicar seu procedimento de modelagem com autômatos híbridos.

Princípio de operação

Para o processo de modelagem e simulação do comportamento deste conversor, as seguintes hipóteses são consideradas:

- A chave e o diodo são considerados ideais.
- Os elementos passivos do circuito são ideais.

Condições de Operação

- Inicialmente, a chave Sw está fechada, o diodo D não está conduzindo, considerando as seguintes condições iniciais³:

$$- i_L(0) = 2 \text{ A},$$

$$- v_C(0) = 10 \text{ V}.$$

- As especificações dos componentes são as seguintes:

$$- R = 5 \Omega,$$

$$- L = 250 \mu \text{ H},$$

$$- C = 100 \mu \text{ F},$$

$$- V_{ref} = 10 \text{ V},$$

$$- E = 20 \text{ V},$$

$$- dv = 0.1 \text{ V representa a banda de histerese}.$$

Modos de Funcionamento

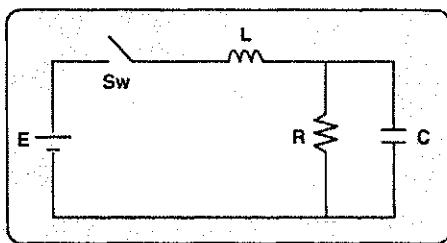


Figura 3.28: Estágio 1 - Descarga do capacitor C

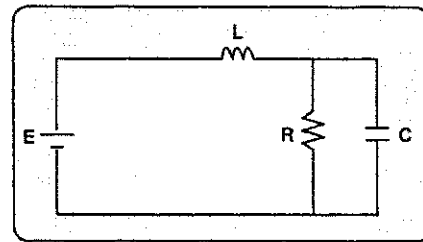


Figura 3.29: Estágio 2 - Armazenamento de energia na indutância L

O ciclo completo de chaveamento deste circuito em regime permanente pode ser dividido em três estágios de operação. Os estágios fisicamente possíveis que resumem o funcionamento do circuito são:

1. *Estágio de descarga do capacitor (C):* este estágio é representado pela Figura 3.28.

Neste estágio, a chave Sw está aberta, o diodo (D) não está conduzindo, logo não há corrente através da indutância (L). A tensão sobre o capacitor C , v_C , decresce linearmente à taxa de $v_C = -\frac{v_C}{CR}$.

³Os subscritos nos nomes das variáveis correspondem aos rótulos dos componentes na Figura 3.27.

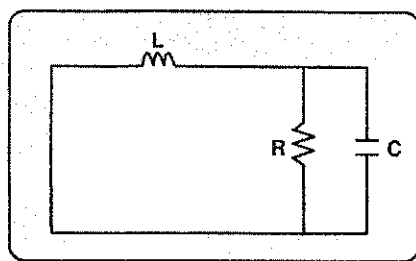


Figura 3.30: Estágio 3 - Transferência de energia da indutância L para o capacitor C

Este estágio termina quando a tensão v_C atinge a banda inferior da histerese ($v_C < V_{ref} - dv$), fazendo com que a chave Sw seja fechada, iniciando o estágio seguinte.

2. *Estágio de carga do capacitor (C)*: este estágio é representado pela Figura 3.29.

Neste estágio, a chave Sw está fechada, o diodo não está conduzindo. A tensão sobre C (v_C) e a corrente sobre a indutância (i_L) crescem linearmente. A taxa de crescimento de v_C é dada pela equação: $\dot{v}_C = \frac{i_L}{C} - \frac{v_C}{CR}$. A taxa de crescimento da corrente através da indutância, i_L , é dada pela equação: $\dot{i}_L = \frac{E}{L} - \frac{v_C}{L}$.

Este estágio termina quando a chave Sw é aberta, iniciando o estágio seguinte. A condição para Sw abrir é quando a tensão na carga atinge a banda superior da histerese, ou seja, $v_C > (V_{ref} + dv)$.

3. *Estágio de descarga linear do indutor (L)*: este estágio é representado pela Figura 3.30.

Quando o interruptor Sw abre, o diodo D conduz. Durante este estágio, i_L decresce e se atingir o valor zero, quando de sua descarga total, o novo modo de operação é o estágio 1. Se a tensão v_C atingir a banda superior da histerese antes que esta condição seja alcançada, o interruptor Sw fecha e o próximo modo de operação é o estágio 2.

As equações que refletem este comportamento é: $\dot{i}_L = -\frac{v_C}{L}$ e $\dot{v}_C = \frac{i_L}{C} - \frac{v_C}{RC}$.

3.4.1 Etapas de Modelagem

As etapas de modelagem são facilmente identificadas com as etapas do algoritmo descritas anteriormente, como a seguir:

Etapa 1 e 2. Os sub-sistemas (componentes (C'_i s)) deste sistema são o conversor de potência e o controlador em malha fechada, conforme ilustrado, na Figura 3.27, logo $N = 2$ com sub-sistemas S_1 e S_2 .

A cadeia de caracteres que representa a estrutura do conversor é $Cadeia_1 = f_v t T i C R$. A partir desta cadeia, são identificados os seguintes elementos: o número de estados discretos (NLOC) é 4, pois $NLOC = 2^D$, onde D é o número de símbolos T ou t na cadeia de caracteres, $D = 2$; o número de variáveis contínuas, CO , é determinado pela quantidade de i ou I e C ou c da cadeia. Neste caso, o sistema é de ordem 2, com $CO = 2$.

Conversor Buck

Para este conversor, construímos o seu respectivo autômato híbrido, conforme as seguintes etapas:

Etapas 3(a). Os componentes que podem causar mudanças no modo discreto são o interruptor (Sw) e o diodo (D), sendo $D = 2$. A corrente que flui através do interruptor Sw é considerada unidirecional.

Etapas 3(b). As variáveis contínuas são i_L e v_C , logo $CO = 2$, $X = \begin{bmatrix} i_L & v_C \end{bmatrix}^T$. Os subscritos nos nomes das variáveis correspondem aos rótulos dos componentes, na Figura 3.27.

Etapas 3(c) e (d). Neste caso, existem quatro modos discretos possíveis de serem modelados, como listados na Tabela 3.3.

Tabela 3.3: Modos de Operação do Conversor

Modo	Sw	D
1	Aberta	Aberto
2	Fechada	Aberto
3	Aberta	Fechado
4	Fechada	Fechado

O projetista deve analisar se todos os modos discretos são fisicamente possíveis. Neste circuito, o modo 4 não é fisicamente possível, porque quando a chave Sw está fechada, o diodo D não está conduzindo, salvo apenas em um pequeno instante de transição em que o diodo D comuta em função da ativação da chave Sw sem atraso. Então, neste caso, os modos discretos são representados por $Q_1 = \{Loc1, Loc2, Loc3\}$ ($NLOC = 3$).

Etapas 3 (e), (f), (g) e (h). A execução destas etapas pode ocorrer simultaneamente como mostrado a seguir:

O conjunto de condições invariantes (H) e o conjunto de equações diferenciais (F) representando as condições de fluxo para todos os modos de operação são:

Loc1: Condição Invariante - $v_C > 0$ & $i_L = 0$

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR} \end{bmatrix} X \quad (3.1)$$

Loc2: Condição Invariante - $i_L > 0$ & $v_C > 0$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} E \quad (3.2)$$

Loc3: Condição Invariante - $i_L > 0$ & $v_C > 0$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X \quad (3.3)$$

Etapas 3(i) A condição inicial é $i_L = 2$ A, $v_C = 10$ V, o vetor de estado é definido como

$$X = \begin{bmatrix} i_L & v_C \end{bmatrix}^T.$$

O ciclo de chaveamento do conversor começa quando Sw está fechada e D não está conduzindo (*Loc2*).

Etapas 4-6 A execução destas etapas pode ser mostrada através da exibição dos eventos e transições definidos no autômato mostrado, na Figura 3.32.

O Controlador

Nesta aplicação, a ação de controle consiste na abertura e fechamento da chave Sw . Os eventos gerados pelo conversor são sincronizados com os eventos do controlador. Neste caso, o controlador tem 2 estados, como mostrados na Tabela 3.4, que são: *Aberto* e *Fechado*, representados por $Q_2 = \{SwAberta, SwFechada\}$. O autômato do controlador gera dois eventos - *Fechar* e *Abrir* - que são enviados para o autômato do conversor.

Tabela 3.4: Modos de Operação do Controlador

Modo	Sw
SwAberta	Aberto
SwFechada	Fechado

O controle do conversor é realizado sobre a tensão de saída (v_C) com o objetivo de mantê-la no patamar determinado por V_{ref} e, para isto, é definido uma banda de histerese para a tensão v_C . Caso a tensão de saída ultrapasse os limites da banda superior (dv) ou inferior ($-dv$) da histerese, é gerado um comando para desligar ou ligar a chave Sw , respectivamente.

A lei de controle por histerese da tensão de saída, ilustrada na Figura 3.31, pode ser expressa da seguinte forma:

$$Sw = \begin{cases} 1 & \text{se } e > dv \\ 0 & \text{se } e \leq -dv \end{cases}$$

onde $e = V_{ref} - v_c$, 0 significa *Abrir* e 1 significa *Fechar* a chave Sw .

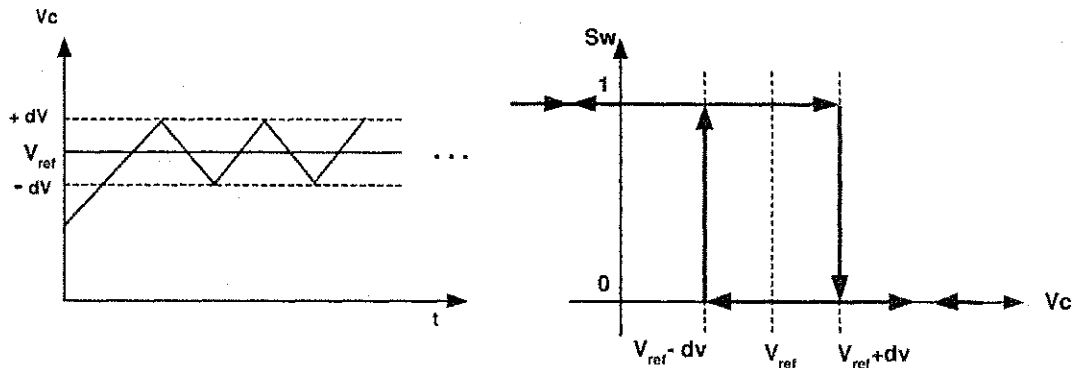


Figura 3.31: Exemplo de controle por histerese

O autômato, representando o controlador, é mostrado na Figura 3.33.

O conjunto de condições invariantes (H) para todos os modos de operação do controlador são:

- $SwAberta$: Condição invariante: $v_c \geq (V_{ref} - dv)$
- $SwFechada$: Condição invariante: $v_c \leq (V_{ref} + dv)$

As variáveis V_{ref} e dv representam a tensão desejada na saída do conversor (referência) e a banda de histerese, respectivamente.

Os modelos de autômatos híbridos descritos acima estão prontos para simulação e análise. O comportamento global do sistema é resultado do autômato produto dos autômatos componentes do sistema: conversor e controlador.

3.4.2 Autômatos Híbridos Resultantes

O autômato híbrido, que representa o conversor Buck da Figura 3.27, está ilustrado na Figura 3.32 cujas locações representam os modos descritos na Tabela 3.5.

O autômato híbrido resultante, representando o controlador, é ilustrado na Figura 3.33.

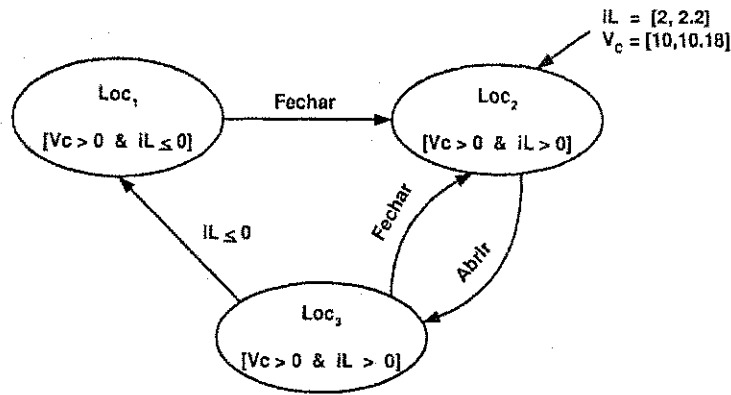


Figura 3.32: Autômato híbrido para o circuito conversor Buck

Tabela 3.5: Descrição das locações do autômato híbrido do conversor Buck

Localção	Descrição
<i>Loc₁</i>	<i>Sw</i> - aberta, <i>D</i> - aberto
<i>Loc₂</i>	<i>Sw</i> - fechada, <i>D</i> - aberto
<i>Loc₃</i>	<i>Sw</i> - aberta, <i>D</i> - fechado

3.4.3 Resultados da Simulação

Serão apresentados os resultados da simulação do modelo do autômato híbrido representado na Figura 3.32. O comportamento do circuito é obtido, resolvendo as equações de estado dos três estágios possíveis em um ciclo de chaveamento.

As formas de ondas características para corrente e tensão do conversor em operação até que o regime permanente seja alcançado são mostradas nas Figuras 3.34 e 3.35. A Figura 3.35 mostra uma nova forma de apresentação destas ondas, com uma dimensão diferente no eixo de corrente (i_L) e tensão (v_C).

Vamos explicar o comportamento do circuito através das locações do autômato, durante aproximadamente um ciclo de operação.

1. Inicialmente, a corrente na indutância L é $i_L = 2$ A e a tensão no capacitor é de $v_C = 10$

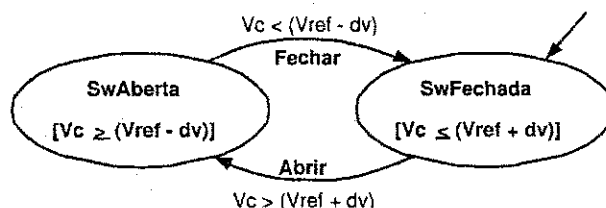


Figura 3.33: Autômato híbrido para o controlador do conversor Buck

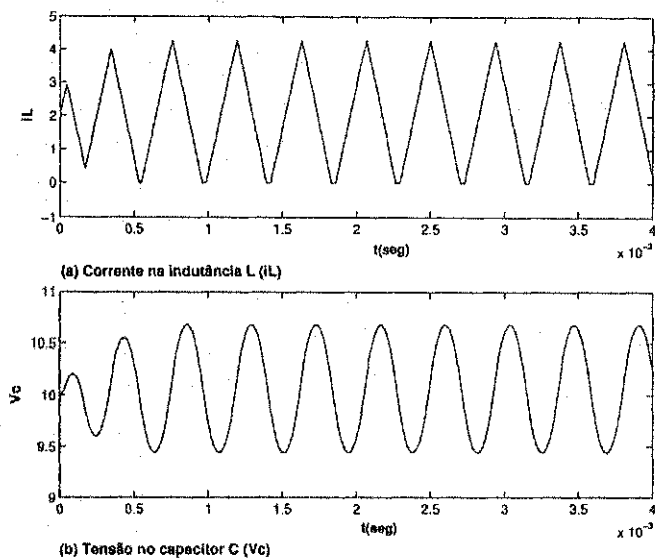


Figura 3.34: Formas de ondas de corrente (i_L) e tensão (v_C).

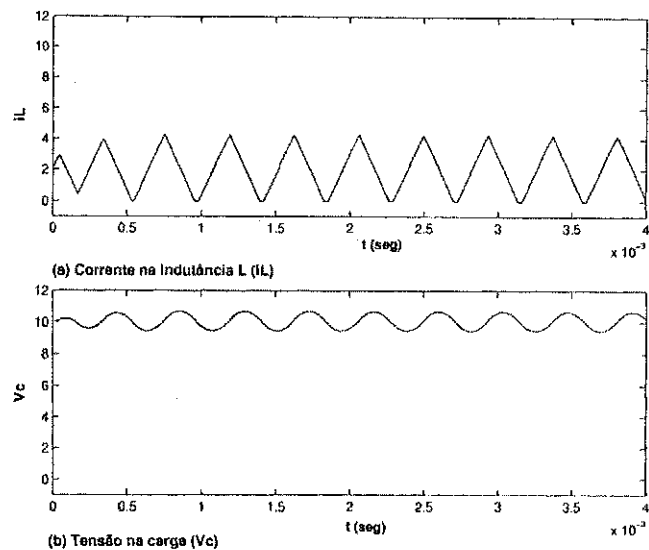


Figura 3.35: Formas de ondas de corrente (i_L) e tensão (v_C).

V , a chave Sw está fechada e o diodo D está bloqueado. O estado inicial do autômato do conversor é a Localização 2 (Loc_2), representando a topologia da Figura 3.29. Durante o período em que a operação permanece neste estado, a tensão v_C e a corrente i_L crescem até o instante em que v_C atinge a banda de histerese e o diodo conduz, alcançando o estado 3. Este estágio está representado na Figura 3.36.

- Neste estágio, representado pela localização Loc_3 do autômato, i_L é maior que a corrente da carga, ou seja, ainda existe corrente passando pelo capacitor C , logo v_C continua aumentando, uma vez que ainda tem energia da indutância L passando para C . Após um período de tempo, a corrente i_L é menor que a corrente na carga, fazendo com que o capacitor C passe a suprir a energia para o resistor R da carga até que a tensão v_C seja menor que a banda inferior de histerese. Neste instante, a chave Sw fecha e ocorre mudança para o estado 2, representado pela localização Loc_2 . De volta à localização 2 (Loc_2), a indutância L começa a ser carregada e v_C continua descarregando até o instante que v_C volta a subir e atinge a banda superior da histerese, fazendo com que a chave Sw seja aberta. O próximo estágio é a localização Loc_3 do autômato. Este comportamento está representado na Figura 3.37.
- Neste estágio, energia da indutância L acaba antes que v_C atinja o patamar inferior da banda de histerese, fazendo com que o diodo D deixe de conduzir. Assim, o estado 1 é alcançado (Loc_1). Neste estado, a corrente i_L é nula e o capacitor C vai manter a corrente da carga e quando a tensão v_C atinge a banda inferior da histerese, a chave Sw é fechada.

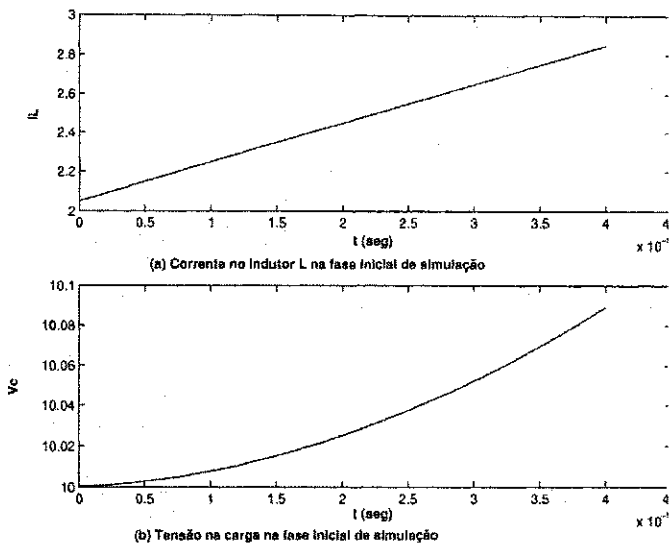


Figura 3.36: Formas de ondas de corrente i_L e tensão v_C para o estágio inicial de simulação, locação Loc_2 .

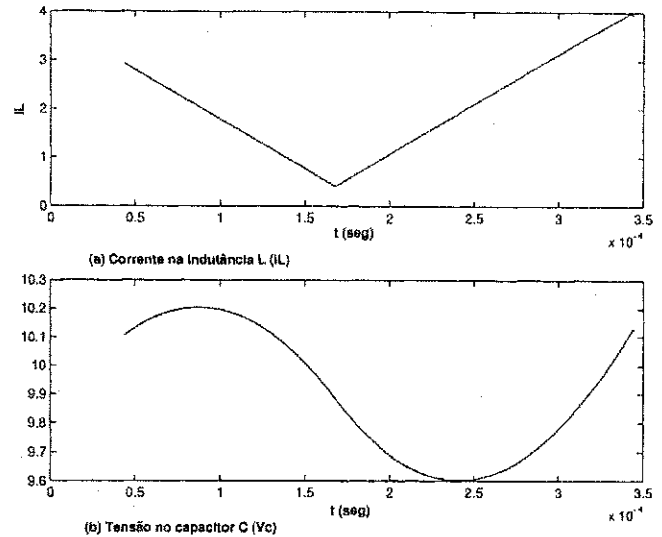


Figura 3.37: Formas de ondas de corrente i_L e tensão v_C para estágio representado pela locação Loc_3 .

Este estágio está representado na Figura 3.38.

4. É necessário um período de tempo para que a corrente da indutância (i_L) atinja a corrente da carga. Isto faz com que a tensão em v_C continue caindo por algum tempo e, posteriormente, quando i_L for igual à corrente da carga, volta a carregar.
5. Enfim, em regime permanente, a tensão máxima na carga é de 10.67 V e a corrente máxima é de 4.27 A, alcançado após, aproximadamente, 0.8 mseg, conforme pode ser visto nas Figuras 3.39 e 3.40.

3.4.4 Simulação de circuitos de eletrônica de potência com outras ferramentas

A simulação, na área de sistemas de eletrônica de potência, constitui-se uma ferramenta de análise que pode ser utilizada para substituir a prototipagem ou experiências em laboratório, especialmente, nas fases iniciais do processo de desenvolvimento do equipamento conversor de energia. Assim, a simulação objetiva antecipar a ocorrência de erros de projetos não verificados na fase de dimensionamento ou especificação do circuito, possibilitando minimizar defeitos e reduzir o tempo para a obtenção de um protótipo que funcione corretamente (ROLIM, 1999).

As características de um circuito são determinadas por sua estrutura e pelos valores dos parâmetros dos seus componentes e o seu comportamento pode ser representado por um conjunto

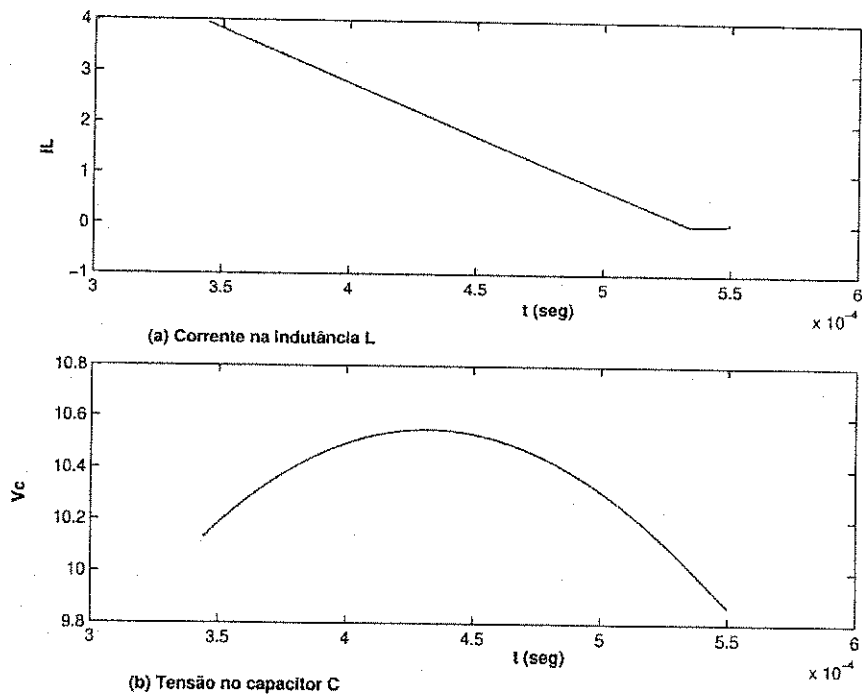


Figura 3.38: Formas de ondas de tensão (v_C) e corrente (i_L) após estado representado pela localização Loc_1 .

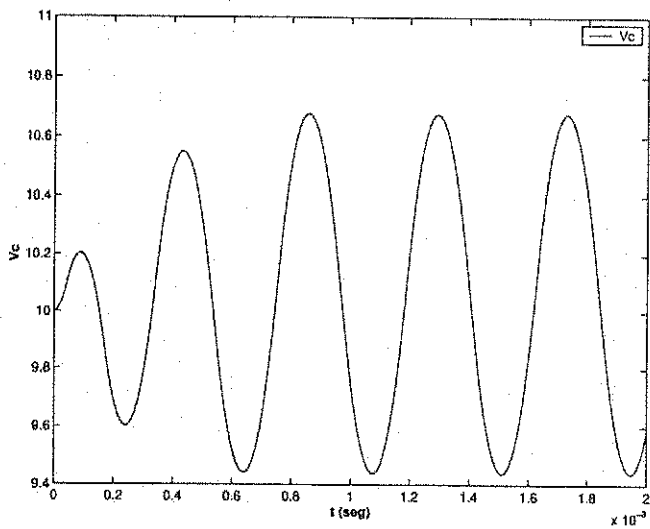


Figura 3.39: Formas de ondas de tensão (v_C) em regime permanente

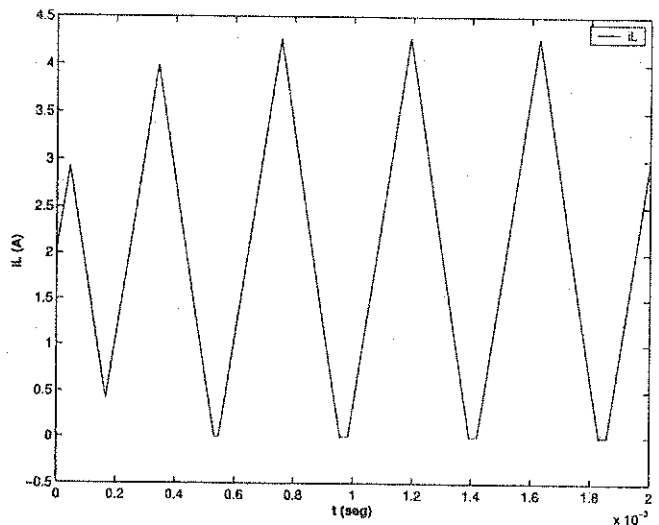


Figura 3.40: Formas de ondas de Corrente i_L em regime permanente

de equações de estado. O problema para simulação de um sistema de eletrônica de potência consiste, basicamente, em modelar o comportamento dos componentes que podem ser representados de forma linear (indutores ou capacitores) ou não-linear (para dispositivos semicondutores). A escolha da ferramenta apropriada para analisar o comportamento de circuitos é uma decisão crítica para a realização de projetos eficientes e bem sucedidos.

Normalmente, para simulação de circuitos de eletrônica de potência, são utilizados programas de computador com o objetivo de avaliar o comportamento desejado e detectar erros no projeto do circuito. Através da análise das formas de onda de corrente e tensão de cada dispositivo, geradas pelo programa de simulação, consegue-se obter uma melhor compreensão do seu funcionamento.

As ferramentas de simulação disponíveis, em sua grande maioria, provêm ao usuário interfaces para diagramas esquemáticos e *layouts* de circuitos impressos, com o objetivo de facilitar a construção do circuito ou reduzir o tempo de modelagem e testes. Os exemplos mais conhecidos de ferramentas para simulação de circuitos são SABER, PSPICE, HSPICE, ICAP3, etc (FRANZ, 1990; ESPINOZA; JO6S, 1994).

Além destas ferramentas de simulação, programas como SIMNON e MATLAB/SIMULINK constituem-se uma opção bastante utilizada para simulação de conversores, especialmente, no projeto de controladores embarcados ou projeto de processadores digitais de sinais, projetos de sistemas de controle e estudos de desempenhos de circuitos (ESPINOZA; JO6S, 1994).

Como alternativa às técnicas de simulação, este trabalho utiliza métodos formais, como forma de se obter o modelo do circuito de potência e técnicas matemáticas para estudo e verificação do seu comportamento.

Ao simular um circuito modelado através de autômatos híbridos, obtêm-se os mesmos resultados que outras ferramentas convencionais para circuitos. Entretanto, a utilização de um modelo matemático, como autômatos híbridos, para representação de circuitos de potência no estudos de suas propriedades possibilita utilizar recursos para obter outras formas de estudos e verificação de comportamento que apenas métodos de simulação, como usualmente é feito por ferramentas convencionais de simulação de circuitos, não provêm.

Uma das limitações de estudo de comportamento, utilizando ferramentas de simulação, é quando se deseja verificar o comportamento do circuito dado um intervalo de condições iniciais. Os métodos de simulações são utilizados para obter uma aproximação do comportamento do sistema para uma determinada condição inicial. Na prática, as condições iniciais, geralmente, não são conhecidas exatamente para um único valor. Sabe-se, entretanto, que estão dentro de um intervalo de valores. Conseqüentemente, ao invés de se obter o resultado para uma única entrada, é necessário realizar simulações para um número elevado de entradas.

Em se tratando de circuitos de potência, a simulação em computador é considerada uma

ferramenta excelente na avaliação do comportamento desejado e na detecção de erros. Porém, estudos da operação de circuitos complexos ou determinados tipos de análise, como os efeitos verificados por um conjunto de condições iniciais, são realizados através de testes exaustivos, ou seja, é necessário fazer uma simulação para cada uma condição inicial desejada, não existindo a garantia na correção global dos resultados, uma vez que pode ser impossível executar simulações para um número elevado de dados.

Existem outras limitações do uso de técnicas de simulação para estudar circuitos de potência, particularmente, quando se deseja analisar circuitos com oscilação na fonte (corrente ou tensão) de entrada. Este tipo de análise é difícil de ser realizado apenas utilizando simulações, uma vez que é necessário caracterizar todos os possíveis comportamentos resultantes de um intervalo de valores de entrada do sistema. É muito difícil simular um sistema com todas as entradas possíveis, pois seria necessário simulações exaustivas para isto.

O modelamento de um circuito de potência, utilizando métodos formais, possibilita especificar o seu comportamento completo de forma bastante realística, representar um conjunto de valores para as condições iniciais e tratar a análise do comportamento destes sistemas, a partir de um intervalo de valores para as entradas e para os valores dos parâmetros dos componentes do sistema. A verificação do comportamento do sistema é realizada sem que seja necessário a execução de um número elevado de simulações.

Resultados obtidos

Com o propósito de verificar se o comportamento expresso pela representação formal obtida, através de autômatos híbridos, é consistente, utilizamos a ferramenta computacional Stateflow (MATHWORKS, 1998) para implementação dos dois autômatos obtidos, conversor e controlador, e simulamos o sistema de acordo com uma determinada condição inicial especificada. O comportamento contínuo de cada locação, a execução de cada transição feita durante as mudanças de estado e ocorrência de eventos enviados pelo controlador foram validados com os resultados obtidos por esta ferramenta. Vale salientar que as formas de ondas obtidas com esta ferramenta, espelhando o comportamento dos autômatos híbridos, consideram características ideais dos componentes do circuito, ou seja, o chaveamento é instantâneo, sem efeitos parasitas, etc.

A validação do modelo também foi realizada comparando o comportamento obtido através das curvas apresentadas com sua simulação equivalente no PSPICE. As formas de ondas para corrente e tensão para o conversor, utilizado como exemplo obtidas pelo PSPICE, estão ilustradas na Figura 3.41. A ferramenta PSPICE foi utilizada como uma outra forma de verificar a consistência dos resultados obtidos, no modelo formal, representando o conversor e verificar se seu comportamento é semelhante com as demais ferramentas utilizadas na simulação de circuito

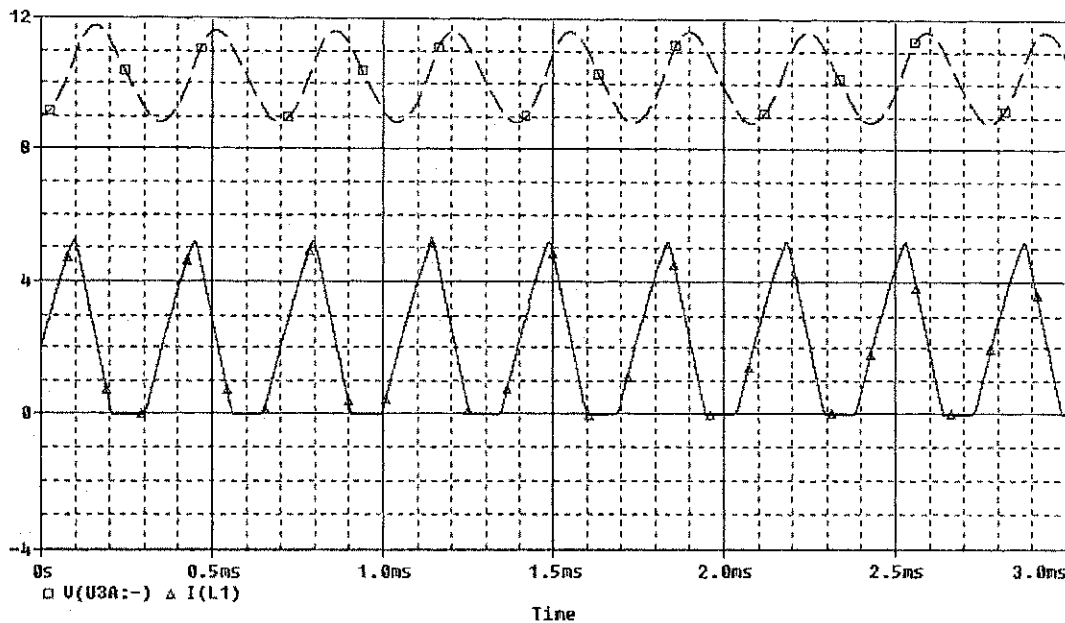


Figura 3.41: Formas de ondas de tensão (v_C) e corrente (i_L) obtidas pelo PSPICE.

de eletrônica de potência. As pequenas possíveis diferenças que possam ser encontradas, comparadas com a Figura 3.34, devem-se ao fato de que os componentes modelados no PSPICE possuem características mais próximas das características reais destes componentes.

3.4.5 Variação paramétrica em componentes do circuito

A utilização de métodos formais para análise de circuitos de potência possibilitou a realização de estudos no conversor Buck acima, considerando casos de variação paramétrica nos seus componentes.

Para construir o modelo do autômato híbrido com variação de parâmetro em seus componentes, as etapas abaixo devem ser seguidas:

1. Obter autômato híbrido original, executando os procedimentos do Algoritmo 3.
2. Para cada locação (LOC_i) do autômato obtido faça
 - (a) Se (Mudança de parâmetro é determinada por uma condição na variável de estado ($x_j \in X$)) então
 - Se (Condição de permanência da locação (H_i) contém testes com esta variável de estado) então
 - Criar uma nova locação para a nova condição, determinada pela variável de estado

FimSe**Senão**

- Criar uma nova locação para cada novo valor de parâmetro

FimSe

- Atualizar as condições de permanência (H_i) das locações originais e novas com os testes que determinam a mudança de parâmetro.
- Alterar as equações dinâmicas de cada nova locação, modificando as devidas referências ao uso do parâmetro alterado.
- Criar uma transição da locação original para a nova locação, cuja guarda é a condição sob a qual o parâmetro deve variar.

FimPara**3. Para** cada nova locação criada do autômato obtido faça

- Criar uma transição para as demais locações existentes e, vice-versa, quando a locação original à nova possuir transições para estas locações.
- Atualizar as guardas destas transições com as condições equivalentes às locações originais.

FimPara

Este procedimento possibilita a linearização por partes no modelo do sistema para a variação nominal dos componentes do circuito. Cada parte desta linearização é representada por uma locação no autômato híbrido. A Figura 3.42 ilustra a linearização em três segmentos do componente L de um determinado circuito, correspondentes às locações L_1 , L_2 , e L_3 do autômato híbrido que modela este circuito.

Componente L do circuito

O procedimento de modelagem, apresentado anteriormente, foi também utilizado para realizar estudos sobre o conversor Buck acima com variação no valor nominal do componente L , provocada pela saturação magnética da indutância.

Esta variação no parâmetro L do circuito representa uma não-linearidade do componente determinada por uma condição determinada pela corrente na indutância, i_L . Na simulação realizada, foi considerada uma mudança no valor de L , quando a corrente i_L atingir um valor de

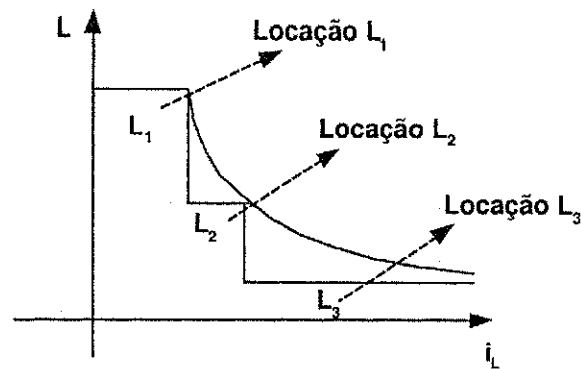


Figura 3.42: Exemplo de linearização por partes do componente L de um dado circuito.

limiar I_{ref} , assumindo um novo valor L_1 . Esta não-linearidade é representada por descontinuidades modeladas através de *saltos autônomos*, fazendo com que haja uma alteração no modelo do autômato híbrido representando o conversor.

As configurações e os modos de operação do conversor são as mesmas apresentadas anteriormente na Seção 3.4, porém foi necessário acrescentar 2 localções ao autômato híbrido original, representando os saltos autônomos provocados pela variação no parâmetro do componente L . As condições de fluxo das localções são alteradas para o novo valor do componente L , sendo substituído por L_1 . O conjunto de condições invariantes (H) e o conjunto de equações diferenciais (F) representando as condições de fluxo para todos os modos de operação são:

Loc1: Condição Invariante - $v_C > 0$ & $i_L = 0$

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR} \end{bmatrix} X \quad (3.4)$$

Loc2: Condição Invariante - $i_L < I_{ref}$ & $v_C > 0$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} E \quad (3.5)$$

Loc3: Condição Invariante - $i_L > I_{ref}$ & $v_C > 0$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L_1} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{1}{L_1} \end{bmatrix} E \quad (3.6)$$

Loc4: Condição Invariante - $i_L < I_{ref}$ & $v_C > 0$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X \quad (3.7)$$

Loc5: Condição Invariante - $i_L > I_{ref}$ & $v_C > 0$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{LI} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X \quad (3.8)$$

O modelo do autômato híbrido resultante para representar esta variação paramétrica consiste de cinco modos discretos possíveis, conforme ilustrado no autômato da Figura 3.43.

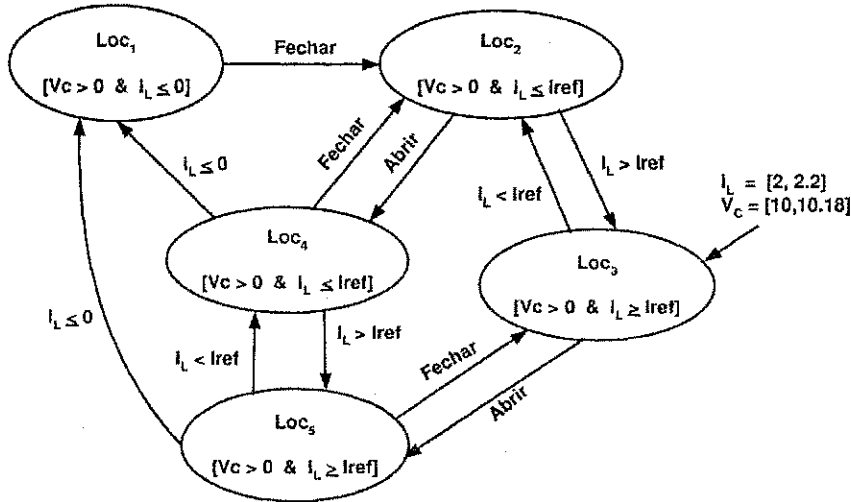


Figura 3.43: Autômato híbrido para o conversor com variação paramétrica em L

As locações do autômato da Figura 3.43 representam os modos descritos, na Tabela 3.6.

Tabela 3.6: Descrição das locações do autômato híbrido do conversor com variação em L

Localização	Descrição
Loc_1	Sw - aberta, D - aberto, L
Loc_2	Sw - fechada, D - aberto, L
Loc_3	Sw - fechada, D - aberto, L_1
Loc_4	Sw - aberta, D - fechado, L
Loc_5	Sw - aberta, D - fechado, L_1

Após a obtenção do modelo do autômato híbridos, foram obtidos os resultados da simulação para o comportamento deste circuito, observando a variação no componente L . O valor da indutância L foi variado para um valor aproximadamente 50 % menor que o valor inicial, quando a corrente alcança um valor maior que 2 A. Este novo modelo e comportamento obtidos tiveram como objetivo modelar, através de autômatos híbridos, não-linearidades de componentes de circuitos ocasionadas por condições determinadas por variáveis de estado do sistema, neste caso, pela corrente i_L .

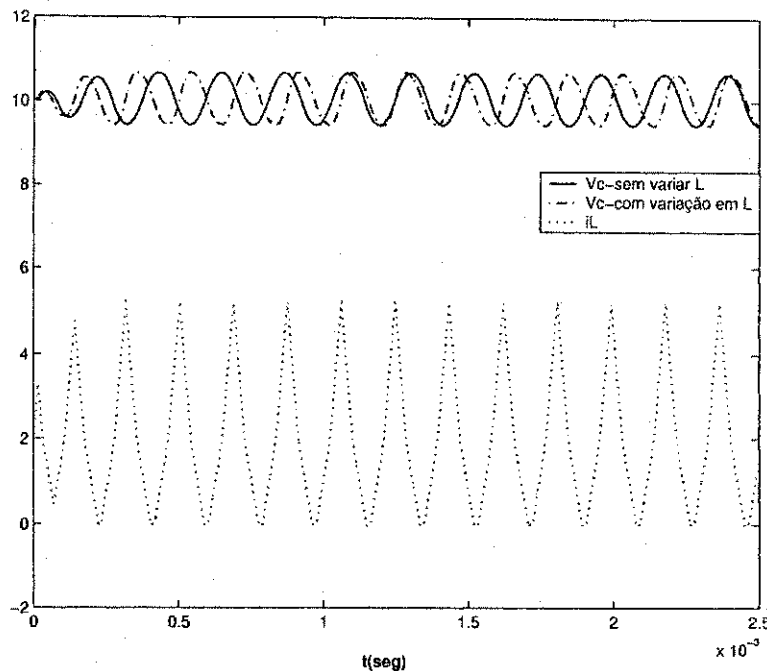


Figura 3.44: Formas de ondas de tensão (v_C) e corrente (i_L) para verificar o comportamento do conversor para variação do parâmetro L .

A variação no valor da indutância L é determinada por uma condição dada pela corrente i_L . Após a alteração do valor da indutância, L , quando a condição de corrente é alcançada, verifica-se que a tensão da carga atinge o regime permanente mais rapidamente que o comportamento com o valor original deste componente e com a diminuição do valor nominal da indutância L , aumenta a frequência de ondulação da tensão v_C , conforme pode ser visto através da diferença entre as tensões em v_C antes e depois da alteração do valor de L , ilustrada na Figura 3.44.

Componente R do circuito

A variação no parâmetro R do circuito representa uma não-linearidade do componente após decorridos um determinado tempo de operação (t_R). Este exemplo é utilizado para ilustrar a influência da variação da carga no comportamento do sistema.

Na simulação realizada, foi considerada uma mudança no valor de R , quando o tempo de operação atingir um valor de limiar t_R , assumindo um novo valor R_1 aproximadamente 50% menor que a resistência inicial. Esta não-linearidade é representada por descontinuidades modeladas através de saltos autônomos, fazendo com que haja uma alteração no modelo do autômato híbrido representando o conversor.

O autômato híbrido original foi modificado por ter sido necessário acrescentar locações que

modelam o comportamento com variação na carga. As condições de fluxo das locações são alteradas para o novo valor da resistência R , sendo substituído por R_1 . O conjunto de condições invariantes (H) e o conjunto de equações diferenciais (F) representando as condições de fluxo para todos os modos de operação são:

Loc1: Condição Invariante - $v_C > 0 \ \& \ i_L = 0 \ \& \ t \leq t_R$

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR} \end{bmatrix} X \quad (3.9)$$

Loc2: Condição Invariante - $i_L > 0 \ \& \ v_C > 0 \ \& \ t \leq t_R$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} E \quad (3.10)$$

Loc3: Condição Invariante - $i_L > 0 \ \& \ v_C > 0 \ \& \ t \leq t_R$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X \quad (3.11)$$

Loc4: Condição Invariante - $i_L > 0 \ \& \ v_C > 0 \ \& \ t > t_R$

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ \frac{1}{C} & -\frac{1}{CR_1} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} E \quad (3.12)$$

Loc5: Condição Invariante - $i_L > 0 \ \& \ v_C > 0 \ \& \ t > t_R$

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR_1} \end{bmatrix} X \quad (3.13)$$

Loc6: Condição Invariante - $i_L \leq 0 \ \& \ v_C > 0 \ \& \ t > t_R$

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR_1} \end{bmatrix} X \quad (3.14)$$

O modelo do autômato híbrido resultante para representar esta variação paramétrica consiste de seis modos discretos possíveis, conforme ilustrado no autômato da Figura 3.45.

As locações do autômato da Figura 3.45 representam os modos descritos na Tabela 3.7.

Foram obtidos os resultados da simulação para o comportamento deste circuito, observando a variação no componente R . O valor da indutância R foi variado para um valor aproximadamente 50 % menor que o valor inicial após decorridos 1.8 mseg de simulação.

Após a alteração do valor da resistência, R , da carga para um decréscimo de 50 %, verifica-se que a tensão máxima da carga em regime permanente é maior que a tensão de saída, quando R

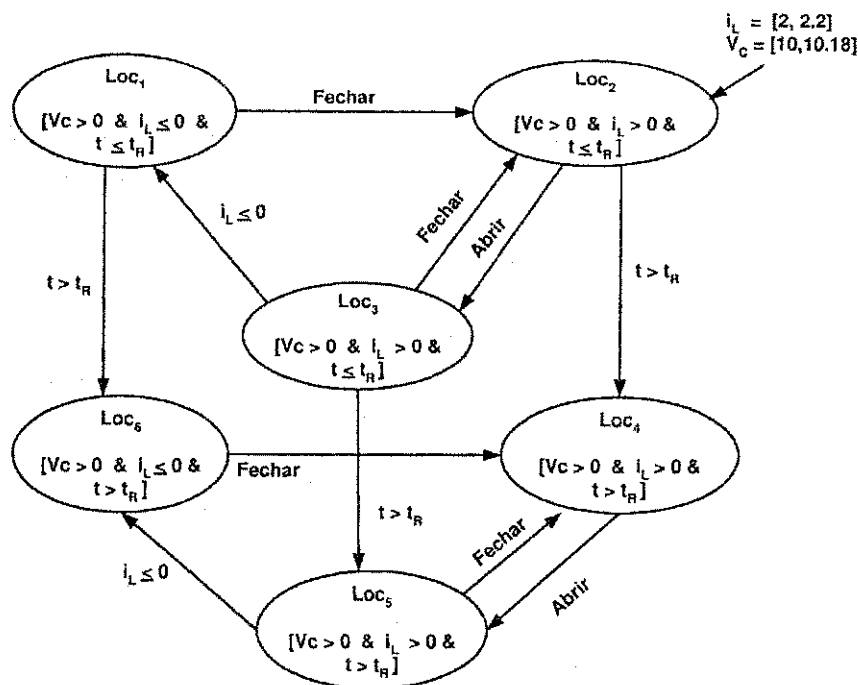


Figura 3.45: Autômato híbrido para o conversor com variação paramétrica em R .

Tabela 3.7: Descrição das locações do autômato híbrido do conversor com variação em L

Localização	Descrição
<i>Loc₁</i>	<i>Sw</i> - aberta, <i>D</i> - aberto, <i>R</i>
<i>Loc₂</i>	<i>Sw</i> - fechada, <i>D</i> - aberto, <i>R</i>
<i>Loc₃</i>	<i>Sw</i> - aberta, <i>D</i> - fechado, <i>R</i>
<i>Loc₄</i>	<i>Sw</i> - fechado, <i>D</i> - aberto, <i>R₁</i>
<i>Loc₅</i>	<i>Sw</i> - aberta, <i>D</i> - fechado, <i>R₁</i>
<i>Loc₆</i>	<i>Sw</i> - aberta, <i>D</i> - aberto, <i>R₁</i>

tem valor original deste componente, uma vez que o armazenamento de energia na indutância L é maior, conforme pode ser visto através da diferença entre as tensões em v_C antes e depois da alteração do valor de R , ilustrada na Figura 3.46.

3.4.6 Circuito conversor com histerese na corrente da carga

O procedimento de modelagem apresentado anteriormente foi também utilizado para realizar estudos sobre o conversor Buck acima com controle via histerese na corrente da carga.

As configurações e os modos de operação do conversor são as mesmas apresentadas anteriormente na Seção 3.4, porém as condições para mudança de estado serão alteradas para considerar

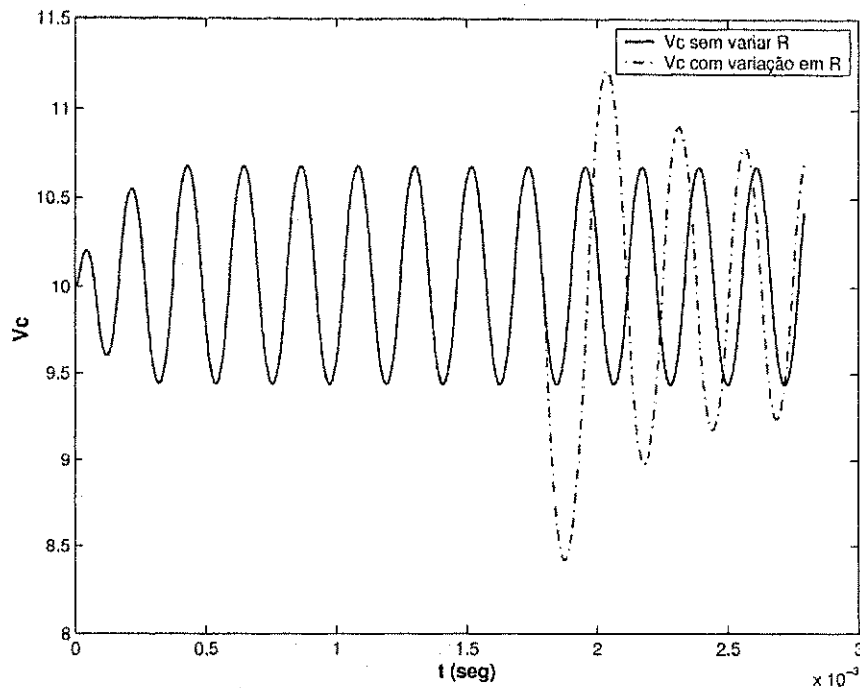


Figura 3.46: Formas de ondas de tensão (v_C) para verificar o comportamento do conversor para $R = 5\Omega$ e para variação do valor de R .

a abertura e fechamento da chave Sw , de acordo com o valor de referência para a corrente na carga.

Para o processo de modelagem e simulação do comportamento deste conversor, as seguintes hipóteses são consideradas: a chave e o diodo são considerados ideais e os elementos passivos do circuito são ideais.

Como condições de operação para este circuito temos:

- Inicialmente, consideramos que a chave Sw está fechada, o diodo D não está conduzindo e temos as seguintes condições iniciais:

- $i_L(0) = 2 \text{ A}$,
- $v_C(0) = 10 \text{ V}$.

- Temos as seguintes especificações dos componentes:

- $R = 5 \Omega$,
- $L = 250 \mu \text{ H}$,
- $C = 100 \mu \text{ F}$,
- $I_{ref} = 1 \text{ A}$,

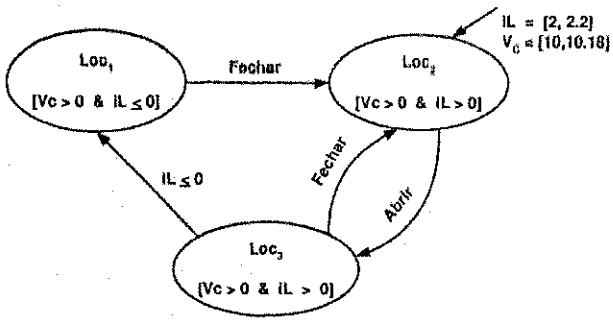


Figura 3.47: Autômato híbrido para o conversor com histerese na corrente da carga.

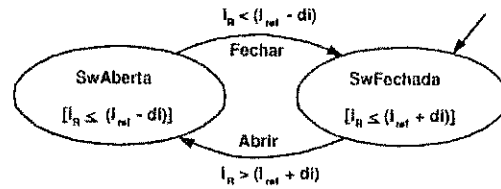


Figura 3.48: Autômato híbrido para o controlador.

- $E = 20 \text{ V}$,
- $di = 0.1 \text{ A}$ representa a banda de histerese.

As condições de fluxo (F) e condições de permanência (H) nas locações do modelo anterior permanecem as mesmas.

A lei de controle por histerese da corrente da carga pode ser expressa da seguinte forma:

$$Sw = \begin{cases} 1 & \text{se } e > di \\ 0 & \text{se } e \leq -di \end{cases}$$

onde $e = I_{ref} - i_R$, 0 significa *Abrir* e 1 significa *Fechar* a chave Sw .

Os modelos resultantes do autômato híbrido do conversor e do controlador estão ilustrados nos autômatos da Figura 3.47 e da Figura 3.48, respectivamente.

As locações do autômato da Figura 3.47 representam os modos descritos na Tabela 3.8.

Tabela 3.8: Descrição das locações do autômato híbrido do conversor com histerese na corrente da carga

Locação	Descrição
Loc_1	Sw - aberta, D - aberto
Loc_2	Sw - fechada, D - aberto
Loc_3	Sw - aberta, D - fechado

Após a obtenção dos modelos dos autômato híbridos, foi obtido, como resultado da simulação para o comportamento deste circuito, as formas de ondas ilustradas, na Figura 3.49, para verificar o comportamento do conversor com controle via histerese na corrente da carga.

O resultado de simulação, aqui apresentado, pode servir de base para avaliar o comportamento do modelo para variação paramétrica de componentes do circuito, assim como foi realizado no modelo do circuito cujo controlador era via histerese na tensão de saída.

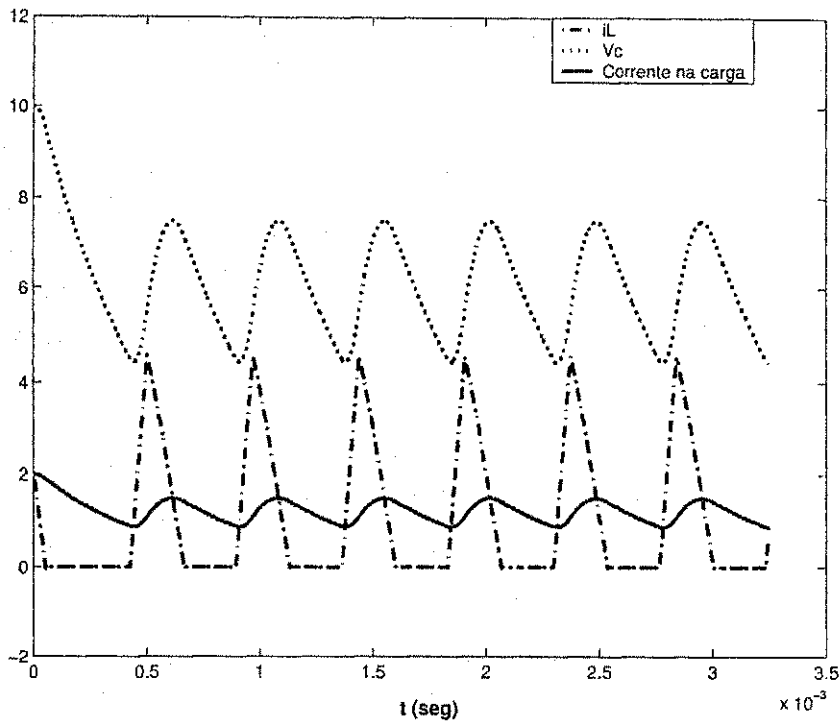


Figura 3.49: Formas de ondas de tensão e corrente na carga e corrente na indutância (i_L).

3.4.7 Processo de refinamento - exemplo com diodo

Nesta Seção, é apresentado um exemplo de processo de refinamento do dispositivo semicondutor *diodo*. Nos casos estudados anteriormente, o diodo era considerado ideal, no exemplo, que será aqui apresentado, este dispositivo é representado como na Figura 3.50.

O processo de refinamento consiste em pesquisar na biblioteca de autômatos híbridos o autômato que representa o refinamento desejado pelo projetista. O refinamento estudado neste exemplo considera as seguintes hipóteses: o diodo sofre queda de tensão na junção PN e queda de tensão devido à resistência dinâmica.

É considerado uma condição de ruptura reversa o diodo, quando a tensão for menor que a

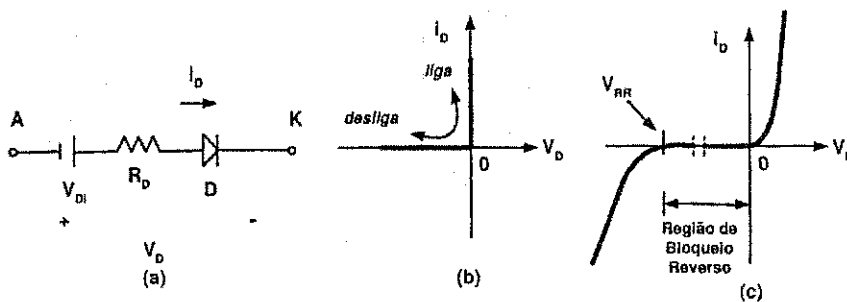


Figura 3.50: Diodo com comportamento mais refinado.

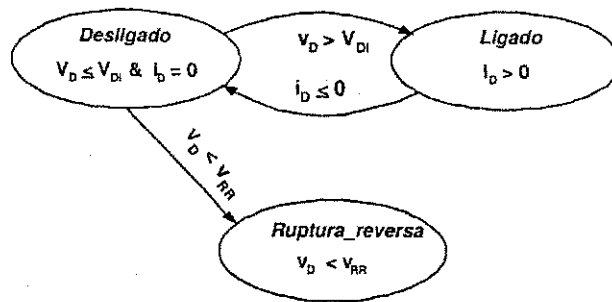


Figura 3.51: Autômato do Diodo com comportamento mais refinado.

taxa de ruptura $V_{RR} = -75$ V.

O autômato híbrido que representa o diodo com comportamento mais refinado, conforme especificado pelas hipótese acima, está ilustrado na Figura 3.51.

Este autômato é pesquisado na biblioteca de autômatos híbridos e trocado pelo autômato anterior, representando comportamento ideal. O comportamento do conversor, representando o refinamento no diodo, após composição com o autômato da chave Sw , está ilustrado na Figura 3.52.

As locações do autômato da Figura 3.52 representam os modos descritos na Tabela 3.9.

Tabela 3.9: Descrição das locações do autômato híbrido do conversor com histerese na corrente da carga

Localção	Descrição
Loc_1	Sw - aberta, D - aberto
Loc_2	Sw - fechada, D - aberto
Loc_3	Sw - aberta, D - fechado
Loc_4	Sw - aberta, D - Ruptura
Loc_5	Sw - fechada, D - Ruptura

3.4.8 Entrada variante no tempo - Aplicação

O procedimento, representado pelo Algoritmo 3, será utilizado, agora, para o conversor Boost, mostrado na Figura 3.53. Este exemplo será utilizado para ilustrar o emprego do método do cálculo dos estados alcançáveis para circuitos com entrada variante no tempo. Aqui, serão mostrados o modelo do autômato híbrido obtido e, na Seção 4.4.1, será mostrado o espaço de estado obtido.

Neste conversor, a corrente na indutância é regulada via controle por histerese. Apresentamos, a seguir, as condições iniciais e os possíveis estados de operação deste circuito para,

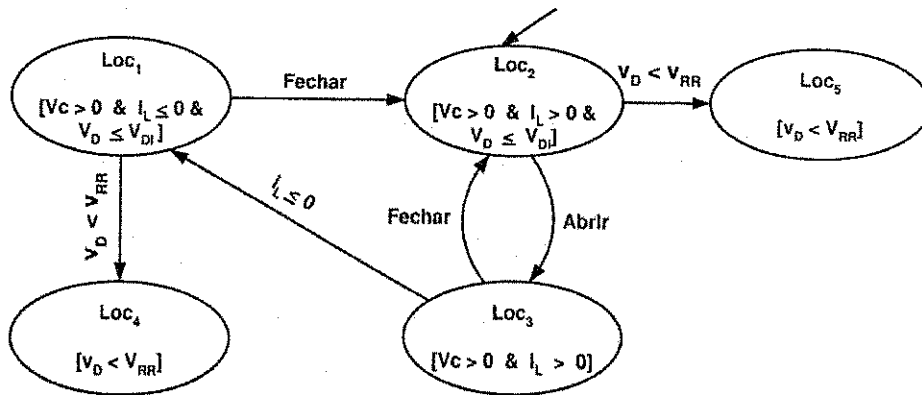


Figura 3.52: Autômato do conversor após refinamento do diodo.

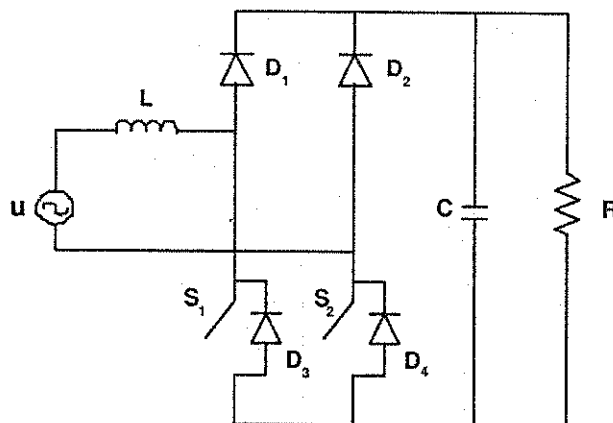


Figura 3.53: Circuito conversor Boost

posteriormente, mostrarmos seu procedimento de modelagem com autômatos híbridos.

Condições de Operação

- Inicialmente, as chaves estão abertas e os diodos não estão conduzindo. Condições iniciais consideradas:

- $i_L(0) = 0 \text{ A}$,

- $v_C(0) = 0 \text{ V}$.

- Temos as seguintes especificações dos componentes:

- $R = 20 \Omega$,

- $L = 2 \text{ m H}$,

- $C = 1 \text{ m F}$,

- $h = 0.5 \text{ A}$ representa a banda de histerese.

A tensão na entrada é variante no tempo, segundo o seguinte comportamento:

$$u = 120\sqrt{2}\text{sen}(120\pi t).$$

A histerese na corrente da indutância é calculada em relação à seguinte referência: $i_t = 15\text{sen}(120\pi t)$.

A Figura 3.54 ilustra os vários modos de operação deste conversor.

- O modo 1 (ver Figura 3.54(a)) ocorre quando a tensão de entrada (u) é positiva e as chaves estão abertas. A corrente flui pelo diodo D_1 , através do capacitor e da carga, e retorna pelo diodo anti-paralelo D_4 .
- O modo 2 (ver Figura 3.54(b)) ocorre quando a tensão de entrada (u) é positiva e as chaves estão fechadas. A corrente de entrada flui pela chave S_1 e volta pelo diodo anti-paralelo D_4 . Ao mesmo tempo, o capacitor descarrega, fornecendo corrente para a carga.
- O modo 3 (ver Figura 3.54(c)) ocorre quando a tensão de entrada (u) é negativa e as chaves estão abertas. A corrente de entrada flui pelo diodo D_2 , pelo capacitor e pela carga, e volta pelo diodo anti-paralelo D_3 .
- O modo 4 (ver Figura 3.54(d)) ocorre quando a tensão de entrada (u) é negativa e as chaves estão fechadas. A corrente de entrada flui pela chave S_2 e volta pelo diodo anti-paralelo D_3 . Ao mesmo tempo, o capacitor descarrega, fornecendo corrente para a carga.

- O modo 5 (ver Figura 3.54(e)) ocorre quando as chaves estão fechadas e os diodos não estão conduzindo. Neste caso, o capacitor descarrega, fornecendo corrente para a carga.

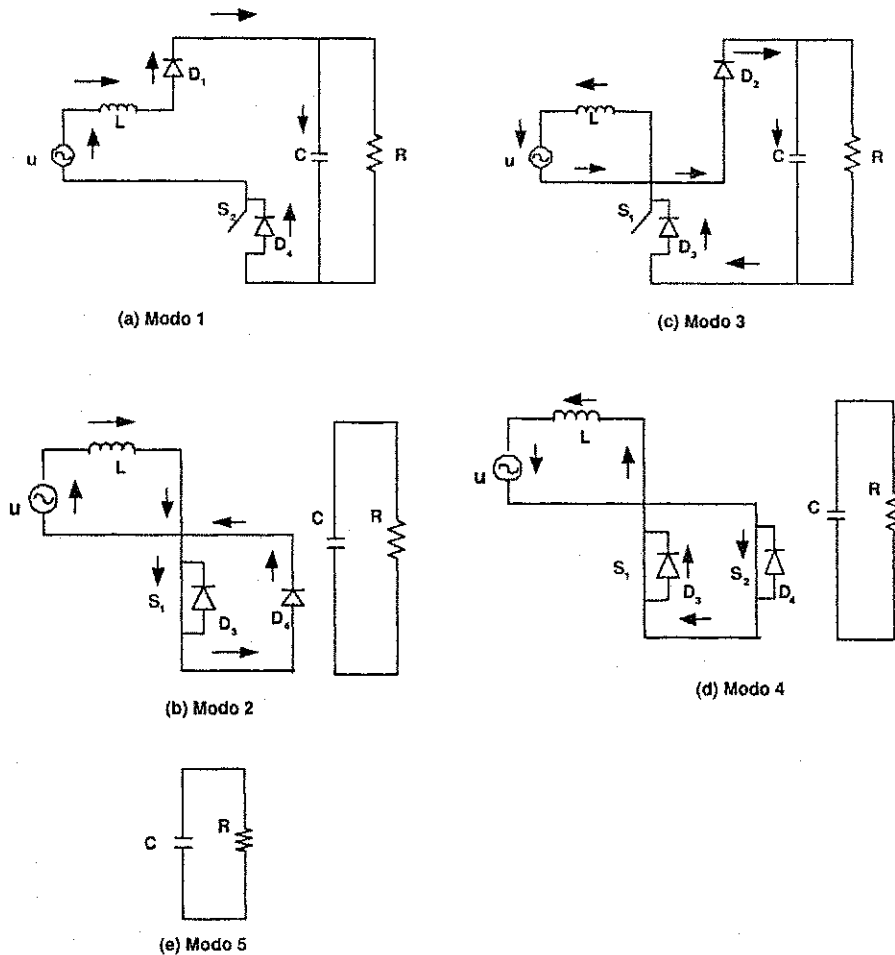


Figura 3.54: Modos de operação do conversor Boost

Modelos dos autômatos híbridos

O conjunto de condições invariantes (H) e o conjunto de equações diferenciais (F) representando as condições de fluxo para todos os modos de operação do conversor e seu controlador são:

Loc1: Condição Invariante - $v_C > 0$ & $i_L > 0$ & S_1/S_2 Abertas

$$\dot{X} = \begin{bmatrix} 0 & -1/L \\ 1/C & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u \quad (3.15)$$

Loc2: Condição Invariante - $i_L > 0$ & $v_C > 0$ & S_1/S_2 Fechadas

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u \quad (3.16)$$

Loc3: Condição Invariante - $i_L \leq 0 \ \& \ v_C > 0 \ \& \ v_C < |u| \ \& \ S_1/S_2$ Abertas

$$\dot{X} = \begin{bmatrix} 0 & -\frac{1}{L} \\ \frac{1}{C} & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u \quad (3.17)$$

Loc4: Condição Invariante - $i_L \leq 0 \ \& \ v_C > 0 \ \& \ v_C < |u| \ \& \ S_1/S_2$ Fechadas

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR} \end{bmatrix} X + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} u \quad (3.18)$$

Loc5: Condição Invariante - $i_L \leq 0 \ \& \ v_C > 0 \ \& \ v_C > |u| \ \& \ S_1/S_2$ Abertas

$$\dot{X} = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{1}{CR} \end{bmatrix} X \quad (3.19)$$

A condição inicial é $i_L = 0 \text{ A}$, $v_C = 0 \text{ V}$, o vetor de estado é definido como $X = [i_L \ v_C]^T$.

O ciclo de chaveamento do conversor começa quando as chaves estão abertas e os diodos D_2 e D_3 não estão conduzindo (Loc1).

A execução destas etapas podem ser mostradas através da exibição dos eventos e transições definidos nos autômato mostrado na Figura 3.55.

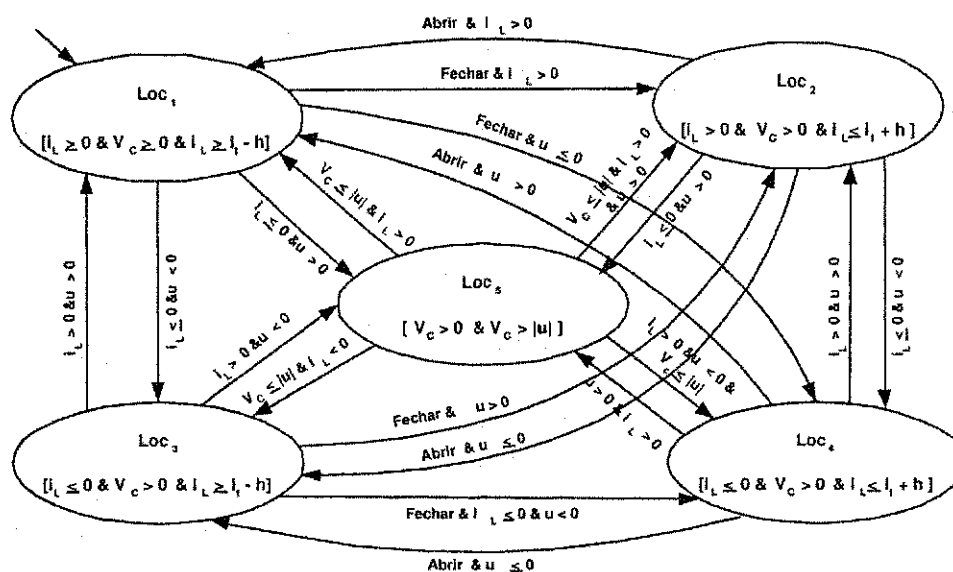


Figura 3.55: Autômato híbrido para o circuito conversor Boost

As locações, no autômato da Figura 3.55, representam os modos descritos, na Tabela 3.10.

Tabela 3.10: Descrição das locações do autômato híbrido do conversor Buck

Locação	Descrição
Loc_1	Chaves (S_1 e S_2) - abertas, D_1 e D_4 - conduzindo
Loc_2	Chaves (S_1 e S_2) - fechadas, D_4 - conduzindo
Loc_3	Chaves (S_1 e S_2) - abertas, D_2 e D_3 - conduzindo
Loc_4	Chaves (S_1 e S_2) - fechadas, D_3 - conduzindo
Loc_5	Chaves (S_1 e S_2) - abertas, Diodos - não conduzindo

A ação de controle consiste na abertura e fechamento das chaves. Os eventos gerados pelo conversor são sincronizados com os eventos do controlador. Neste caso, o controlador tem 2 estados, como mostrados na Tabela 3.11, que são: *Aberta* e *Fechada*, representados por $Q_2 = \{Aberta, Fechada\}$. O autômato do controlador gera dois eventos - *Fechar* e *Abrir* - que são enviados para o autômato do conversor.

Tabela 3.11: Modos de Operação do Controlador

Modo	Sw
Aberta	Aberta
Fechada	Fechada

O controle do conversor é realizado sobre a corrente na indutância (i_L) com o objetivo de mantê-la no patamar definido por i_t . Os comandos para ligar e desligar as chaves são determinados quando i_L ultrapassa o limite inferior ($-h$) ou superior (h) da banda de histerese, respectivamente.

O autômato, representando o controlador, é mostrado na Figura 3.56.

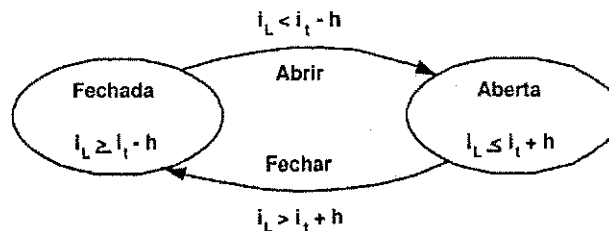


Figura 3.56: Autômato híbrido para o controlador do conversor Buck

O conjunto de condições invariantes (H) para todos os modos de operação do controlador são:

- *Aberta*: Condição invariante: $i_L \geq (i_t - h)$
- *Fechada*: Condição invariante: $i_L \leq (i_t + h)$

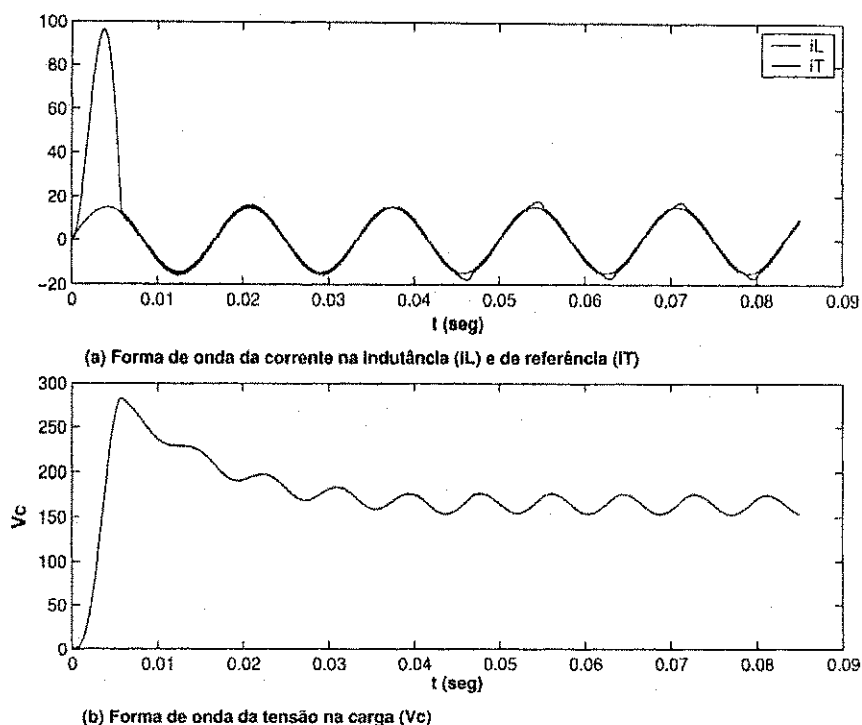


Figura 3.57: Formas de ondas de corrente (i_L) e tensão (v_C)

Resultados da Simulação

Serão apresentados os resultados da simulação do modelo do autômato híbrido representado na Figura 3.55.

As formas de ondas características para corrente e tensão do conversor em operação até que o regime permanente seja alcançado são mostradas, nas Figuras 3.57.

3.5 Conclusões

Este capítulo apresentou um resumo sobre circuitos eletrônicos de potência, com seus componentes principais e regimes de funcionamento. Posteriormente, introduzimos um procedimento de modelagem para obtenção de autômatos híbridos que representam o comportamento de circuitos de potência. Para ilustrar, apresentamos o procedimento proposto, etapa-a-etapa, para obtenção do autômato híbrido a partir de um exemplo de um circuito de eletrônica de potência, representado por um conversor Buck, bem como os autômatos obtidos com seus respectivos resultados de simulações.

Para validar o comportamento obtido por autômatos híbridos, simulamos este mesmo circuito com a ferramenta PSPICE, bastante usual para simulação de circuitos de eletrônica de potência, e apresentamos as formas de ondas para todas as simulações realizadas. O comportamento do

modelo formal obtido é consistente com o comportamento do conversor modelado pela ferramenta PSPICE, validando, assim, o modelo formal escolhido e o procedimento de modelagem desenvolvido neste trabalho.

Foram apresentados, também, os modelos de autômatos híbridos para estudos de variação paramétrica nos componentes L e R do conversor estudado e seus respectivos resultados de simulação. Finalmente, foi apresentado um exemplo de processo de refinamento do componente diodo e o autômato resultante deste processo.

O modelo formal autômato híbrido não será utilizado apenas como mais uma ferramenta de simulação, mas como um primeiro passo para a verificação formal e o desenvolvimento de um procedimento automático para reprojeto do controlador, utilizando métodos formais para circuitos de eletrônica de potência.

Capítulo 4

Verificação de propriedades de circuitos de eletrônica de potência

Foi apresentado, no capítulo anterior, um procedimento sistemático para a obtenção de um modelo de autômato híbrido que representa o comportamento de um circuito de eletrônica de potência. Este autômato representa a especificação formal do circuito e será utilizado em procedimentos de verificação e análise de propriedades, bem como em procedimentos para reprojeto do controlador. O procedimento de verificação que é tratado nesta tese utiliza técnicas de pesquisa no espaço de estado através do cálculo dos estados alcançáveis do sistema. No caso de sistemas híbridos, que possuem espaço de estado infinito, é necessário encontrar uma abstração finita que os represente, para que possam ser utilizados em procedimentos de verificação de propriedades. Assim, é necessário utilizar técnicas de aproximação dos estados alcançáveis para possibilitar a utilização de métodos computáveis e decidíveis, na verificação de propriedades e reprojeto do controlador de circuitos eletrônicos de potência.

Neste capítulo, é apresentado o procedimento de verificação de propriedades para circuitos, utilizando a técnica de aproximação baseada na abordagem apresentada, no Capítulo 2. Além disto, são apresentados os tipos de propriedades de circuitos possíveis de serem verificadas e como estas propriedades podem ser especificadas pelo projetista. A abordagem utilizada foi implementada para um estudo de caso, no qual são apresentados os resultados obtidos para o cálculo do espaço alcançável obtido e análise de propriedades para este exemplo.

4.1 Verificação Formal

Eletrônica de Potência é uma tecnologia utilizada no processamento da energia elétrica, visando obter maior eficiência e qualidade. Os métodos empregados em Eletrônica de Potência baseiam-

se na utilização de dispositivos semicondutores operados em regime de chaveamento para realizar o controle do fluxo de energia e a conversão de formas de onda de tensões e correntes entre fontes e cargas (BOSE, 1992).

Foi observado um grande avanço na tecnologia de eletrônica de potência, durante as duas últimas décadas, e um aumento nas suas áreas de aplicações, tais como industrial, comercial, residencial, militar, etc (LEE, 2001). As inovações nesta área têm feito com que os circuitos de eletrônica de potência tornem-se cada vez mais complexos e, conseqüentemente, suas aplicações requerem garantias de confiabilidade antes da implementação do projeto de um circuito. Durante o projeto de um circuito, o estudo de suas operações e a observação de faltas na busca da confiabilidade é feito através da análise do comportamento, verificando se o mesmo atende às especificações do projetista.

Como dito anteriormente, o método mais utilizado para verificar o comportamento de um circuito é a realização de testes e simulações digitais de seu modelo, operando sob uma dada condição inicial. A utilização de ferramentas de simulação como único recurso de análise do circuito, raramente possibilita identificar todas as potenciais falhas do sistema ou observar seu comportamento para um intervalo de condições iniciais.

Verificação de propriedades é a tarefa de determinar se o sistema está construído de acordo com a especificação dada pelo projetista e, para isto, o projetista pode utilizar técnicas matemáticas para especificação e análise de sistemas. *Verificação formal* consiste em determinar se o sistema preenche ou não um conjunto de requisitos pré-estabelecidos, utilizando uma notação matemática formal.

O processo de verificação formal possibilita a detecção de erros nas fases recentes de desenvolvimento do sistema e previne que os erros se propaguem nas fases posteriores. Dado que a verificação pode ser realizada em todas as fases do ciclo de desenvolvimento do sistema, essa verificação possibilita um gerenciamento contínuo e maior compreensão sobre a qualidade e progresso do desenvolvimento deste sistema. Além disto, pode prover ao usuário um visão incremental do desempenho do sistema com a possibilidade de realização de ajustes o mais recentemente possível. Embora a verificação não possa garantir correteude total, uma vez que o modelo formal do comportamento do sistema pode abstrair informações que seriam necessárias na sua análise, pode contribuir para a melhor compreensão do sistema, revelando inconsistências e incompletudes que não seriam detectadas de outra forma.

No desenvolvimento de um sistema, é muito importante que este seja o mais completo possível e livre de erros. Quando se inicia a fase de projeto de um sistema, geralmente, utiliza-se de uma linguagem informal para enumerar seus requerimentos. O analista faz uma "tradução" deste pedido em uma linguagem mais formal, a qual é chamada de especificação formal. Especificação

é o processo de descrever um sistema e suas propriedades desejadas. *Especificação formal* é uma especificação expressa em uma linguagem cujo vocabulário, sintaxe e semânticas são formalmente definidos e têm base matemática. A especificação formal é utilizada para que as propriedades do sistema sejam descritas de maneira precisa e para que possam ser compreendidas de forma única pelas pessoas envolvidas no projeto do sistema (ROCH, 2000).

A utilização de métodos de especificação formal, para o desenvolvimento de sistemas, apresenta algumas características relevantes (VYATKIN; HANISCH, 2000):

- Permite abstrair a especificação do contexto da implementação;
- Estabelece uma notação precisa, a qual pode ser facilmente entendida por qualquer pessoa;
- Permite investigar possíveis mudanças que um sistema pode sofrer de um modo muito mais rápido e barato;
- Permite a realização de provas sobre a correção da implementação do sistema;
- Permite a realização de provas sobre as propriedades do sistema.

Em resumo, a especificação formal de um sistema é a descrição do que o sistema deve fazer, sem dizer como será feito. É importante destacar, no entanto, que a exatidão do processo de verificação formal depende da exatidão com a qual o sistema é modelado e especificado, ou seja, da complexidade tanto do modelo do sistema quanto da sua especificação formal.

A Figura 4.1 ilustra um processo de desenvolvimento de um sistema de eletrônica de potência. Pode-se notar a existência de uma etapa de análise e verificação, em contraste com um ambiente, onde à modelagem do sistema seguiriam testes e simulações baseados na especificação dos parâmetros de desempenho do circuito para alguns casos escolhidos. Para o processo de verificação formal, é necessária a formalização da especificação do comportamento desejado e, após a verificação formal do modelo e constatado que o mesmo satisfaz às propriedades especificadas, o circuito pode, então, ser implementado.

Os métodos de verificação formal, normalmente, fazem uso dos seguintes tipos de especificação formal (VYATKIN; HANISHCH, 2001):

- *propriedades de segurança*, descrevendo a presença ou ausência de estados perigosos nas possíveis trajetórias do sistema;
- ausência de estados, nos quais o sistema não pode mais evoluir - denominado *impasse (deadlock)* - caso onde o sistema alcança um estado não terminal do qual não pode mais sair; ou *vivacidade (livelock)*, na qual o sistema fica, indefinidamente, alternando sua operação entre estados não terminais;

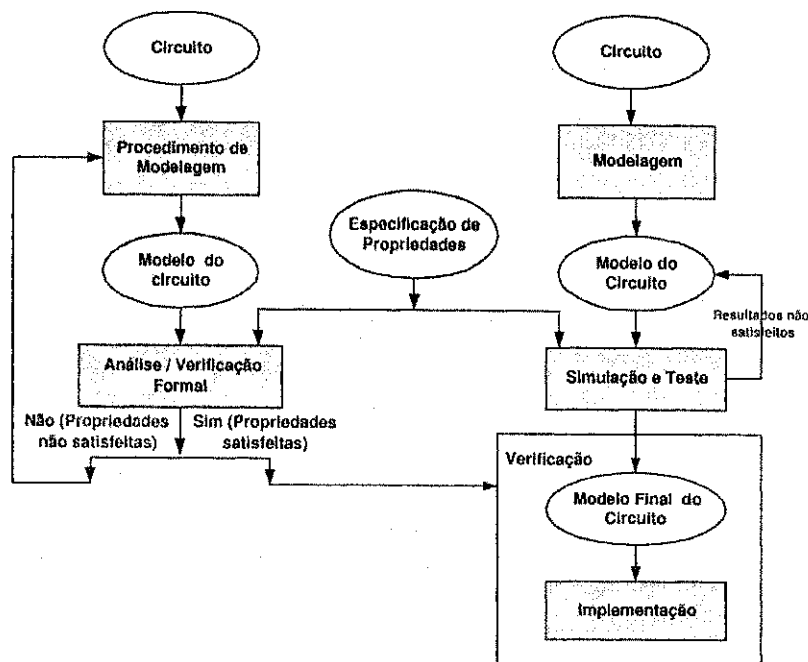


Figura 4.1: Verificação formal no processo de desenvolvimento de um circuito de eletrônica de potência.

- *reatividade*, que consiste em assegurar que uma certa combinação de entradas sempre causa a reação esperada.

4.2 Métodos para Verificação Formal

Os dois principais métodos para a verificação de propriedades de um sistema são *verificação automática (model checking)* e *sistema de prova dedutiva*.

4.2.1 Verificação Automática

Verificação automática é uma técnica de verificação formal que, tendo como base um modelo finito de um sistema, é utilizada para verificar se uma dada propriedade é satisfeita naquele modelo.

A tarefa principal de um algoritmo de *verificação automática* é, recursivamente, calcular o conjunto de estados que satisfazem uma determinada propriedade. Este tipo de técnica exige um *modelo finito* para representação do sistema, ou seja, o sistema a ser verificado representado por um modelo com número finito de possíveis estados alcançáveis e transições entre estes estados. A partir do espaço de estados deste modelo, é realizada uma busca exaustiva para comprovar a validade de uma dada especificação (KOWALEWSKI, 2002).

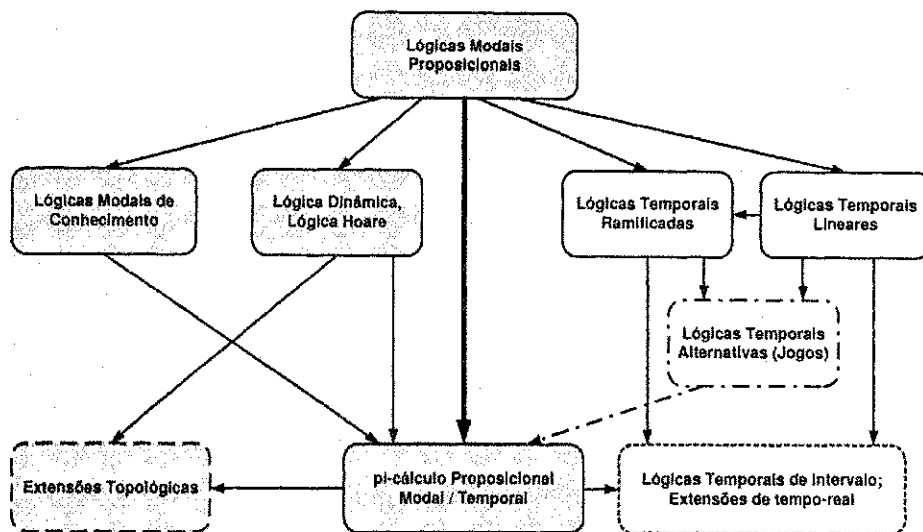


Figura 4.2: Família de lógicas computacionais.

Para sistemas híbridos, isto é realizado através de um procedimento comumente chamado de *verificação simbólica*, uma vez que, sobre um espaço de estado infinito, é necessário uma forma simbólica finita de representar conjuntos de estados e operações sobre eles, ou seja, utilizar uma forma de enumeração explícita para representar o espaço de estado.

Especificação Formal

O processo de verificação formal requer um modelo computacional do sistema e uma forma de especificar as propriedades que correspondem, normalmente, às exigências de comportamento temporal, funcional, desempenho, dentre outras, deste sistema (CLARKE et al., 2001; DAVOREN; NERODE, 2000).

Uma maneira formal para descrever especificações de comportamento de sistemas consiste na utilização de lógicas. A Figura 4.2 mostra um diagrama da família de lógicas utilizadas para especificação formal de sistemas.

As setas sólidas indicam relações de inclusão ou classificação entre lógicas, nas quais tudo que é expresso na primeira pode também ser expresso na segunda. Retângulos com linhas sólidas indicam lógicas utilizadas diretamente sobre modelos de sistemas e retângulos com linhas pontilhadas indicam lógicas que requerem alguma extensão ou adaptação de modelos de sistemas.

Lógicas são utilizadas na especificação formal de sistemas por possuírem sintaxes e semânticas rigorosas para representação de modelos de sistemas e por estabelecerem uma correspondência não ambígua entre o sistema e a sua forma de representação. Dentre os tipos de lógicas existentes, as *lógicas modais* possibilitam expressar noções de necessidade e possibilidade nas afirmações e asserções do sistema em relação ao tempo, servindo de base a muitas outras

lógicas, dentre as quais estão incluídas as lógicas temporais. As lógicas mais utilizadas nas especificações de propriedades, dentre as existentes, são baseadas naquelas que pertencem à sub-família de *lógicas temporais*.

Lógica temporal é uma extensão da lógica convencional que incorpora operadores especiais para relacionar a validade das fórmulas lógicas à evolução do sistema. Com lógica temporal, é possível especificar formalmente comportamentos de sistemas ao longo do tempo. A especificação é feita com proposições lógicas que expressam propriedades e relações do sistema no passado, presente e futuro.

Existem dois grandes grupos de lógica temporal: lógica temporal linear (LTL) e lógica ramificada (CTL). Em LTL, os operadores descrevem propriedades presentes em todas as possíveis trajetórias do sistema; em CTL, os operadores temporais inferem sobre diferentes trajetórias a partir de um dado estado (DAVOREN; NERODE, 2000). As demais lógicas para verificação automática de sistemas são derivadas destas aqui citadas.

Detalhes sobre lógicas e suas construções de fórmulas para verificação automática podem ser encontrados em Davoren e Nerode (2000), Tabuada e Pappas (2003).

4.2.2 Sistemas de Prova Dedutiva

Sistema de Prova Dedutiva ou método dedutivo é o processo de encontrar uma prova de uma propriedade, a partir do uso de axiomas do sistema. O processo de prova consiste de etapas que envolvem axiomas e regras, definições derivadas e lemas intermediários.

Neste método, o projetista não somente tem que especificar o comportamento do sistema e as propriedades em uma lógica apropriada, mas também encontrar uma seqüência de argumentos apropriados para provar tais propriedades. Este processo depende muito mais da criatividade e experiência do projetista do que métodos de verificação algorítmica (HERRMANN; KRUMM, 1999).

Ao contrário de métodos de verificação automática, métodos dedutivos têm um escopo bem mais abrangente. Eles são aplicáveis às fórmulas de lógica de especificação, expressando comparação relacional e propriedades de continuidade. Além do mais, as inferências de um sistema de prova são válidas sobre uma grande variedade de classes de modelos formais para os quais o sistema está incluído. Métodos dedutivos podem ser utilizados não somente para a verificação de propriedades individuais, consideradas uma por vez, mas também para um número maior de propriedades, possibilitando construir uma *teoria dedutiva* ou *base de conhecimento* de fórmulas-verdade num modelo.

Implementações de métodos de verificação dedutiva tendem a ser semi-automatizadas, combinando um certo grau de pesquisa de prova automatizada, construção de prova interativa com o

usuário e verificação de prova automatizada. Porém, a vantagem de utilizar este tipo de método refere-se à possibilidade de lidar diretamente com espaço de estados infinitos, não sendo restrito a sistemas com espaços de pesquisa finitos.

Especificação Formal

Várias lógicas foram desenvolvidas para serem utilizadas em métodos de verificação dedutiva, consistindo de lógicas combinadas com operadores temporais, para prover um único formalismo para descrição de componentes de sistema e especificação de propriedades de sistemas; por exemplo, lógicas temporais de ações TLA+ e cTLA.

cTLA é uma extensão da Lógica Temporal de Ações (TLA), conhecida como TLA composicional. Nesta lógica, os sistemas são modelados por sistemas de transições e uma especificação, representando as propriedades do sistema, descritas por fórmulas em lógica temporal linear. Além disso, cTLA reconhece as extensões de TLA para sistemas de tempo real e sistemas híbridos, introduzindo a noção de processos. Neste caso, uma especificação de propriedade é estruturada em definições modulares de tipos de processos, nos quais os processos podem representar partes de implementações do sistema. Detalhes sobre especificação e verificação de sistemas híbridos baseados em cTLA podem ser encontrados nos trabalhos de Herrmann e Krumm (1999, 1997).

TLA+ é também uma extensão da lógica TLA (LAMPART, 1993) cuja sintaxe possibilita exprimir uma notação modular de grandes fórmulas em TLA. Esta lógica suporta a descrição e verificação de sistemas que são compostos de um conjunto de componentes orientados-à-implementação. Já existem extensões desta lógica para estudos de sistemas tempo real e híbridos (HERRMANN; KRUMM, 1997). Detalhes sobre esta lógica podem ser encontrados no artigo de Herrmann e Krumm (1998).

4.2.3 Verificação automática - lógica ACTL

Nesta tese, foi adotado o método de verificação algorítmica, conhecido como *model checking* (MACMILLAN, 1992), na análise de propriedades de sistemas. Esta técnica foi escolhida por ser possível, automaticamente, realizar a verificação formal de propriedades e prover a possibilidade de gerar contra-exemplos, caso a propriedade não tenha sido satisfeita.

Por ser uma técnica de verificação utilizada com sucesso e de forma bastante sedimentada na área da Ciência da Computação, procuramos empregá-la para análise de circuitos de potência. Para isto, é obtida uma representação de espaço de estado do sistema, utilizando técnicas de aproximação direta, conforme apresentadas no Capítulo 2, e, assim, executar procedimentos de verificação simbólica automática sobre o conjunto alcançável obtido.

Neste trabalho, podem ser verificados os seguintes tipos de propriedades de circuitos de potência: segurança, invariância, alcançabilidade e percurso (*trace*), a serem detalhadas na Seção 4.3.1. As propriedades de invariância e de percurso são especificadas, utilizando-se a lógica ACTL.

A ACTL é uma sub-classe da lógica temporal CTL para especificação e verificação de propriedades. É uma lógica utilizada para especificar a evolução do sistema em termos do valor verdade das proposições atômicas satisfeitas no conjunto de estados alcançáveis do sistema (CLARKE et al., 2001). Uma fórmula ACTL consiste de operadores lógicos usuais, proposições, operadores temporais e quantificadores de caminho.

Os operadores temporais especificam a evolução do sistema, ao longo de um caminho, no espaço alcançável. Existem três operadores temporais que foram utilizados neste trabalho: F, G e U, uma vez que a realização de pesquisa, no espaço de estado, não possibilita tratar de aspectos temporais em suas propriedades.

- O operador F (no futuro) diz que a propriedade é satisfeita em algum estado no futuro, incluindo o estado corrente.
- O operador G (sempre) diz que a propriedade é satisfeita sempre, ou seja, para todos os estados ao longo do caminho.
- O operador U (até) envolve duas propriedades: fUg requer que g seja satisfeita em algum estado no futuro e que f seja satisfeito em todos os estados ao longo do caminho até a ocorrência de g .

Utilizando-se a lógica ACTL, é possível especificar propriedades através de quantificadores de caminho e operadores temporais, como na lógica CTL. Porém, em se tratando de quantificadores de caminho, só é possível utilizar quantificadores universais, ou seja, só é permitido o quantificador A em suas fórmulas. Os operadores temporais devem ser sempre precedidos por um quantificador de caminho.

A Tabela 4.1 resume os operadores temporais e quantificadores possíveis de serem utilizados na lógica ACTL.

Tabela 4.1: Tabela com operadores ACTL

Quantificadores de caminho	Operadores temporais
A - para todos os caminhos	F - No Futuro
	G - Sempre
	U - Até

A sintaxe de uma fórmula ACTL f ou g é dada pela seguinte gramática:

Gramática ACTL: $f \rightarrow ap, \sim ap, f \& g, f \vee g, AFf, AGf, AfUg$, onde ap representa uma proposição atômica, \sim representa o operador lógico unário *não*, \vee e $\&$ representam os operadores lógicos binários *ou* e *e*, respectivamente.

4.3 Análise de Alcançabilidade

Após a obtenção do modelo formal do sistema híbrido e da especificação formal de seu comportamento, realiza-se a verificação de propriedades para analisar se este sistema se comporta conforme esperado.

O procedimento de verificação algorítmico mais importante para sistemas híbridos é a *análise de alcançabilidade*, apresentado anteriormente. Esta técnica consiste em verificar se, para um dado sistema híbrido, um certo estado híbrido (estado discreto e uma região do espaço de estado contínuo) ou um conjunto de estados híbridos que são alcançáveis, a partir do estado híbrido inicial. Este problema é muito importante porque muitos problemas ou propriedades de sistemas podem ser reduzidos a um problema de alcançabilidade deste tipo.

O procedimento de verificação utiliza como entrada o conjunto alcançável e a especificação de propriedade para verificar se as propriedades desejadas são válidas ou não. Vale salientar que as propriedades possíveis de serem verificadas através destes métodos são conhecidas como *propriedades de segurança ou de invariância* (WONG-TOI, 1997; ASARIN; DANG; MALER, 2000), nas quais uma dada propriedade é ou não satisfeita, se sua especificação for ou não satisfeita, no espaço de estado alcançável do sistema.

Uma forma de aplicar a metodologia de verificação algorítmica para sistemas híbridos é utilizar métodos efetivos para caracterizar conjuntos alcançáveis de dinâmicas contínuas, chamados de *métodos de verificação simbólica*. Estes métodos são executados sobre partições do espaço de estado infinito de sistemas híbridos, e a manipulação simbólica é executada sobre descrições finitas dos conjuntos infinitos no espaço de estado n -dimensional. A representação sintática finita de sub-conjuntos do espaço de estado X de autômatos híbridos (sub-conjuntos em R^n) consiste, neste trabalho, de conjuntos de poliedros ortogonais. Estes poliedros ortogonais serão aproximados em poliedros de grade.

Conforme mencionado anteriormente, devido às dificuldades em calcular com exatidão os conjuntos alcançáveis de um autômato híbrido, neste trabalho, são utilizados métodos de aproximação dos estados alcançáveis, através de poliedros ortogonais de grade. O poliedro ortogonal de grade é uma aproximação das trajetórias dos estados híbridos alcançados do sistema, dada uma condição inicial.

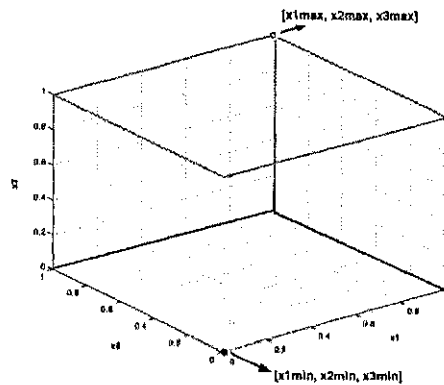


Figura 4.3: Exemplo de representação de um poliedro ortogonal.

Os motivos que levaram à escolha de poliedros ortogonais para representar regiões do espaço de estado contínuo são os seguintes:

- Poliedros ortogonais são objetos geométricos mais fáceis de descrever e manipular do ponto de vista computacional.
- Poliedros ortogonais admitem representações canônicas, baseadas em pares de ordenadas representando seus vértices ou através de restrições booleanas. Assim sendo, é possível realizar implementações eficientes de testes de intersecção, pertinência e equivalência, necessários para os algoritmos de cálculo dos estados alcançáveis e verificação de propriedades.
- Poliedros ortogonais de grade são conjuntos fechados sob a operação de união. Deste modo, o conjunto de estados alcançáveis acumulados após a execução do algoritmo é representado como um poliedro único, facilitando o teste de terminação do algoritmo.

Poliedros ortogonais podem ser especificados através de dois formatos diferentes:

1. Uma lista com os pares de coordenadas de todos seus vértices e
2. Um conjunto de combinações Booleanas de desigualdades lineares para todas suas faces.

Neste trabalho, um poliedro ortogonal será especificado como uma lista de todos seus vértices da seguinte forma: $[x_1^{min} \dots x_n^{min}], [x_1^{max} \dots x_n^{max}]$, onde n é o número de variáveis de estado do sistema. Um exemplo de representação de um poliedro ortogonal para $n = 3$ está ilustrado na Figura 4.3.

Tendo como base os algoritmos de alcançabilidade aproximados para sistemas híbridos, apresentados no Capítulo 2, e a forma de representação simbólica que será utilizada nesta tese, é apresentado o problema de verificação para autômatos híbridos e os algoritmos de verificação abstrato para realização de análise de comportamento para sistemas híbridos.

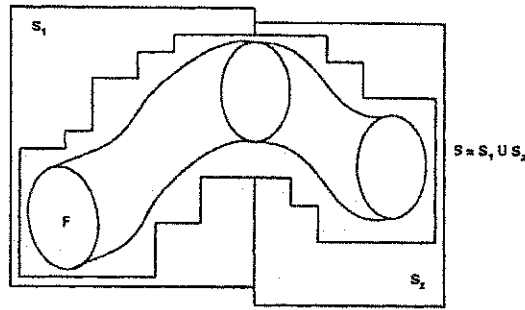


Figura 4.4: Ilustração do problema de verificação de propriedades, dados F e o conjunto S .

4.3.1 Verificação de propriedades - Problema e Algoritmos

Considere o autômato híbrido A , conforme definido na Seção 2.3.1. Seja $F = \{(q, F_q) | q \in Q \wedge F_q \subseteq X\}$ que representa um conjunto de condições iniciais para todas as variáveis de estado admissíveis do sistema. O problema de verificação de propriedades, considerado neste trabalho, trata de propriedades de segurança ou invariância, conforme ilustrado na Figura 4.4 e definido a seguir.

Definição 2 *Problema de verificação de propriedade:* Dado um conjunto $S = \{(q, S_q) | q \in Q \wedge S_q \subseteq X\}$, deseja-se verificar se o autômato híbrido A satisfaz S , ou seja, a propriedade é satisfeita se todas as trajetórias de A permanecem na região definida em S .

A verificação de propriedades tratada, neste trabalho, utiliza procedimentos de análise de alcançabilidade do espaço de estado do sistema híbrido estudado, que podem ser de dois tipos: procedimento de análise de alcançabilidade para frente ou para trás. O método de alcançabilidade para frente consiste em começar a pesquisa no espaço de estados a partir de um conjunto inicial F de estados, calcular seus sucessores até que haja a intersecção deste conjunto com uma dada região final ou até que o conjunto de todos estados alcançáveis seja calculado. O método de alcançabilidade para trás consiste em realizar a pesquisa a partir de uma dada região e calcular seus predecessores até que o conjunto de condições iniciais seja alcançado.

A análise realizada neste trabalho trata dos seguintes tipos de propriedades:

- 1 - **Propriedade de Segurança:** Para verificar este tipo de propriedade, é necessária a especificação do comportamento não desejado, representado por uma região do espaço de estado, considerado como conjunto de *maus* estados ou estados não-seguros. Este conjunto é descrito através de uma restrição, representada por uma conjunção de desigualdades lineares, que é transformada em uma região representada por um poliedro.

Provar que o autômato híbrido A satisfaz a propriedade de segurança é equivalente a provar que A nunca alcança o conjunto $B = \{(q, B_q) | q \in Q \wedge B_q = X \setminus S_q\}$, onde $X \setminus S_q$

denota o complemento de S_q , que representa o conjunto de estados não-seguros. No modelo de autômato híbrido considerado neste trabalho, os conjuntos H_q , $G_{qq'}$, S_q e F_q são representados através de poliedros ortogonais em \mathbb{R}^n , $\forall q, q' \in Q$.

Formato da propriedade: *Restrição*, onde *Restrição* consiste de uma lista de desigualdades lineares que representa uma região do espaço de estado.

Exemplos de propriedades de segurança possíveis de serem especificadas para este trabalho são:

- $i_L > 5$
Representa o comportamento não-desejado para o modelo do sistema. Neste caso, não é desejável que a corrente i_L assuma valores maiores do que 5 A, durante todo ciclo de operação do sistema.
- $v_C < 0$
Neste caso, não é desejável que, em todo espaço alcançável, o comportamento do sistema apresente tensão com valores menores do que 0.
- $v_D < v_{RR} \vee i_D > i_{Lim}$
Esta propriedade representa o seguinte teste de verificação: se a tensão no diodo (v_D) ultrapassou o limite estabelecido pelo dispositivo para suportar tensão reversa (v_{RR}) ou se a corrente no diodo (i_D) ultrapassou o limite suportado pelo dispositivo (i_{Lim}).

Este procedimento de verificação está ilustrado no Algoritmo 4.

Este procedimento pode ser explicado da seguinte forma:

A propriedade a ser verificada é representada através de uma restrição representada por uma conjunção de desigualdades (variável *Restrição* no Algoritmo 4), representando a região de estados não-seguros no espaço de estados. Esta restrição é traduzida para a forma de um poliedro ortogonal, ou seja, uma estrutura com um conjunto de todos os vértices do poliedro que a representa. Isto é realizado pela função $Transf_poliedro(Restrição)$.

A condição inicial de execução do procedimento é representada pela variável F , ou seja, F é o subconjunto inicial do espaço de estados. O conjunto alcançável é inicializado com o conjunto inicial $P^0 := F$. Em cada iteração, o operador sucessor-contínuo, δ_c , é aplicado a P^k , resultando no conjunto de estados alcançáveis pela dinâmica contínua. Em seguida, o operador sucessor-discreto δ_d é aplicado para obter o conjunto de estados alcançáveis pelas transições discretas. Se a intersecção do conjunto calculado P^k com o conjunto de estados

Algoritmo 4 Algoritmo Verificação de Propriedade de Segurança

```

Input (Restrição,  $F$ );
 $B := \text{Transf\_poliedro}(\text{Restrição});$ 
 $P^0 := F$ 
repita
   $k = 0, 1, 2 \dots$ 
  se  $P^k \cap B \neq \emptyset$  então
    retorne Sistema não seguro. Parar.
  fim se
   $P_c := \delta_c(P^k);$ 
   $P_d := \delta_d(P_c);$ 
   $P^{k+1} := P^k \cup P_c \cup P_d;$ 
até  $P^{k+1} = P^k;$ 
retorne Sistema Seguro.

```

não-seguros B é não-vazia, então o algoritmo retorna, como resultado, que o sistema não satisfaz a propriedade especificada, através da mensagem "*Sistema não seguro*", e é encerrado. Se o algoritmo termina e o conjunto não-seguro, B , não é alcançado, então foi provado que o sistema satisfaz a propriedade de segurança, retornando a mensagem "*Sistema seguro*".

Este algoritmo é semelhante ao algoritmo de verificação automática básico implementado em muitas ferramentas de verificação, tais como KRONOS (YOVINE, 1997), para autômatos temporizados, e Hytech (HENZINGER; HO; WONG-TOI, 1997), para autômatos híbridos lineares, porém, nesse caso, as dinâmicas contínuas consideradas são mais genéricas devido ao método de aproximação aqui considerado, já explicado no Capítulo 2, podendo considerar dinâmicas com sinais de entrada incertas.

2 - Propriedades de Invariância: O procedimento para verificar este tipo de propriedade consiste em determinar se uma dada restrição especificada através de fórmulas em ACTL (CHUTINAN; KROGH, 1999) é satisfeita sobre o conjunto alcançável do sistema.

Formato da propriedade: *Operadores*(*Restrição*), onde

- *Operadores* são operadores temporais e quantificadores da lógica ACTL.
- *Restrição* é uma conjunção de desigualdades lineares que representa uma região do espaço de estado alcançado.

Exemplos de propriedades possíveis de serem expressas são:

- $AG(-0.1 \leq i_L \leq 2)$

Esta propriedade especifica que, em todo espaço alcançável do sistema, a condição $(-0.1 \leq i_L \leq 2)$ deve ser sempre satisfeita.

- $AF(i_L > 3)$

Esta propriedade verifica se, a partir da condição inicial F , há algum estado alcançável tal que $(i_L > 3)$.

- $A(i_L < 2)U(v_C > 10)$

Esta propriedade verifica se, a partir da condição inicial F , os estados alcançáveis satisfazem a condição $(i_L < 2)$ até que um estado com $(v_C > 10)$ seja alcançado.

- $AG(v_{Sw} < v_{RR})$

Esta propriedade verifica se, a partir da condição inicial F , a tensão no dispositivo semicondutor, representado por Sw , é sempre menor que a tensão de ruptura reversa em todo estado alcançável, ou seja, não há condições, durante a operação do circuito, que causem uma falha no dispositivo Sw .

- $AF(i_D > I_{DMax})$

Esta propriedade verifica se, para uma dada condição inicial F , existe algum estado alcançável tal que a corrente no diodo D (i_D) ultrapassa o limite suportado por este dispositivo (I_{DMax}).

O procedimento de verificação para este tipo de propriedade está ilustrado no Algoritmo 5.

A propriedade a ser verificada é representada por uma fórmula em ACTL (variável $Prop$ do Algoritmo 5). Os operadores temporais e quantificadores de caminho são identificados na fórmula ($Op.Prop$) e atribuídos à variável Op para ser utilizada, quando do teste para determinar se a propriedade é ou não satisfeita no modelo do sistema. A parte da fórmula contendo a restrição com a conjunção de desigualdades ($Restrição.Prop$) é traduzida para forma de poliedro ortogonal e atribuída à variável Reg , representando a região no espaço alcançável a ser analisada. O conjunto alcançável é inicializado com o conjunto de condições iniciais F ($P^0 := F$).

A cada iteração, são calculados os conjuntos sucessores contínuos e discretos e verificado se o conjunto alcançado obtido satisfaz à propriedade especificada, conforme o operador Op especificado na fórmula.

- Se $Op = AG$ e a intersecção do conjunto alcançável com a região Reg especificada for vazia ou se $Op = \sim AG \vee \sim AF$ e a intersecção do conjunto alcançável com a região

Algoritmo 5 Algoritmo Verificação de propriedade de invariância

```

Input (Prop, F);
Op ← Op.Prop;
Reg ← Transf_poligono(Restrição.Prop);
flag := 0;
P0 := F;
repita
  k = 0, 1, 2...
  caso Op = AG
    se Pk ∩ Reg = ∅ então
      return Propriedade não satisfeita. Parar.
    fim se
  caso Op = ∼ AG ∨ Op = ∼ AF
    se Pk ∩ Reg ≠ ∅ então
      return Propriedade não satisfeita. Parar.
    fim se
  caso Op = AF
    se Pk ∩ Reg ≠ ∅ então
      flag := 1;
      return Propriedade satisfeita. Parar.
    fim se
  fim-caso
  Pc := δc(Pk);
  Pd := δd(Pc);
  Pk+1 := Pk ∪ Pc ∪ Pd;
até Pk+1 = Pk;
se Op = AF & flag = 0 então
  Propriedade não satisfeita;
senão
  Propriedade satisfeita.
fim se

```

especificada na propriedade (Reg) não for vazia, então a propriedade não foi satisfeita, retornando a mensagem "*Propriedade não satisfeita*" e o procedimento é encerrado. A verificação falha, neste caso, porque para $Op = AG$ a condição determinada por Reg deve ser sempre verdadeira em todo espaço alcançável e para $Op = \sim AG \vee \sim AF$, em nenhum conjunto sucessor calculado a condição determinada por Reg deve ser verdadeira.

- Se $Op = AF$ e a intersecção do conjunto alcançável com a região especificada na propriedade (Reg) não for vazia, então a propriedade foi satisfeita e o procedimento é encerrado, pois em qualquer iteração, a partir da condição inicial, se o conjunto alcançável corresponde à condição determinada por Reg , então a propriedade é satisfeita e retorna a mensagem "*Propriedade satisfeita*".
- Se o algoritmo termina, $Op = AF$ e nenhum conjunto alcançável satisfaz a condição determinada por Reg , então a propriedade não foi satisfeita, retornando a mensagem "*Propriedade não satisfeita*". Caso contrário, a propriedade foi satisfeita, retornando a mensagem "*Propriedade satisfeita*".

3 - Propriedade de Alcançabilidade: Neste tipo de propriedade, deseja-se verificar se a partir de uma condição inicial F , uma determinada região do espaço de estado é alcançável. A propriedade a ser verificada (variável $Prop$ do Algoritmo 6) é representada por restrições, como condição Booleana, especificando as regiões inicial e final do espaço alcançável da seguinte forma: [*Restrição_inicial, Restrição_final*].

Exemplos de propriedades possíveis de serem verificadas são:

- $[(-0.1 \leq i_L \leq 2 \ \& \ v_c < 10.1), (1 \leq i_L \leq 3 \ \& \ 10.11 < v_c < 10.5)]$.
Esta propriedade verifica se o estado representado pela restrição $(1 \leq i_L \leq 3 \ \& \ 10.11 < v_c < 10.5)$ é alcançado pelas trajetórias, a partir de $(-0.1 \leq i_L \leq 2 \ \& \ v_c < 10.1)$.
- $[(0 \leq i_L \leq 2 \ \& \ 9 \leq v_c < 10.1)], (1 \leq i_L \leq 3 \ \& \ v_{Sw} < v_{RR})]$.
Esta propriedade verifica se, a partir da região dada pela restrição $(0 \leq i_L \leq 2 \ \& \ 9 \leq v_c < 10.1)$, a região representando uma condição de ruptura reversa na tensão sobre a chave Sw é alcançada, $(1 \leq i_L \leq 3 \ \& \ v_{Sw} < v_{RR})$.

O procedimento de verificação para este tipo de propriedade está ilustrado no Algoritmo 6.

Algoritmo 6 Algoritmo Verificação da Propriedade de Alcançabilidade

```

Input (Prop, F);
Reg := Transf_poliedro(Fim.Prop);
F := Transf_poliedro(Ini.Prop);
P0 := F;
repita
  k = 0, 1, 2 ...
  se Pk ∩ Reg ≠ ∅ então
    return Propriedade satisfeita. Parar.
  fim se
  Pc := δc(Pk);
  Pd := δd(Pc);
  Pk+1 := Pk ∪ Pc ∪ Pd;
até Pk+1 = Pk;
return Propriedade não satisfeita.

```

A variável *Prop*, cujo conteúdo é a propriedade a ser verificada, é decomposta em duas regiões do espaço de estado, representando a condição inicial (*F*) e a região final a ser alcançada (*Reg*).

Este algoritmo é semelhante ao Algoritmo 4, que realiza a verificação de propriedades de segurança, excetuando o tipo de teste que determina se a propriedade é ou não satisfeita. A cada iteração, o conjunto alcançável é calculado através da aplicação dos operadores sucessores contínuo e discreto, tendo como condição inicial o conjunto *F*. Se a intersecção do conjunto calculado com a região final especificada (*Reg*) for não-vazia, então o algoritmo retorna como resultado que a propriedade foi satisfeita, através da mensagem *Propriedade satisfeita* e é encerrado. Se o algoritmo termina e a região *Reg* não foi alcançada, então a propriedade não foi satisfeita e o algoritmo retorna como mensagem *Propriedade não satisfeita*.

4 - Propriedade de percurso (*trace*): Além dos tipos de propriedades acima descritas, este trabalho possibilita verificar propriedades que tratam dos estados discretos ou locações do autômato híbrido modelado. O procedimento, aqui introduzido para realizar este tipo de verificação, estende os tipos de propriedades tratadas na abordagem direta para sistemas híbridos, possibilitando tratar, além de regiões do espaço alcançável, os estados discretos alcançados durante a operação do sistema.

A sintaxe de uma fórmula *Prop* para verificar este tipo de propriedade é dada pela seguinte

gramática:

- $Prop \rightarrow ap, \sim ap, AFf, AGf, AfUg$, onde
 - ap representa uma proposição atômica.
 - \sim representa o operador lógico unário *não*.
 - As fórmulas com os operadores AF , AG e AU são fórmulas na lógica ACTL, cuja semântica já foi explicada anteriormente.
 - f e g são listas de desigualdades lineares, representando regiões do espaço de estado ou representam listas de locações do autômato da seguinte forma: $Loc = [n_1, \dots, n_m]$, onde m é o número de locações e $n_i, i = 1, \dots, m$, é a identificação das locações.
- $Prop \rightarrow Loc = [n_1, \dots, n_m]$ ou $Loc = n \ \& \ Restrição$, onde
 - m é o número de locações e $n_i, i = 1, \dots, m$ é a identificação das locações;
 - $Restrição$ é uma lista de desigualdades lineares.

Este tipo de propriedade possibilita verificar se um determinado estado discreto (locação) foi alcançado, se uma sequência de locações foi executada, se uma determinada restrição é satisfeita sempre que uma locação é alcançada, se um estado discreto, representando uma condição de ruptura reversa modelada, foi alcançado, constituindo, assim, a ocorrência de uma falha no dispositivo semicondutor do circuito, dentre outras propriedades.

Exemplos destas propriedades podem ser:

- $AGLoc = 2$
Verifica se a locação 2 do autômato é o único estado discreto alcançável em todo espaço de estado calculado, ou seja, a operação do sistema permanece sempre no estado 2 do autômato.
- $AFLoc = Rup$
 Rup é a locação no modelo que representa um estado de ruptura reversa no dispositivo. Neste caso, verifica se, a partir da condição inicial, a locação Rup do autômato é inevitavelmente alcançada, representando, assim, a ocorrência de falha no dispositivo. Como exemplo, Rup pode representar a locação 4 da Figura 3.52. Se este estado for alcançado, é constatada a falha no diodo D do conversor modelado.
- $Loc = 2 \ \& \ i_L > 0$
Verifica se, na locação 2 do autômato híbrido, os valores da corrente são sempre positivos.

- $Loc = [0, 1, 2]$

Esta propriedade verifica, se o ciclo de operação, representado pela seqüência de estados do autômato ($\{0, 1, 2\}$), é sempre executada. Este tipo de especificação possibilita verificar a corretude do ciclo de operação do circuito modelado.

O procedimento para a verificação deste tipo de propriedade está ilustrada no Algoritmo 7.

A propriedade a ser verificada é representada pela variável $Prop$ do Algoritmo 7. Os operadores temporais e quantificadores de caminho são identificados na fórmula ($Op.Prop$) e atribuídos à variável Op , a ser utilizada nos testes para verificar se a propriedade é ou não satisfeita. É identificada, na propriedade especificada, a referência aos estados discretos ($Loc.Prop$) e atribuídas à variável Loc . A parte da propriedade, contendo as desigualdades lineares, representando a restrição ($Restrição.Prop$), é traduzida em um poliedro e atribuída à variável Reg do algoritmo. A locação inicial do autômato é atribuída à variável $Localc$ que será atualizada sempre que um novo estado discreto for alcançado e a região inicial é representada pelo conjunto F .

A cada iteração, são calculados os conjuntos sucessores contínuos e discretos; caso haja mudança de estado discreto, a variável $Localc$ será atualizada para o novo estado discreto alcançado. A propriedade especificada é satisfeita da seguinte forma:

- Se $Op = AG$ e a locação alcançada for diferente da locação especificada na propriedade ou se $Op = \sim AG \vee \sim AF$ e a locação alcançada for igual à locação especificada na propriedade, então a propriedade não foi satisfeita, retornando a mensagem "*Propriedade não satisfeita*" e o procedimento é encerrado. A verificação falha, neste caso, porque para $Op = AG$ todas as locações alcançadas devem ser sempre igual a Loc e para $Op = \sim AG \vee \sim AF$, em nenhum conjunto calculado, a locação alcançada deve ser igual a Loc .
- Se $Op = AF$ e a locação alcançada for igual à locação especificada na propriedade (Loc), então a propriedade foi satisfeita e o procedimento termina, pois em qualquer iteração, a partir da condição inicial, se a locação alcançada corresponde à locação determinada por Loc , então a propriedade é satisfeita e retorna a mensagem "*Propriedade satisfeita*".
- Se não foi especificado operadores em ACTL e as regiões da locação Loc especificada não correspondem à condição determinada por Reg , então a propriedade não

Algoritmo 7 Algoritmo Verificação de Percurso *Trace*

```

Input (Prop, F);
Loc ← Loc.Prop; % Locação especificada na propriedade
Op ← Op.Prop; % Operador lógico
Reg ← Transf_poliedro( Restrição.Prop); % Região do espaço de estado
Localc := q; flag := 0; i = 1; n = tamanho(Loc); % Inicializações
P0 := F;
repita
  k = 0, 1, 2...
  se  $\sim \text{empty}(Op)$  então
    caso Op = AG
      se Localc ≠ Loc então
        return Propriedade não satisfeita. Parar.
      fim se
      caso Op =  $\sim AG \vee Op = \sim AF$ 
        se Localc = Loc então
          return Propriedade não satisfeita. Parar.
        fim se
        caso Op = AF
          se Localc = Loc então
            flag := 1;
            return Propriedade satisfeita. Parar.
          fim se
        fim-caso
      senão
        se Localc = Loc & Pk ∩ Reg ≠ ∅ então
          return Propriedade não satisfeita. Parar.
        fim se
        se (n > 1) ∧ (Localc ≠ Loc(i)) então
          Propriedade não satisfeita. Parar.
        fim se
        i = i + 1;
        Se (i > n) então i = 1; fim se;
      fim se
      Pc := δc(Pk);
      Pd := δd(Pc);
      Se Pc ∩ Hq = ∅ então Localc := q; fim se; % Mudança de locação
      Pk+1 := Pk ∪ Pc ∪ Pd;
    até Pk+1 = Pk;
  se Op = AF & flag = 0 então
    Propriedade não satisfeita;
  senão
    Propriedade satisfeita.
  fim se

```

é satisfeita e retorna a mensagem "*Propriedade não satisfeita*" e o procedimento é encerrado.

- Se o algoritmo termina, $Op = AF$ e nenhum conjunto alcançável corresponde à locação determinada por Loc , então a propriedade não foi satisfeita, retornando a mensagem "*Propriedade não satisfeita*". Caso contrário, a propriedade foi satisfeita, retornando a mensagem "*Propriedade satisfeita*".

4.3.2 Algoritmo de Verificação para Trás

Os algoritmos de verificação, vistos anteriormente, realizam o cálculo do espaço alcançável para frente, durante as etapas de verificação. Estes algoritmos podem ser adaptados para a análise de alcançabilidade para trás. Neste caso, os operadores Δ_c e Δ_d são os operadores predecessores contínuos e predecessores discretos, respectivamente, super-aproximados.

Para calcular $\Delta_c(q, F)$, utiliza-se o algoritmo para $\delta_c(q, F)$ com a dinâmica reversa, ou seja, f_q é trocado por $-f_q$. O operador predecessor discreto é calculado da seguinte forma: $\Delta_{q'q}(q, F) = (q', R_{q'q}^{-1}(F) \cap G_{q'q} \cap H_{q'})$, onde $R_{q'q}^{-1} : 2^X \rightarrow 2^X$ é a inversa de $R_{q'q}$, definida como $R_{q'q}^{-1}(X) = \{x' | \exists x \in X, x = R_{q'q}(x')\}$.

A partir de $\Delta_{q'q}(q, F)$, pode-se calcular a aproximação de sucessores discretos como

$$\Delta_d(q, F) = \bigcup_{q' \in Q} \Delta_{q'q}(q, F)$$

O Algoritmo 8 ilustra o procedimento de verificação para trás para o caso de propriedades de segurança. Este algoritmo calcula o conjunto alcançável para trás do conjunto de maus estados especificados, \mathcal{B} , e verifica se o mesmo intercepta o conjunto inicial F . Os demais algoritmos para os outros tipos de propriedades, apresentados na Seção 4.3.1, são adaptados para a utilização do operador predecessor Δ . Neste caso, o conjunto inicial (P^0) é inicializado com as regiões especificadas em cada propriedade (Reg), os predecessores contínuos e discretos são calculados, em cada iteração dos procedimentos, até que a região especificada pela condição inicial do sistema (F) seja alcançada ou o procedimento seja encerrado, quando da satisfação ou não da propriedade.

4.4 Estudo de Caso

O circuito conversor Buck, cujo procedimento de modelagem está ilustrado no Capítulo 3 e mostrado na Figura 3.27, foi adotado como exemplo de aplicação para o cálculo do espaço alcançável e dos procedimentos de verificação de propriedades.

Algoritmo 8 Algoritmo de Verificação para Trás

```

Input (Restrição);
 $B := \text{Transf\_poliedro}(\text{Restrição});$ 
 $P^0 := B;$ 
repita
   $k = 0, 1, 2 \dots$ 
  se ( $P^k \cap F \neq \emptyset$ ) então
    retorne Sistema não seguro. Parar.
  fim se
   $P_c := \Delta_c(P^k);$ 
   $P_d := \Delta_d(P_c);$ 
   $P^{k+1} := P^k \cup P_c \cup P_d;$ 
até  $P^{k+1} = P^k$ 
retorne Sistema seguro.

```

Apesar de sua simplicidade, este exemplo foi utilizado por possibilitar a implementação completa de todos os métodos utilizados neste trabalho, incluindo o reprojeto do controlador, através dos recursos de programação do MATLAB (HANSELMAN; LITTLEFIELD, 1997), permitindo a visualização completa de todos os resultados obtidos. Outro motivo para utilizar este exemplo foi a inexistência de ferramentas que tratem sistemas de ordem elevada e implementem procedimentos para a síntese do controlador. Ademais, é evidente a necessidade de ferramentas prontas para estudar sistemas híbridos, tratando dinâmicas mais genéricas e possibilitando tanto a análise de propriedades quanto ao reprojeto do controlador de forma sistemática.

Vale salientar que a implementação, ilustrada para um exemplo simples, valida os métodos aqui apresentados, podendo, portanto, serem estendidos e aplicados a exemplos de circuitos mais complexos e de dimensões mais elevadas, demonstrando sua abrangência e generalidade.

4.4.1 Espaço alcançável

Para a realização de estudos nos modelos de circuitos de potência, seja verificação de propriedades ou reprojeto do controlador, é necessário a obtenção e cálculo dos conjuntos alcançáveis destes modelos, dado um conjunto de condições iniciais.

Os métodos de alcançabilidade e de aproximação, apresentados no Capítulo 2, foram implementados para o exemplo do conversor Buck, modelado no Capítulo 3, considerando seus diferentes casos de operação. Além dos casos modelados, foi implementado o método de obtenção dos estados alcançáveis, considerando entrada incerta através da fonte de tensão do conversor.

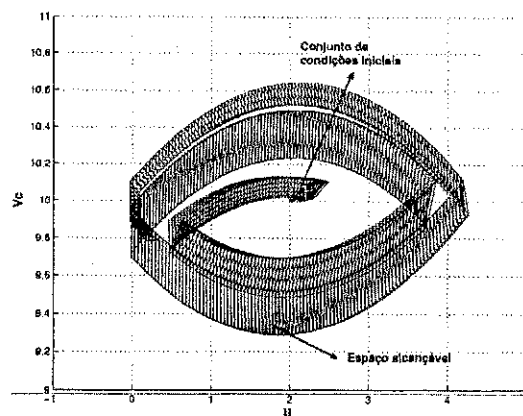


Figura 4.5: Conjunto de estados alcançáveis do circuito controlado via histerese na tensão de saída.

Os resultados obtidos são apresentados a seguir. A condição inicial para estes casos é a região $F = \begin{bmatrix} 2 & 10 \\ 2.2 & 10.2 \end{bmatrix}$. A consistência destes resultados foi verificada, através da simulação de condições extremas de cada exemplo, cujos resultados e respostas mostraram-se compatíveis com os obtidos pela técnica aqui utilizada e implementada.

Circuito conversor com controle por histerese da tensão de saída

O espaço alcançável para o circuito conversor com controle por histerese da tensão de saída, cujo processo de modelagem foi visto na Seção 3.4, foi calculado e o resultado obtido está ilustrado na Figura 4.5.

Circuito conversor com variação paramétrica na indutância L

O espaço alcançável para o circuito conversor com controle por histerese da tensão de saída com variação paramétrica na indutância L , cujo processo de modelagem e resultado de simulação foi visto na Seção 3.4.5, foi calculado e o resultado obtido está ilustrado na Figura 4.6.

Circuito conversor com variação paramétrica na resistência da carga R

O espaço alcançável para o circuito conversor com controle por histerese da tensão de saída com variação paramétrica na resistência da carga R , cujo processo de modelagem e resultado de simulação foi visto na Seção 3.4.5, foi calculado e o resultado obtido está ilustrado na Figura 4.7.

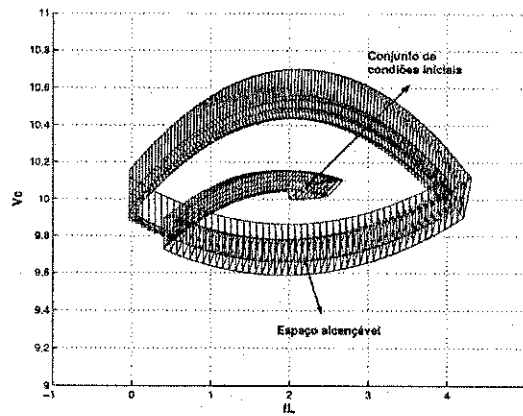


Figura 4.6: Conjunto de estados alcançáveis do circuito com variação no parâmetro L .

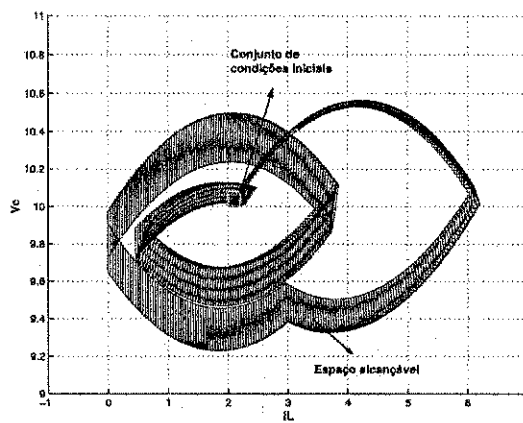


Figura 4.7: Conjunto de estados alcançáveis do circuito com variação na resistência da carga (R).

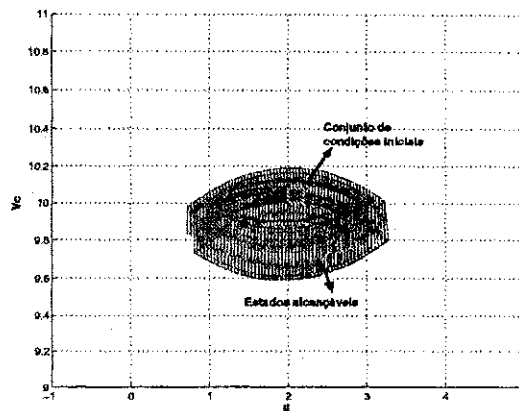


Figura 4.8: Conjunto de estados alcançáveis do circuito considerando entrada incerta na fonte de tensão.

Circuito conversor com incerteza na fonte

O espaço alcançável para o circuito conversor com controle por histerese da tensão de saída com variação da tensão da fonte foi calculado, segundo procedimento apresentado na Seção 2.6 e o resultado obtido está ilustrado na Figura 4.8.

A dinâmica do sistema considerada é $\dot{x} = Ax + Bu$, onde a matriz A em cada estado segue a mesma dinâmica para o circuito com entrada constante. Porém, a entrada do sistema, modelada pela fonte de tensão, é representada pela variação dentro do intervalo $u \in [19, 21]$ V.

Circuito conversor com controle por histerese da corrente da carga

O espaço alcançável para o circuito conversor controlado via histerese na corrente da carga, cujo processo de modelagem foi visto na Seção 3.4, foi calculado e o resultado obtido está ilustrado, na Figura 4.9.

A mudança da variável controlada, mesmo considerando o mesmo conjunto de condições iniciais, alterou bastante o resultado do espaço alcançável obtido, como pode ser observado, na Figura 4.9.

Conversor com entrada variante no tempo

O circuito utilizado para exemplificar a obtenção do espaço alcançável com entrada variante no tempo foi ilustrado e modelado na Seção 3.4.8, cuja estrutura está ilustrado, na Figura 3.53.

Para calcular $\delta(F)$, considerando o sinal de entrada variante no tempo, é necessário transformar o sistema de estado original em um sistema autônomo.

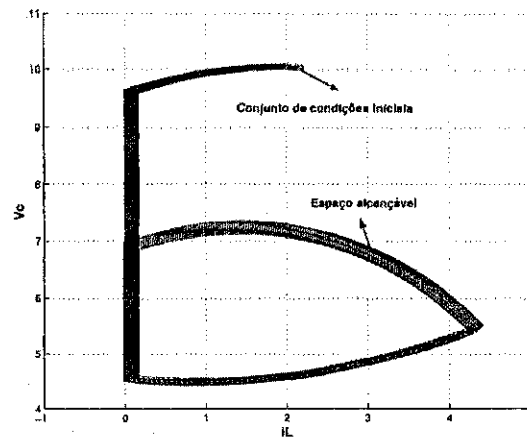


Figura 4.9: Conjunto de estados alcançáveis do circuito com controle via histerese na corrente.

A obtenção do espaço de estado para o conversor Boost é possível, transformando o sistema em um sistema autônomo conforme descrito a seguir.

O novo vetor de estado tem a seguinte forma: $x_e(t) = \begin{bmatrix} x(t) \\ x_a(t) \end{bmatrix}$, onde $x(t)$ é o vetor de estado do sistema e x_a é o vetor de estado do sistema autônomo.

A nova equação de estado, representando o comportamento do sistema, é:

$$\begin{cases} \dot{x}_e(t) = A_e x_e(t) \\ y(t) = C_e x_e(t) \end{cases} \quad (4.1)$$

onde $A_e = \begin{bmatrix} A & BC_a \\ 0 & A_a \end{bmatrix}$ e $C_e = [C \quad DC_a]$.

Considerando o sistema representado pelo conversor Boost apresentado na Seção 3.4.8, a fonte de entrada, $u(t)$, é representada por um sinal senoidal da forma: $u(t) = 120\sqrt{2}\text{sen}(\omega t)$. As matrizes do sistema autônomo do sinal de entrada são as seguintes:

- $A_a = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}$,
- $C_a = [\omega \quad 0]$.

onde $\omega = 120\pi$.

O novo sistema, cuja solução é independente do vetor de entrada, é dado por:

$$\begin{cases} \dot{x}_e(t) = \begin{bmatrix} \dot{x} \\ \dot{x}_a \end{bmatrix} = \begin{bmatrix} A & BC_a \\ 0 & A_a \end{bmatrix} \begin{bmatrix} x \\ x_a \end{bmatrix}, \\ y = [C \quad 0] \begin{bmatrix} x \\ x_a \end{bmatrix} \end{cases} \quad (4.2)$$

onde:

- x é o vetor de estado do sistema original, ou seja, $x(t) = \begin{bmatrix} i_L \\ v_C \end{bmatrix}$,
- x_a é o vetor de estado do sistema autônomo do sinal de entrada,
- A, B, C são matrizes do sistema original.
- A_a e C_a são as matrizes do sistema do sinal de entrada.

A condição inicial é representada pela região $F = \begin{bmatrix} 0 & 0 \\ 0.2 & 0.1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$.

As dinâmicas do sistema autônomo obtido, independente da variação do sinal de entrada, e o conjunto de condições invariantes para cada locação do autômato são as seguintes:

Loc1: Condição Invariante - $v_C > 0$ & $i_L > 0$ & S_1/S_2 Abertas

$$\dot{x}_e = \begin{bmatrix} 0 & -\frac{1}{L} & \frac{\omega}{L} & 0 \\ \frac{1}{C} & -\frac{1}{CR} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} x_e \quad (4.3)$$

Loc2: Condição Invariante - $i_L > 0$ & $v_C > 0$ & S_1/S_2 Fechadas

$$\dot{x}_e = \begin{bmatrix} 0 & 0 & \frac{\omega}{L} & 0 \\ 0 & -\frac{1}{CR} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} x_e \quad (4.4)$$

Loc3: Condição Invariante - $i_L \leq 0$ & $v_C > 0$ & $v_C < |u|$ & S_1/S_2 Abertas

$$\dot{x}_e = \begin{bmatrix} 0 & -\frac{1}{L} & \frac{\omega}{L} & 0 \\ \frac{1}{C} & -\frac{1}{CR} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} x_e \quad (4.5)$$

Loc4: Condição Invariante - $i_L \leq 0$ & $v_C > 0$ & $v_C < |u|$ & S_1/S_2 Fechadas

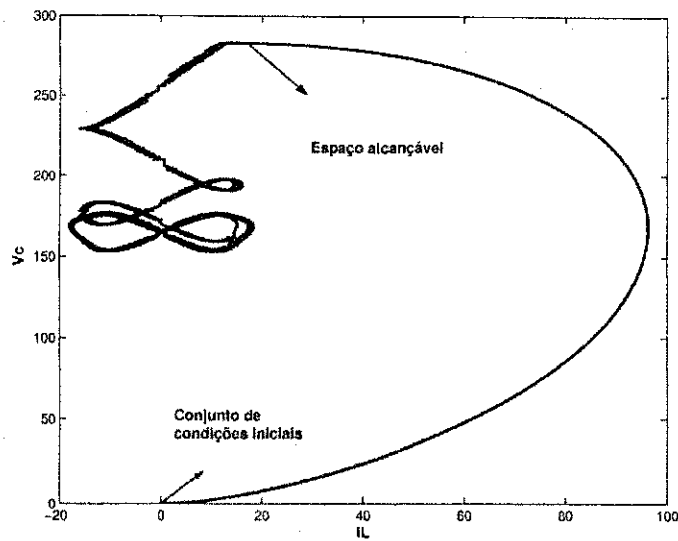


Figura 4.10: Conjunto de estados alcançáveis do circuito com entrada variante no tempo.

$$\dot{x}_e = \begin{bmatrix} 0 & -\frac{1}{L} & \frac{e}{L} & 0 \\ \frac{1}{C} & -\frac{1}{CR} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} x_e \quad (4.6)$$

Loc5: Condição Invariante - $i_L \leq 0 \ \& \ v_C > 0 \ \& \ v_C > |u| \ \& \ S_1/S_2$ Abertas

$$\dot{x}_e = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{CR} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\omega^2 & 0 \end{bmatrix} x_e \quad (4.7)$$

O espaço alcançável para o circuito com entrada variante no tempo, cujo processo de modelagem foi visto na Seção 3.4.8, foi calculado e o resultado obtido está ilustrado na Figura 4.10.

4.4.2 Verificação de propriedades - conversor Buck

O procedimento de verificação de propriedades implementado foi executado para o circuito Buck com controle por histerese da tensão de saída, sobre o espaço alcançável ilustrado na Figura 4.5, com condição inicial representado pela região $F = \begin{bmatrix} 2 & 10 \\ 2.2 & 10.2 \end{bmatrix}$.

1. Propriedades de Segurança:

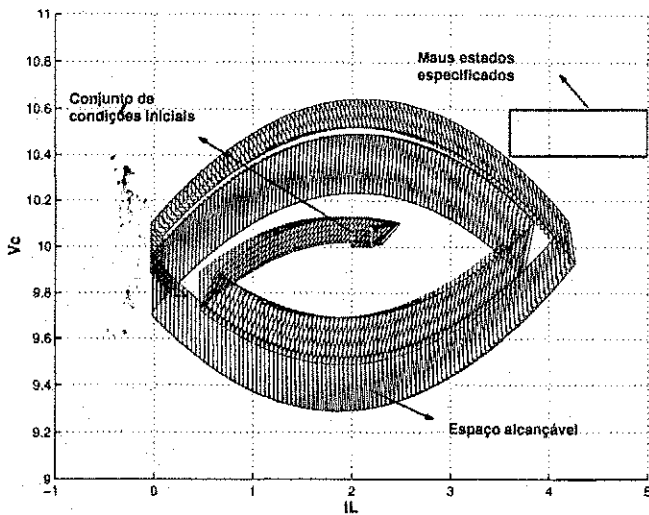


Figura 4.11: Verificação de propriedade de segurança - Sistema seguro

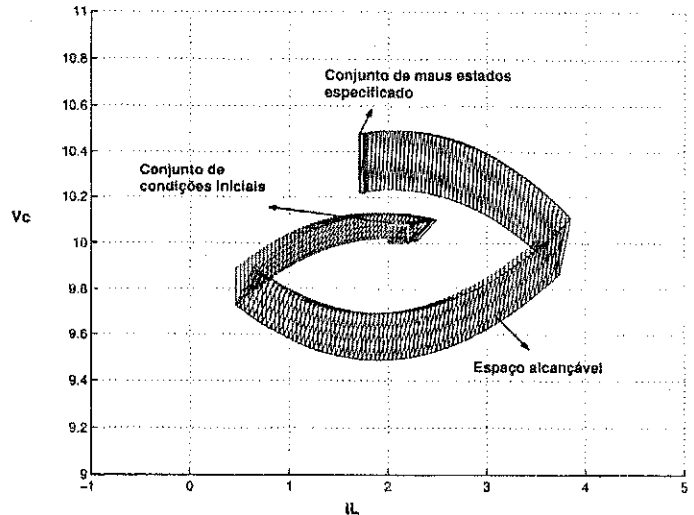


Figura 4.12: Verificação de propriedade de segurança - Sistema não seguro

O procedimento de verificação foi executado para os seguintes exemplos de propriedades de segurança:

(a) $i_L > 3.5 \ \& \ v_C > 10.4$

Esta especificação representa o comportamento não desejado do sistema. Assim, deseja-se verificar se, no conversor modelado, a corrente i_L nunca será superior a 3.5 A e a tensão não será maior que 10.4 V. O sistema foi considerado seguro para esta propriedade. Para ilustrar a validade do procedimento executado, a Figura 4.11 mostra a região não desejada e o espaço de estado do sistema.

(b) $1.75 < i_L < 1.9 \ \& \ 9.75 < v_C < 9.88$

Neste caso, deseja-se verificar se, no conversor modelado, a corrente i_L nunca assumirá valores no intervalo determinado pela restrição $1.75 < i_L < 1.9$ e a tensão não assumirá valores no intervalo determinado pela restrição $9.75 < v_C < 9.88$. O sistema foi considerado inseguro para esta propriedade, conforme ilustrado na Figura 4.12.

(c) $v_{sw} > v_{RR}$

Neste caso, deseja-se verificar se, nos estados de bloqueio da chave S_w , a tensão da chave é menor que a tensão reversa suportada por este dispositivo. O sistema foi considerado seguro, ou seja, não há condições de falha do dispositivo S_w , conforme ilustrado na Figura 4.13.

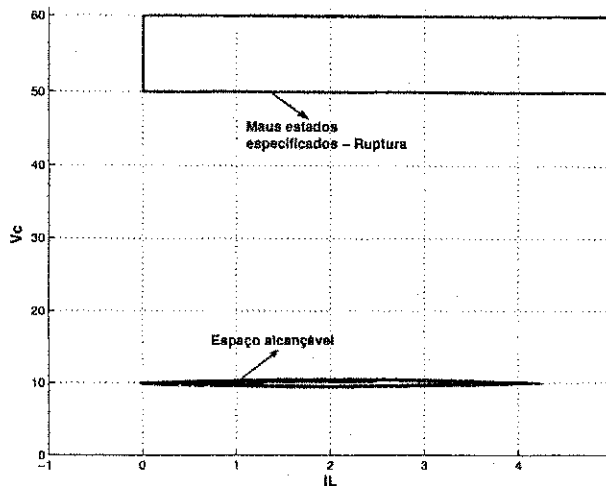


Figura 4.13: Verificação de propriedade de segurança - Sistema seguro

2. Propriedades de Alcançabilidade:

O procedimento de verificação foi executado para os seguintes exemplos de propriedades de alcançabilidade:

(a) $[(2.0 < i_L < 2.2 \ \& \ 10 < v_C < 10.2), (1.78 < i_L < 2 \ \& \ 9.49 < v_C < 9.7)]$

Esta propriedade é utilizada para verificar se o estado representado pela restrição $(1.78 < i_L < 2 \ \& \ 9.49 < v_C < 9.7)$ é alcançado pelas trajetórias, a partir de $(2.0 < i_L < 2.2 \ \& \ 10 < v_C < 10.2)$. As regiões que representam cada uma destas restrições são: Região Inicial é $Ini = \begin{bmatrix} 2.0 & 10.0 \\ 2.2 & 10.2 \end{bmatrix}$ e a região final é

$Final = \begin{bmatrix} 1.78 & 9.49 \\ 2.0 & 9.7 \end{bmatrix}$. Esta propriedade foi satisfeita no processo de verificação de propriedades. Para comprovar o resultado da verificação, a Figura 4.14 mostra a região final alcançável no espaço de estado do sistema.

(b) $[(2.0 < i_L < 2.2 \ \& \ 10 < v_C < 10.2), (-0.9 < i_L < 0.4 \ \& \ 9.0 < v_C < 9.2)]$

Esta propriedade é utilizada para verificar se o estado representado pela restrição $(-0.9 < i_L < 0.4 \ \& \ 9.0 < v_C < 9.2)$ é alcançado pelas trajetórias a partir de $(2.0 < i_L < 2.2 \ \& \ 10 < v_C < 10.2)$. As regiões que representam cada uma destas restrições são: Região Inicial $Ini = \begin{bmatrix} 2.0 & 10.0 \\ 2.2 & 10.2 \end{bmatrix}$ e a região final é $Final =$

$\begin{bmatrix} -0.9 & 9.0 \\ 0.4 & 9.2 \end{bmatrix}$. Esta propriedade não foi satisfeita no processo de verificação de propriedades, conforme pode ser visto na Figura 4.15.

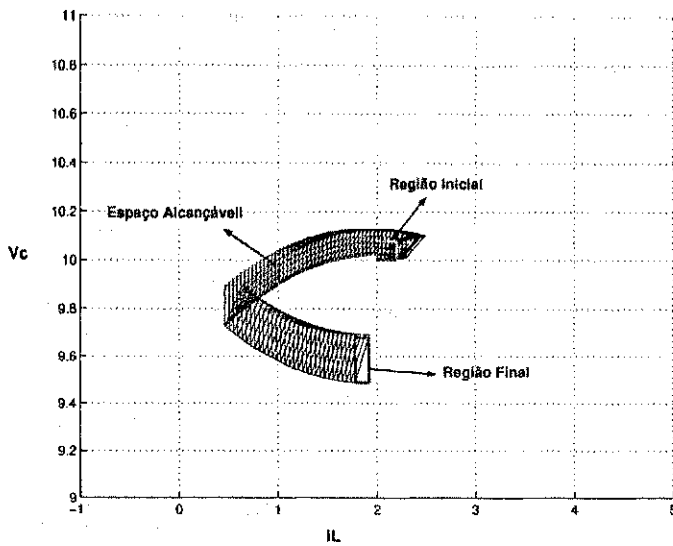


Figura 4.14: Verificação de propriedade de alcançabilidade - Estado alcançável

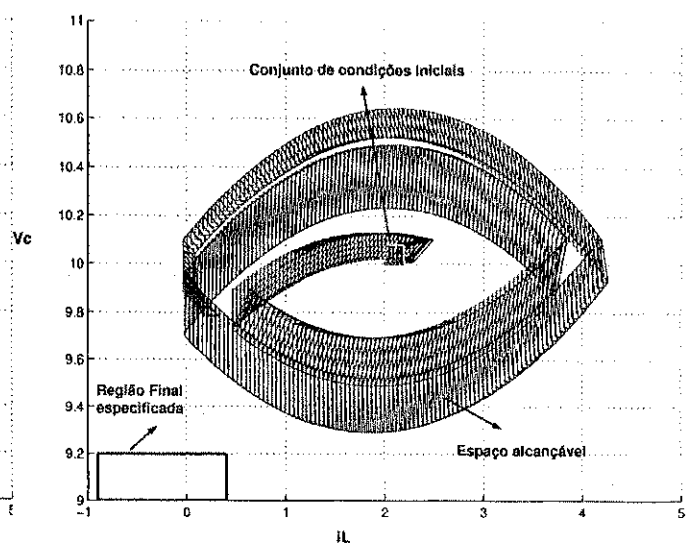


Figura 4.15: Verificação de propriedade de alcançabilidade - Estado não-alcançável

(c) $[(2.0 < i_L < 2.2 \ \& \ 10 < v_C < 10.2), (v_{Sw} < v_{RR})]$

Esta propriedade é utilizada para verificar se o estado representado pela restrição $(v_{Sw} < v_{RR})$ é alcançado pelas trajetórias, a partir de $(2.0 < i_L < 2.2 \ \& \ 10 < v_C < 10.2)$. O valor para a tensão reversa do dispositivo Sw é $v_{RR} = 50 \text{ V}$. As regiões que representam cada uma destas restrições são: Região Inicial $Ini = \begin{bmatrix} 2.0 & 10.0 \\ 2.2 & 10.2 \end{bmatrix}$

e a região final é $Final = \begin{bmatrix} 2.5 & 50.0 \\ 5 & 60 \end{bmatrix}$. Esta propriedade não foi satisfeita no processo de verificação de propriedades, conforme pode ser visto, na Figura 4.16.

3. Propriedades de Invariância:

O procedimento de verificação foi executado para os seguintes exemplos de propriedades de invariância:

(a) $AG(i_L < 3.0)$

Esta especificação é utilizada para verificar se, em todo espaço alcançável do sistema, a corrente i_L é sempre menor que 3 A. Esta propriedade não é satisfeita, conforme ilustrado na Figura 4.17. Pode ser observado um destaque, na Figura, que representa a região no espaço alcançável onde a corrente (i_L) não satisfaz a propriedade de invariância especificada, ou seja, a região que viola a condição especificada e i_L atinge um valor maior que 3.0 A.

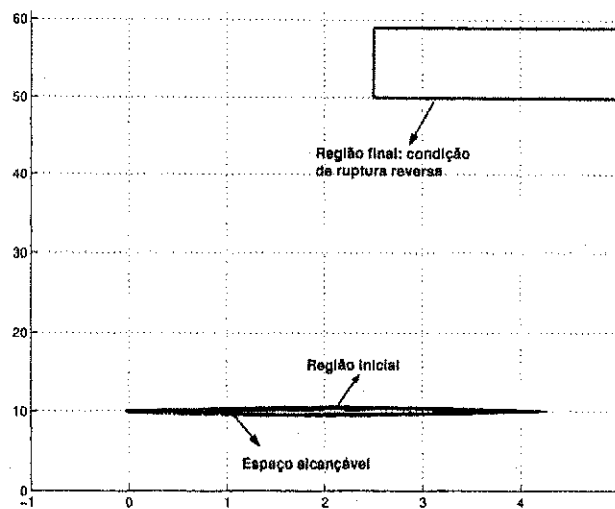


Figura 4.16: Verificação de propriedade de alcançabilidade - Estado não-alcançável

(b) $AF(9 < v_C < 10)$

Esta especificação é utilizada para verificar se, a partir da condição inicial, existe algum estado alcançável, tal que a restrição ($9 < v_C < 10$) seja satisfeita. Esta propriedade é satisfeita pelo sistema modelado, conforme ilustrado na Figura 4.18.

(c) $AG(i_D < i_{Max})$

Esta especificação é utilizada para verificar se, a partir da condição inicial, em todo estado alcançável, a corrente que passa no diodo D é maior que a corrente máxima por ele suportada (i_{Max}). Esta propriedade é satisfeita pelo sistema modelado, conforme ilustrado na Figura 4.19, pois não existe estados alcançáveis, cuja corrente no diodo seja maior que i_{Max} .

4. Propriedades de percurso (*trace*):

O procedimento de verificação foi executado para os seguintes exemplos de propriedades:

(a) $AF(Loc = 3)$

Esta especificação é utilizada para verificar se, a partir da condição inicial, a locação 3 do autômato será inevitavelmente alcançada.

(b) $Loc = 2 \ \& \ (1 < i_L < 2.5)$

Esta especificação é utilizada para verificar se, na locação 1 do autômato, os valores de corrente devem estar sempre no intervalo ($1 < i_L < 2.5$).

(c) $Loc = [0, 1, 2]$

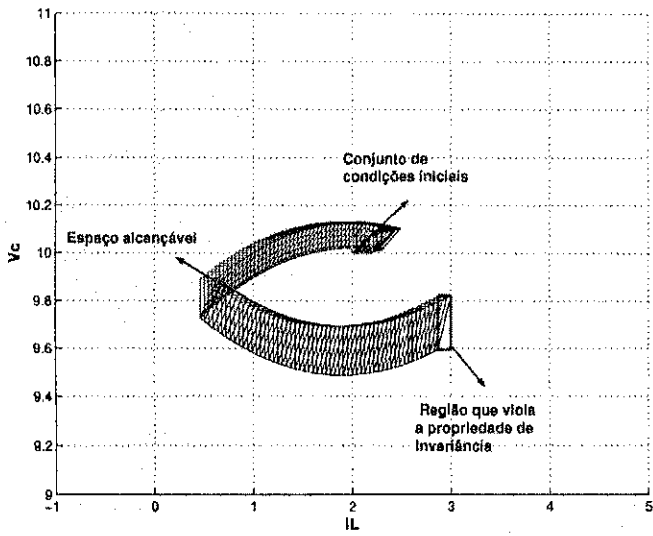


Figura 4.17: Verificação de propriedade de invariância - Propriedade não satisfeita

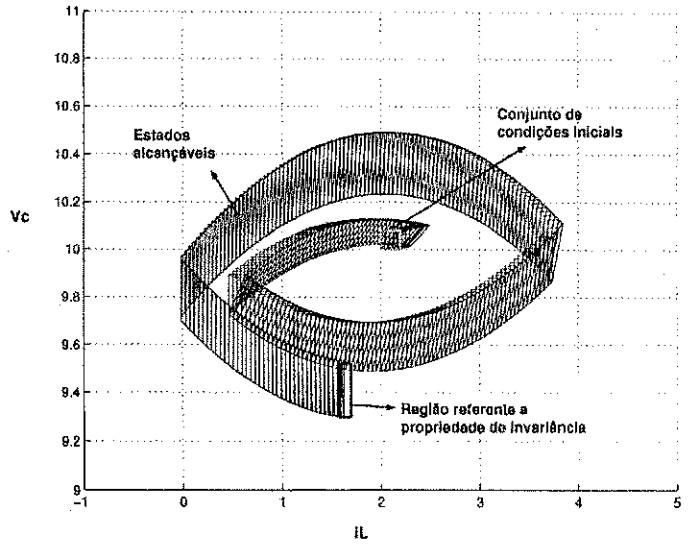


Figura 4.18: Verificação de propriedade de invariância - Propriedade satisfeita

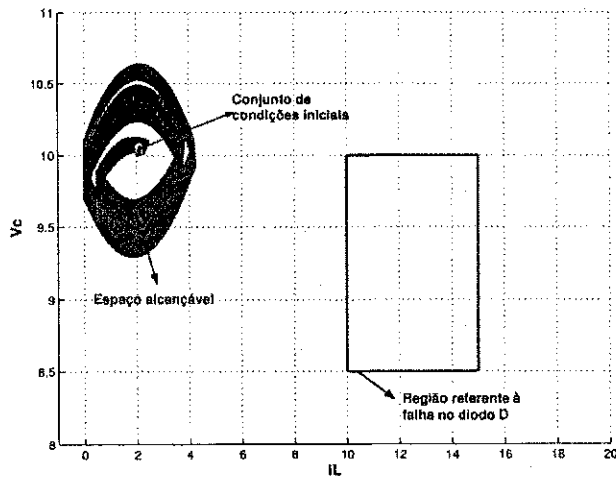


Figura 4.19: Verificação de propriedade de invariância - Propriedade satisfeita

Esta propriedade verifica se, para o autômato do conversor modelado, o ciclo de operação, compreendendo a seqüência de locações 0, 1 e 2, é sempre obedecido. O sistema satisfaz a propriedade especificada, garantindo a corretude do modelo, uma vez que esta seqüência é a esperada para este conversor.

O algoritmo de verificação forneceu, como resposta, que o sistema satisfaz a estes exemplos de propriedade de percurso.

O procedimento de verificação de propriedades sobre o espaço de estado foi executado para as propriedades, acima relacionadas, e os resultados obtidos estão mostrados na Tabela 4.2.

Tabela 4.2: Resultado da verificação de propriedades no sistema conversor

Propriedade	Resultado	Tempo decorrido (em segundos)
1(a)	Seguro	87.330
1(b)	Não seguro	26.360
1(c)	Seguro	89.883
2(a)	Satisfeita	13.020
2(b)	Não satisfeita	88.320
2(c)	Não satisfeita	8.6170
3(a)	Não satisfeita	13.410
3(b)	Satisfeita	42.680
3(c)	Satisfeita	87.3660
4(a)	Satisfeita	2.7740
4(b)	Satisfeita	3.0040
4(c)	Satisfeita	88.278

4.5 Conclusões

Este capítulo apresentou a importância da utilização de métodos formais para análise e verificação de propriedades de circuitos de eletrônica de potência. Além disto, foi apresentado um resumo sobre os tipos de métodos de verificação existentes. Dos tipos existentes, esta tese trata do método de verificação simbólica de sistemas híbridos para a realização de procedimentos de verificação de propriedades, através de métodos de alcançabilidade. Finalmente, os procedimentos de alcançabilidade e verificação foram ilustrados para um estudo de caso do circuito conversor Buck e mostrados os resultados da verificação de propriedades de segurança, alcançabilidade e invariância.

O próximo capítulo apresentará o método de reprojeto de controladores, sob o qual este trabalho se baseia, para a obtenção sistemática de controladores para circuitos de eletrônica de potência.

Capítulo 5

Reprojeto de controladores de sistemas híbridos

No capítulo anterior, foram apresentados os procedimentos para verificação automática de propriedades para sistemas modelados por autômatos híbridos. Estes procedimentos baseiam-se em pesquisas sobre o espaço de estados do sistema, representados, simbolicamente, por poliedros ortogonais através de procedimentos de aproximação. A aproximação do espaço de estado do sistema foi realizada para que os procedimentos de verificação de sistemas híbridos fossem executados de forma algorítmica, ou seja, para que fossem decidíveis.

Neste capítulo, é apresentado o procedimento para o reprojeto do controlador de circuitos de potência, baseado nos métodos de análise de alcançabilidade apresentados anteriormente. Esta tese trata, particularmente, de controlador chaveado, no qual a planta é chaveada de um modo de operação para outro, obedecendo a uma dada especificação do controlador que opera em malha fechada com a referida planta. Finalmente, é apresentado um estudo de caso para ilustrar a aplicação do procedimento de reprojeto, apresentado neste capítulo.

5.1 Introdução

O objetivo da análise de propriedades é verificar se uma dada propriedade é satisfeita no sistema projetado, enquanto que o objetivo do *reprojeto do controlador* é obter controladores para que o sistema controlado satisfaça uma dada especificação.

O sistema original é modelado por um autômato híbrido, sobre o qual será aplicado o procedimento de reprojeto do controlador. Um diagrama simplificado, ilustrando a utilização de autômatos híbridos para análise e controle de sistemas, é mostrado na Figura 5.1. Dada uma especificação e circuito de potência, executa-se o procedimento para modelagem do conversor,

apresentado no Capítulo 3, obtendo como resultado sua estrutura, diferentes configurações (determinadas pelos diferentes estados das chaves), vetor de estado e equações dinâmicas. Todas estas informações são utilizadas pelo procedimento de obtenção do modelo do autômato híbrido que representa o comportamento do circuito especificado. Uma vez obtido o autômato híbrido, são executados os procedimentos de aproximação dos estados alcançáveis sobre o qual serão executados os métodos de verificação de propriedade e reprojeto do controlador.

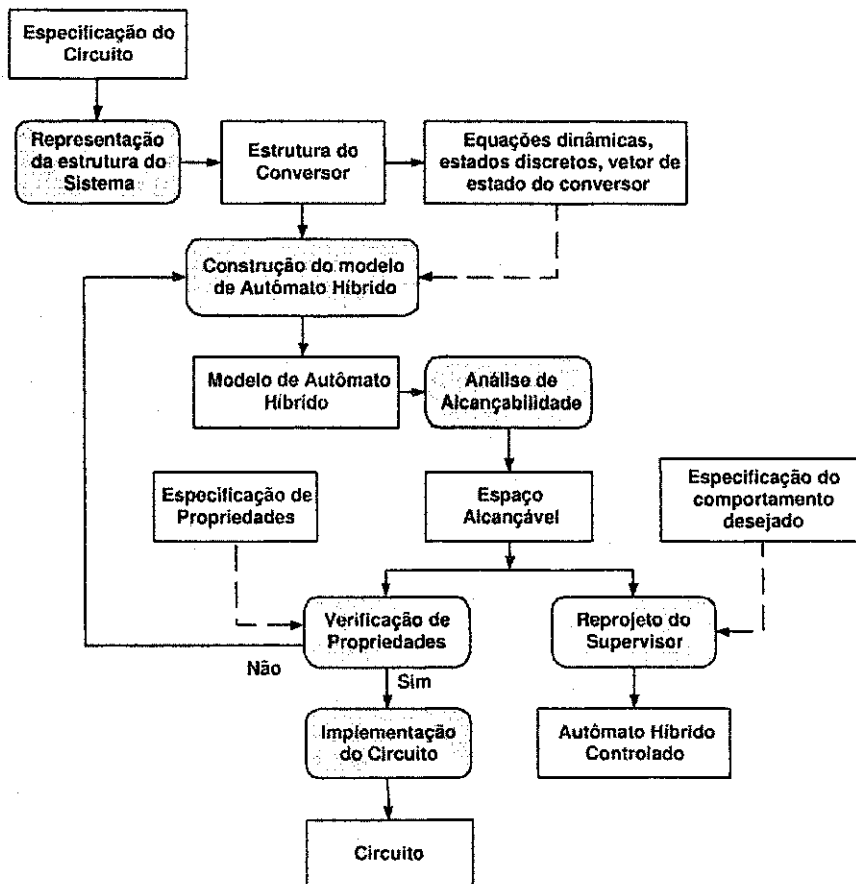


Figura 5.1: Diagrama representando a utilização de autômatos híbridos para análise e controle de sistemas híbridos

O método de reprojeto utilizado, neste trabalho, resulta em um controlador, cujo objetivo é garantir que todos os estados alcançáveis do autômato híbrido permaneçam dentro de um conjunto de estados seguros. Este conjunto de estados seguros representa a especificação do comportamento desejado, dada pelo projetista, chamada de *especificação de segurança* e representada por uma região do espaço de estado deste autômato. O controlador obtido deve gerar ações de controle para que todos os estados alcançáveis, a partir da condição inicial, sempre permaneçam dentro da especificação de segurança (WONG-TOI, 1997). Métodos que tratam de especificação de segurança são chamados de métodos para solução de *problema de segurança*

do controlador.

É importante ressaltar que o método apresentado, nesta tese, não tem o intuito de tratar problemas de controle contínuo, tais como problemas de estabilidade. É assumido que o controlador resolve localmente estes tipos de problemas. Assim, qualquer ação de controle contínuo deve ser considerada como parte da planta a ser controlada. Assume-se que os tempos de chaveamento, as constantes de tempo do sistema e os chaveamentos ocorridos não levam o sistema em malha fechada à instabilidade (LIBERZON; MORSE, 1999). Este trabalho abordará unicamente problemas caracterizados como *problemas de segurança*, cujas especificações de controle e procedimento de reprojeto tratam de regiões do espaço de estado do sistema.

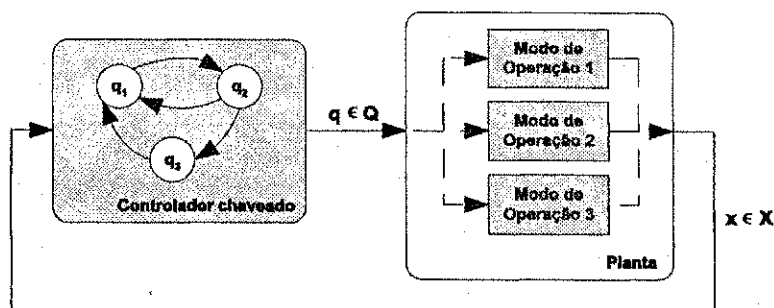


Figura 5.2: Estrutura de controle para sistemas chaveados

A Figura 5.2 ilustra o esquema de controle da classe de sistemas chaveados que é considerada neste trabalho. O sistema, denominado de *planta*, pode estar em um dos seus vários modos de operação, cujo comportamento é estabelecido por diferentes dinâmicas contínuas. O sistema pode chavear entre estes diferentes modos, quando uma determinada condição for satisfeita ou uma região do espaço de estado for alcançada. A escolha entre os modos de operação é feita por um controlador que observa, continuamente, o estado da planta e decide qual modo selecionar.

O controlador é modelado por um autômato com um conjunto Q de estados, no qual cada estado é identificado como um modo da planta. O controlador pode ser definido pela função $c: Q \times X \rightarrow 2^Q$, ou seja, quando o modo corrente é q e o estado da planta é x , o controlador pode escolher permanecer em q ou chavear para um dos vários outros modos. $c(q, x)$ contém todos os modos para os quais é fisicamente possível chavear de (q, x) . Os modos de chaveamento, q , podem expressar uma mudança discreta, gerada pelo controlador, ou podem representar uma mudança no modo de operação quando o estado da planta atinge uma região do espaço de estado, provocando uma transição sobre a qual o controlador não pode evitar (mudança não-controlável). Os tipos de transições possíveis de serem especificadas no modelo do sistema controlado serão detalhadas na Seção 5.2.1.

O resultado do processo de reprojeto é a obtenção de um controlador mais restritivo que

permite a planta permanecer em um modo ou chavear para um outro modo, se este não alcançar um "mau" estado especificado. Neste trabalho, a especificação considerada para o reprojeto do controlador trata apenas de propriedades de invariância, ou seja, o sistema deve evitar maus estados, atuando apenas na especificação dada (DANG, 2000; ASARIN; DANG; MALER, 2000). O sistema combinado, que é o produto do controlador e planta, é visto como um autômato híbrido e o algoritmo de reprojeto é executado sobre este autômato.

Na Figura 5.3, está ilustrado o esquema para o procedimento de reprojeto do controlador. Uma vez obtido o espaço alcançável do modelo de autômato híbrido do sistema e a especificação desejada, o procedimento de reprojeto é executado para obter o modelo de autômato híbrido controlado, representando o sistema original em malha fechada obedecendo à especificação do projetista.

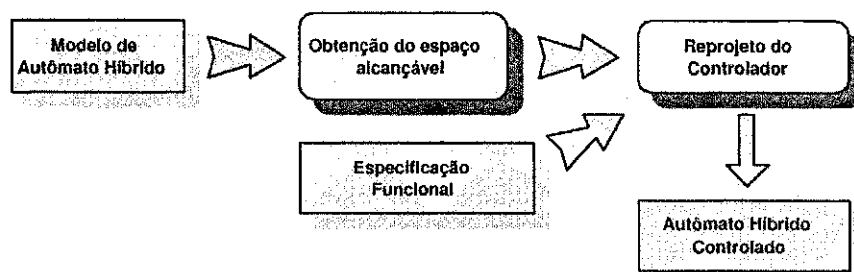


Figura 5.3: Diagrama representando procedimento de reprojeto do supervisor

As técnicas de alcançabilidade para autômatos híbridos são agora utilizadas para resolver o seguinte problema: dado um autômato híbrido, restringir as guardas das transições e condições de permanência (invariantes das locações) para que todas as trajetórias do sistema satisfaçam uma dada especificação de segurança. Este problema é semelhante à formulação da teoria de autômatos do problema de síntese de controladores: dado um sistema com vários modos contínuos, encontrar regras para chavear entre estes modos (baseado em observações de estado), restringindo o comportamento do sistema para um conjunto pré-especificado do espaço de estado, o qual pode ser considerado como o conjunto de *bons* estados.

5.1.1 Comportamento Restritivo

A realização do procedimento de reprojeto do controlador para um autômato híbrido A consiste na obtenção de um autômato A' mais restritivo que o original cujo comportamento obedece a uma dada especificação de segurança.

Para dois autômatos híbridos $A = \langle X, Q, f, H, G \rangle$ e $A' = \langle X, Q, f, H', G' \rangle$, diz-se que A' é mais restritivo, em se tratando de comportamento, que A , denotado por $A' \preceq A$, se $H' \subseteq H$ e

$G' \subseteq G$, ou seja, $H'_q \subseteq H_q$ e $G'_{qq'} \subseteq G_{qq'}$, $\forall q, q' \in Q$.

O conjunto de trajetórias de A^* , autômato que representa o comportamento máximo possível de ser controlado, está incluído no conjunto de trajetórias de A e é obtido, através da restrição do comportamento do autômato original (A), ou seja, $A^* \preceq A$.

5.2 Problema de Reprojeto

O problema de reprojeto do controlador para evitar *maus estados*, tratando-se de um problema de "estado proibido", é definido da seguinte forma: sejam A um autômato híbrido conforme definição 2.3.1 e $S \subseteq Q \times X$ um conjunto que representa a especificação do controlador, o objetivo do procedimento de reprojeto do controlador é encontrar o máximo autômato $A^* \preceq A$ tal que todas as trajetórias de A^* permaneçam em S .

A solução utilizada, nesta abordagem, é baseada no cálculo do máximo conjunto P^* de estados "vencedores". O conjunto P^* contém os estados do autômato, para os quais o controlador, através de chaveamento apropriado, garante que todas as trajetórias do sistema controlado estão dentro do conjunto S . Então, o autômato A^* e a regra de chaveamento podem ser derivados de P^* , que será chamado de *conjunto de invariante máximo*.

A existência e unicidade da solução de A^* pode ser definida da seguinte forma:

O conjunto P^* é o conjunto de todos os estados que admitem uma trajetória dentro da especificação S , ou seja, A^* é um autômato híbrido, cujas trajetórias sempre permanecem em S . O procedimento para calcular P^* remove deste conjunto os estados dos quais é possível sair da especificação. Como resultado, qualquer autômato maior que A^* conterá estados fora de P^* , dos quais o sistema sai de S depois de um tempo finito. Assim sendo, A^* é o autômato que possui o comportamento máximo possível de ser controlado, obtido a partir do autômato que representa o sistema original A , cujas trajetórias não saem de S .

5.2.1 Conjunto Invariante Máximo

O reprojeto do controlador consiste na obtenção do conjunto invariante máximo (P^*) que é o sub-conjunto do espaço de estado do sistema, sobre o qual o controlador atua para garantir que as trajetórias do sistema controlado permanecem dentro da especificação S . Para isto, as transições do autômato híbrido A são classificadas em controláveis e não-controláveis e são criados operadores para calcular o conjunto P^* , a partir do espaço alcançável do sistema.

Tipos de Transições

O modelo de autômato híbrido, considerado neste trabalho para a obtenção do controlador, possui dois tipos de transições:

- *Transições controláveis* ($T_c \subseteq Q \times Q$): são transições geradas pelo controlador.
- *Transições não-controláveis* ($T_u \subseteq Q \times Q$): são transições geradas pelo ambiente, sobre as quais o controlador não tem controle algum, tais como: a ocorrência da ação humana no sistema (operador aperta um botão) ou a condução de um diodo (determinado pelas condições das variáveis de estado do sistema). Estes tipos de transições podem ser utilizadas para descrever a passagem de uma região para outra do espaço de estado, quando sistemas lineares são utilizados para aproximar dinâmicas não-lineares ou quando ocorre uma mudança no modo de operação de acordo com alguma condição causada pelas variáveis de estado do sistema.

No modelo de autômato híbrido a ser considerado, o controlador não tem domínio sobre as ações do ambiente, ou seja, se uma transição não-controlável e uma transição controlável estão habilitadas em um dado estado, então as transições não-controláveis têm prioridade mais elevada que as controláveis. O problema de reprojeto é formulado para encontrar um controlador, que restrinja as ações controláveis, de tal forma que o sistema controlado sempre permaneça dentro de um dado conjunto S , apesar do comportamento do ambiente.

A caracterização das transições de um modelo de autômato híbrido de circuitos de potência em controláveis e não-controláveis depende, naturalmente, do nível de abstração que foi considerado para a obtenção deste modelo. No caso da realização de um processo de refinamento de componentes deste circuito, o novo modelo de autômato híbrido obtido não expressará o mesmo conjunto de transições controláveis e não-controláveis do autômato anterior. Estes conjuntos serão alterados de acordo com o nível de refinamento realizado no(s) componente(s) do modelo e quais componentes foram refinados, uma vez que novas condições de representação de mudanças de estados foram inseridas ou alteradas.

Operadores para cálculo de P^*

Serão utilizadas, neste capítulo, as notações introduzidas, no Capítulo 2, relacionadas às evoluções contínuas de autômatos híbridos.

O cálculo do conjunto invariante máximo, P^* , é a etapa principal do algoritmo de reprojeto. Para calcular P^* , são definidos os seguintes operadores de alcançabilidade: operador *predecessor*

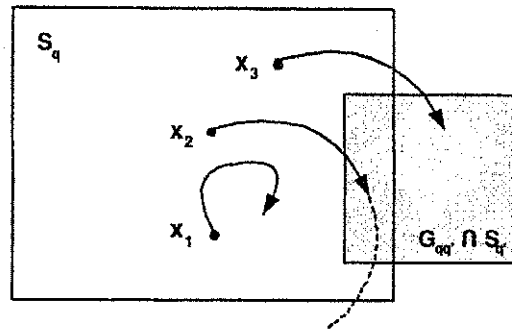


Figura 5.4: Cálculo de S'_q através da caracterização dos operadores π_q^∞ e U_q

ilimitado no tempo, operador *Until* e operador *predecessor de uma-etapa*. Sejam Z e Y subconjuntos de X , $T = \mathbb{R}$ o domínio do tempo, $t \in T$, $x, x' \in X$, $q, q' \in Q$, e $\alpha : T \rightarrow X$ a solução de $\dot{x} = f_q(x)$.

Operador Predecessor ilimitado ($\pi_q^\infty : 2^X \rightarrow 2^X$): é definido para o conjunto $Z \subseteq X$ como

$$\pi_q^\infty(Z) = \{x \mid x \xrightarrow{q, \infty} Z\}$$

O conjunto $\pi_q^\infty(Z)$ é o conjunto de estados, a partir dos quais é possível continuar indefinidamente com a dinâmica do estado q , enquanto ainda permanece em Z .

Pode ser observado, na Figura 5.4, que a trajetória do ponto x_1 permanece sempre em S_q , enquanto que a trajetória de x_2 eventualmente sai de S_q depois de algum tempo. Assim, $x_1 \in \pi_q^\infty(S_q)$, porém $x_2 \notin \pi_q^\infty(S_q)$.

Operador Until ($U_q : 2^X \times 2^X \rightarrow 2^X$): é definido para os conjuntos $Z, Y \subseteq X$ como

$$U_q(Z, Y) = \{x \mid \exists t, x \xrightarrow{q, t} Y\}$$

O conjunto $U_q(Z, Y)$ é o conjunto de estados, a partir dos quais é possível continuar com a dinâmica f_q e permanecer dentro de Z até que Y seja alcançado.

Pode ser observado na Figura 5.4 que a trajetória do ponto x_2 permanece em S_q até alcançar a região que representa $G_{qq'} \cap S_{q'}$, enquanto que a trajetória de x_3 sai de S_q antes de alcançar a região $G_{qq'} \cap S_{q'}$. Assim, $x_2 \in U_q(S_q, G_{qq'} \cap S_{q'})$, porém $x_3 \notin U_q(S_q, G_{qq'} \cap S_{q'})$.

Operador Predecessor de uma-etapa ($\pi : 2^{Q \times X} \rightarrow 2^{Q \times X}$): é definido para um conjunto

$S = (q_1, S_1) \cup \dots \cup (q_m, S_m)$, onde m é o número de estados discretos em Q . Assim, $\pi(S)$ pode ser definido como

$$\pi(S) = \{(q, x) \mid x \xrightarrow{q, \infty} S \vee (\exists t \in T \exists q' \in Q \exists x' \in X, x \xrightarrow{q, t} x' \wedge x' \in G_{qq'} \wedge (q', x') \in S)\}$$

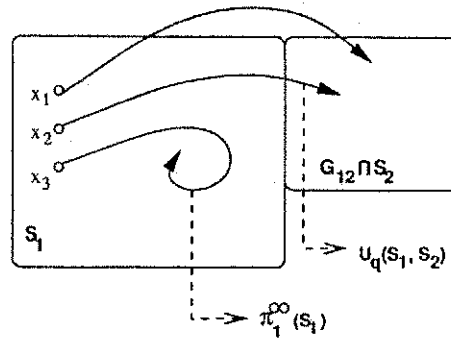


Figura 5.5: Cálculo de S'_1 de S_1 e S_2 .

A interpretação para o operador *predecessor de uma-etapa*, π , é que um estado (q, x) está em $\pi(S)$, se existe uma trajetória infinita sem descontinuidades, começando de (q, x) , sempre permanecendo em S , ou é possível permanecer em S , por algum tempo, e executar uma transição para um outro estado (q', x') que ainda permanece em S .

O operador $\pi(S)$ pode ser expresso através dos operadores \mathcal{U}_q e π_q^∞ como $\pi(S) = (q_1, S'_1) \cup \dots \cup (q_m, S'_m)$, onde para todo q

$$S'_q = \pi_q^\infty(S_q) \cup \bigcup_{q' \neq q} \mathcal{U}_q(S_q, G_{qq'} \cap S_{q'}) \quad (5.1)$$

A Figura 5.4 ilustra o cálculo de $S'_{q'}$ utilizando os operadores \mathcal{U}_q e π_q^∞ . Pode ser notado que os pontos x_1 e x_2 pertencem a $S'_{q'}$, uma vez que x_1 invariavelmente nunca sai de S_q , ou seja, $x_1 \in \pi_q^\infty(S_q)$, e x_2 permanece em S_q até que $G_{qq'} \cap S_{q'}$ seja alcançado, ou seja, $x_2 \in \mathcal{U}_q(S_q, G_{qq'} \cap S_{q'})$. O ponto x_3 não pertence à região calculada pelo operador π , pois $x_3 \notin S_{q'}$.

A utilização destes operadores está ilustrada na Figura 5.5. Nesta Figura, o ponto $x_3 \in \pi_1^\infty(S_1)$, o ponto $x_2 \in \mathcal{U}_1(S_1, (G_{12} \cap S_2))$ e o ponto x_1 não está em nenhum destes conjuntos; portanto, $x_1 \notin S'_1$.

As setas com linhas preenchidas, nas Figuras 5.4, e 5.5 representam as trajetórias do sistema, segundo a dinâmica f_q , tendo como condição inicial x_1, x_2 e x_3 .

Quando todas as transições são controláveis, para o sistema permanecer em S , a estratégia do controlador seria esperar a passagem de um tempo $t > 0$ e, então, executar uma transição. Todavia, na presença de transições não-controláveis, deve ser observado que em algum tempo $t' < t$ o ambiente pode executar uma transição que levaria o sistema a sair da especificação S .

Assim sendo, se $(q, q') \in T_u$, então de qualquer estado (q, x) , tal que $x \in G_{qq'}$, o ambiente pode forçar uma transição para (q', x) e, se $(q', x) \notin S$, então isto fará com que o sistema viole

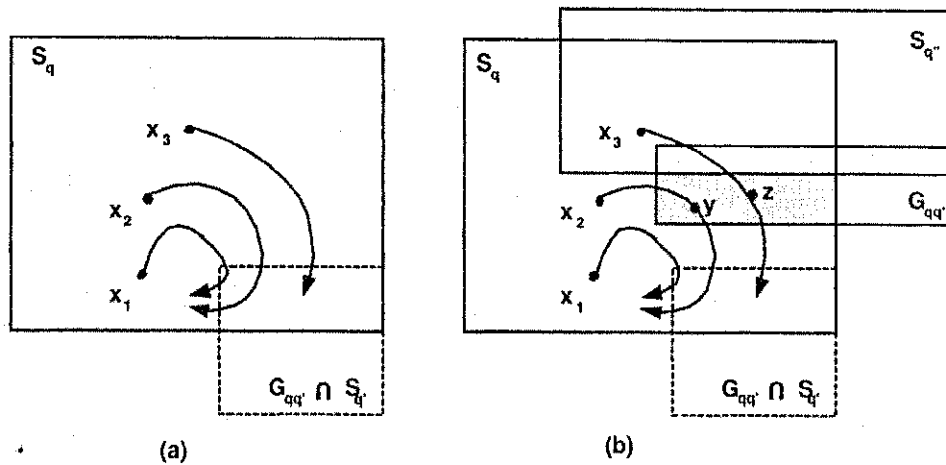


Figura 5.6: (a) Operador π para transições controláveis (T_c) (b) Operador π para transições controláveis (T_c) e não-controláveis (T_u)

a especificação de segurança. Devido a isto, o operador π será calculado da seguinte maneira:

Seja $S = (q_1, S_1) \cup \dots \cup (q_m, S_m)$, o cálculo de $\pi(S) = (q_1, S'_1) \cup \dots \cup (q_m, S'_m)$ é realizado em duas etapas:

1. Calcular $\widehat{S} = (q_1, \widehat{S}_1) \cup \dots \cup (q_m, \widehat{S}_m)$, onde

$$\widehat{S}_q = S_q \setminus \bigcup_{(q,q') \in T_u} G_{qq'} \cap \overline{S}_{q'}$$

onde $\overline{S}_{q'}$ representa o complemento de $S_{q'}$. A obtenção de \widehat{S} consiste na remoção de S de todos os estados (q, x) , dos quais o ambiente pode levar o sistema para algum estado (q', x) para fora de S .

2. Para todo estado $q \in Q$, calcular S'_q da seguinte maneira:

$$S'_q = \pi_q^\infty(\widehat{S}_q) \cup \bigcup_{q,q' \notin T_u} \mathcal{U}_q(\widehat{S}_q, G_{qq'} \cap \widehat{S}'_{q'}) \quad (5.2)$$

A Figura 5.6 ilustra o cálculo do operador π para transições controláveis e não-controláveis. Na Figura 5.6(a), (q, q') é a única transição a partir de q , e esta transição é controlável. Pode ser observado que os pontos x_1, x_2 e x_3 estão todos em $\pi(S)$, uma vez que suas trajetórias permanecem invariavelmente em S_q ou permanecem lá até que a região $G_{qq'} \cap S_{q'}$ seja alcançada. Ao adicionar uma transição não-controlável (q, q'') ao sistema, retiramos da região de $\pi(S)$ o retângulo sombreado, conforme mostrado na Figura 5.6(b). Esta região consiste dos pontos em S_q , a partir dos quais o ambiente pode forçar uma transição para q'' e fazer com que o sistema

saia de S . Portanto, \widehat{S}_q é o resultado da remoção deste retângulo de S_q . Neste caso, os pontos x_2 e x_3 não pertencem a $\pi(S)$.

Com a caracterização do operador π , será apresentado, a seguir, o algoritmo para o cálculo do conjunto máximo invariante P^* .

5.3 Procedimentos para reprojeto do controlador

Dado $Z \subseteq X$ e um conjunto $S = \{(q, S_q) | q \in Q \wedge S_q \subseteq X\}$, o primeiro passo para obter o conjunto P^* é calcular o conjunto $\pi(S)$. O Algoritmo 9 caracteriza o operador $\pi(S)$ e foi construído a partir da Eq. 5.2.

Algoritmo 9 Cálculo de $\pi(S)$

```

 $P^0 := \emptyset;$ 
para todo  $q \in Q$  faça
   $Z := \pi_q^\infty(S_q);$ 
  para todo  $q' \neq q$  faça
     $Z := Z \cup \mathcal{U}_q(S_q, G_{qq'} \cap S_{q'});$ 
  fim para
   $P := P \cup (q, Z);$ 
fim para
retorne  $P$ 

```

O Algoritmo 9 utiliza os operadores \mathcal{U}_q e π_q^∞ que são baseados no cálculo de estados alcançáveis, apresentado no Capítulo 2. Neste algoritmo, o resultado é um sub-conjunto do espaço alcançável para cada q , cujas trajetórias não saem da especificação S . O algoritmo efetivo de reprojeto é obtido, utilizando as técnicas de aproximação, também apresentadas no Capítulo 2, para sub-aproximar o conjunto resultante, a partir do cálculo destes operadores. O cálculo da sub-aproximação é necessário nos algoritmos de reprojeto de controladores, para garantir que o conjunto invariante máximo calculado seja um subconjunto do conjunto invariante P^* .

Cálculo do operador \mathcal{U}_q . O conjunto (Z, Y) é o conjunto de estados, a partir dos quais o sistema pode alcançar Y , enquanto permanece em Z . Assim, (Z, Y) pode ser caracterizado como o conjunto de predecessores contínuos de (q, Y) , $\Delta_c(q, Y)$, tendo Z como condição de permanência em q . Como dito anteriormente, o cálculo dos predecessores contínuos equivale a calcular os sucessores contínuos com a dinâmica reversa, ou seja, $\dot{x} = -f_q$.

Cálculo do operador π_q^∞ . O conjunto $\pi_q^\infty(Z)$ pode ser obtido, removendo de Z os estados

a partir dos quais a dinâmica contínua f_q pode levar o sistema à região \bar{Z} , onde \bar{Z} é o complemento de Z . Assim, $\pi_q^\infty = Z \setminus \Delta(\bar{Z})$.

Depois de calculado, o operador π pode ser sub-aproximado através de poliedros convexos, conforme procedimentos mostrados no Capítulo 2.

5.3.1 Algoritmo para cálculo do conjunto invariante máximo

O Algoritmo 10 descreve as etapas para o cálculo do conjunto P^* , representando o conjunto invariante máximo.

Algoritmo 10 Cálculo de P^*

$P^0 := S \cap H;$

repita

$k = 0, 1, 2 \dots$

$P^{k+1} := P^k \cap \pi(P^k);$

até $P^{k+1} = P^k$

$P^* := P^k;$

Pode ser observado que, neste algoritmo, os estados em S , que não estão em H , não são admitidos pelo modelo. Assim, o algoritmo inicializa o cálculo do conjunto invariante com $P^0 = S \cap H$. O algoritmo termina com a obtenção do ponto fixo P^* . O conjunto P^* contém todos os estados, para os quais existem controladores chaveados que podem prevenir o sistema de sair da especificação S .

Uma vez obtido o conjunto P^* , o autômato A é restrito a P^* da seguinte forma: o autômato $A^* = \langle X, Q, f, H^*, G^* \rangle$, onde $\forall q, q' \in Q, H_q^* = \{x | (q, x) \in P^*\}$ e $G_{qq'}^* = G_{qq'} \cap H_q^* \cap H_{q'}^*$ é a solução do problema de *reprojeto de segurança do controlador*. O autômato A^* é o máximo autômato, cujas trajetórias nunca saem de S .

5.4 Controlador Chaveado

O controlador chaveado derivado de A^* é definido pela lei de controle $c : Q \times X \rightarrow 2^Q$ da seguinte forma:

$$c(q, x) = \{q' | (q' = q \wedge x \in H_q^*) \vee (q' \neq q \wedge x \in G_{qq'}^*)\}$$

A lei de controle, que define o controlador, não depende unicamente do estado contínuo, mas também do estado discreto do sistema.

Este controlador chaveado é semelhante à noção de *supervisor menos restritivo* de Ramadge e Wonham (1989), ou seja, $\forall(q, x) \in Q \times X$, todos os outros controladores chaveados, que mantêm o sistema em S , têm suas leis de malha fechada contidas em c .

Em circuito de eletrônica de potência, o reprojeto do controlador pode ser implementado, por exemplo, através de acréscimos de condições ou testes, representando as restrições da especificação do controlador, no programa do micro-controlador que opera com o circuito.

5.5 Estudo de Caso

Ainda não existe ferramenta disponível que realize a síntese ou reprojeto do controlador para sistemas híbridos. No entanto, implementamos no MATLAB todos os procedimentos para o reprojeto do controlador de forma sistemática, baseado na abordagem de Asarin, Dang e Maler (2000), para a aplicação introduzida, no Capítulo 3. O procedimento foi executado para o conversor com controle por histerese da tensão da carga e para o conversor com controle por histerese da corrente da carga.

5.5.1 Conversor com controle por histerese da tensão da carga

Os resultados obtidos são os seguintes: o conjunto P^* obtido para a planta, representada pelo conversor da Figura 3.27, está ilustrado nas Figuras 5.7 e 5.8, que representa o conjunto P^* para as locações Loc_2 e Loc_3 do modelo da planta.

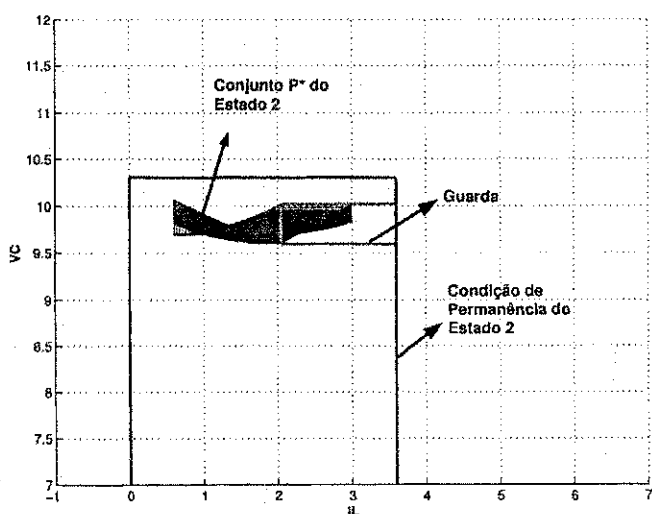


Figura 5.7: Conjunto P^* da locação 2.

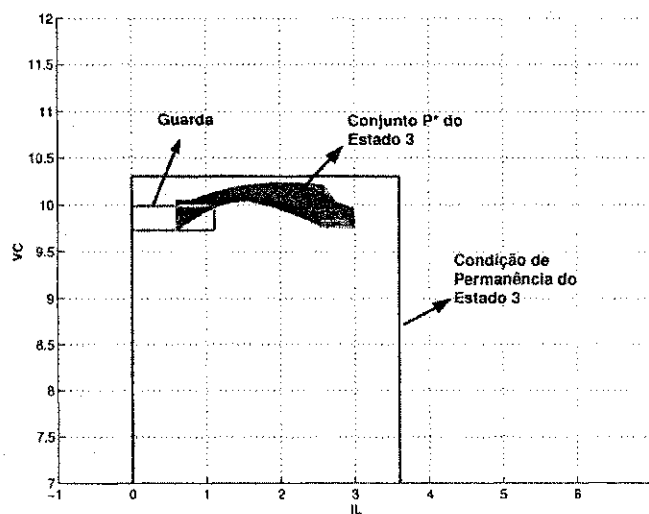


Figura 5.8: Conjunto P^* da locação 3.

O controlador foi obtido para a seguinte especificação: a tensão de saída deve obedecer à

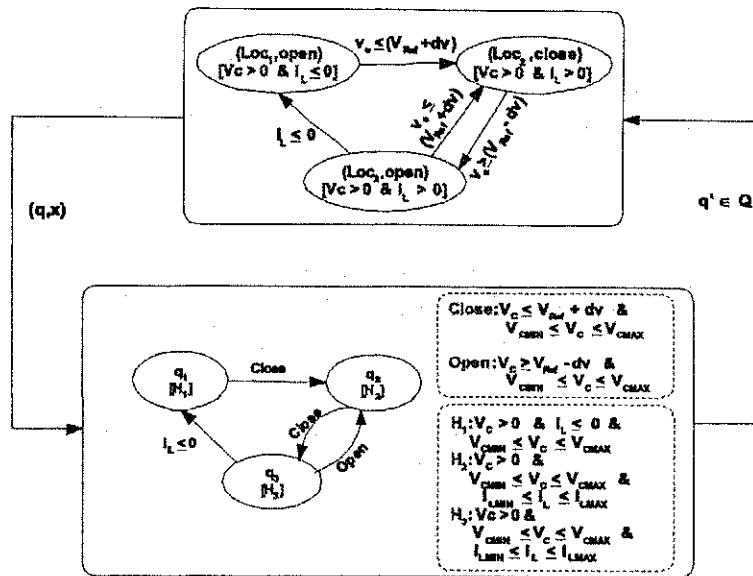


Figura 5.9: Controlador obtido após execução do procedimento de reprojeto

condição $V_{CMIN} \leq V_C \leq V_{CMAX}$, onde $V_{CMIN} = 9.6 \text{ V}$ e $V_{CMAX} = 10.05 \text{ V}$, e a corrente na indutância deve obedecer a seguinte condição: $I_{MIN} \leq i_L \leq I_{MAX}$, onde $I_{MIN} = 0.6 \text{ A}$ e $I_{MAX} = 3 \text{ A}$. O sistema controlado pelo controlador sintetizado está ilustrado na Figura 5.9.

O autômato $A^* = \langle Q, X, H^*, G^*, f \rangle$ é a solução do problema do reprojeto do controlador. A função de realimentação, representando o controlador $c : Q \times X \rightarrow 2^Q$, pode ser derivada de A^* , representado na Figura 5.9.

5.5.2 Conversor com controle por histerese da corrente da carga

Os resultados obtidos para a mudança na variável de controle são os seguintes: o conjunto P^* obtido para a planta está ilustrado, nas Figuras 5.10 e 5.11, que representa o conjunto P^* para as locações Loc_2 e Loc_3 do modelo da planta.

O controlador foi obtido para a seguinte especificação: a corrente na carga (i_R) deve obedecer a condição $I_{MIN} \leq i_R \leq I_{MAX}$, onde $I_{MIN} = 0.7 \text{ A}$ e $I_{MAX} = 1.3 \text{ A}$.

O algoritmo de reprojeto foi executado e o sistema controlado pelo controlador sintetizado está ilustrado, na Figura 5.12.

5.6 Conclusões

Este Capítulo apresentou uma solução para o problema de reprojeto de segurança do controlador baseado na abordagem de Asarin, Dang e Maler (2000). O reprojeto do controlador consiste

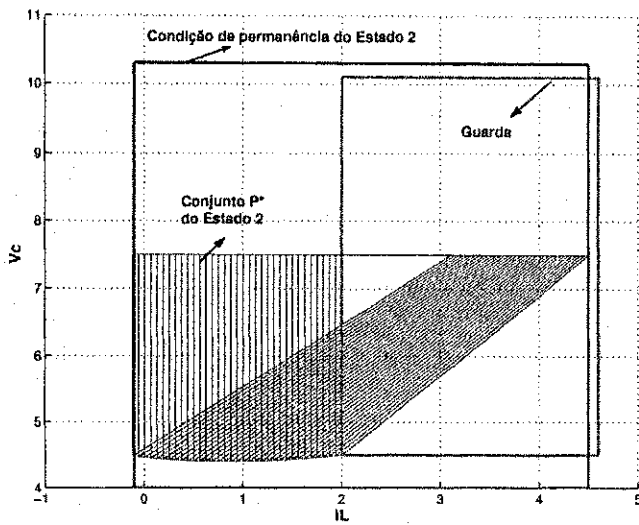


Figura 5.10: Conjunto P^* da localização 2.

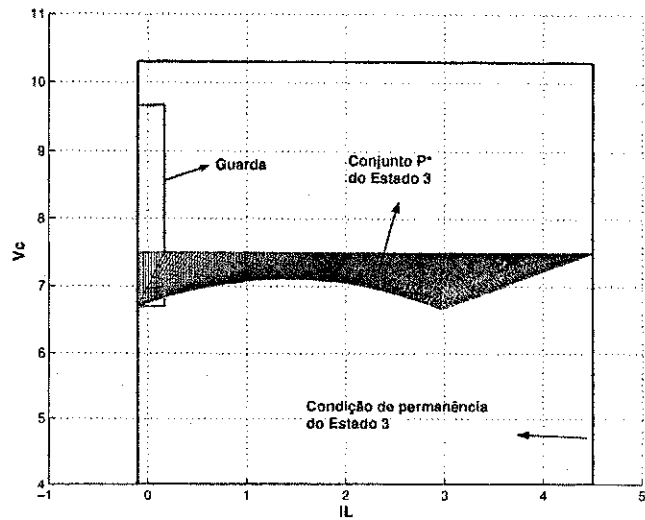


Figura 5.11: Conjunto P^* da localização 3.

em restringir o autômato A , representando o sistema, ao conjunto P^* , caracterizado como o conjunto invariante máximo. Este conjunto é efetivamente calculado, utilizando as técnicas de alcançabilidade aproximadas apresentadas no Capítulo 2. O algoritmo para reprojeto de controladores considera a existência de dois tipos de transições no modelo do autômato híbrido original: transições controláveis e não-controláveis. Finalmente, foi apresentado um estudo de caso com a aplicação da abordagem apresentada para um conversor Buck.

O método para reprojeto do controlador, apresentado neste trabalho, utiliza uma abordagem simbólica para tratar diretamente o espaço de estados de sistemas híbridos com o propósito de tornar o processo de obtenção do controlador o mais sistemático possível.

O método trata modelos contínuos com dinâmicas lineares, consideradas mais genéricas que alguns trabalhos existentes nesta área, e se constituem adequadas para a classe de conversores que este trabalho propõe-se estudar.

Em relação a alguns trabalhos desenvolvidos nesta área, podem ser citados os trabalhos de Alur e Dill (1994b) que aplicaram procedimentos para síntese de controladores numa classe de autômatos temporizados. Henzinger et al. (1995) trataram do problema de controle para uma classe de autômatos retangulares, na qual a dinâmica das variáveis é definida, utilizando apenas limites inferiores e superiores de valores em \mathbb{Q} . Trabalhos tratando de dinâmica contínua não-trivial são extremamente limitados, muitos deles não têm garantias de decidibilidade do procedimento de obtenção do controlador. Os trabalhos de Tomlin e Sastry (1999) utilizam teoria dos jogos para resolverem o problema da síntese do controlador para dinâmicas contínuas com controle contínuo com etapa variante no tempo. Cury e Krogh (1999) tratam de dinâmica não-trivial, utilizando um autômato aproximado como representação finita do autômato original.

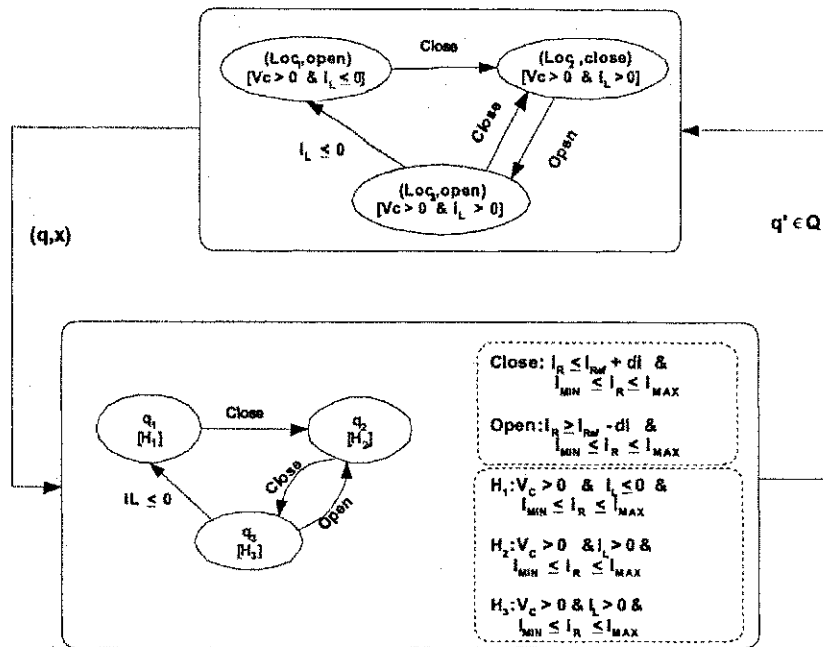


Figura 5.12: Controlador obtido após execução do procedimento de reprojeto

O procedimento deste trabalho é baseado na abordagem direta de estudo de sistemas híbridos (ASARIN ODED MALER; SIFAKIS, 1998), não sendo, portanto, necessária a obtenção de um sistema de transição finito que represente o sistema original. A aproximação do sistema híbrido para uma representação finita pode fazer com que informações importantes não sejam representadas.

No próximo capítulo, serão apresentadas as conclusões gerais, com as contribuições e perspectivas de futuros possíveis trabalhos originados desta tese.

Capítulo 6

Conclusões e propostas para futuros trabalhos

Este trabalho tratou das possibilidades de utilização de métodos formais, no estudo de circuitos de eletrônica de potência.

Inicialmente, foi desenvolvido um procedimento sistemático de obtenção do modelo matemático que captura o comportamento do sistema sob a forma de autômato híbrido.

A partir da descrição formal do comportamento do sistema, demonstrou-se a possibilidade de verificação da correção do projeto do sistema, analisando se seu comportamento corresponde à sua especificação.

A análise de propriedades de um sistema é um procedimento fundamental no auxílio aos projetistas durante as etapas de desenvolvimento de sistemas, permitindo a detecção de faltas e erros nos estágios iniciais, além de possibilitar uma maior confiabilidade em sua operação.

O uso de métodos formais em projetos de circuito de eletrônica de potência contribui na obtenção de projetos corretos em aplicações críticas, além de possibilitar a economia de tempo e custo no processo de sua implementação.

6.1 Contribuições

Este trabalho contribui ao mostrar como utilizar métodos formais para estudo, na área de eletrônica de potência. A aplicação de técnicas matemáticas, através da utilização de métodos formais e procedimentos algorítmicos conhecidos na Ciência da Computação, para modelagem, análise do comportamento e reprojeto de um novo controlador, na área de eletrônica de potência, constitui-se numa nova área de aplicação, utilizando a abordagem de sistemas híbridos.

6.1.1 Modelagem

Este trabalho introduz um procedimento sistemático para, a partir de um modelo de circuito de eletrônica de potência, obter um modelo matemático que captura o comportamento híbrido inerente a este sistema, através de autômatos híbridos. O modelo de autômato híbrido obtido é baseado na representação da estrutura do conversor, resultante de um método de montagem de blocos de elementos (*jogo de quebra-cabeças*) que representam os componentes deste conversor. Além disto, o procedimento de modelagem permite tratar refinamentos de dispositivos semicondutores do circuito.

A obtenção de soluções eficientes e de procedimentos decidíveis e algorítmicos para questões relacionadas à análise de comportamento e ao controlador de circuitos de potência não pode prescindir de tratar a generalidade de sua dinâmica contínua. Os métodos aqui utilizados para cálculo dos estados alcançáveis, verificação de propriedades e reprojeto do controlador tratam eficientemente dinâmicas não-triviais e consideram sistemas de dimensões relativamente elevadas.

Os procedimentos para obtenção do espaço alcançável híbrido possibilitam tratar:

- Conjuntos de condições iniciais,
- Incertezas na entrada da fonte do circuito e
- Variações nos valores nominais dos parâmetros dos seus componentes.

Além disto, como contribuição relevante para estudos de sistemas híbridos, foi introduzido um método de obtenção do espaço alcançável de sistemas híbridos que possibilita a representação de fontes de entrada variante no tempo, no modelo do circuito.

6.1.2 Verificação de Propriedades

Normalmente, a análise que se faz sobre um modelo de circuito eletrônico de potência é realizada através de simulações e testes exaustivos. Ferramentas como a simulação não permitem avaliar o desempenho global do circuito para determinados tipos de problemas, tais como: (a) a análise de propriedades a partir de um intervalo de valores para as condições iniciais, (b) o tratamento na variação dos valores nominais dos parâmetros do circuito e (c) incertezas na fonte de entrada do sistema. Para casos como estes, a abordagem de simulação é insatisfatória, porque alguns erros podem ser detectados depois da implementação do sistema, sendo, assim, difíceis de serem consertados. Além disto, testes exaustivos não garantem que todas as condições de operação do sistema foram cobertas.

Os métodos aqui apresentados, de utilização de técnicas formais para estudar o comportamento de circuitos de eletrônica de potência, tanto para verificação de propriedades quanto para

o reprojeto do controlador, basearam-se na abordagem direta para estudo de sistemas híbridos. A principal vantagem da utilização da abordagem direta é que procedimentos para verificação e reprojeto do controlador podem ser realizados não apenas para uma classe restrita de autômatos híbridos, mas também para dinâmicas não-triviais de representação do comportamento contínuo, notadamente, dinâmicas lineares.

Nesta tese, foram implementados procedimentos de verificação de autômatos híbridos, tratando os seguintes tipos de propriedades: segurança, invariância, alcançabilidade e percurso. As propriedades são especificadas formalmente, utilizando lógica temporal ACTL.

Os procedimentos de verificação de propriedades, implementados nesta tese, deram as seguintes contribuições para o estudo de circuitos de potência:

- Permite tratar não apenas regiões do espaço de estado, mas também dos estados discretos alcançáveis do sistema.
- Possibilita ampliar os tipos de propriedades tratadas por métodos de abordagem direta para análise de sistemas híbridos.

Em algumas ferramentas utilizadas para estudo de sistemas híbridos, como Checkmate, Hytech, Verishift e Shift, existe a possibilidade de modelar dinâmicas não-triviais. Entretanto, estas ferramentas não permitem representar incerteza na entrada dos modelos de autômatos híbridos, e, tampouco, representar entradas variantes no tempo.

Os tipos de propriedades, tratadas neste trabalho, possibilitam verificar alguns aspectos do comportamento de circuitos de eletrônica de potência, tais como:

- As condições de operação que põem em risco algum componente do circuito.
- O nível de tensão, sobre o qual uma chave comutaria.
- A compatibilidade dos valores de tensão na carga com um determinado limiar especificado.
- A existência de alguma condição inicial que comprometa o comportamento do circuito.
- As condições de falha em dispositivos semicondutores.
- A existência de alguma condição de operação que, caso haja uma falta de energia durante a operação do circuito, provoque uma situação de falta ou erro no circuito.
- A existência de alguma condição em que a tensão ou corrente, sobre determinado componente do circuito, pode ultrapassar os limites estabelecidos na especificação do projeto.

6.1.3 Reprojeto de Controladores

O problema de reprojeto do controlador é encontrar o autômato híbrido não-bloqueante máximo, de modo que toda trajetória do autômato do controlador pertença a um subconjunto do espaço de estado do autômato híbrido que representa o circuito.

O problema de reprojeto, considerado neste trabalho, trata de controladores chaveados para circuitos de eletrônica de potência com relação a uma especificação de segurança. O objetivo do controlador é garantir que todas as configurações alcançáveis levem a um conjunto de configurações seguras, ou seja, dentro de um conjunto que representa a especificação do controlador.

Os procedimentos de reprojeto do controlador para circuitos de potência, implementados nesta tese, são baseados na abordagem direta para estudo de sistemas híbridos (DANG, 2000; ASARIN; DANG; MALER, 2001). Esta implementação é relevante, principalmente do ponto de vista prático, uma vez que ainda não existe disponível ferramentas que tratem da síntese do controlador de sistemas híbridos, baseando-se na pesquisa direta do espaço de estado do sistema.

6.2 Futuros Trabalhos

Como proposta para futuros trabalhos, sugerimos a implementação de um modelo de circuito que considere dinâmicas não-lineares mais complexas para verificação do comportamento do circuito. Além disto, poder-se-ia ser desenvolvido um procedimento que possibilite a obtenção do autômato híbrido, diretamente, a partir da obtenção da estrutura do conversor.

Em se tratando de verificação de propriedades, o procedimento deste trabalho poderia ser estendido para tratar propriedades que modelem elementos adicionais que possam influenciar o comportamento de circuitos de potência, tais como temperatura e potência, aqui não tratados. Neste sentido, poderia ser considerado, também, formas para tratar propriedades temporais do comportamento de circuitos.

Em se tratando de problemas de reprojeto do controlador, os procedimentos, aqui implementados, podem ser estendidos para tratar de outros tipos especificações, abordando propriedades temporais ou tratar de abordagens baseadas em teoria dos jogos, conforme sugerido em Asarin Oded Maler e Sifakis (1998), nas quais o controlador e o ambiente podem ter movimentos simultâneos, para o contexto de eletrônica de potência. Além disto, os procedimentos de reprojeto podem ser generalizados para tratarem de dinâmicas contínuas não-lineares.

Além disto, devido às dificuldades encontradas em algumas das ferramentas de sistemas híbridos para tratarem os problemas de circuitos de potência aqui estudados, sugerimos a construção de uma ferramenta mais completa e genérica, baseada nos procedimentos aqui apresentados, incluindo sistemas de dimensões mais elevadas aos aqui estudados. Esta ferramenta deve integrar

uma programa de projeto de circuito, um verificador de modelo para sistemas híbridos e uma interface para permitir ao usuário especificar as propriedades que deseja verificar, e fazer uma série de modelagens de circuitos de maior complexidade. Seria uma ferramenta nova para os projetistas de circuitos, demonstrando, também, que a técnica de verificação formal tem mais um campo de aplicação útil.

Apêndice A

Glossário

Para facilitar a compreensão e consulta do assunto exposto, apresentamos um glossário com os principais termos utilizados nesta tese.

Alcançabilidade

O problema de alcançabilidade pode ser formulado da seguinte forma: sejam dois estados σ e σ' no espaço de estado infinito S de um autômato híbrido H . Assim, σ' é alcançável, a partir de σ , se existe uma execução ou uma trajetória de H que inicia em σ e termina em σ' .

Alfabeto

É um conjunto não-vazio de símbolos, no qual um determinado símbolo é uma entidade não definida formalmente.

Aproximação

É um método para representar, simbolicamente, de forma explícita e enumerativa, subconjuntos dos números reais.

Autômatos

Autômatos são modelos matemáticos de máquinas de estados que têm entradas e saídas discretas e que reconhecem um conjunto de palavras sobre um dado alfabeto. Um autômato pode ser visto como uma entidade de controle que tem uma variável interna que representa seu estado.

Autômato híbrido

Autômato híbrido é uma máquina de estado finito estendida com um conjunto de variáveis reais e equações diferenciais que são definidas diferentemente para cada estado (discreto) ou configuração da máquina de estado finito. As equações diferenciais descrevem a especificação dinâmica da evolução do sistema, naquela configuração particular. O estado do autômato muda instantaneamente, através de uma transição discreta ou através de uma atividade contínua. Os arcos modelam uma mudança abrupta no comportamento dinâmico do sistema, ocasionando a

transição de uma configuração para outra.

Autômato híbrido Poliedro invariante

É um autômato híbrido, cuja condição de guarda é uma desigualdade linear e a condição de invariante para qualquer locação é representada por um poliedro convexo definido pela conjunção dos complementos das condições de guarda.

Autômato híbrido linear

É um caso especial de autômato híbrido, no qual as taxas de mudança das variáveis contínuas são constantes, ou seja, as condições de fluxo, invariantes e relações de transição podem ser definidas por expressões lineares sobre o conjunto de variáveis contínuas do sistema (X). Vale ressaltar a interpretação especial para o termo linear neste contexto. Mais especificamente, para os modos de controle, a condição de fluxo é definida por uma equação diferencial da forma $\dot{x} = k$, onde k é uma constante definida para cada variável contínua em X , e a invariante de cada locação é definida por uma igualdade ou desigualdade linear em X . Para cada transição, o conjunto de guardas consiste de fórmulas lineares sobre X , para cada variável.

Autômato Temporizado

É um caso especial de autômato híbrido linear, em que cada variável contínua cresce uniformemente com o tempo (com inclinação 1) e pode ser considerado um *relógio*. Uma transição discreta tem as seguintes funções: *zera* o relógio ou não altera seu valor.

Autômato Retangular

É um caso especial de autômato híbrido linear, em que as condições de fluxo são independentes dos modos de controle e as variáveis são independentes aos pares. A condição de fluxo tem a forma $\dot{x} = [a, b]$ para cada variável contínua $x \in X$. A condição invariante e a relação de transição são descritas, através de predicados lineares que também correspondem a retângulos n -dimensionais.

Bissimulação

Um bisimulação é uma relação de equivalência que provê uma descrição abstrata do sistema sobre o espaço quociente. Se a bisimulação tiver um número finito de classes de equivalência, então o sistema quociente é um autômato finito. Este autômato equivale a uma representação tempo-abstrato do comportamento contínuo do sistema híbrido. O sistema quociente, representado pelo autômato finito, preserva as propriedades de alcançabilidade do sistema híbrido original. Assim, um sistema híbrido pode ser transformado em um autômato finito, usando um bisimulação finita, sendo possível utilizar ferramentas padrões de verificação de propriedades para autômatos aos sistemas híbridos.

Cadeia de Caracteres

Também conhecida como *palavra* ou *sentença*, uma cadeia de caracteres sobre um alfabeto

é uma seqüência finita de símbolos (do alfabeto) justapostos. Normalmente, a cadeia *vazia* é representada pelo símbolo ε .

Classes de equivalência e Partições

Da teoria dos conjuntos, sabemos que para um dado conjunto B e um conjunto de subconjuntos A_1, A_2, \dots, A_n de B , os subconjuntos serão uma partição de B se e somente se:

1. $A_1 \cup A_2 \cup \dots \cup A_n = B$
2. $i \neq j \Rightarrow A_i \cap A_j = \emptyset$

Para definirmos os elementos que pertencem a cada um dos conjuntos, definimos relações $R \subseteq B \times B$. Se uma relação R é reflexiva, simétrica e transitiva, então temos uma *relação de equivalência*. Uma relação de equivalência $R \subseteq B \times B$ induz uma partição em B , na qual cada subconjunto A_i é formado por elementos equivalentes entre si e não equivalentes aos outros subconjuntos.

Condições iniciais

Condições que devem ser encontradas antes que a operação ou série de operações do sistema possam começar.

Corretude

Corretude é o conceito que diz, genericamente, que o sistema faz o que foi especificado para fazer.

Decidibilidade

Um problema é considerado decidível, se existe um procedimento computacional que decidirá em um número finito de etapas, se um sistema alcança um determinado estado ou satisfaz uma determinada propriedade. Um problema é indecidível, se não existe um algoritmo que tenha como entrada uma instância do problema e determine, se a resposta para instância é "sim" ou "não". Um procedimento é *semi-decidível* produzem a resposta correta, se eles terminam, mas sua terminação não é garantida.

Eventos

Representam o início ou fim das atividades em determinados instantes de tempo de um sistema, no qual não são consumidos tempo nem recursos. Portanto, são instantes de observação e de mudança de estado do sistema que inicia uma atividade que não estava em curso antes.

Impasse (*deadlock*)

Impasse ocorre quando, em um dado tempo, não há mais estados possíveis de serem alcançados a partir do estado corrente, ou seja, a execução pára, não existindo condições de continuação. Enfim, um impasse corresponde à ausência de estados, nos quais o sistema não pode mais evoluir.

Invariante

Um invariante é um tipo de condição que é sempre satisfeita para um conjunto de variáveis do sistema.

Especificação Formal

Especificação é o processo de descrever um sistema e suas propriedades desejadas. Especificação formal é uma especificação expressa em um linguagem, cujo vocabulário, sintaxe e semântica são formalmente definidos e têm uma base matemática.

As propriedades do sistema incluem o comportamento funcional, comportamento temporal, características de desempenho ou estrutura interna.

Estado

Um estado é uma instância da totalidade dos valores das variáveis no sistema, incluindo o fluxo de controle. Todas as operações no sistema são transições de um estado para outro.

Linguagens formais

As linguagens formais são representadas por alfabetos de símbolos (letras ou dígitos). É considerado um conjunto de cadeias formadas por símbolos pertencente a um alfabeto.

Lógica Temporal

É uma extensão da lógica convencional (proposicional) que incorpora operadores especiais para relacionar a validade das fórmulas lógicas à evolução do sistema. Com este tipo de lógica, pode-se especificar (ou verificar) como componentes, protocolos, objetos, módulos, procedimentos e funções se comportam ao longo do tempo. A especificação é feita com comandos de lógica temporal que fazem declarações sobre propriedades e relações no passado, presente e futuro. Existem dois tipos principais de lógica temporal: linear com tempo discreto (LTL) e com ramificações (CTL).

- **LTL - Lógica Temporal Linear**

Lógica temporal linear é uma linguagem de declarações sobre computações.

- **CTL - Lógica temporal ramificada**

É uma lógica temporal de ramificação sobre árvores computacionais.

Máquina de estado finito

Máquina de estado finito (MEF) é um modelo orientado, seqüencial para especificação funcional, consistindo de um conjunto de estados (Q), um conjunto de entradas (I), uma função de transição que especifica o próximo estado (δ), dado o estado corrente e entrada, uma função de saída que especifica ações associadas com transições (O), o estado inicial (Q_0) e o estado final. Pode ser denotado pela tupla $MEF = \langle Q, I, \delta, O, Q_0 \rangle$.

Métodos formais

Métodos formais referem-se à aplicação de métodos matemáticos rigorosos para especificação, desenvolvimento, verificação e teste de sistemas.

Modelo

Um modelo é uma réplica ou uma abstração das características essenciais de um sistema. Modelo, então, vem a ser uma representação simplificada de alguma parte da realidade, sendo a natureza do problema aquela que vai determinar que tipo de abstração será a mais apropriada.

Multi-conjunto

Multi-conjunto é semelhante a um conjunto, no entanto podem existir várias aparências do mesmo elemento em um multi-conjunto.

Definição formal: *Um multi-conjunto m , sobre um conjunto não-vazio A , é uma função $m : A \mapsto \mathbb{N}$. O inteiro não-negativo $m(x) \in \mathbb{N}$, é o número de aparências do elemento x no multi-conjunto m .*

Um multi-conjunto m é escrito freqüentemente como uma soma formal $\sum_{x \in A} m(x) \cdot x$. $m(x)$ é chamado de coeficiente de A . Um elemento $x \in A$ pertence a um multi-conjunto m se, e somente se, $m(x) \geq 0$. Exemplo: Um multi-conjunto $m_1 = \{a, c, c, e\}$. A soma formal de m_1 é $1'a + 2'c + 1'e$.

Operador lógico

Um operador lógico, também denominado *conectivo lógico*, é um operador sobre um conjunto de proposições. Podem ser unários ou binários e são definidos, conforme a seguinte lista:

- *Negação*: denotado pelos símbolos \sim ou \neg
- *E*: denotado pelos símbolos $\&$ ou \wedge
- *Ou*: denotado pelo símbolo \vee
- *Se-Então*: denotado pelos símbolos \rightarrow ou $::=$
- *Se-Somente-Se*: denotado pelos símbolos \leftrightarrow ou \Leftrightarrow

Poliedro

Um poliedro em \mathbb{R}^n é a região do espaço limitada por um conjunto finito de polígonos tal que cada aresta do polígono é compartilhada exatamente por um outro polígono (chamado de *polígono adjacente*). Os vértices e arestas do polígono são os vértices e arestas do poliedro e os polígonos são as faces do poliedro.

A fronteira de um poliedro é uma coleção finita de polígonos convexos limitados (as faces do polígono) tal que: (1) as faces se interceptam propriamente; (2) a vizinhança de cada ponto é topologicamente um disco(círculo) aberto e (3) a superfície é conexa.

Poliedros convexos

Poliedros convexos são chamados de *politopos*. Poliedros convexos podem ser caracterizados de duas maneiras:

1. Os ângulos diedrais são convexos, ou seja, menores que 180° . Um ângulo diedral com relação a uma aresta e de um poliedro P é o ângulo entre os planos formados pelas faces de P que compartilham e .
2. A soma dos ângulos das faces em cada vértice é menor que 360° .

Predicados

Também conhecida como *função de valor Booleano*, um predicado sobre um conjunto S é uma função P tal que para cada elemento $a \in S$, ou $P(a) = TRUE$ ou $P(a) = FALSE$, onde $TRUE$ e $FALSE$ são chamados de valores verdade. Predicados sobre um conjunto S são especificados como expressões, com valores verdadeiro ou falso, quando variáveis na expressão são trocadas por símbolos de S . Assim, tomando como exemplo, a expressão $x < 5$ especifica um predicado sobre \mathbb{N} , ou seja,

$$P(x) = \begin{cases} 1 & \text{se } x = 0, 1, 2, 3, 4 \\ 0 & \text{caso contrário.} \end{cases}$$

Proposição

Uma *proposição* é uma sentença declarativa que possui valor lógico *verdadeiro* ou *falso*.

Proposição atômica

É uma proposição que não contém operadores lógicos em sua sentença.

Prova de teoremas

Prova de teoremas é o processo de encontrar uma prova de uma propriedade a partir de axiomas do sistema. Etapas na prova apelam para os axiomas e regras e, possivelmente, definições derivadas e lemas intermediários. Ao contrário da técnica de verificação automática de modelos (*model checking*), prova de teorema pode lidar diretamente com espaços de estado infinitos. Consiste de técnicas como indução estrutural para provar sobre domínios infinitos.

Regra de produção

Uma regra de produção (α, β) é representada por $\alpha ::= \beta$ e define as condições de geração das palavras de uma linguagem.

Retroação

Retroação (feedback) é a influência que as saídas do sistema exercem sobre as suas entradas no sentido de ajustá-las ou regulá-las ao funcionamento do sistema.

Sistema

Um sistema é um conjunto de elementos dinamicamente relacionados entre si, formando uma atividade para atingir um objetivo, operando sobre entradas (informação, energia ou matéria) e fornecendo saídas (informação, energia ou matéria) processadas.

Sistemas Chaveados

São sistemas dinâmicos híbridos que consistem de uma família de subsistemas contínuos ou discretos e uma regra que determina o chaveamento entre eles. O comportamento de chaveamento destes sistemas podem ser gerados pela mudança da dinâmica em diferentes regiões de operação.

Sistemas Contínuos

São sistemas nos quais as variáveis de estado mudam continuamente no tempo.

Sistemas a eventos discretos

Um sistema a evento discreto (SED) é um sistema dinâmico, cuja mudança de estado ocorre em pontos discretos no tempo, em decorrência de eventos isolados.

Sistemas híbridos

Sistemas híbridos são sistemas que apresentam comportamento constituído por dinâmicas contínuas e discretas interagindo entre si.

Sistema de Transição Quociente (STQ)

São sistemas simplificados derivados do sistema original, agregando os estados de maneira apropriada, segundo um critério determinado pelo projetista. Um STQ é considerado uma aproximação conservativa no sentido de que, para toda trajetória no sistema híbrido original, há uma trajetória no STQ correspondente ao conjunto de estados da trajetória, no sistema híbrido original.

Sistemas Zeno

São sistemas híbridos que exibem infinitamente muitos chaveamentos em um intervalo finito de tempo.

Sub-aproximação

É um tipo de aproximação interna do espaço de estados alcançável, ou seja, o conjunto aproximado resultante está contido no conjunto alcançável.

Super-aproximação

É um tipo de aproximação externa do espaço de estados alcançável, ou seja, o conjunto aproximado resultante contém o conjunto alcançável.

Teoria de Controle Supervisório

Desenvolvida por Ramadge e Wonham (1989), baseia-se na modelagem do sistema em malha aberta (planta) e na especificação do comportamento desejado por linguagens controláveis. O supervisor tem a função de exercer uma ação de controle restritiva sobre a planta, de modo

a restringir seu comportamento conforme especificado. Esta abordagem permite a síntese automática de controladores de forma mais independente da experiência do projetista. Além disso, quando a especificação não pode ser cumprida, tem-se a garantia de construir um supervisor que satisfaz parte do comportamento desejado de forma minimamente restritiva.

Tupla

É uma lista, consistindo de um conjunto ordenado de elementos. A ordem, na qual aparecem os elementos de um tupla, é relevante e seus elementos não precisam ser distintos. Por exemplo, uma n -tupla: $\langle a_1, \dots, a_n \rangle$ é uma tupla com n elementos, $\langle 4, 1, 4, 2 \rangle$ é uma 4-tupla. Uma 2-tupla é chamada de um par ordenado e uma 3-tupla é chamada de *tripla ordenada*.

Verificação

Verificação é a tarefa de determinar se o sistema foi construído de acordo com sua especificação. O processo de verificação consiste na avaliação da conformidade de um sistema frente a uma dada especificação ou requerimento. Quando realizado com base em uma notação matemática, este processo é denominado *verificação formal*.

Verificação automática de modelos (*Model Checking*)

É uma técnica utilizada na verificação de sistemas com comportamento finito, e verificar se uma propriedade desejada é satisfeita naquele modelo. É utilizado para determinar automaticamente, se as propriedades são satisfeitas pelo modelo.

Verificação simbólica de modelos (*Symbolic Model Checking*)

Técnica que utiliza cálculo de ponto fixo sobre uma relação de transição de estado para determinar a alcançabilidade de um dado estado. Este método utiliza uma representação simbólica da relação de transição, como, por exemplo, expressão booleana, através de BDD's (diagramas de decisão binária), e restrições lineares, representadas por poliedros convexos.

Vivacidade

É a situação na qual o sistema fica indefinidamente alternando entre estados não terminais.

Validação

Validação é o ato de verificação para ter certeza de que um sistema faz o que lhe é esperado para ter certeza da corretude do sistema.

Bibliografia

ALUR, R.; COURCOUBETIS, C.; HALBWACHS, N.; HENZINGER, T.A.; HO, P.-H.; NICOLLIN, X.; OLIVERO, A.; SIFAKIS, J.; YOVINE, S. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, v. 138, p. 3–34, 1995.

ALUR, R.; COURCOUBETIS, C.; HENZINGER, T.A.; HO, P.-H. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: GROSSMAN, R. et al. (Ed.). *Hybrid Systems I*. [S.l.]: Springer-Verlag, 1993, (Lecture Notes in Computer Science 736). p. 209–229.

ALUR, R.; DILL, D. The theory of timed automata. *Theoretical and Computer Science*, v. 126, n. 2, p. 183–235, 1994.

ALUR, R.; DILL, D. L. A theory of timed automata. *Theoretical Computer Science*, v. 126, p. 183–235, 1994.

ALUR, R.; HENZINGER, T.; HO, P.-H. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, p. 181–201, 1996.

ANTSAKLIS, P.; LEMMON, M. D.; STIVER, J. A. *Hybrid Systems II*. [S.l.]: Springer-Verlag, 1995. (Lecture Notes in Computer Science).

ANTSAKLIS, P. A brief introduction to the theory and applications of hybrid systems. *Proceedings of the IEEE*, v. 88, n. 7, p. 879–887, jul 2000.

ANTSAKLIS, P.; STIVER, J. A.; LEMMON, M. D. Hybrid systems modeling and autonomous control systems. *Hybrid Systems*, Springer-Verlag, p. 266–285, dez 1993. New York.

ASARIN, E.; BOURNEZ, O.; DANG, T.; MALER, O. Approximate reachability analysis of piecewise-linear dynamical systems. *Hybrid Systems: Computation and Control*, Springer-Verlag, p. 20–31, 2001.

ASARIN, E.; DANG, T.; MALER, O. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE. Special Issue "Hybrid System: Theory & Applications"*, v. 88, n. 7, p. 1011–1025, jul 2000.

ASARIN, E.; DANG, T.; MALER, O. d/dt: A verification tool for hybrid systems. *Proceedings of the Computer Aided Verification*, Springer-Verlag, p. 365–370, jul 2002. Copenhagen, Denmark. (Lecture Notes in Computer Science 2404)

ASARIN, E.; MALER, O.; PNUELI, A.; SIFAKIS, J. Controller synthesis for timed automata. *IFAC Symposium on System Structure and Control*, p. 469–474, 1998.

KURZANSKI, A. B.; VARAIYA, P. Ellipsoidal techniques for reachability analysis. In *Hybrid Systems: Computation and Control*, Springer Verlag, v. 1790, p. 202–214, jan 2000.

BACK, J. G. A.; MYERS, M. A dynamical simulation facility for hybrid systems. *Hybrid Systems*, Springer-Verlag, p. 255–267, dez 1993. New York.

BAIL, J. L.; ALLA, H.; DAVID, R. Hybrid petri nets. In *Proceedings of the 1st International European Control Conference, ECC91*, p. 1472–1477, 1991.

BAINOV, D. D.; SIMEONOV, P. S. *Systems with impulse effect*. [S.I.]: Ellis Horwood, 1989. Chichester, England.

BARNES, S. A.; LEFLEY, P. W. Finite state machine brings high frequency, adaptative control to power electronics applications. in *Proc. IEEE Control*, p. 974–979, 1999.

BENGTSSON, J.; LARSEN, K. G.; LARSSON, F.; PETTERSSON, P.; WANG, Y.; WEISE, C. New generation of uppaal. In *Proceedings of International Workshop on Software Tools for Technology Transfer*, p. 43–52, jul 1998. Aalborg, Denmark.

BENVENISTE, A. Compositional and uniform modelling of hybrid systems. In: *Hybrid Systems III. Verification and Control*. [S.I.]: Springer-Verlag, 1996, (Lecture Notes in Computer Science, 1066). p. 41–51.

BENVENISTE, A.; BERRY, G. Special issue on the synchronous approach to reactive and real-time systems. In: *Proceedings of the IEEE*. [S.I.]: Springer-Verlag, 1991, (Lecture Notes in Computer Science, v. 9).

BONIFÁCIO, A. L.; MOURA, A. V. Modeling and parameters synthesis for an air traffic management system. *3rd International Conference on Formal Methods in Computer Aided Design (FMCAD'00)*, p. 316–334, 2000.

- BOSE, B. K. Recent advances in power electronics. *IEEE Transactions on Power Electronics*, v. 7, n. 1, p. 2–10, 1992.
- BOTCHKAREV, O. Ellipsoidal techniques for verification of hybrid systems. URL: <http://robotics.eecs.berkeley.edu/olegb/VeriSHIFT/>, jan 2000.
- BOTCHKAREV, O.; TRIPAKIS, S. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. *HSCC*, Springer, v. 1790, p. 73–88, mar 2000.
- BRANICKY, M. S. Stability of switched and hybrid systems. *Proc. IEEE Conf. Decision and Control*, p. 3498–3503, dez 1994. Lake Buena Vista, FL.
- BRANICKY, M. S. General hybrid dynamical systems: Modeling, analysis and control. *Hybrid Systems III - Verification and Control*, Springer-Verlag (Lecture Notes in Computer Science 1066), p. 186–200. 1996.
- BRANICKY, M. S. Topology of hybrid systems. *Proc. IEEE Conf. Decision and Control*, p. 2309–2314, dez 1996. San Antonio.
- BROCKETT, R. W. Hybrid models for motion control systems. *Essays in Control: Perspectives in the Theory and its Applications*, p. 29–53, 1993.
- CAINES, P. E.; WEI, Y.-J. The hierarchical lattices of a finite machine. *Systems and Control Letters*, v. 25, p. 257–263, 1995.
- CHUTINAN, A.; KROGH, B. Computing polyhedral approximations to flow pipes for dynamic systems. In *The 37th IEEE Conference on Decision and Control: Session on Synthesis and Verification of Hybrid Control Laws (TM-01)*, 1998.
- CHUTINAN, A.; KROGH, B. Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations. In *Hybrid systems: Computation and Control, Second International Workshop*, Springer-Verlag, p. 76–90, 1999.
- CHUTINAN, A.; KROGH, B. H. Checkmate verification tool. <http://www.ece.cmu.edu/krogh/CheckMate/verify/content.html>, 2000.
- CHUTINAN, A.; KROGH, B. H. Computing approximating automata for a class of hybrid systems. *Mathematical and Computer Modeling of Dynamical Systems*, v. 6, n. 1, p. 30–50, mar 2000.
- CLARKE, E. M.; GRUMBERG, O.; JHA, S.; LU, Y.; VEITH, H. Progress on the state explosion problem in model checking. *Informatics*, Springer-Verlag, p. 176–194, 2001.

CLARKE, E. M.; WING, J. M. Formal methods: State of the art and future directions. *ACM Workshop on Strategic Directions in Computing Research, ACM Computing Surveys*, v. 28, n. 4, p. 626–643, dez 1996.

COFER, D. D. Affordable implementation of reliable avionics systems. *IEEE Proceedings of the 36th Conference on Decision and Control*, p. 2423–2428, dez 1997. San Diego, California, USA.

CURY, J. E. R.; KROGH, B. H. Synthesizing supervisory controllers for hybrid systems. *Journal of the Society of Instrument and Control Engineers (SICE)*, v. 38, n. 3, p. 161–168, 1999.

CURY, J. E. R.; KROGH, B. H.; NIINOMI, T. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *in Proceedings of the 34th IEEE Conference on Decision and Control*, p. 141–146, 1995.

DANG, T. *Verification and Synthesis of Hybrid Systems*. Tese (Doutorado) — Institut National Polytechnique de Grenoble, VERIMAG, out 2000.

DAVID, R.; ALLA, H. Petri nets for modelling of dynamic systems - a survey. *Automatica - A Journal of IFAC - The International Federation of Automatic Control*, Pengamon Ed., v. 30, p. 175–202, feb 1994.

DAVIS, M. H. A. *Markov Models and Optimization*. [S.l.]: Chapman and Hall, 1993. London, England.

DAVOREN, J. M.; NERODE, A. Logics for hybrid systems. *Proceedings of the IEEE*, v. 88, n. 7, p. 985–1010, jul 2000.

DEMONGODIN, I.; KOUSSOULAS, N. T. Differential petri nets: Representing continuous systems in a discrete-event world. *IEEE Transactions on Automatic Control*, v. 44, n. 3, p. 573–578, abr 1998.

DESHPANDE, A.; GOLLU, A.; SEMENZATO, L. The shift programming language and run-time system for dynamic networks of hybrid automata. *Technical Report - Department of Electrical Engineering and Computer Sciences*, 1995. University of California - Berkeley - USA.

ENGELL, S.; KOWALEWSKI, S.; SCHULZ, C.; STURSBERG, O. Continuous-discrete interactions in chemical processing plants. *Proceedings of the IEEE, Special on Hybrid Systems: Theory and Applications*, v. 88, n. 7, p. 1050–1068, jul 2000.

ESPINOZA, J.; JO6S, G. Power converter system simulation using high level languages. *IEEE Power Electronics Society*, p. 79–84, ago 1994. Québec, Canada.

EZZIN, J.; HADDAD, A. H. Controllability and observability of hybrid systems. *International Journal of Control*, v. 49, n. 6, p. 2045–2055, jun 1989.

FRANZ, G. A. Multilevel simulation tools for power converters. *IEEE APEC'90*, p. 629–633, 1990.

GIUA, A.; USAI, E. High-level hybrid petri nets: A definition. In *Proceedings of the 35th Conference on Decision and Control*, p. 148–150, dez, 1996. Kobe, Japan.

GROSSMAN, R. L.; NERODE, A.; RAVN, A.; RISCHER, H. Hybrid systems - introduction. *Hybrid Systems*, Springer-Verlag, p. 266–285, dez 1993. New York.

HANSELMAN, D.; LITTLEFIELD, B. *MATLAB - User's Guide - Version 5.05*. [S.l.]: Prentice Hall, Inc., 1997. New Jersey.

HE, K.; LEMMON, M. Modeling hybrid control systems using programmable timed petri nets. *European Journal of Automation, APII-JESA, Journal europeen des systemes automatisees*, v. 32, p. 1187–1208, 1998.

HENZINGER, T. The theory of hybrid automata. In: *Proceedings of the 11th Annual Symposium on Logic in Computer Science*. [S.l.]: IEEE Computer Society Press, 1996. p. 278–292.

HENZINGER, T. Discrete abstractions of hybrid systems. In: *Proceedings IEEE*. [S.l.]: IEEE Computer Society Press, 2000. v. 88, p. 971–984.

HENZINGER, T.; HO, P.-H.; WONG-TOI, H. A user guide to HYTECH. In: BRINKSMA, E. et al. (Ed.). *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*. [S.l.]: Springer-Verlag, 1995, (Lecture Notes in Computer Science 1019). p. 41–71.

HENZINGER, T.; HO, P.-H.; WONG-TOI, H. HYTECH: a model checker for hybrid systems. In: GRUMBERG, O. (Ed.). *CAV 97: Computer-aided Verification*. [S.l.]: Springer-Verlag, 1997, (Lecture Notes in Computer Science 1254). p. 460–463.

HENZINGER, T.; KOPKE, P. W.; PURI, A.; VARAIYA, P. What's decidable about hybrid automata? In: *Proceedings of the 27th Annual Symposium on Theory of Computing*. [S.l.]: ACM Press, 1995. p. 373–382.

HERRMANN, P.; GRAW, G.; KRUMM, H. Compositional specification and structured verification of hybrid systems in CTLA. *Proceedings of the 1st IEEE International Symposium*

on *Object-Oriented Real-Time Distributed Computing*, IEEE Computer Society Press, p. 335–340, 1998.

HERRMANN, P.; KRUMM, H. Specification of hybrid systems in CTLA+. *Proceedings of the 5th International Workshop on Parallel & Distributed Real-Time Systems - (WPDRTS'97)*, IEEE Computer Society Press, p. 212–216, out 1997.

HERRMANN, P.; KRUMM, H. Formal hazard analysis of hybrid systems in ctna. *18th IEEE Symposium on Reliable Distributed Systems (SRDS'99)*, IEEE Computer Society Press, p. 68–77, out 1999.

HO, P.-H. *Automatic Analysis of Hybrid Systems*. Tese (Ph. D. thesis) — Cornell University, Faculty of the Graduate School of Cornell University, ago 1995.

JI, Y.; CHIZECK, H. J. Controllability, stabilizability and continuous-time markovian jump linear quadratic control. *IEEE Transactions on Automatic Control*, v. 35, n. 7, p. 777–788, 1990.

JOHNSON, T. L. Analytic models of multistage processes. *Proceeding of the IEEE Conference on Decision and Control*, dez 1981. San Diego.

JOSEPH, M. *Formal Techniques in Real-Time and Fault-Tolerant Systems*, Springer-Verlag, v. 331, 1989. New York.

KARNOPP, D. C.; MARGOLIS, D. L.; ROSENBERG, R. C. *System Dynamics - A Unified Approach (2nd Edition)*. [S.I.]: J. Wiley, 1990. New York - NY.

KARNOPP, D. C.; ROSENBERG, R. C. *System Dynamics - A Unified Approach*. [S.I.]: J. Wiley, 1975. New York - NY.

KASSAKIAN, J. G.; SCHLECHT, M. F.; VERGHESE, G. C. *Principles of Power Electronics*. [S.I.]: Addison-Wesley Publishing Company, 1992. (Electrical Engineering).

KLOAS, M.; FRIESEN, V.; SIMONS, M. Smile - a simulation environment for energy systems. *Proceeding of the 5th International IMACS - Symposium on Systems Analysis and Simulation (SAS'95)*, v. 18-19, p. 503–506, 1995.

KOUTSOUKOS, X. D.; ANTSAKLIS, P. J. Computational issues in intelligent control: Discrete-event and hybrid systems. *Soft Computing and Intelligent Systems: Theory and Practice*, Academic Press, 1999.

KOWALEWSKI, S. Introduction to the analysis and verification of hybrid systems. in *Proc. Modelling, Analysis and Design of Hybrid Systems*, Springer-Verlag, v. 279, p. 153–171, 2002.

KULKARNI, S. R.; RAMADGE, P. J. On the existence and complexity of convergent on-line decision rules. *In Proceedings of the IEEE 34th Conference on Decision and Control*, p. 3022-3027, dez 1995.

LAFFERRIERE, G. A. Discontinuous stabilizing feedback using partially defined lyapunov functions. *in Proceedings of the IEEE Conference on Decision and Control*, p. 3487-3491, dez 1994.

LAMPORT, L. Hybrid systems in TLA+. In: GROSSMAN A. NERODE, A. P. R. R.; RISCHEL, H. (Ed.). *Proceedings of Hybrid Systems*. [S.l.]: Springer, 1993. (Lecture Notes in Computer Science), p. 36-59. Berlim, German.

LEE, F. C. Power electronics research - a paradigma shift. *COBEP 2001 - 6th Brazilian Power Electronics Conference*, nov 2001. Brasil.

LEFLEY, S. A. B. P. W.; SMITH, G. A. The finite state machine for control of high-frequency power electronics. *in Power Engineering Journal*, p. 305-311, dez 2000.

LEMMON, M. D.; HE, K. X.; MARKOVSKY, I. Supervisory hybrid systems. *IEEE Control Systems*, v. 4, n. 19, p. 42-55, jun 1999.

LENNARTSON, B. E. B.; TITTUS, M. Hybrid systems in process control. *In Proc. of the 33rd CDC*, p. 3587-3592, 1994. Orlando, FL, USA.

LIBERZON, D.; MORSE, A. S. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine*, v. 5, n. 19, p. 59-70, 1999.

MACMILLAN, K. *Symbolic Model Checking: An Approach to the State Explosion Problem*. Tese (Doutorado) — Carnegie Mellon University, Pittsburgh, PA, USA, maio 1992.

MALER, O. Control from computer science. *IFAC - 5th Symposium Nonlinear Control (NOLCOS '2001)*, p. 357-365, jul 2001. Saint-Petersburg, Russia.

MALER, O.; MANNA, Z.; PNUELI, A. From timed to hybrid systems. In: BAKKER, J. W. de et al. (Ed.). *Real Time: Theory in Practice*. [S.l.]: Springer-Verlag, 1992, (Lecture Notes in Computer Science 600). p. 447-484.

MANNA, Z.; PNUELI, A. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. New York: Springer Verlag, 1992.

MANNA, Z.; PNUELI, A. Verifying hybrid systems. *Hybrid Systems*, p. 4-35, 1993. Reims, France.

MATHWORKS. The mathworks inc. <http://www.mathworks.com>, 2002.

MATHWORKS, I. T. *Stateflow: For Use With Simulink, User's Guide*. Version 1. [S.l.], 1998.

MENEZES, P. B. *Linguagens Formais e Autômatos*. 3a. edição. ed. [S.l.]: Editora Sagra Luzzato, 2000. ISBN 85-241-0554-2.

MIRANDA, M. V. C.; LIMA, A. M. N. Using hibrid automata to model power electronic circuits. *COBEP'2001 - 6th Brazilian Power Electronics Conference*, p. 1-6, nov 2001. Florianopolis, SC, Brasil.

MOHAN, N.; UNDELAND, T. M.; ROBBINS, W. P. *Power Electronics - Converters, Applications and Design*. [S.l.]: Jon Wiley and Sons, Inc., 1995. New York, USA.

MORSE, A. S. *Supervisory control of families of linear set-point controllers - Part 1: Exact matching*. [S.l.]: Preprint, 1993.

MORSE, A. S. *Supervisory control of families of linear set-point controllers - Part 2: Robustness*. [S.l.]: Preprint, 1994.

MOSTERMAN, P. J.; BISWAS, G. Modeling discontinuous behavior with hybrid bond graphs. In: *Proceedings of the 9th International Workshop on Qualitative Reasoning about Physical Systems*. [S.l.: s.n.], 1995. p. 139-147. Amsterdam, The Netherlands.

MOSTERMAN, P. J.; BISWAS, G. A formal hybrid modeling scheme for handling discontinuities in physical system models. *Proceeding of AAAI*, p. 985-990, ago 1996. Portland, Oregon.

MOSTERMAN, P. J.; BISWAS, G. Model based diagnosis of dynamic systems. *Seventh Journees du L.I.P.N.*, Pengamon Ed., p. 143-154, set 1997. University of Paris-Nord, Villetaneuse, France.

MOSTERMAN, P. J.; BISWAS, G. A theory of dscontinuities in dynamc physical systems. *Journal of the Franklin Institute*, v. 334B, n. 6, 1997.

Sriram Narasimhan and Gautam Biswas and Gabor Karsai and Tal Pasternak and Feng Zhao
NARASIMHAN, S.; BISWAS, G.; KARSAI, G.; PASTERNAK, T.; ZHAO, F. Building observers to address fault isolation and control problems in hybrid dynamic systems. *International Conference on Systems, Man and Cybernetics*, p. 2393-2398, 2000. Nashville, TN.

NERODE, A.; KOHN, W. Models for hybrid systems: Automata, topologies, stability. *Hybrid Systems*, Springer-Verlag, p. 2309-2314, dez 1993. New York.

NICOLLIN, X.; SIFAKIS, J.; YOVINE, S. From atp to timed graphs and hybrid systems. In: BAKKER, J. W. de et al. (Ed.). *Real Time: Theory in Practice*. [S.l.]: Springer-Verlag, 1992, (Lecture Notes in Computer Science 600). p. 549–572.

PAVLIDIS, T. Stability of systems described by differential equations containing impulses. *Proceeding of IEEE Transactions on Automatic Control*, v. 12, n. 1, p. 43–45, 1967.

PELETIES, P.; DECARLO, R. Asymptotic stability of m-switched systems using lyapunov-like functions. In *Proceedings of the American Control Conference*, p. 1679–1684, jun 1991.

PEPYNE, D. L.; CASSANDRAS, C. G. Optimal control of hybrid systems in manufacturing. *Proceeding of the IEEE, Special Issue on Hybrid Systems: Theory and Applications*, v. 88, n. 7, p. 1108–1123, jul 2000.

PETERS, J. F.; PEDRYCZ, W. *Engenharia de Software*. [S.l.]: Editora Campus, 2001. ISBN 85-352-0746-5.

PETTERSSON, S. *Analysis and Design of Hybrid Systems*. Tese (Doutorado) — Chalmers University of Technology, Department of Signals and Systems, 1999.

PIQUET, H.; FOCH, H.; DEMNI, H. E. Une méthode de conception en électronique de puissance, à l'aide d'éléments caractérisés. *EPF'2000 - 8th Colloque Electronique de Puissance du Futur*, nov 2000. Lille, France.

PURI, A.; VARAIYA, P. Verification of hybrid systems using abstraction. In *Hybrid Systems II: Computation and Control*, A. Nerode, P. Antsaklis, W. Kohn and S. Sastry, v. 999, p. 359–369, jan 1995.

PURI, A.; VARAIYA, P. Decidability of hybrid systems with rectangular differential inclusions. In: *Computer Aided Verification, CAV'94*. [S.l.]: Springer-Verlag, 1994, (Lecture Notes in Computer Science 944). p. 54–104.

RAISCH, J.; O'YOUNG, S. D. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control, Special Issue on Hybrid Control Systems*, v. 43, n. 4, p. 569–573, 1998.

RAMADGE, P. J. G.; WONHAM, W. M. The control of discrete event systems. *Proceedings of the IEEE*, v. 77, n. 1, p. 81–98, jan 1989.

ROCH, S. Extended computation tree logic. In: *Proceedings of the Workshop on Concurrency, Specification and Programming*. [S.l.], p. 75–80, out 2000. Berlim, German.

- ROLIM, J. D. P. *Programas para Simulação Digital de Conversores Estáticos*. Dissertação (Mestrado) — Universidade Federal da Paraíba, COPELE, 1999. Paraíba - Brasil.
- SAIEDIAN, H. An invitation to formal methods. *IEEE Computer*, v. 29, n. 4, p. 16–30, abr 1996.
- SIEGELMAN, H. T.; SONTAG, E. D. Turing computation with neural nets. *Applied Math. Letters*, v. 4, p. 77–80, 1991.
- SILVA, A. R. J.; MARTINS, D. C. Numerical simulation of power static converters: Diagonalization method of the dynamical matrix. in *Proceeding of IEE Power Electronics and Variable Speed Drives*, v. 429, p. 508–512, set 1996.
- Silva, B. I. and Stursberg, O. and Krogh, B. H. and Engell, S. SILVA, B. I.; STURSBURG, O.; KROGH, B. H.; ENGELL, S. An assessment of the current status of algorithmic approaches to the verification of hybrid systems. *Proceedings on the 40th IEEE Conference on Decision and Control*, p. 2867–2874, 2001.
- SILVA, E. R. C. da. Eletrônica de potência. *Apostila do Departamento de Engenharia Elétrica*, abr 1999. Univ. Federal da Paraíba, CCT.
- SOMMERVILLE, I. *Engenharia de Software*. Sexta edição. [S.l.]: Addison Wesley, 2003. ISBN 85-88639-07-6.
- STIVER, J. A.; ANTSAKLIS, P. J.; LEMMON, M. D. A logical des approach to the design of hybrid control systems. *Math. and Computer Modeling, Special Issue on Discrete Event Systems*, p. 55–76, jun 1996.
- STROMBERG, J. E. *A mode switching modelling philosophy*. Tese (Doutorado) — Department of Electrical Engineering - Linkoping University, Linkoping - Sweden, 1994.
- STROMBERG, J. T. J. E.; SODERMAN, U. Variable causality in bond graphs caused by discrete effects. *Proceeding of the International Conference in Bond Graphs Caused by Discrete Effects*, p. 115–119, 1993. San Diego, CA.
- STURSBURG, O.; KOWALEWSKI, S. Approximating switched continuous systems by rectangular automata. *Proc. 5th European Control Conference*, set 1999. Karlsruhe, Germany.
- TABUADA, P.; PAPPAS, G. J. Model checking LTL over controllable linear systems is decidable. *Hybrid Systems : Computation and Control*, Springer-Verlag, v. 2623, abr 2003. (Lecture Notes in Computer Science). Prague, Czech Republic.

TAVERNINI, L. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods, and Applications*, v. 11, p. 665–683, jun 1987.

TITTUS, M.; LENNARTSON, B. Hierarchical supervisory control for batch processes. In *Proceeding on the 37th IEEE Conference on Decision and Control*, p. 3251–3255, 1998. Tampa, FL, USA.

TOMLIN, J. L. C.; SASTRY, S. Controllers for reachability specifications fo hybrid systems. *Automatica*, v. 35, n. 3, mar 1999.

VARAIYA, P. Reach set computation using optimal control. In *Proceeding KIT Workshop*, p. 377–383, 1998.

VYATKIN, V.; HANISCH, H.-M. Practice of modeling and verification of distributed controllers using signal net systems. *Proceedings on the Workshop on Concurrency, Specification and Programming*, p. 335–350, Humboldt-University, out 2000. Berlin, German.

VYATKIN, V.; HANISHCH, H.-M. Application of visual specifications for verification of distributed controllers. *Proceedings of the IEEE Systems, Man, and Cybernetic Conference*, p. 646–651, 2001.

WIMPEY, D. G. *Finite-State control of discrete-time continuous processes: an automata motivated approach*. Tese (Ph. D. thesis) — Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, jun 1982.

WITSENHAUSEN, H. S. A class of hybrid-state continuous-time dynamic systems. *Proceeding of IEEE Transactions on Automatic Control*, v. 11, n. 2, p. 161–167, 1966.

WONG-TOI, H. The synthesis of controllers for linear hybrid automata. In: *Proceedings of the 36th IEEE Conference on Decision and Control*. [S.I.]: IEEE Computer Society Press, 1997. p. 4607–4612. San Diego, CA.

YOURDON, E. *Análise Estruturada Moderna*. Décima edição - tradução da terceira edição americana. [S.I.]: Editora Campus, 1990. ISBN 85-7001-615-8.

YOVINE, S. A verification tool for real-time systems. *Systems Tools for Technology Transfer*, v. 1, n. 1, p. 123–133, 1997.