

Jefferson Dantas Pinheiro Rolim

Programas Para Simulação Digital de Conversores Estáticos

Dissertação Submetida à Coordenação do Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Paraíba – Campus II como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica.

ÁREA DE CONCENTRAÇÃO: PROCESSAMENTO DA ENERGIA

Cursino Brandão Jacobina, Dr. Ing.

Orientador

Antonio Marcus Nogueira Lima, Dr.

Orientador

Campina Grande, Paraíba, Brasil

©Jefferson Dantas Pinheiro Rolim, Abril de 1999



R748p Rolim, Jefferson Dantas Pinheiro
Programas para simulacao digital de conversores estaticos / Jefferson Dantas Pinheiro Rolim. - Campina Grande, 1999.
108 f. : il.

Dissertacao (Mestrado em Engenharia Eletrica) - Universidade Federal da Paraiba, Centro de Ciencias e Tecnologia.

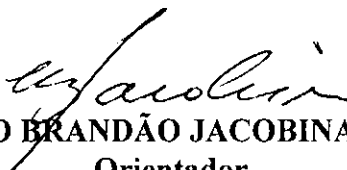
1. Engenharia Eletrica 2. Processamento de Energia 3. Conversores Estaticos 4. Simulacao Digital 5. Dissertacao I. Jacobina, Cursino Brandao, Dr. II. Lima, Antonio Marcus Nogueira, Dr. III. Universidade Federal da Paraiba - Campina Grande (PB) IV. Título

CDU 681.3.06(043)

PROGRAMAS PARA SIMULAÇÃO DIGITAL DE CONVERSORES ESTÁTICOS

JEFFERSON DANTAS PINHEIRO ROLIM

Dissertação Aprovada em 12.04.1999



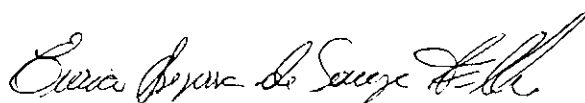
PROF. CURSINO BRANDÃO JACOBINA, Dr.Ing., UFPB
Orientador



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



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



PROF. EURICO BEZERRA DE SOUZA FILHO, D.Sc., UFPE
Componente da Banca

CAMPINA GRANDE - PB
Abril - 1999

Dedicatória

Aos meus pais João e Zuleide, pela dedicação, amor e incentivo.

Agradecimentos

A Deus, que durante todos os dias de minha vida tem iluminado os meus caminhos.

Aos meus pais João e Zuleide, e aos meus irmãos Eduardo(in memorian), Danielle e Daniel, que me apoiaram desde o início da minha trajetória acadêmica, dando a mim forças para alcançar as metas desejadas.

A minha namorada, por sua compreensão, apoio e carinho.

Ao CNPq pelo financiamento de uma parte deste trabalho.

Aos professores Cursino Brandão Jacobina, Antônio Marcus Nogueira Lima, pela orientação e colaboração sem as quais não seria possível a realização deste trabalho.

Ao professores Edson Roberto Cabral da Silva e Eurico Bezerra de Sousa Filho, pelo apoio dado durante a realização deste trabalho.

Aos amigos Alexandre Cunha, Luiz Antônio, Talvanes Meneses e Fabiano Salvadori pelo apoio e sugestões dados durante a realização deste trabalho.

Resumo

Neste trabalho, apresentam-se programas para a simulação de conversores estáticos. Na implementação destes programas avaliam-se diferentes metodologias de simulação e diferentes modelos de representação para os interruptores de potência. No programa SICON1 o conversor é modelado através de um conjunto de equações algébricas, sendo possível dois tipos de topologia para a simulação do conversor, uma topologia fixa (a configuração do conversor permanece a mesma e os parâmetros que representam os dispositivos semicondutores variam) e uma topologia variável (a configuração do conversor muda de acordo com os estados aberto e fechado dos semicondutores). No programa SICON2 o conversor é modelado através de equações de estado e usa-se uma topologia fixa na qual o período de amostragem é variável. São apresentados diversos resultados obtidos a partir do uso do SICON1 e do SICON2.

Abstract

On this working, there are programs for a simulation of static converters. On the implementation of these programs, different kinds of methodologies of simulation and different models of representation for the interrupters of power are analysed. On the program SICON1, the converter is modeled by a set of algebraic equations, being possible two kinds of topology for the simulation of converter: a fixed topology (the configuration of the converter remains the same and the parameters that represent the semiconductor device varies), and a variable topology (the configuration of the converter changes according to the open and close semiconductor state). On the program SICON2, the converter is modeled by the state equations and uses a fixed topology in that the sampling time is variable. A lot of results obtained from the use of SICON1 and SICON2 are presented.

Sumário

1	Introdução Geral	1
1.1	Organização do Trabalho	4
2	Estado da Arte	5
2.1	Introdução	5
2.2	Simulação em Eletrônica de Potência	5
2.2.1	Variáveis de Estado	5
2.2.2	Análise Nodal	6
2.2.3	Análise Nodal Modificada	7
2.3	Conclusão	8
3	Análise de Circuitos	9
3.1	Introdução	9
3.2	Análise de Nós de Redes Lineares Invariantes	9
3.2.1	Análise de redes resistivas	11
3.3	Resolução de Circuitos Através de Equações de Estado	13
3.3.1	Análise de Percurso Fechado	14
3.3.2	Análise de Cortes	15
3.3.3	Equações de Estado Para Redes Lineares Invariantes	17
3.4	Conclusão	21
4	Programa SICON1	22
4.1	Introdução	22
4.2	Os Modelos dos Semicondutores	22
4.2.1	A Topologia Fixa	22
4.2.2	A Topologia Variável	23
4.2.3	O Chaveamento dos Semicondutores	23
4.3	Modelos de Capacitores e Indutores	24
4.4	Utilização dos Modelos	27

4.5	Escolha do Período de amostragem	28
4.6	Estrutura do Programa	29
4.6.1	Entrada de Dados	29
4.6.2	Fluxograma do Programa	32
4.7	Resultados de Simulação	33
4.7.1	Conversor Buck	33
4.7.2	Retificador Trifásico Não Controlado	35
4.7.3	Inversor Trifásico	36
4.8	Conclusão	41
5	Programa SICON2	42
5.1	Introdução	42
5.2	Estrutura do Programa SICON2	42
5.2.1	Entrada de Dados	42
5.2.2	A modelagem dos dispositivos semicondutores	43
5.2.3	Solução da equação de estado contínua	43
5.2.4	A escolha do período de amostragem	43
5.2.5	Interpolação	46
5.2.6	Procedimento Para Montar a Matriz F	47
5.2.7	Exemplo de Montagem da Matriz F	47
5.2.8	Fluxograma do Programa	48
5.3	Resultados de Simulação	49
5.3.1	Conversor QRC Com Comutação a Corrente Nula (QRC-ZCS)	49
5.3.2	Conversores CC/CC PWM Soft-Single-Switched (SSS)	51
5.3.3	Inversor Monofásico em Ponte	62
5.3.4	Retificador Monofásico em Ponte	64
5.4	Conclusão	67
6	Análise Comparativa	68
6.1	Introdução	68
6.2	Conversor CC-CC	68
6.2.1	Resultados Obtidos Com o Programa SICON2	70
6.2.2	Resultados Obtidos Com o Programa SACSO	70
6.2.3	Resultados Obtidos Com o Programa PSPICE	70
6.3	Retificador Monofásico	70
6.3.1	Resultados Obtidos Com o Programa SICON1	72
6.3.2	Resultados Obtidos Com o Programa SACSO	72

6.3.3	Resultados Obtidos Com o Programa PSPICE	72
6.4	Conclusões	75
7	Conclusão Geral	76
7.1	Sugestões Para Trabalhos Futuros	78
8	LISTAGEM DOS PROGRAMAS SICON1 e SICON2	79
8.1	PROGRAMA SICON1	79
8.2	Subrotina para a Conversores CC/CC e CC/CA (topologia fixa)	85
8.3	Subrotina para a Conversores CA/CC e CA/CA (topologia fixa)	87
8.4	Subrotina para a Conversores CC/CC (topologia variável)	89
8.5	Subrotina para a Conversores CA/CC e CA/CA (topologia variável) . .	92
8.6	Programa SICON2	94
8.7	Subrotina para circuitos CC/CC	98
8.8	Subrotina para a conversão CA/CC	100
8.9	Function fsc2 (Cálculo do Período de Amostragem)	103

Lista de Figuras

3.1	Braço de ordem k	11
3.2	Exemplos de árvores do grafo ξ	14
3.3	Percursos fechados fundamentais para a árvore escolhida de um grafo.	15
3.4	Cortes fundamentais para a árvore escolhida de um dado grafo.	16
3.5	Braços típicos com fontes independentes	19
4.1	Símbolo de um diodo	24
4.2	Símbolo de um SCR	24
4.3	Símbolos dos transistores NPN e PNP	24
4.4	Capacitor linear e invariante no tempo	25
4.5	Modelo para o capacitor segundo a aproximação Forward Euler	26
4.6	Modelo para o capacitor segundo a aproximação Backward Euler	26
4.7	Modelo para o capacitor segundo a aproximação Trapezoidal	26
4.8	Indutor linear e invariante no tempo	27
4.9	Modelo para o indutor segundo a aproximação Forward Euler	27
4.10	Modelo para o indutor segundo a aproximação Backward Euler	27
4.11	Modelo para o indutor segundo a aproximação Trapezoidal	28
4.12	Conversor Buck	28
4.13	Modelo resistivo adotado para o conversor buck	28
4.14	Tela de entrada para a escolha do tipo de conversor	30
4.15	Tela de entrada para a escolha do tipo de topologia	30
4.16	Tela para entrar com a quantidade de nós, com o número de elementos e com o nó de referência.	30
4.17	Tela para a escolha do tipo de elemento a ser inserido no circuito conversor.	31
4.18	Tela para a entrada dos parâmetros para um capacitor.	31
4.19	Fluxograma do programa SICON1	32
4.20	Representação do conversor Buck	33
4.21	Curvas teóricas para o conversor Buck	34
4.22	Resultados obtidos na simulação do conversor Buck	35

4.23	Representação do Retificador Trifásico Não Controlado	36
4.24	Formas de onda para o retificador trifásico não controlado	37
4.25	Resultados obtidos na simulação do Retificador Trifásico	38
4.26	Representação do Inversor Trifásico	39
4.27	Formas de onda para o inversor trifásico	39
4.28	Resultados obtidos na simulação do Inversor Trifásico	40
5.1	Tela de entrada para a escolha do tipo de conversor	44
5.2	Segunda tela de apresentação do programa SICON2	44
5.3	Tela para a entrada dos parâmetros para um capacitor.	44
5.4	Representação da interpolação	46
5.5	Exemplo de conversor	47
5.6	Configuração do conversor (árvore, elementos de enlace, elementos de árvore, e sentido dos elementos)	48
5.7	Fluxograma do programa SICON2	49
5.8	Conversor QRC-ZCS	50
5.9	Disposição dos componentes para o conversor QRC-ZCS	50
5.10	Resultado para o conversor QRC-ZCS.	51
5.11	Conversor Buck PWM SSS	52
5.12	Formas de onda teóricas para o Conversor Buck PWM SSS	54
5.13	Disposição dos elementos no conversor Buck PWM SSS	54
5.14	Resultados obtidos para o conversor Buck PWM SSS	55
5.15	Formas de onda experimentais para o Conversor Buck PWM-SSS; Tempo: $1,56\mu S/div$, Tensões : 20 V/div, Is: 0,55 A/div, ILR: 1,1 A/div	56
5.16	Conversor Quadrático PWM-SSS	57
5.17	Formas de onda teóricas para o Conversor Quadrático PWM SSS	59
5.18	Disposição dos elementos no conversor Quadrático PWM- SSS	59
5.19	Resultados obtidos para o conversor Quadrático PWM SSS	60
5.20	Formas de onda experimentais para o Conversor Quadrático PWM SSS	61
5.21	Inversor Monofásico em Ponte	62
5.22	Algumas curvas teóricas para o inversor monofásico em ponte	63
5.23	Disposição dos elementos do inversor monofásico em ponte	63
5.24	Resultados obtidos para o inversor monofásico em ponte	64
5.25	Retificador Monofásico em Ponte	65
5.26	Disposição dos elementos do retificador monofásico em ponte	66
5.27	Resultados obtidos para o retificador monofásico em ponte	66

6.1	Conversor CC-CC ressonante	68
6.2	Formas de onda teóricas para o Conversor CC-CC ressonante	69
6.3	Resultado para o conversor CC-CC obtidos com o programa SICON2.	70
6.4	Resultado para o conversor CC-CC obtidos com o programa SACSO;(a) tensão no capacitor,(b)corrente na carga	71
6.5	Resultado para o conversor CC-CC obtidos com o programa PSPICE;(a) corrente na carga,(b) tensão no capacitor	71
6.6	Retificador monofásico com carga RL	72
6.7	Formas de onda teóricas para o Retificador Monofásico com Carga RL	73
6.8	Resultado para o Retificador Monofásico obtidos com o programa SICON1	73
6.9	Resultado para o Retificador Monofásico obtidos com o programa SACSO;(a)corrente de carga,(b)tensão de carga	74
6.10	Resultado para o Retificador Monofásico obtidos com o programa PS- PICE;(a)corrente de carga,(b)tensão de carga	74

Capítulo 1

Introdução Geral

A Eletrônica de Potência tem um papel importante no controle de sistemas industriais e vem se tornando uma das mais importantes áreas da Engenharia Elétrica [1][2]. Na Eletrônica de Potência, os circuitos a serem estudados são, na sua grande maioria, os conversores estáticos de potência. Estes por sua vez dividem-se em: conversores CC-CC, conversores CA-CC, conversores CC-CA, e conversores CA-CA.

Na década de 70 e especialmente na de 80, o uso industrial de conversores estáticos de potência experimentou um grande avanço com o desenvolvimento de novos semicondutores de Potência (MOSFET, GTO e IGBT) e novas configurações de conversores, proporcionando uma maior eficiência no uso dos mesmos.

A necessidade de reduzir perdas, aumentar a eficiência e conservar a energia em processos industriais, vem proporcionando um aumento do uso da eletrônica em tais processos, e este aumento, por sua vez, exige o emprego de ferramentas que permitam o estudo de configurações e modos de operação de conversores de energia.

O desenvolvimento de programas de simulação, consiste na mais importante ferramenta usada para a concepção e dimensionamento de sistemas de conversão estática de energia [3][4][5][6][7]. Através da observação das formas de onda de corrente e tensão de cada dispositivo, geradas pelo programa de simulação, consegue-se obter uma melhor análise do seu funcionamento. A simulação pode ser usada para substituir experiências em laboratório, especialmente durante os primeiros passos do desenvolvimento do conversor de energia, e isto permite antecipar a ocorrência de condições de operação externas, minimizar defeitos e reduzir o tempo para a obtenção de um protótipo que funcione corretamente.

Na simulação em computadores, todas as tensões e correntes inerentes ao conversor podem ser investigadas, usualmente com mais eficiência do que no desenvolvimento do "hardware" em laboratório [8]. Variações na performance do conversor produzidas por

mudanças de parâmetros, também podem ser acompanhadas mais facilmente com o uso de simulações.

Vários programas têm sido desenvolvidos com o objetivo de simular circuitos na área de Eletrônica de Potência. Programas como o PSPICE [9], Electronic Workbench, SACSO[10], e o PROSCES [11] são exemplos desses programas. A maioria destes programas foram concebidos para uma simulação completa do conversor (análise transitória, análise espectral, inserção de módulos de controle, etc). Mesmo considerando-se a existência desses programas ainda há espaço para a investigação de novas metodologias de simulação que permitam reduzir a complexidade computacional, melhorar a precisão e flexibilizar o estudo de novas topologias de conversores estáticos.

O desenvolvimento de um programa de simulação próprio permite o domínio completo do código fonte o que facilita a realização de modificações, correções de falhas e a inclusão de novas funcionalidades. Neste sentido, este trabalho propõe o desenvolvimento de programas simplificados para o uso em simulações de conversores estáticos de potência. Os programas são "toolboxes" desenvolvidos dentro do ambiente MATLAB [12].

As principais razões para o uso do ambiente do MATLAB, são as seguintes:

- Facilidade na manipulação de complexas funções matemáticas;
- Disponibilidade de várias funções gráficas;
- Programação de fácil entendimento.

Nos "toolboxes" desenvolvidos ou a configuração do circuito se modifica de acordo com os estados aberto ou fechado dos semicondutores (topologia variável), ou a configuração permanece a mesma e os parâmetros que representam os semicondutores se modificam (topologia fixa). Nos toolboxes desenvolvidos usando topologia fixa, usa-se um resistor com resistência pequena para representar o dispositivo semiconductor em estado de condução e um resistor com resistência grande para representar o dispositivo semiconductor em estado de bloqueio. A dispersão de resistência grande e pequena pode levar a dispersão das constantes de tempo do circuito, o que leva, mesmo com a utilização de um período de amostragem variável, a um passo de amostragem pequeno. A topologia variável diminui esta dispersão na medida em que não se utiliza um resistor com resistência grande para representar o dispositivo semiconductor no seu estado de bloqueio.

O primeiro "toolbox" desenvolvido é o SICON1. O programa SICON1 utiliza a teoria de resolução de circuitos através da análise de nós de redes lineares invariantes, com o modelo do conversor sendo desenvolvido exclusivamente por equações algébricas. O programa permite o estudo de qualquer tipo de conversor estático a semiconductor,

com muito pouco conhecimento a priori do funcionamento do conversor. No programa SICON1 podemos simular o conversor utilizando topologia fixa ou topologia variável, e em ambos os casos usou-se um período de amostragem fixo. O primeiro passo para descrever o conversor estático diz respeito a escolha do tipo de conversor (conversor CC-CC, conversor CA-CC, conversor CC-CA, conversor CA-CA). O segundo passo corresponde a escolha do tipo de representação adotada para simular o conversor (topologia fixa ou topologia variável). O terceiro passo diz respeito a entrada da quantidade de nós, do número de elementos, e do nó de referência. O quarto passo corresponde a escolha do tipo de componente a ser inserido no circuito conversor, juntamente com seus parâmetros.

O segundo "toolbox" desenvolvido é o SICON2. O programa SICON2 utiliza um modelo para o conversor baseado em equações de estado, que são determinadas através da análise cortes e de percursos fechados. As variáveis de estado são as correntes nos indutores e as tensões nos capacitores. O circuito é simulado com o uso de uma topologia fixa, onde os dispositivos semicondutores são representados por resistências grandes ou pequenas, relativas aos estados aberto e fechado, respectivamente. O período de amostragem utilizado na simulação é variável [13], escolhido automaticamente a partir dos autovalores da matriz dinâmica, e a determinação dos instantes de mudança de estado dos semicondutores é realizada por meio de uma interpolação linear. O primeiro passo para descrever o conversor estático diz respeito a escolha do tipo de conversor (conversor CC-CC, conversor CA-CC, conversor CC-CA, conversor CA-CA). O segundo passo diz respeito a entrada da quantidade de nós, número de braços, número de capacitores em árvore, número de indutores em enlace, número de diodos e interruptores em enlace, número de diodos e interruptores em árvore, número de resistências em enlace, e o número de condutâncias em árvore. O terceiro passo corresponde a escolha do tipo de componente a ser inserido no circuito conversor, juntamente com seus parâmetros.

O programas SICON1 e SICON2 foram implementados com os recursos de programação do MATLAB. O programas são de fácil entendimento, e podem ser usado no estudo das diversas concepções de conversores estáticos.

O desenvolvimento de tais programas não objetiva competir com outros programas já desenvolvidos, mas sim colocar em disponibilidade novos "toolboxes" para os usuários do MATLAB, voltados para a área de eletrônica de potência.

São apresentados diversos resultados de simulação, mostrando a eficiência dos programas desenvolvidos.

1.1 Organização do Trabalho

No capítulo 2, é feita uma revisão bibliográfica acerca do desenvolvimento de programas de simulação voltados para o estudo de circuitos na área de Eletrônica de Potência.

No capítulo 3, são apresentadas as ferramentas teóricas, necessárias para o correto desenvolvimento tanto do programa SICON1 como o do SICON2. Estas ferramentas consistem na análise de nós de redes lineares invariantes, e na análise de equações de estado obtidas através da análise de percurso fechado.

No capítulo 4 é feita a descrição do programa SICON1. Inicialmente é feita uma análise sobre as possíveis representações para os dispositivos semicondutores. Logo após, são tratados os possíveis modelos para os capacitores e indutores. Em seguida é mostrada toda a estrutura do programa SICON1. E para finalizar são tratados alguns resultados obtidos através de simulação. Tenta-se mostrar a eficiência deste primeiro "toolbox" voltado para o estudo de conversores estáticos de energia.

No capítulo 5 é feita a descrição do programa SICON2. Inicialmente, são feitas considerações sobre o funcionamento do programa. Logo após, são realizados todos os comentários da estrutura do SICON2, tratando sobre a modelagem do dispositivo semicondutor, a solução da equação de estado contínua, a escolha do período de amostragem, a interpolação necessária para determinar o instante correto da mudança de estado dos dispositivos semicondutores, e também são feitos comentários sobre o fluxograma do programa. Por fim, são tratados alguns resultados obtidos através de simulação. Em todo o capítulo, tenta-se mostrar o funcionamento deste segundo "toolbox".

No capítulo 6 temos uma comparação entre resultados obtidos com os programas SICON1 e SICON2, e resultados obtidos com o SACSO e o PSPICE.

No capítulo 7 as conclusões sobre os resultados obtidos durante o trabalho são sumarizadas e propostas de continuidade do trabalho são apresentadas.

Capítulo 2

Estado da Arte

2.1 Introdução

Uma breve revisão bibliográfica sobre as técnicas empregadas na simulação de conversores estáticos de energia, presentes na área de Eletrônica de Potência, é apresentada neste capítulo. Uma atenção especial é dada para os métodos de análise de circuitos usados nas ferramentas de simulação.

2.2 Simulação em Eletrônica de Potência

Todos os programas de simulação de circuitos possuem as seguintes etapas: formulação das equações, e solução das equações. Os programas de simulação diferem, basicamente, em como estas equações são formuladas, e não como são resolvidas. Isto é devido ao fato de que diversas características de um programa podem ser influenciadas pelo método usado para a formulação das equações, dentre estas características pode-se citar: esforço despendido na programação, exigências no armazenamento de informações, o tempo para a especificação do problema e a velocidade de execução. Uma vez escolhido o método de formulação, a estrutura dos dados do programa fica basicamente determinada. A solução das equações formuladas é feita por um método de integração numérica, adequado ao tipo de formulação escolhida.

2.2.1 Variáveis de Estado

O método de variáveis de estado [14] tem sido bastante empregado na concepção e análise de circuitos em Eletrônica de Potência. A idéia básica consiste em formular equações de estado para o circuito em estudo e depois resolvê-las através de métodos

de aproximação numérica.

O método de análise por variáveis de estado leva a um conjunto de equações para o circuito, em termos de tensões e correntes de certos braços, conhecidas como variáveis de estado do circuito. Normalmente, as variáveis de estado são as tensões sobre os capacitores e as correntes sobre os indutores. Esse conjunto de equações é usualmente obtido usando a teoria dos grafos, onde uma árvore apropriada para cada configuração do circuito tem que ser determinada. As variáveis do circuito, não caracterizadas como variáveis de estado, são encontradas a partir de um sistema de equações algébricas, dependentes das variáveis de estado já obtidas.

Vários programas de simulação, que usam tal técnica, foram desenvolvidos para aplicações em Eletrônica de Potência. Exemplos típicos incluem: CIRCUIT [15], SCRIPT [16], ATOSEC [17], CAP [18], o COSMIR [19], e outros programas de simulação similares [20]. Estas ferramentas da simulação são similares em termos de potencialidade. Todos eles escrevem as equações do estado a partir da teoria dos grafos. Todos os dispositivos semicondutores são modelados como ideais, ou seja, como chaves abertas quando estão bloqueados, e como chaves fechadas quando estão em condução. Portanto, um circuito a ser simulado será representado por um conjunto de equações de estado lineares, com o um conjunto de equações para cada modo de operação do circuito. Desta forma, o circuito será simulado através da escolha e resolução de um conjunto apropriado de equações em função do tempo. Desde que as equações de estado, durante um modo de operação, são lineares e invariantes no tempo, elas podem ser resolvidas através de métodos analíticos [18] ou através de uma integração numérica direta [17]. Muitos destes programas utilizam interface gráfica para tornar mais amigável o uso dos mesmos.

2.2.2 Análise Nodal

Este tipo de análise compreende a aplicação da lei de Kirchhof aos nós do circuito a ser estudado, obtendo-se um conjunto de equações que representam o comportamento do circuito. As ferramentas de simulação baseadas na análise nodal foram desenvolvidas, originalmente, para aplicações na área de Sistemas de Potência. Os programas mais populares nessa área são o EMTP [21] e sua versão alternativa ATP [22]. O EMTP é um programa que foi originalmente desenvolvido para a simulação de transitórios eletromagnéticos em Sistemas de Potência. O uso da versão ATP do EMTP, permite, com uma maior facilidade, a simulação de circuitos na área de Eletrônica de Potência [23]. Isto deve-se ao fato de que tal versão permite o modelamento de uma grande quantidade de dispositivos semicondutores. As principais dificuldades encontradas no

uso de tais programas são: uso de um passo de cálculo fixo e oscilações numéricas devidas aos métodos empregados para resolver as equações formuladas.

Outro programa, mais recentemente desenvolvido, é o PECAN [24]. Uma importante característica do PECAN diz respeito a facilidade que o mesmo possui em permitir a simulação de sistemas em Eletrônica de Potência em malha fechada. No PECAN a implementação dos componentes e da malha fechada são feitas de formas distintas.

2.2.3 Análise Nodal Modificada

Alguns problemas com relação a análise nodal, dentre eles o tratamento aproximado para as fontes de tensão, podem ser eliminados através da análise nodal modificada [25]. Tal técnica é baseada na análise nodal tradicional, no entanto, esta nova abordagem inclui fontes de tensão e vários elementos dependentes de corrente.

Existe um grande número de programas de simulação que adotam esta técnica. Exemplos típicos, incluem: PSPICE (MicroSim), HSPICE (Meta-Software), ICAP3 (IntuSoft), SmartSPICE (Silvaco), DSPICE (Dazix Inc), IG-SPICE (A.B.Associates), I-SPICE (NCSS timesharing).

Quase todos estes programas de simulação usam métodos diretos para resolver as equações formuladas usando o método da Análise Nodal Modificada. Estes métodos, geralmente, usam integração numérica para converter o conjunto de equações diferenciais-algébricas não lineares em um conjunto de equações diferença não lineares. Estas equações são resolvidas usando o método iterativo de Newton-Raphson.

Outro programa que se utiliza de tal técnica é o Saber [26](Analogy Inc). Este programa de simulação possui diversas características interessantes. A primeira delas diz respeito ao programa ser um simulador analógico/digital. A segunda característica, de grande importância, corresponde ao fato da possibilidade de tal programa conseguir separar em modelos distintos o circuito a ser simulado, isto o torna mais flexível do que os simuladores da família SPICE. Uma outra característica corresponde ao fato do programa usar algoritmos numéricos iguais aos do SPICE e outros de características próprias.

Outra ferramenta de simulação que usa a Análise Nodal Modificada, foi desenvolvida por Sudha [27]. Este simulador modela o dispositivo semicondutor como um indutor binário. A maior dificuldade deste método corresponde ao fato do mesmo não usar um modelo muito preciso para os dispositivos semicondutores.

Embora a Análise Nodal Modificada seja mais completa do que a Análise Nodal, como técnica de simulação, algumas dificuldades no seu uso estão presentes, dentre elas: um conjunto maior de variáveis se faz necessário, a análise de pólos e zeros não é

feita de forma simples, e algumas análises mais específicas só são possíveis através de técnicas muito complexas.

2.3 Conclusão

Neste capítulo foi apresentado uma revisão bibliográfica sobre as técnicas empregadas na simulação de conversores estáticos de energia, presentes na área de Eletrônica de Potência. Observa-se que no desenvolvimento de um programa de simulação voltado para esta área deve-se ter a concepção de que o principal passo corresponde a escolha de método adequado para a formulação das equações do circuito. Várias características importantes de um programa de simulação dependem do método escolhido para formular tais equações.

Capítulo 3

Análise de Circuitos

3.1 Introdução

Neste capítulo são apresentadas as ferramentas de análise de circuitos utilizadas para a implementação dos programas de simulação de conversores estáticos descritos neste trabalho. A primeira ferramenta a ser descrita consiste numa análise de nós de redes lineares invariantes [28]. A segunda diz respeito a resolução de circuitos através de equações de estado [28].

3.2 Análise de Nós de Redes Lineares Invariantes

A utilização da análise de nós de redes lineares invariantes permite a representação da estrutura do conversor através de um sistema de equações algébricas lineares.

Considerando uma rede qualquer contendo nt nós e b braços. Numa rede com tal configuração há b tensões de braço e b correntes de braço a serem determinadas.

Para construir a representação algébrica da rede tomamos arbitrariamente um nó para referência. Esse nó de referência é normalmente chamado de terra. Este nó será definido como nt , e os demais são identificados com $1, 2, 3, \dots, n$, onde $n = nt-1$.

As n equações algébricas lineares, obtidas pela aplicação da lei de Kirchhof para correntes a todos os nós, exceto o terra, constituem um conjunto de equações lineares independentes.

Pode-se afirmar que o sistema tem a seguinte forma matricial:

$$A_j = 0 \tag{3.1}$$

na qual j representa o vetor corrente de braço cuja dimensão é b ; isto é,

$$j = \begin{bmatrix} j_1 \\ j_2 \\ \vdots \\ j_b \end{bmatrix}$$

e $A = (a_{ik})$ é uma matriz $n \times b$ definida por

$$a_{ik} = \begin{cases} 1 & \text{se o braço } k \text{ sai do nó } i \\ -1 & \text{se o braço } k \text{ entra no nó } i \\ 0 & \text{se o braço } k \text{ não é incidente com o nó } i \end{cases}$$

Aj é um vetor de dimensão n , pois, quando escrevemos que a componente de ordem i do vetor Aj é igual a zero, simplesmente estamos afirmando que a soma de todas as correntes de braço saindo do nó i é zero. A matriz A é denominada de matriz incidência reduzida.

A equação (3.1) representa um sistema de n equações linearmente independentes com $j_1, j_2, \dots, e j_b$ variáveis. Considerando que sempre temos $b > n$, pode-se concluir que a matriz incidência reduzida A é de ordem completa.

As tensões de e_1, e_2, \dots, e_n serão usadas como variáveis da análise dos nós, e estas por sua vez são as tensões dos nós $1, 2, \dots, n$ medidas em relação ao nó terra. A lei de Kirchhof para as tensões garante que as tensões de nó são definidas sem ambigüidade. Ao calcular a tensão de qualquer nó em relação à terra através de uma soma algébrica de tensões ao longo de um caminho desde o nó terra até o nó em questão, a lei das tensões de Kirchhof garante que a soma vai ser independente do caminho escolhido.

As tensões de braço podem ser obtidas apartir das tensões de nó pela equação

$$v = A^T e \quad (3.2)$$

na qual v é um vetor de ordem k , sendo $k = 1, 2, \dots, b$, correspondente as tensões de braço, A^T é a matriz $b \times n$ que é a transposta da matriz incidência reduzida A , e e corresponde ao vetor tensões dos nós, de ordem n .

O vetor v e e são definidos da seguinte forma:

$$v_k = \begin{cases} e_i & \text{se o braço } k \text{ sai do nó } i \\ -e_i & \text{se o braço } k \text{ entra no nó } i \end{cases}$$

$$e = [e_1 \ e_2 \ \dots \ e_n]^T$$

Por outro lado, se o braço k sai do nó i e entra no nó j , teremos:

$$v_k = e_i - e_j$$

Em redes lineares invariantes, todos os elementos, exceto as fontes, são lineares e invariantes. O problema geral de análise de nós é combinar as equações de braço de elementos lineares invariantes com as duas equações básicas

$$\begin{aligned} A_j &= 0 \\ e \\ v &= A^T e \end{aligned}$$

As equações resultantes formam um sistema de equações diferenciais lineares com n variáveis de rede, e_1, e_2, \dots, e_n . Como o modelo adotado para a análise dos circuitos inclui somente elementos resistivos e fontes, as equações resultantes serão algébricas lineares.

3.2.1 Análise de redes resistivas

Considerando uma rede resistiva linear invariante, com b braços e nt nós, na qual um braço típico é mostrado na Figura 3.1.

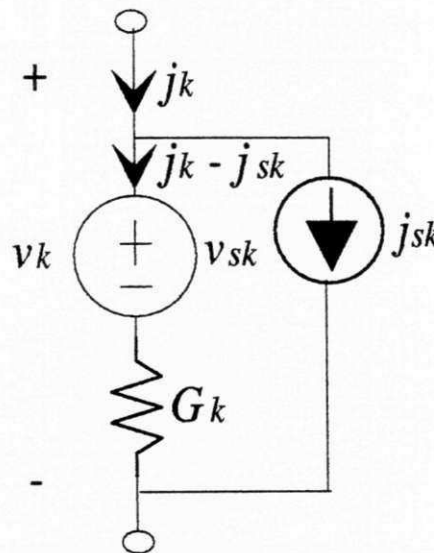


Figura 3.1: Braço de ordem k

As equações de braço são da forma

$$v_k = R_k j_k + v_{sk} - R_k j_{sk} \tag{3.3}$$

ou, de modo equivalente,

$$j_k = G_k + j_{sk} - G_k v_{sk} \quad (3.4)$$

na qual $k = 1, 2, \dots, b$.

Em notação matricial temos, da equação (3.4),

$$j = Gv + j_s - Gv_s \quad (3.5)$$

onde G é chamada de matriz condutância de braço. Ela é uma matriz diagonal de ordem b ; isto é,

$$G = \begin{bmatrix} G_1 & 0 & \dots & \dots & \dots & 0 \\ 0 & G_2 & 0 & \dots & \dots & 0 \\ \vdots & 0 & \ddots & \dots & \dots & 0 \\ \vdots & \dots & \dots & \ddots & \dots & 0 \\ \vdots & \dots & \dots & \dots & \ddots & 0 \\ 0 & \dots & \dots & \dots & \dots & G_b \end{bmatrix}$$

Os vetores j_s e v_s são vetores fonte de dimensão b ; isto é,

$$j_s = \begin{bmatrix} j_{s1} \\ j_{s2} \\ \vdots \\ j_{sb} \end{bmatrix}$$

$$v_s = \begin{bmatrix} v_{s1} \\ v_{s2} \\ \vdots \\ v_{sb} \end{bmatrix}$$

Multiplicando pela esquerda a equação (3.5) pela matriz A , substituindo v por $A^T e$ e usando (3.1), obtemos

$$AGA^T e + Aj_s - AGv_s = 0 \quad (3.6)$$

ou

$$AGA^T e = AGv_s - Aj_s \quad (3.7)$$

Na equação (3.7), AGA^T é uma matriz quadrada $n \times n$, enquanto que AGv_s e $-Aj_s$ são vetores de dimensão n . Introduzindo as seguintes notações:

$$Y_n = AGA^T \quad (3.8)$$

$$i_s = AGv_s - Aj_s \quad (3.9)$$

A equação (3.7) torna-se:

$$Y_n e = i_s \quad (3.10)$$

O conjunto de equações representado em (3.10) é normalmente chamado de equações de nó. Y_n é chamada de matriz admitância de nó, e i_s , é o vetor fonte de corrente de nó.

Os vetores v_s e j_s , que especificam as fontes de braços, são fornecidos; portanto, o vetor fonte de corrente de nó i_s é também conhecido. Assim, (3.10) relaciona o vetor e de dimensão n com a conhecida $n \times n$ matriz Y_n e o conhecido vetor i_s de dimensão n .

Uma vez obtido e , é fácil obter as b tensões de braço v e obter as b correntes de braço j .

3.3 Resolução de Circuitos Através de Equações de Estado

A resolução de circuitos através de equações de estado é uma outra técnica utilizada na simulação dos conversores. As equações de estado para o conversor são obtidas por meio da análise de cortes e de percursos fechados.

Para a compreensão da técnica a ser utilizada na resolução das equações do circuito [28] é necessário definir os conceitos de árvore e enlace. Define-se por árvore um subgrafo ligado contendo todos os nós do grafo [28] e não possuindo caminhos fechados. Define-se como enlace os braços pertencentes ao grafo e que não fazem parte da árvore.

Um grafo normalmente tem muitas árvores. Na Figura 3.2 mostra-se um exemplo de algumas árvores de um grafo ligado ξ .

Escolhida uma árvore T , cada enlace associado a esta árvore irá definir um *percurso fechado fundamental*, ou seja, o percurso fechado formado pelo enlace e pelo único caminho de árvore entre os nós desse enlace. Todos os braços de árvore de T junto com alguns enlaces definem um único corte de ξ . Este corte é chamado de *corte fundamental* associado ao braço de árvore.

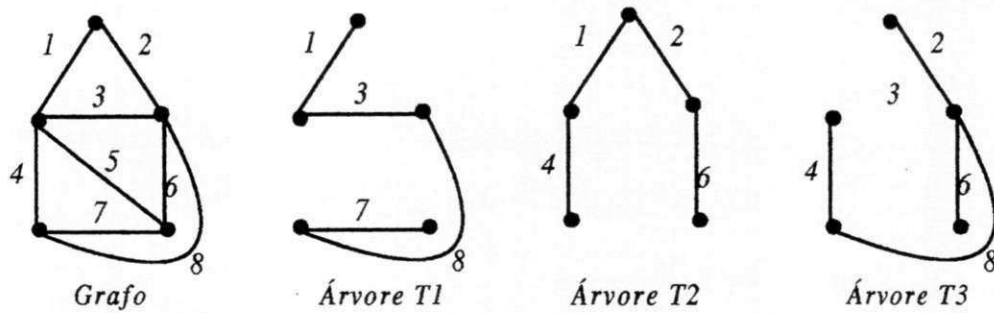


Figura 3.2: Exemplos de árvores do grafo ξ

3.3.1 Análise de Percurso Fechado

Considerando um grafo ligado com b braços e n_t nós, e tomando uma árvore arbitrária T . Há $n = n_t - 1$ braços de árvore e $l = b - n$ enlases.

A numeração dos braços é feita como segue: primeiro os enlases de 1 a l , depois os braços de $l + 1$ até b . Cada enlace define um percurso fechado fundamental, isto é, o percurso fechado formado pelo enlace e pelo único caminho de árvore entre os nós desse enlace. Isto está ilustrado na Figura 3.3 por um simples grafo com $b = 8$, $n_t = 5$, $n = 4$ e $l = 4$.

Para aplicar a lei de Kirchhoff para tensões a cada percurso fechado fundamental, adota-se um sentido de referência para o percurso fechado que concorda com o sentido de referência do enlace que define o percurso fechado fundamental, e isto é visto na Figura 3.3.

A forma matricial da lei de Kirchhoff para tensões, para o circuito da Figura 3.3, é escrita como segue:

$$\begin{array}{l}
 l \text{ percursos} \\
 \text{fechados}
 \end{array}
 \left\{ \begin{array}{l}
 \left[\begin{array}{cccc|cccc}
 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & -1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 0 & -1 & 1 & 1 \\
 0 & 0 & 0 & 1 & 0 & -1 & 1 & 0
 \end{array} \right]
 \begin{array}{l}
 v_1 \\
 v_2 \\
 v_3 \\
 v_4 \\
 v_5 \\
 v_6 \\
 v_7 \\
 v_8
 \end{array}
 \end{array} \right. = \begin{array}{l}
 \left[\begin{array}{l}
 0 \\
 0 \\
 0 \\
 0
 \end{array} \right]
 \end{array}$$

l enlases n braços de árvore

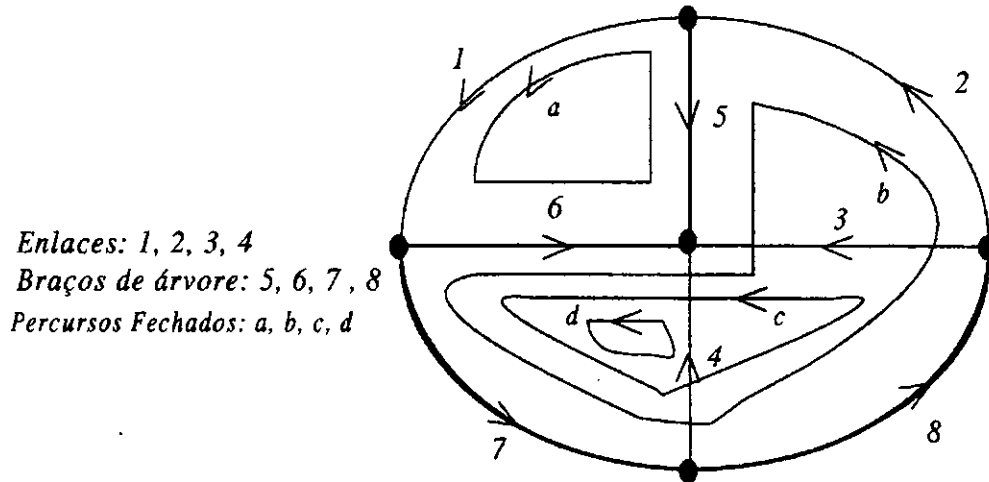


Figura 3.3: Percursos fechados fundamentais para a árvore escolhida de um grafo.

A equação matricial descrita acima, pode ser escrita da seguinte forma:

$$Bv = 0 \tag{3.11}$$

na qual B é uma matriz $l \times b$ chamada de *matriz percurso fechado fundamental*.

Uma vez que cada percurso fechado fundamental inclui apenas um enlace, e sendo iguais as orientações do percurso fechado e do enlace, numerando-se os enlaces com $1, 2, \dots, l$ e os braços de árvore com $l + 1, l + 2, \dots, b$, a matriz B toma a seguinte forma:

$$B = \left[\begin{array}{c|c} 1_l & F \end{array} \right] \tag{3.12}$$

onde 1_l significa a matriz unitária de ordem l e F significa uma matriz retangular de l linhas e n colunas.

3.3.2 Análise de Cortes

A análise de cortes é dual a análise de percursos fechados. Primeiro, toma-se uma árvore T , em seguida, numeram-se os braços como antes, onde os enlaces vão de 1 a l ,

e os braços de árvore vão de $l + 1$ até b . Cada braço de árvore define um único corte fundamental, e este corte é feito de enlaces e de um braço de árvore que é o que define o corte. Na Figura 3.4 tem-se o mesmo grafo ξ e a mesma árvore T destacados na Figura 3.3. Os quatro cortes fundamentais estão também mostrados.

Numeram-se os cortes como segue: o corte 1 está associado com o braço de árvore 5; o corte 2 com o braço de árvore 6, etc. Por analogia, com as convenções anteriores, para cada corte fundamental, adota-se um sentido de referência para o corte que concorda com o sentido de referência do braço de árvore que define o corte.

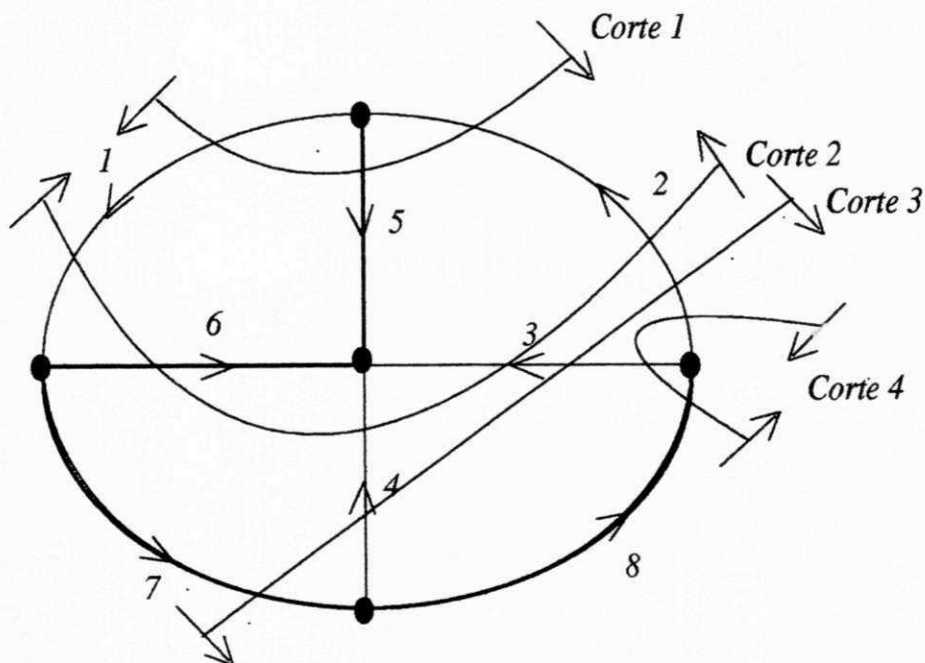


Figura 3.4: Cortes fundamentais para a árvore escolhida de um dado grafo.

Aplicando a lei de Kirchooff para corrente ao grafo da Figura 3.4, temos a seguinte

representação matricial para os cortes em destaque

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} j_1 \\ j_2 \\ j_3 \\ j_4 \\ j_5 \\ j_6 \\ j_7 \\ j_8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

A equação matricial descrita acima, pode ser escrita da seguinte forma:

$$Qj = 0 \quad (3.13)$$

na qual Q é uma matriz $n \times b$ chamada de matriz corte fundamental. Como antes, observa-se que ela é da forma:

$$Q = \left[E \quad : \quad 1_n \right] \quad (3.14)$$

onde E é uma matriz $n \times l$ com elementos -1 , $+1$, e 0 . 1_n é uma matriz unitária $n \times n$.

3.3.3 Equações de Estado Para Redes Lineares Invariantes

O primeiro passo para a aplicação da técnica de resolução de circuitos através de equações de estado é a escolha de uma árvore para o circuito em análise. Neste trabalho considera-se que no circuito em análise não existem percursos fechados contendo unicamente capacitores, nem cortes puramente indutivos. Isto restringe a classe dos circuitos, todavia uma abordagem mais geral resultaria em equações muito complexas.

A escolha da árvore deve ser feita de modo a contemplar todos os elementos capacitivos, sem os elementos indutivos. A árvore poderá ser completada adicionando-se resistores em localizações específicas. Os elementos indutivos são agrupados nos enlaces.

Em geral, é conveniente separar os braços em quatro subconjuntos, quais sejam: os enlaces resistivos, os enlaces indutivos, os braços de árvore capacitivos e os braços de árvore resistivos. As equações da lei de Kirchhoff para as tensões aplicadas aos percursos fechados são $Bv = 0$, ou

$$\begin{bmatrix} 1_1 & \vdots & F \end{bmatrix} \begin{bmatrix} v_R \\ v_L \\ \text{---} \\ v_C \\ v_G \end{bmatrix} = 0 \quad (3.15)$$

na qual v_R , v_L , v_C e v_G são subvetores representando, respectivamente, as tensões dos enlaces resistivos, dos enlaces indutivos, dos braços de árvore capacitivos e dos braços de árvore resistivos.

As equações da lei de Kirchhoff para correntes aplicadas aos cortes fundamentais são $Qj = 0$, ou

$$\begin{bmatrix} -F^T & \vdots & 1_n \end{bmatrix} \begin{bmatrix} j_R \\ j_L \\ \text{---} \\ j_C \\ j_G \end{bmatrix} = 0 \quad (3.16)$$

na qual j_R , j_L , j_C e j_G são subvetores representando, respectivamente, correntes nos enlaces resistivos, enlaces indutivos, braços de árvore capacitivos e braços de árvore resistivos.

Por conveniência, supõe-se que as fontes independentes localizadas em enlaces são fontes de tensão e que as fontes independentes localizadas em braços de árvore são fontes de corrente. Isto certamente não impõe restrição ao método, pois é fácil transformar fontes independentes de tensão em fontes de corrente, e vice-versa. Estas equações de braço são escritas na forma seguinte, com as novas notações definidas abaixo:

$$v_R = R_R + e_R \quad (3.17)$$

$$v_L = L \frac{di_L}{dt} + e_L \quad (3.18)$$

$$j_C = C \frac{dv_C}{dt} + i_C \quad (3.19)$$

$$j_G = G_G v_G + i_G \quad (3.20)$$

As matrizes R_R , L , C e G_G são todas matrizes de parâmetros de braço que significam, respectivamente, matriz resistência de enlace, matriz indutância de enlace, matriz capacitância de braço de árvore e matriz condutância de braço de árvore da rede. Os vetores e_R , e_L , i_C e i_G correspondem as fontes independentes. Os sentidos de referência usados ao escrever as equações (3.17), (3.18), (3.19), (3.20) são indicados na Figura 3.5.

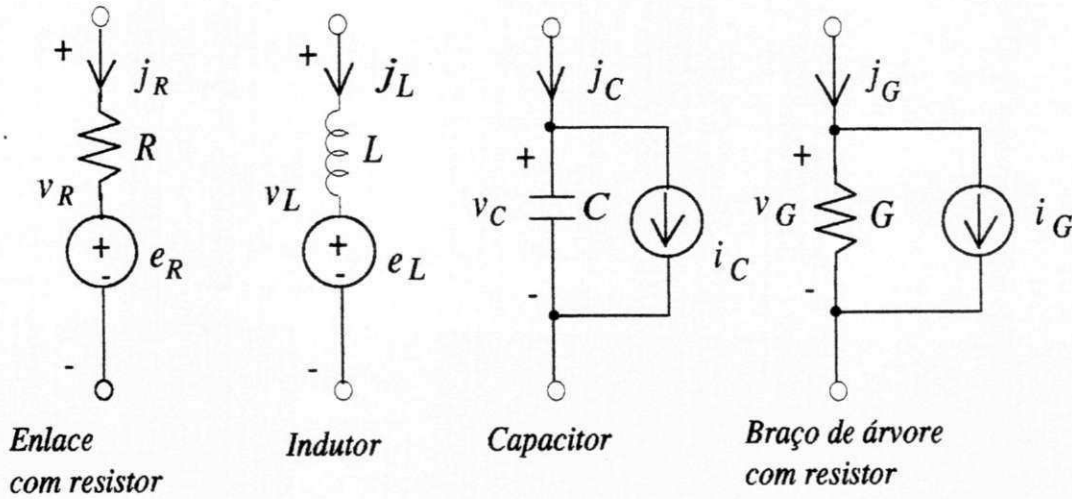


Figura 3.5: Braços típicos com fontes independentes

Combinando as equações (3.15), (3.16), (3.17), (3.18), (3.19) e (3.20), e eliminando-se todas as variáveis que não sejam variáveis de estado nem fontes, pode-se escrever as equações (3.15) e (3.16) como:

$$\begin{bmatrix} v_R \\ v_L \end{bmatrix} = -F \begin{bmatrix} v_C \\ v_G \end{bmatrix} = - \begin{bmatrix} F_{RC} & F_{RG} \\ F_{LC} & F_{LG} \end{bmatrix} \begin{bmatrix} v_C \\ v_G \end{bmatrix} \quad (3.21)$$

$$\begin{bmatrix} j_C \\ j_G \end{bmatrix} = F^T \begin{bmatrix} j_R \\ j_L \end{bmatrix} = \begin{bmatrix} F_{RC}^T & F_{LC}^T \\ F_{RG}^T & F_{LG}^T \end{bmatrix} \begin{bmatrix} j_R \\ j_L \end{bmatrix} \quad (3.22)$$

Combinando-se as equações (3.17), (3.18), (3.19), (3.20) com as equações (3.21) e (3.22) obtém-se

$$R_R j_R = -F_{RC} v_C - F_{RG} v_G - e_R \quad (3.23)$$

$$L \frac{dj_L}{dt} = -F_{LC} v_C - F_{LG} v_G - e_L \quad (3.24)$$

$$C \frac{dv_C}{dt} = F_{RC}^T j_R + F_{LC}^T j_L - i_C \quad (3.25)$$

$$G_G v_G = F_{RG}^T j_R + F_{LG}^T - i_G \quad (3.26)$$

Na forma de estado estas equações são:

$$\frac{dx}{dt} = Ax + Bu \quad (3.27)$$

onde,

$$A = \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}^{-1} \begin{bmatrix} -Y_t & H_t \\ -H_t^T & -Z_t \end{bmatrix};$$

$$B = - \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}^{-1} B_t;$$

$$x = \begin{bmatrix} v_C \\ j_L \end{bmatrix};$$

$$u = \begin{bmatrix} i_C \\ i_G \\ e_R \\ e_L \end{bmatrix}$$

Os termos são definidos como segue:

$$\begin{aligned} Y_t &= F_{RG}^T R_t^{-1} F_{RC} \\ Z_t &= F_{LG} G_t^{-1} F_{LG}^T \\ H_t &= F_{LC}^T - F_{RC}^T R_t^{-1} F_{RG} R_G F_{LG}^T \\ R_t &= R_R + F_{RG} R_G F_{RG}^T \\ G_t &= G_G + F_{RG}^T G_R F_{RG} \\ B_t &= \begin{bmatrix} 1 & -F_{RC}^T R_t^{-1} F_{RG} R_G & F_{RC}^T R_t^{-1} & 0 \\ 0 & -F_{LG} G_t^{-1} & -F_{LG} G_t^{-1} F_{RG}^T G_R & 1 \end{bmatrix} \\ R_G &= G_G^{-1} \\ G_R &= R_R^{-1} \end{aligned} \quad (3.28)$$

Escrevendo a equação (3.27) como

$$\dot{x}(t) - Ax(t) = Bu(t)$$

e pré-multiplicando ambos os lados desta equação por e^{-At} , obtém-se

$$e^{-At} [\dot{x}(t) - Ax(t)] = \frac{d}{dt} [e^{-At} x(t)] = e^{-At} Bu(t)$$

Integrando a equação anterior entre 0 e t, resulta

$$e^{-At} x(t) = x(0) + \int_0^t e^{-A\tau} Bu(\tau) d\tau$$

ou

$$x(t) = e^{At} x(0) + \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau \quad (3.29)$$

A resolução da equação (3.27), ou seja, a equação (3.29), fornecerá as correntes nos indutores e as tensões nos capacitores. A partir destas variáveis de estado obtêm-se tensões e correntes nos demais braços do circuito. Para que isto possa ser feito, faz-se necessário definir toda a estrutura do circuito (número de nós, definição da árvore, número de elementos em árvore, número de elemento de enlace, etc).

3.4 Conclusão

Neste capítulo são apresentadas as técnicas de análise de circuitos necessárias para o desenvolvimento dos programas propostos neste trabalho. Numa primeira etapa foi realizado o estudo da análise de nós de redes lineares invariantes, onde verifica-se que as equações são simples de serem obtidas. Verifica-se que o conjunto de equações obtidas são puramente algébricas. Numa segunda etapa, foi feita a análise de equações de estado, empregadas para a análise de circuitos. Neste caso, torna-se evidente o maior grau de complexidade das equações.

Capítulo 4

Programa SICON1

4.1 Introdução

Neste capítulo, é tratado o programa SICON1, empregado para a simulação de conversores estáticos.

O programa SICON1 utiliza a teoria de resolução de circuitos através da análise de nós de redes lineares invariantes [28].

O objetivo é apresentar a formulação matemática, o algoritmo de simulação, o programa propriamente dito, assim como todos os resultados obtidos com o programa.

4.2 Os Modelos dos Semicondutores

Diante das diversas possibilidades de modelar os semicondutores, a escolha desse modelo se torna importante no desenvolvimento do programa objetivando um melhor desempenho da simulação [29][30][31].

No programa SICON1 adotam-se duas metodologias na simulação do circuito conversor. Na primeira delas a topologia do circuito permanece fixa, enquanto que na segunda a topologia do circuito muda na medida em que os dispositivos semicondutores entram em estado de condução ou bloqueio. O chaveamento do estado de bloqueio para condução e vice-versa, dependerá das características específicas de cada dispositivo do circuito.

4.2.1 A Topologia Fixa

O programa SICON1 quando executado na opção por uma topologia fixa faz com que a configuração do circuito permaneça a mesma e os parâmetros que representam os

semicondutores se modifiquem de acordo com os estados aberto ou fechado dos semicondutores. O dispositivo semicondutor quando em estado de condução é representado por um resistor com uma resistência de pequeno valor, e quando em estado de bloqueio é representado por um resistor com elevada resistência. No programa desenvolvido com este tipo técnica, observa-se que as dimensões das matrizes são constantes, o que torna bastante simples o processo de escrever o programa.

4.2.2 A Topologia Variável

O programa SICON1 quando executado na opção por uma topologia variável faz com que configuração do circuito se modifique de acordo com os estados aberto ou fechado dos semicondutores. Nesta técnica, o dispositivo semicondutor é retirado do circuito quando não está conduzindo. Esta retirada permite uma redução da estrutura do circuito, visto que a quantidade de caminhos fechados diminui. Quando o dispositivo semicondutor está conduzindo o mesmo é tratado como um resistor com uma resistência de pequeno valor. Este tipo de concepção permite uma análise do circuito muito próxima do caso ideal, ou seja, aproximando-se de uma configuração em que o dispositivo semicondutor é tratado como uma chave ideal. Em termos de programação, observa-se uma complexidade maior em relação ao programa desenvolvido com o uso da topologia fixa, pois as dimensões das matrizes que representam a estrutura do conversor estão sempre mudando, tornando-se necessário o desenvolvimento de subrotinas que alterem automaticamente estas dimensões.

4.2.3 O Chaveamento dos Semicondutores

O programa permite a simulação de conversores utilizando como dispositivos semicondutores diodos, SCRs e transistores.

O diodo conduz quando a corrente e a tensão sobre ele são positivas de anodo para catodo. Quando a corrente se anula e a tensão é negativa, o diodo estará no estado de bloqueio. O teste da corrente e da tensão sobre o diodo é feito em cada período de amostragem. O símbolo de um diodo é mostrado na Figura 4.1

A determinação do estado do SCR e do transistor é um pouco mais complexa, pois é necessário considerar seus comandos. Os dois componentes assumem o estado de bloqueio quando a tensão sobre seus terminais é negativa. No entanto, considerações sobre o estado de condução se fazem necessárias.

O SCR conduz se sua tensão é positiva e se ele recebe um sinal de disparo. Caso o sinal de disparo não exista, este se mantém em estado de bloqueio. Uma vez que o SCR



Figura 4.1: Símbolo de um diodo

se encontra no estado de condução, ele só muda de estado na passagem da corrente por zero. o símbolo de um SCR é mostrado na Figura 4.2.

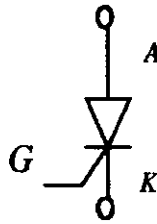


Figura 4.2: Símbolo de um SCR

O transistor conduz se sua tensão é positiva e se nele existe um sinal de polarização de base. Caso o sinal de polarização de base não exista, o dispositivo assume o estado de bloqueio. Temos dois tipos de transistor, o NPN e o PNP, e a identificação do fluxo de corrente determina o tipo de transistor a ser utilizado no circuito. Os símbolos dos transistores NPN e PNP são mostrados na figura 4.3

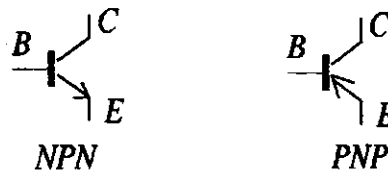


Figura 4.3: Símbolos dos transistores NPN e PNP

O uso do programa SICON1 com os dispositivos semicondutores descritos anteriormente, não impede que o usuário inclua outros dispositivos. Para tal inclusão se faz necessário definir no programa a subrotina que represente o comportamento do mesmo.

4.3 Modelos de Capacitores e Indutores

Normalmente, a existência de capacitores e indutores em um circuito dá origem a necessidade de resolução de equações diferenciais. Para que as equações a serem implementadas no programa SICON1 sejam algébricas, possibilitando uma fácil compreensão,

os modelos a serem considerados para os capacitores e indutores assumem um papel importante. Pode-se encontrar modelos diversos para os capacitores e indutores [32]. Algumas possibilidades são tratadas nesta seção.

Considere o capacitor mostrado na Figura 4.4. Nós podemos expressar a tensão sobre o mesmo em termos da integral da sua corrente, ou seja:

$$v(t + \Delta t) = v(t) + \frac{1}{C} \int_t^{t+\Delta t} i(\tau) d\tau \quad (4.1)$$

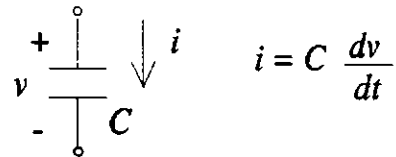


Figura 4.4: Capacitor linear e invariante no tempo

A integral da equação 4.1 pode ser aproximada de várias formas, dentre elas podemos citar:

$$\int_t^{t+\Delta t} i(\tau) d\tau \simeq \begin{cases} \Delta t \times i(t) & \text{Forward Euler (FE)} \\ \Delta t \times i(t + \Delta t) & \text{Backward Euler (BE)} \\ \frac{\Delta t}{2} [i(t) + i(t + \Delta t)] & \text{Trapezoidal (TR)} \end{cases} \quad (4.2)$$

Inserindo as aproximações 4.2 em 4.1, tem-se:

$$v(t + \Delta t) \simeq v(t) + \frac{1}{C} \cdot \begin{cases} \Delta t \times i(t) \\ \Delta t \times i(t + \Delta t) \\ \frac{\Delta t}{2} [i(t) + i(t + \Delta t)] \end{cases} \quad (4.3)$$

As aproximações 4.3 podem ser escritas como:

$$v(t + \Delta t) \simeq v(t) + \Delta t \cdot \begin{cases} \frac{i(t)}{C} & (FE) \\ \frac{i(t+\Delta t)}{C} & (BE) \\ \frac{i(t)+i(t+\Delta t)}{2C} & (TR) \end{cases} \quad (4.4)$$

A aproximação (FE) pode ser representada eletricamente, conforme a Figura 4.5. Neste caso o capacitor será representado por uma fonte de tensão independente.

A aproximação (BE) pode ser representada eletricamente por um resistor em série com uma fonte de tensão independente, conforme mostrado na Figura 4.6.

A aproximação (TR) pode ser representada eletricamente por um resistor em série com uma fonte de tensão independente, conforme mostrado na Figura 4.7.

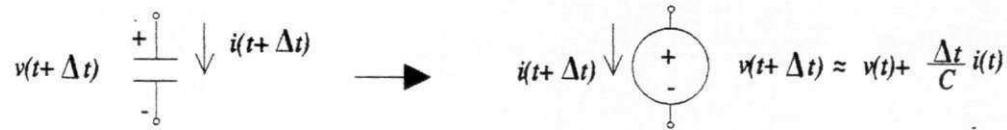


Figura 4.5: Modelo para o capacitor segundo a aproximação Forward Euler

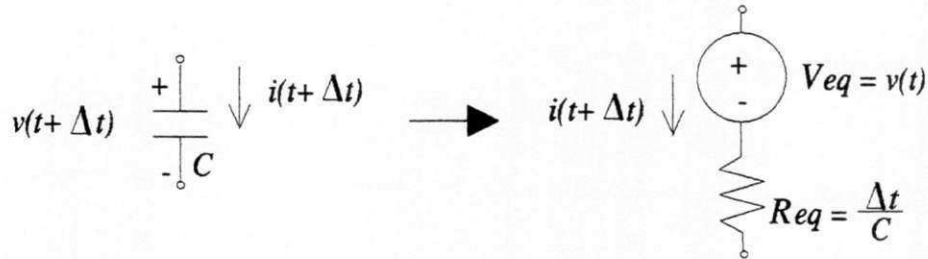


Figura 4.6: Modelo para o capacitor segundo a aproximação Backward Euler

Considere o indutor mostrado na Figura 4.8. A equação básica para a tensão sobre os terminais do mesmo é a seguinte:

$$v(t) = L \frac{di(t)}{dt} \quad (4.5)$$

$$i(t + \Delta t) = i(t) + \frac{1}{L} \int_t^{t+\Delta t} v(t) dt \quad (4.6)$$

Aplicando a aproximação (FE) na integral da equação 4.6 obtém-se:

$$i(t + \Delta t) \approx i(t) + \frac{\Delta t}{L} v(t) \quad (4.7)$$

A aproximação (FE) pode ser representada eletricamente, conforme a Figura 4.9.

Aplicando a aproximação (BE) na integral da equação 4.6 obtém-se:

$$i(t + \Delta t) = i(t) + \frac{\Delta t}{L} v(t + \Delta t) \quad (4.8)$$

A aproximação (BE) pode ser representada eletricamente, conforme mostrado na Figura 4.10.

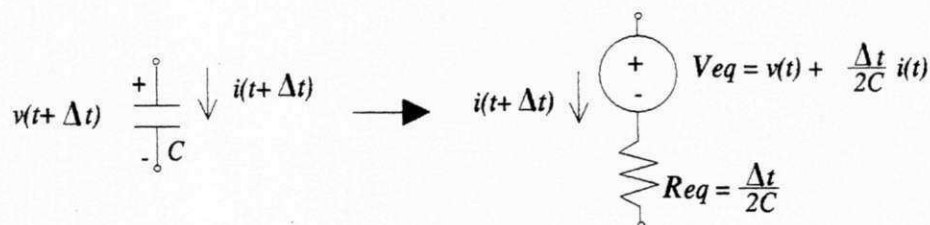


Figura 4.7: Modelo para o capacitor segundo a aproximação Trapezoidal.

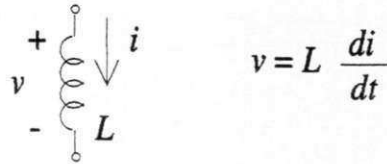


Figura 4.8: Indutor linear e invariante no tempo

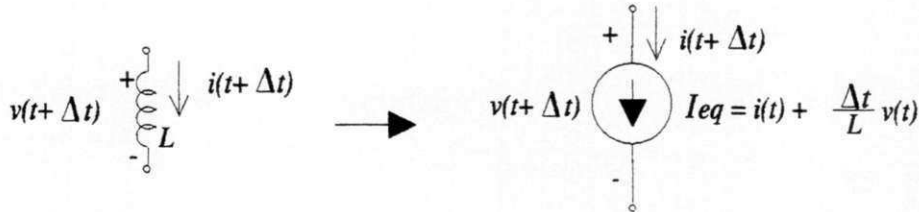


Figura 4.9: Modelo para o indutor segundo a aproximação Forward Euler

Aplicando a aproximação (TR) na integral da equação 4.6 obtém-se:

$$i(t + \Delta t) = i(t) + \frac{\Delta t}{2L} v(t) + \frac{\Delta t}{2L} v(t + \Delta t) \quad (4.9)$$

A aproximação (TR) pode ser representada eletricamente, conforme mostrado na Figura 4.11.

Estas aproximações produzem resultados precisos com um Δt tendendo a zero. No programa SICON1 foram adotadas as aproximações (BE) na implementação dos capacitores e dos indutores.

4.4 Utilização dos Modelos

Na Figura 4.12 temos um conversor que servirá de exemplo de aplicação dos modelos estudados. Na Figura 4.13 temos a troca dos componentes do conversor pelos seus respectivos modelos em redes resistivas. Os dispositivos semicondutores presentes na Figura 4.12 são substituídos por resistores na Figura 4.13 (S está associado a $G5$ e Do a

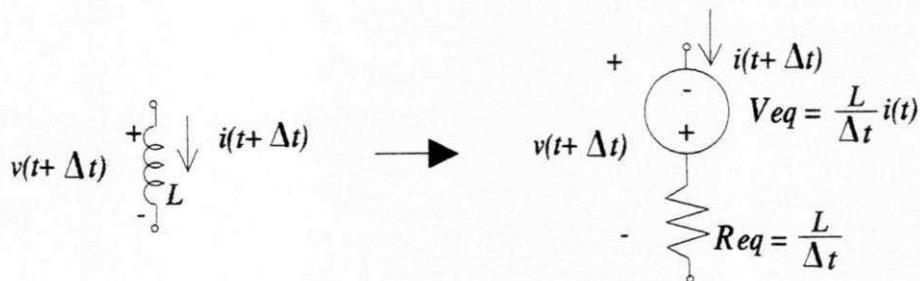


Figura 4.10: Modelo para o indutor segundo a aproximação Backward Euler

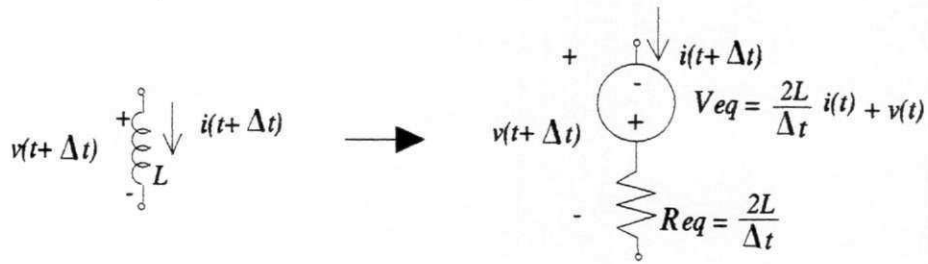


Figura 4.11: Modelo para o indutor segundo a aproximação Trapezoidal

G6). Os indutores e capacitores foram substituídos por um dos seus modelos descritos anteriormente (L_r esta associado a V_3 e G_3 , L_o a V_4 e G_4 , C_r a V_2 e G_2). A ordem para a numeração dos braços é a seguinte: fontes de tensão e corrente, capacitores e indutores, dispositivos semicondutores, e por último os resistores.

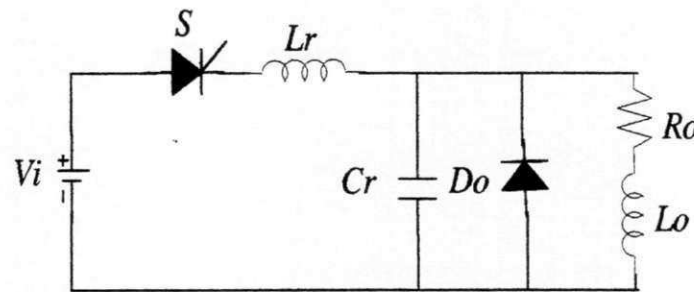


Figura 4.12: Conversor Buck

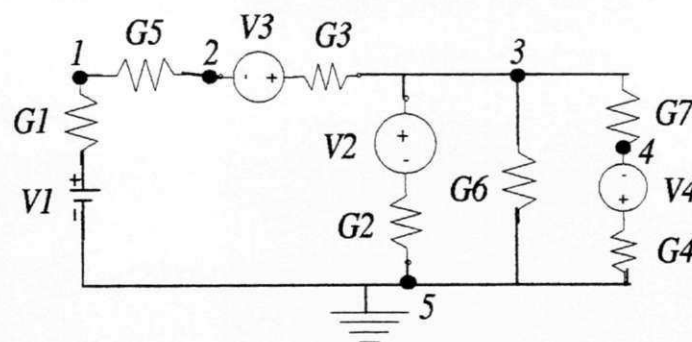


Figura 4.13: Modelo resistivo adotado para o conversor buck

4.5 Escolha do Período de amostragem

O período de amostragem para o programa SICON1 é fixo, ou seja, ele não muda em função das várias etapas de funcionamento do conversor. A escolha do período de amostragem é feita pelo usuário, e deve ser o menor possível (uma fração pequena

do tempo final de simulação), para que se possa verificar, da melhor forma possível, verificar comportamento do circuito, além de permitir que as aproximações para os capacitores e indutores se aproximem do ideal.

4.6 Estrutura do Programa

O programa SICON1 foi estruturado de maneira a permitir que o usuário tenha uma completa visão do funcionamento do mesmo. A sequência de utilização é contruída de forma a permitir que o usuário tenha um mínimo possível de conhecimento sobre o funcionamento do conversor.

O programa SICON1 será executado através do comando *sicon1*, realizado a partir da tela principal do MATLAB.

Antes de executar o programa é necessário que o usuário numere os nós e os braços do circuito conversor em estudo. A ordem a ser seguida para numerar os braços é a descrita na seção anterior.

4.6.1 Entrada de Dados

A primeira etapa do programa corresponde a escolha do tipo de circuito conversor, o que é mostrado na primeira tela do programa SICON1, conforme descrito na Figura 5.1. Uma subrotina é elaborada para cada tipo de conversor escolhido (conversor CC-CC, conversor CC-CA, conversor CA-CC, conversor CA-CA), e de acordo com a escolha o programa executa a subrotina correspondente.

A segunda etapa diz respeito a escolha da representação a ser adotada para simular o conversor. Teremos o conversor simulado segundo uma topologia fixa ou segundo uma topologia variável. Essa escolha é feita através da tela apresentada na Figura 4.15.

A terceira etapa diz respeito a entrada da quantidade de nós do circuito, do número de elementos, e do nó de referência. Esta entrada de dados é feita através da tela de apresentação mostrada na Figura 4.16.

A quarta etapa corresponde a escolha do tipo de componente a ser inserido no conversor. Esta escolha é feita através da tela mostrada na Figura 4.17. De acordo com o tipo de elemento, tem-se a determinação dos respectivos parâmetros, como exemplo, observa-se a entrada dos parâmetros para um capacitor, de acordo com a Figura 4.18.

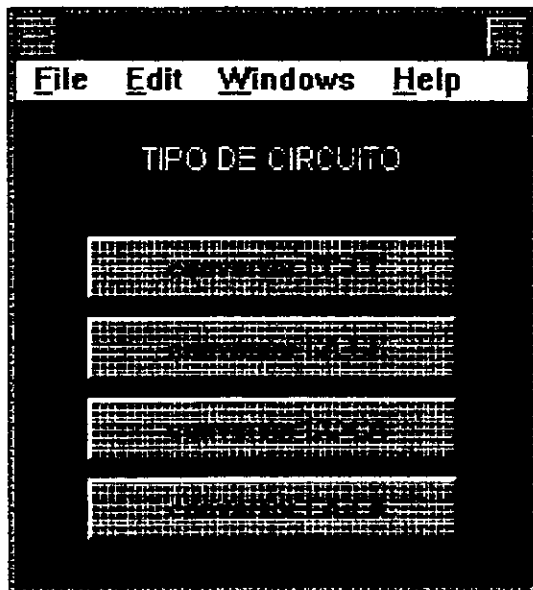


Figura 4.14: Tela de entrada para a escolha do tipo de conversor

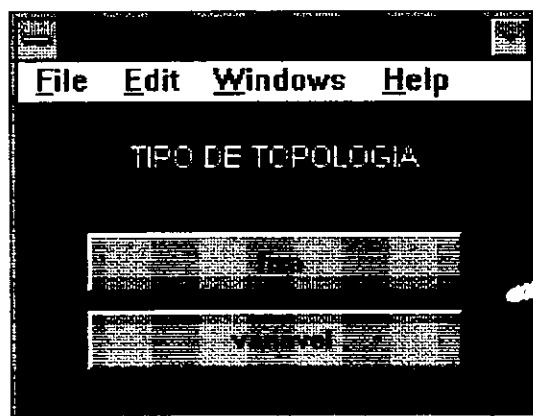


Figura 4.15: Tela de entrada para a escolha do tipo de topologia

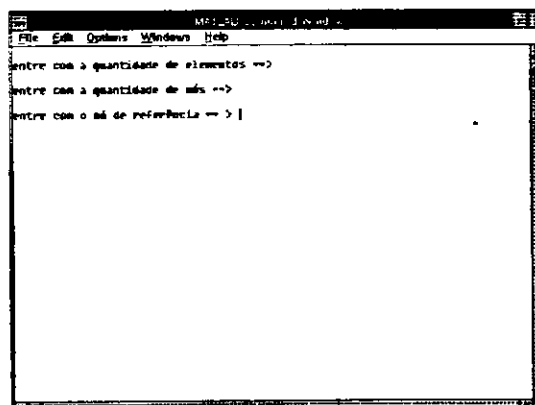


Figura 4.16: Tela para entrar com a quantidade de nós, com o número de elementos e com o nó de referência.

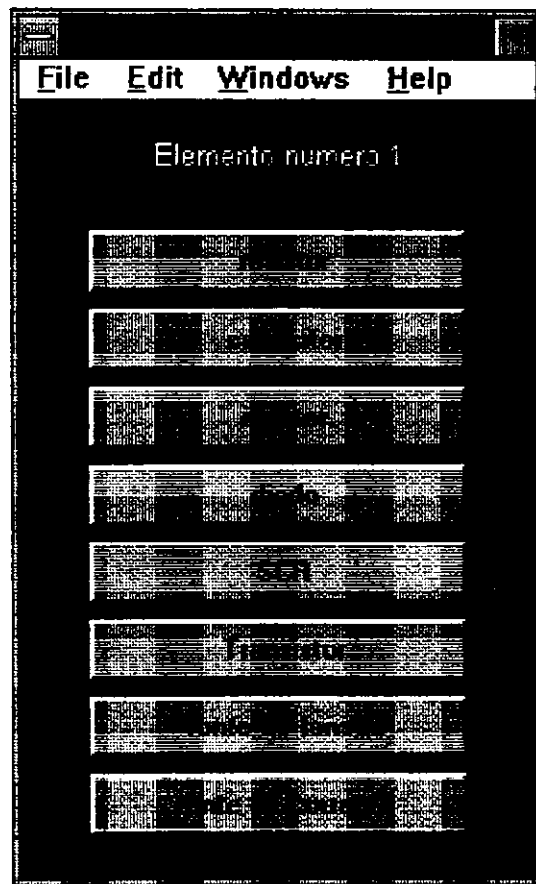


Figura 4.17: Tela para a escolha do tipo de elemento a ser inserido no circuito conversor.

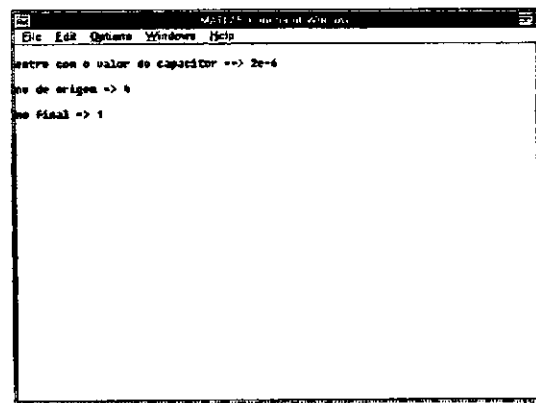


Figura 4.18: Tela para a entrada dos parâmetros para um capacitor.

4.6.2 Fluxograma do Programa

O fluxograma do programa SICON1 é mostrado na Figura 4.19. O programa é escrito diretamente na linguagem de programação do MATLAB, seguindo o que é descrito pelo fluxograma. As etapas de funcionamento do programa são sintetizadas através deste fluxograma.

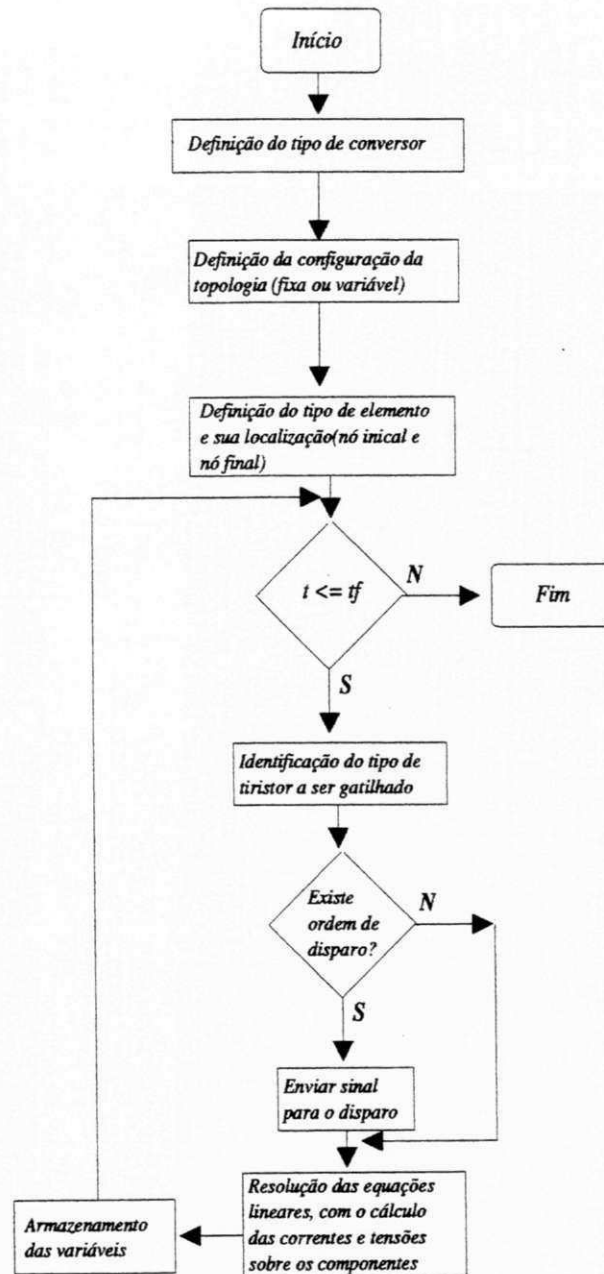


Figura 4.19: Fluxograma do programa SICON1

4.7 Resultados de Simulação

Os resultados obtidos com o programa SICON1 são descritos nesta seção. São mostrados os resultados obtidos pela simulação de três conversores. Cabe citar que serão mostrados resultados obtidos com o uma topologia variável como os obtidos com uma topologia fixa.

4.7.1 Conversor Buck

Na Figura 4.20 tem-se a a representação básica do conversor Buck a comutação com corrente nula. Pode-se dividir a operação do conversor segundo quatro estágios, descritos a seguir:

Estágio 1 $[t_0, t_1]$: Quando a chave S é ligada, a corrente cresce linearmente na chave, com di/dt limitado pelo indutor ressonante. O intervalo termina quando a corrente da chave atinge o valor da corrente de carga (I_o), quando o diodo deixa de conduzir.

Estágio 2 $[t_1, t_2]$: Neste período temos um ciclo ressonante entre L_R e C_R . O circuito é ressonante até i_s atingir zero em t_2 , onde a chave S passa a ficar aberta.

Estágio 3 $[t_2, t_3]$: Com a chave S aberta, o capacitor C_R descarrega sobre a carga, até atingir a tensão zero em t_3 .

Estágio 4 $[t_3, t_4]$: Com a tensão em C_R igual a zero, o diodo D_o passa a conduzir a corrente de carga. Este período é conhecido como período de roda-livre.

A partir do que foi descrito acima, temos as curvas teóricas mostradas na figura 4.21.

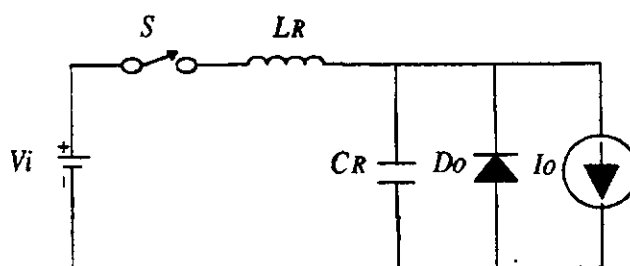


Figura 4.20: Representação do conversor Buck

Usou-se o programa SICON1 na sua opção por simulação por topologia fixa para a análise e simulação do circuito. Os valores dos componentes, usados na simulação do conversor, são os seguintes:

$$V_i = 100 \text{ V}$$

$$I_o = 10 \text{ A}$$

$$L_R = 100 \mu\text{H}$$

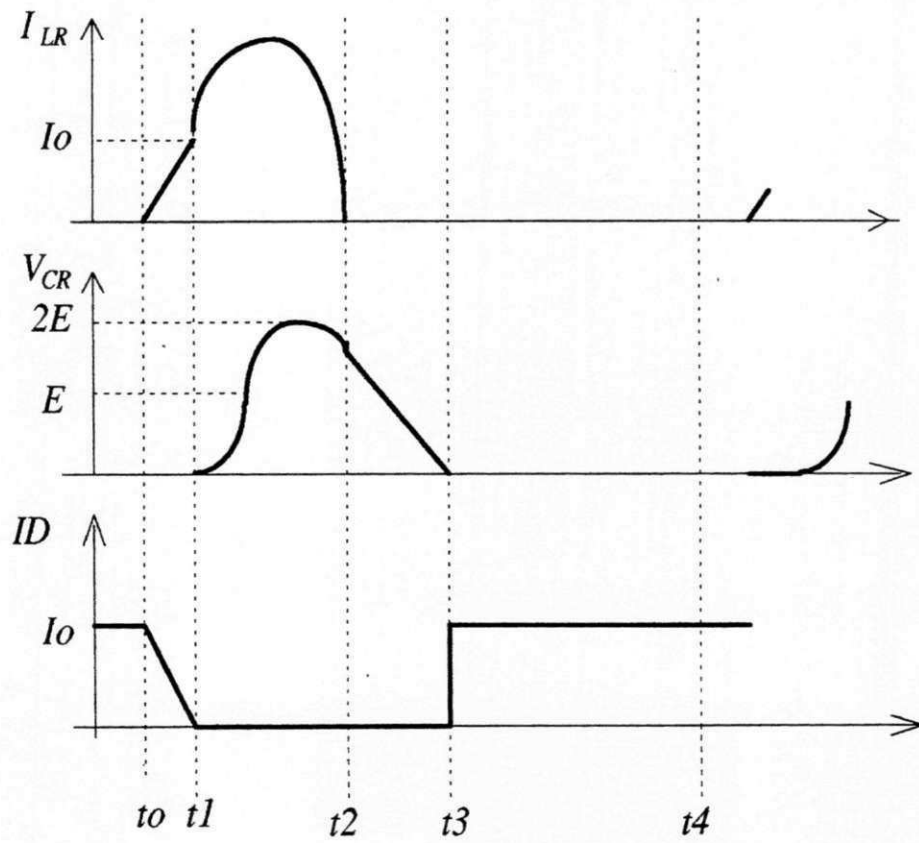


Figura 4.21: Curvas teóricas para o conversor Buck

$$C_R = 20 \mu\text{F}$$

$$f_s = 1 \text{ KHz (frequência de chaveamento)}$$

Na Figura 4.22 temos a representação de algumas curvas obtidas com a simulação do conversor. Observa-se que estes resultados estão de acordo com a análise teórica, conseqüentemente, refletem todas as etapas de funcionamento do conversor.

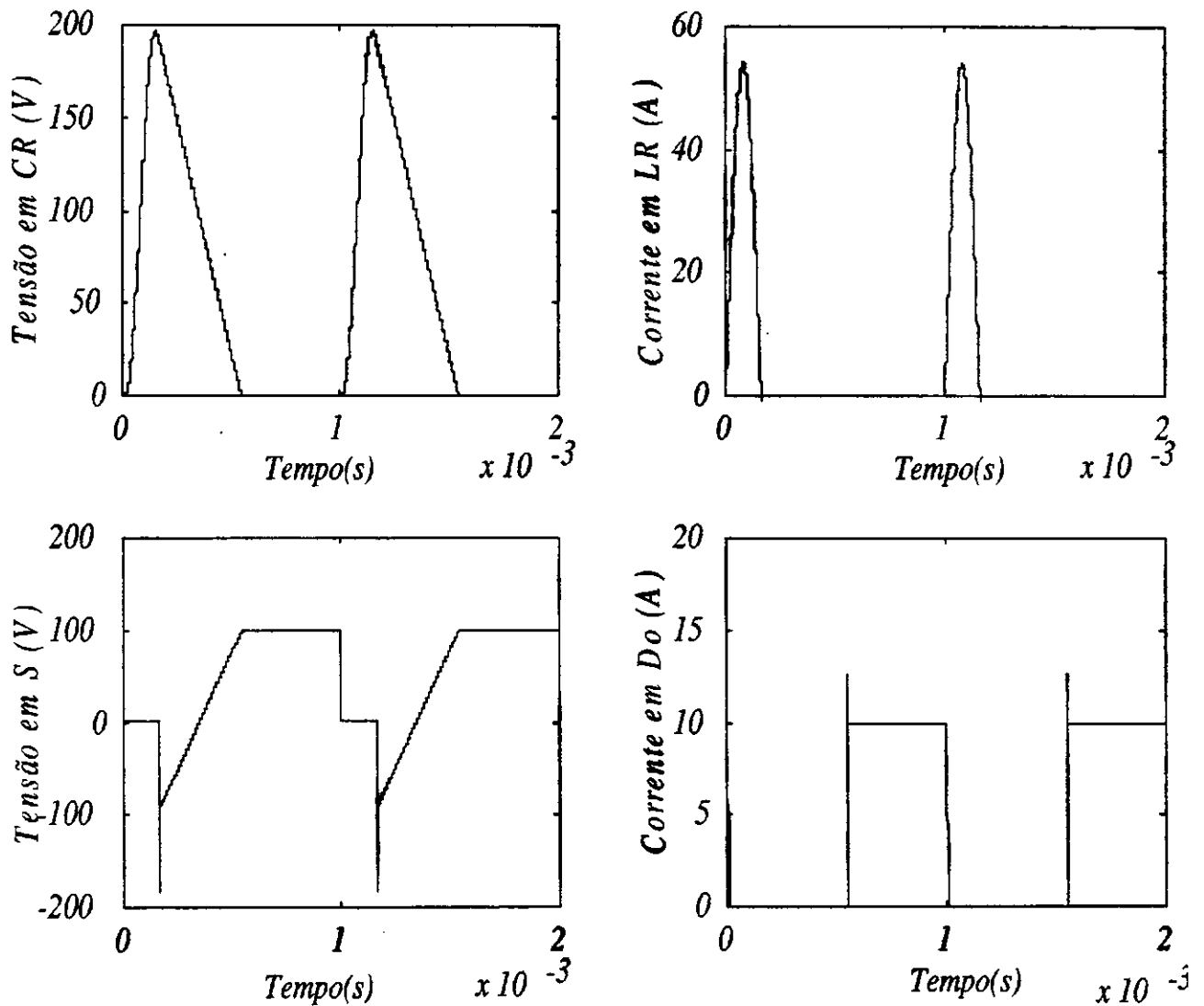


Figura 4.22: Resultados obtidos na simulação do conversor Buck

4.7.2 Retificador Trifásico Não Controlado

Na Figura 4.23 tem-se a a representação básica do retificador trifásico não controlado. Os dispositivos semicondutores utilizados no circuito são exclusivamente diodos. Os diodos que conduzem são aqueles conectados às duas fases de alimentação com a maior tensão entre eles, naquele instante, ou seja, quando v_A é a fase mais positiva $D1$ conduz,

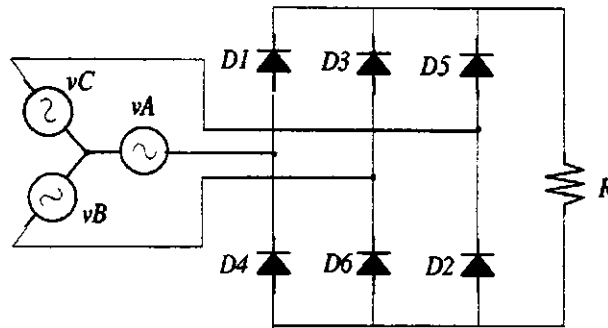


Figura 4.23: Representação do Retificador Trifásico Não Controlado

e durante esse período primeiro v_B é mais negativa com o diodo $D6$ conduzindo, até que v_C torne-se mais negativa, quando a corrente em $D6$ comuta para $D2$. A tensão na carga segue uma tensão senoidal de seis pulsos, durante um ciclo, sendo: $v_A - v_B$, $v_A - v_C$, $v_B - v_C$, $v_B - v_A$, $v_C - v_A$, $v_C - v_B$, todos tendo o valor máximo da tensão de linha, isto é, $\sqrt{3}$ vezes a tensão de fase. A forma de onda de tensão no diodo V_{D1} pode ser determinada como a diferença entre a tensão da fase A e a tensão no topo da carga em relação ao neutro (N) da fonte. O pico de tensão reversa que aparece sobre o diodo é o valor máximo da tensão de linha. Algumas formas de onda teóricas para o retificador trifásico não controlado são mostradas na Figura 4.24.

Os valores dos elementos do circuito, para efeito de simulação, são os seguintes:

$$v_A = 200 \text{ sen}(wt)$$

$$v_B = 200 \text{ sen}(wt - 2\pi/3)$$

$$v_C = 200 \text{ sen}(wt - 4\pi/3)$$

$$R = 20 \Omega$$

O programa SICON1 foi executado na sua opção por topologia variável e na Figura 4.25 temos a representação de algumas curvas obtidas com a simulação do conversor.

Observa-se que os resultados obtidos com o programa SICON1, refletem o comportamento teórico do retificador, mostrado na Figura 4.24.

4.7.3 Inversor Trifásico

O inversor trifásico a ser simulado é o apresentado na Figura 4.26. As chaves estão numeradas em uma sequência tal, que seus disparos produzem tensões numa sequência de fases positiva, V_{AB} , V_{BC} e V_{CA} . Existem duas possibilidades de operação: uma em que apenas duas chaves conduzem simultaneamente, outra em que há três chaves em condução. Na primeira possibilidade, supõe-se que ao final de cada período de 120° um circuito de comutação desliga o tiristor apropriado. As correntes de carga são quase

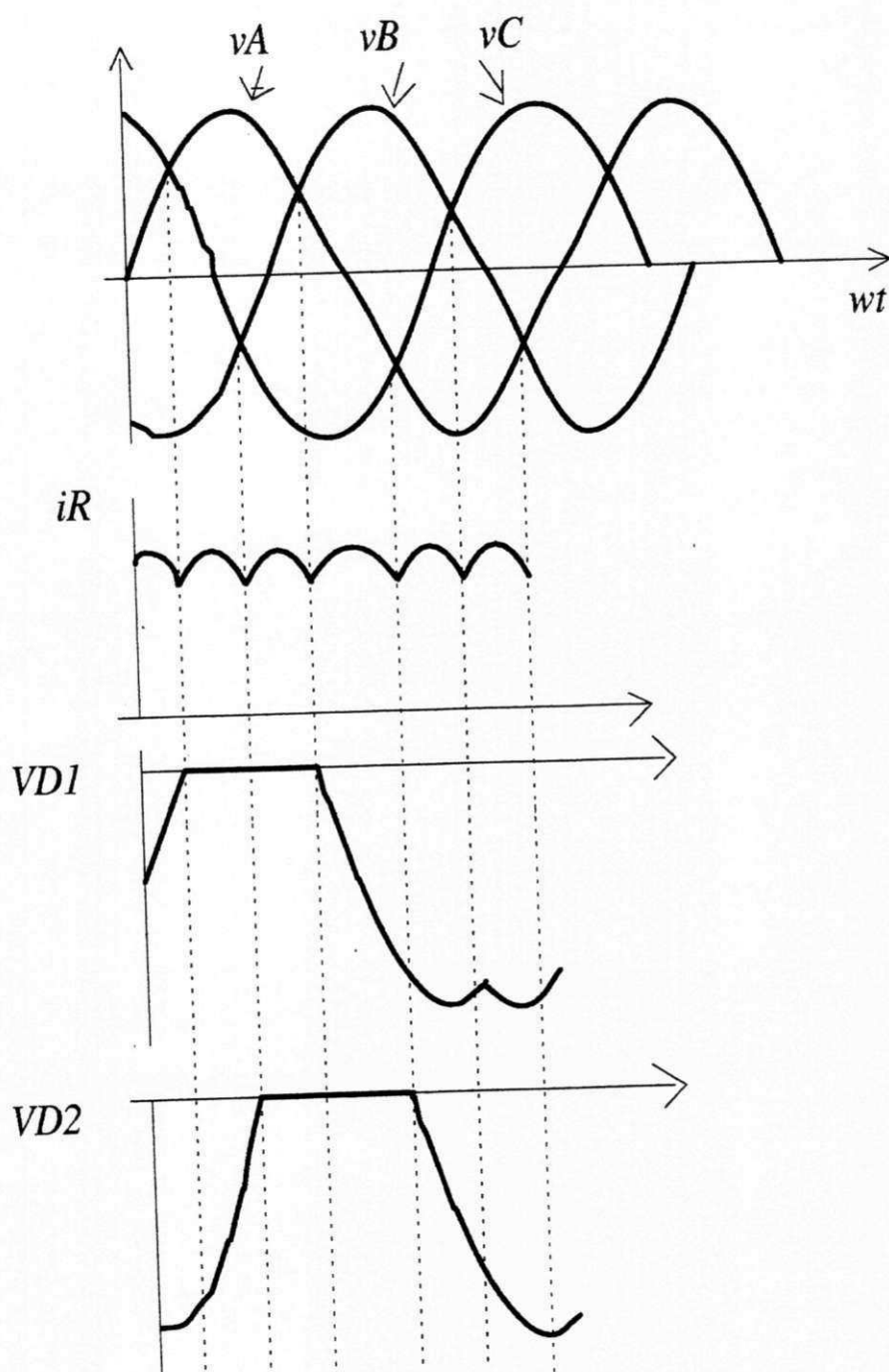


Figura 4.24: Formas de onda para o retificador trifásico não controlado

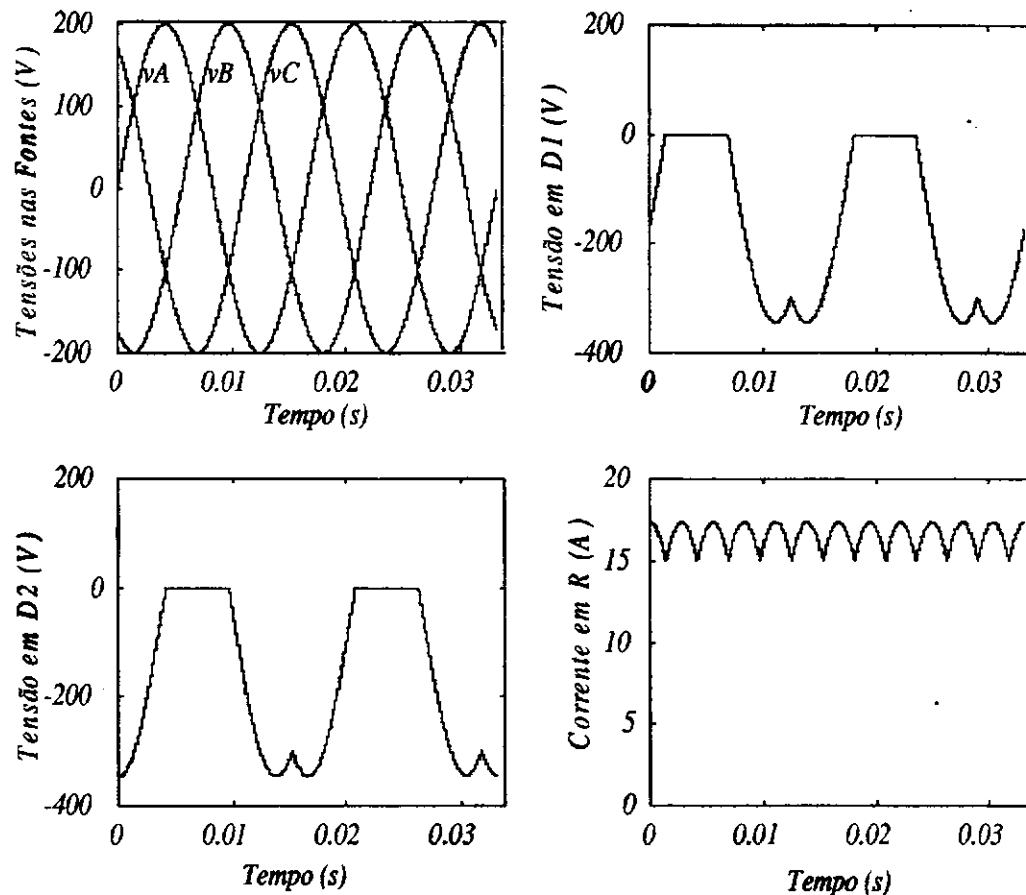


Figura 4.25: Resultados obtidos na simulação do Retificador Trifásico

quadradas, com cada tiristor conduzindo a corrente de carga por um terço do ciclo. A forma de onda em degraus para uma tensão de linha será modificada se qualquer indutância estiver presente na carga, pois a transferência da corrente de carga para os diodos efetivamente manterá as chaves fechadas por um período maior do que 120° .

Algumas formas de onda teóricas, para a possibilidade de condução de 120° , são mostradas na Figura 4.27.

A possibilidade de condução de 120° para o inversor trifásico, foi a escolhida para a simulação. O programa SICON1 foi executado na sua opção por topologia fixa, onde as demais características do circuito são as seguintes:

$$V_i = 100 \text{ V}$$

$$R = 20 \ \Omega$$

$$f_s = 500 \text{ Hz (frequência de chaveamento)}$$

Os resultados obtidos após a simulação são mostrados na Figura 4.28

As curvas apresentadas na Figura 4.28 reproduzem os resultados teóricos mostrados na Figura 4.27, refletindo, também, todas as etapas de funcionamento do circuito.

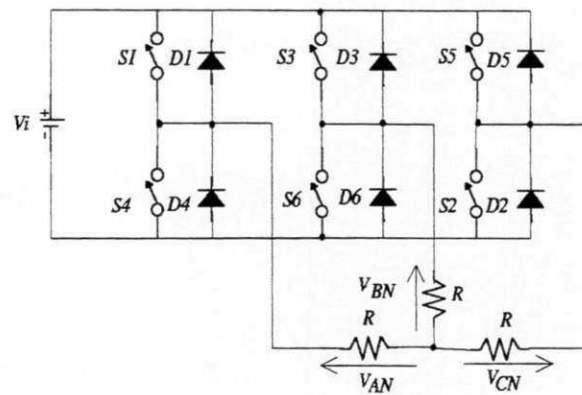


Figura 4.26: Representação do Inversor Trifásico

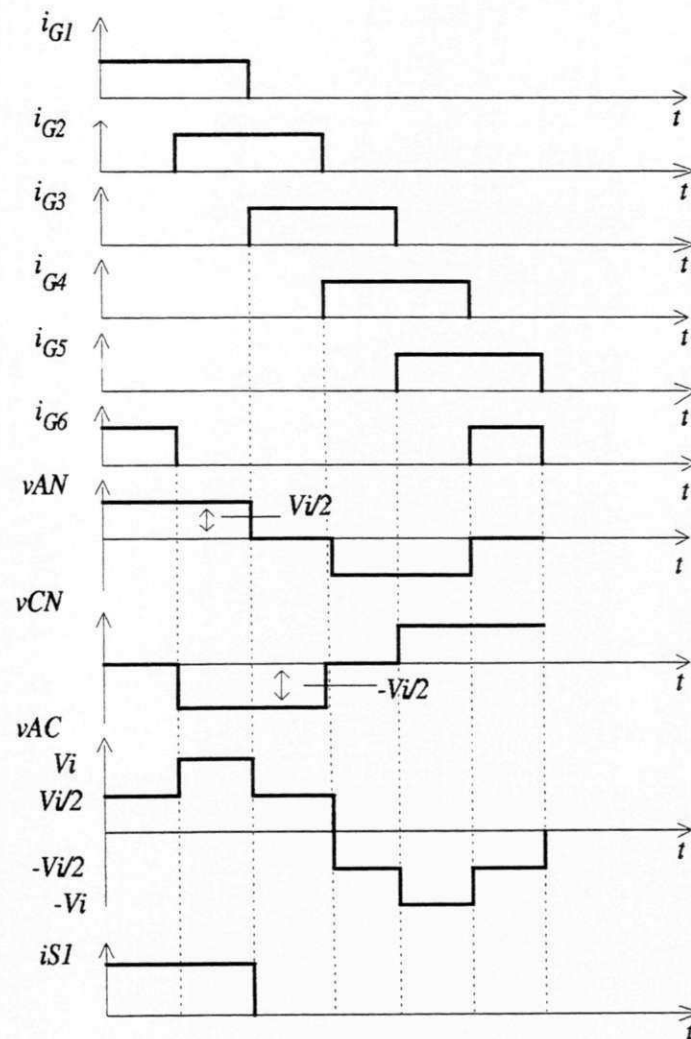


Figura 4.27: Formas de onda para o inversor trifásico

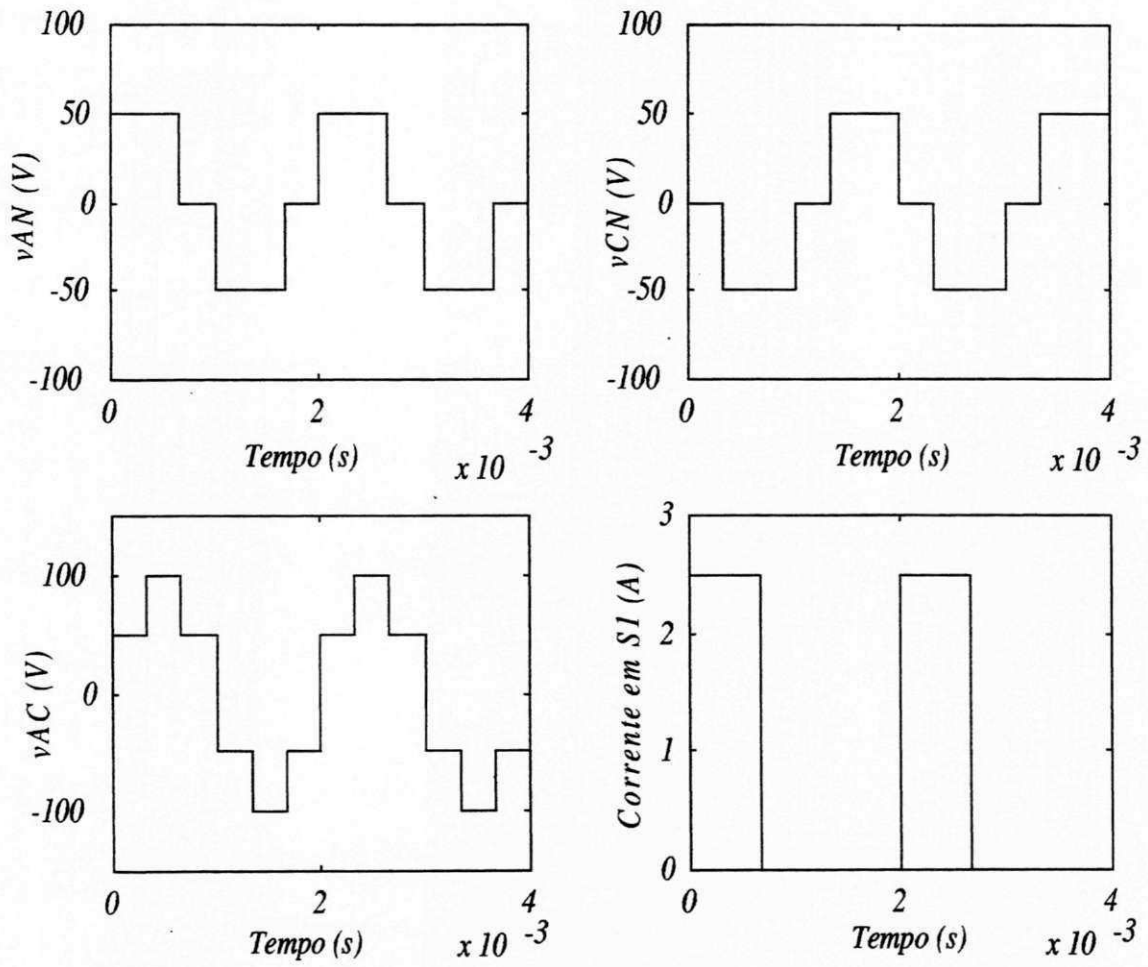


Figura 4.28: Resultados obtidos na simulação do Inversor Trifásico

4.8 Conclusão

Neste capítulo foi apresentado o programa SICON1, um toolbox na área de Eletrônica de Potência, desenvolvido no ambiente MATLAB. O programa baseia-se no modelamento do conversor estático a partir de equações oriundas de uma análise de nós de redes lineares invariantes. Os indutores e capacitores são representados por modelos lineares, o que torna as equações do programa mais simples. O período de amostragem do programa é fixo e deve ser o menor possível, para que as aproximações para os capacitores e indutores se aproximem do ideal. Os resultados obtidos com o programa SICON1 executado tanto com o uso de uma topologia fixa, como com o uso de uma topologia variável, refletem as análises teóricas, ou seja, todas as etapas de funcionamento dos circuitos simulados. O programa SICON1 executado na opção por topologia fixa fornece um tempo de simulação menor do que o programa sendo executado na opção por topologia variável, levando-se em conta um mesmo circuito a ser simulado. Isto deve-se ao fato do maior grau de complexidade das subrotinas desenvolvidas para a implementação do programa por topologia variável.

Capítulo 5

Programa SICON2

5.1 Introdução

Neste capítulo, é tratado o programa SICON2 empregado na simulação digital de conversores estáticos.

O programa SICON2 utiliza a teoria de resolução de circuitos através de equações de estado.

O objetivo é mostrar todo o desenvolvimento adotado para a elaboração do programa.

São apresentados vários resultados de simulação usando o programa SICON2.

5.2 Estrutura do Programa SICON2

O programa consta basicamente de dois módulos, o primeiro corresponde a entrada da estrutura do conversor e o segundo corresponde a simulação do conversor.

5.2.1 Entrada de Dados

A primeira etapa do programa corresponde a escolha do tipo de circuito conversor, o que é mostrado na primeira tela do programa SICON2, conforme descrito na Figura 5.1. Uma subrotina é elaborada para cada tipo de conversor escolhido (conversor CC-CC, conversor CC-CA, conversor CA-CC, conversor CA-CA), e de acordo com a escolha o programa executa a subrotina correspondente.

A segunda etapa diz respeito a entrada do número de braços do circuito, número de nós, número de capacitores em árvore, número de indutores em enlace, número de diodos e interruptores em enlace, número de diodos e interruptores em árvore

número de resistências em enlace, e o número de condutâncias em árvore, assim como a matriz F . Esta entrada de dados é feita através da tela de apresentação mostrada na Figura 5.2.

A terceira etapa corresponde a definição dos valores dos elementos do conversor. Como exemplo, observa-se a entrada dos parâmetros para um capacitor, de acordo com a Figura 5.3.

No segundo modulo do programa é realizada a simulação do conversor em estudo. Neste modulo é feita a resolução das equações de estado que representam o conversor, determinação das variáveis de saída, a interpolação para a determinação do momento de mudança de estado dos interruptores e dos diodos, e o teste para determinar se os interruptores e os diodos estão em condição de conduzir.

5.2.2 A modelagem dos dispositivos semicondutores

No programa SICON2, o dispositivo semicondutor fechado ou aberto é representado por uma resistência pequena ou grande, respectivamente. As matrizes F , L e C (definidas no capítulo 2) são constantes e as matrizes R e G (definidas no capítulo 2) dependem do estado do dispositivo: aberto ou fechado.

5.2.3 Solução da equação de estado contínua

A solução da equação de estado contínua $\frac{dx}{dt} = Ax + Bu$ é obtida com a utilização da forma discreta equivalente. A equação de estado equivalente se escreve:

$$x(k+1) = F(k)x(k) + H(k)u(k) \quad (5.1)$$

onde $F(k)$ e $H(k)$ são expressos da seguinte forma:

$$\begin{aligned} F(k) &= e^{ATe} \\ H(k) &= B \int_0^{Te} e^{At} dt \end{aligned}$$

onde Te corresponde ao período de amostragem

5.2.4 A escolha do período de amostragem

Na escolha do período de amostragem alguns cuidados devem ser tomados, conforme descrito abaixo:

- Para uma dada precisão o tempo de cálculo deve ser o menor possível.

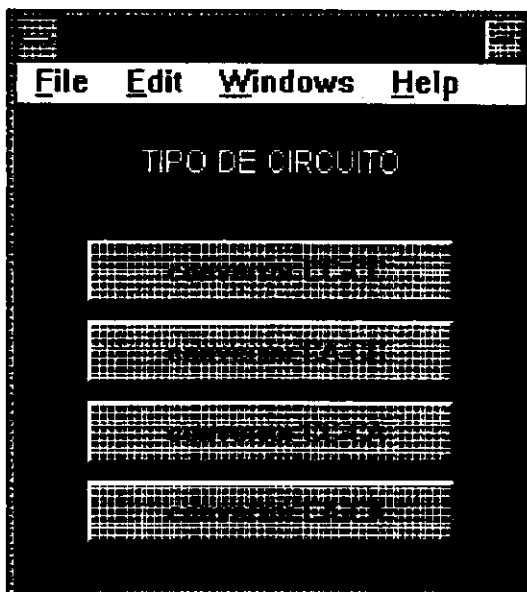


Figura 5.1: Tela de entrada para a escolha do tipo de conversor

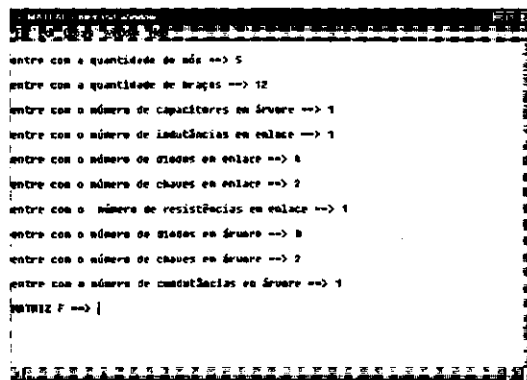


Figura 5.2: Segunda tela de apresentação do programa SICON2

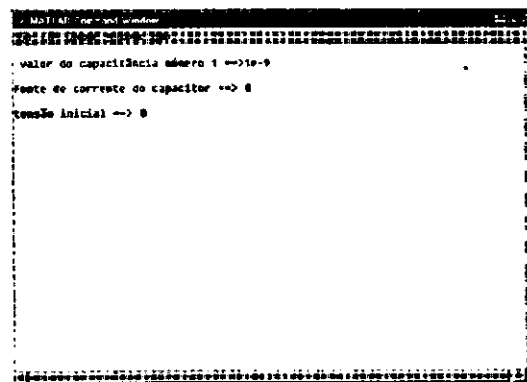


Figura 5.3: Tela para a entrada dos parâmetros para um capacitor.

- Deve-se garantir que não haja perda de nenhuma comutação de um dado semiconductor.
- Obter todas as informações necessárias sobre o circuito a ser estudado com a menor quantidade possível de pontos de cálculo.

Na simulação de circuitos a semiconductor, a determinação com precisão do instante da comutação constitui um aspecto importante. Neste tocante, o valor do período de amostragem ou passo de cálculo, desempenha um papel essencial.

Pode-se ter um período de amostragem constante [33] ou um período de amostragem variável [13]. No método a passo de amostragem constante, o período de amostragem é obtido, geralmente, tomando-se uma fração da menor constante de tempo do circuito, considerando-se todas as possibilidades de estado dos dispositivos semicondutores. No método a passo variável a idéia principal consiste em se obter um passo de cálculo adaptativo com a dinâmica do circuito a ser simulado. Pode-se otimizar o cálculo, fazendo com que o período de amostragem seja determinado em função dos autovalores da matriz A .

Método a passo variável

Pode-se otimizar o cálculo, fazendo com que o período de amostragem seja determinado em função dos autovalores de A .

A dinâmica do sistema de estado $\frac{dx}{dt} = Ax + Bu$ é caracterizada pela matriz A . Os autovalores de A fornecem as constantes de tempo e as frequências próprias do sistema. Para que o sistema discreto equivalente represente de maneira correta o sistema contínuo original, o período de amostragem adequado será função da parte imaginária e real dos autovalores de A [13], já que os autovalores de A refletem a dinâmica do circuito.

Para um autovalor real (λ), escolhe-se um período de amostragem Te tal que:

$$Te = \frac{1}{k_1 * |\lambda|} \quad (5.2)$$

Para autovalores complexos conjugados ($\lambda, \bar{\lambda}$), o período de amostragem Te é escolhido da seguinte maneira:

$$Te = \frac{|(2 * \pi) / \text{Im}(\lambda)|}{k_2} \quad (5.3)$$

onde k_1 e k_2 dependem da precisão desejada.

Estudos mostram que os valores adequados para k_1 e k_2 situam-se entre 3 e 5 para k_1 , e entre 20 e 50 para k_2 [13].

É comum existir uma grande dispersão de autovalores, com a presença de grandes e pequenos autovalores, associados, respectivamente, aos modos rápidos e lentos (dominantes) do circuito. Para não se utilizar períodos de amostragem muito pequenos, pode-se, por exemplo, desprezar os autovalores grandes ou utilizar uma média dos períodos de amostragem associados a cada autovalor (média geométrica, por exemplo). No programa SICON2 existem estas duas possibilidades.

5.2.5 Interpolação

Para melhor detectar o instante exato de abertura ou fechamento do semiconductor, utilizou-se no programa uma interpolação linear dos valores de corrente ou de tensão sobre eles. Esta interpolação é calculada em função dos valores no instante k e no instante $k + 1$. A interpolação é ilustrada na Figura 5.4 .

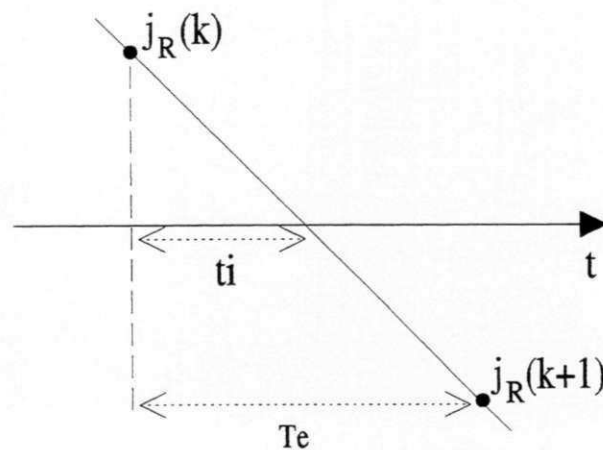


Figura 5.4: Representação da interpolação

Detectada uma mudança de estado no semiconductor, a equação de estado será recalculada, a partir do instante k , com um novo período de amostragem dado por:

$$t_i = \frac{-T e^{j_R(k)}}{[j_R(k+1) - j_R(k)]} \quad (5.4)$$

Onde $j_R(k)$ é a corrente sobre o dispositivo semiconductor no instante k , e $j_R(k+1)$ é a corrente sobre o dispositivo semiconductor no instante $k + 1$.

Outra possibilidade de algoritmo para se obter um período de amostragem variável, assim como uma maneira diferente de detectar-se o instante de mudança de estado é mostrado em [34].

5.2.6 Procedimento Para Montar a Matriz F

Os seguintes passos devem ser seguidos para montar a matriz F :

- a) Escolhe-se uma árvore que contenha todos os nós do circuito e não possua nenhum percurso fechado.
- b) Os capacitores e as condutâncias devem pertencer à árvore, enquanto que as indutâncias e resistências devem pertencer aos enlaces.
- c) As fontes de corrente presentes no circuito devem fazer parte da árvore, enquanto que as fontes de tensão devem constituir elemento de enlace.
- d) A ordem de numeração para os elementos deve ser a seguinte: primeiro numera-se os elementos de enlace e depois os elementos de árvore. A numeração dos elementos de enlace segue a seguinte ordem: primeiro numera-se os diodos, em seguida os tiristores, depois as resistências e por último as indutâncias. A numeração dos elementos em árvore deve ser a seguinte: primeiro numera-se os capacitores, em seguida diodos, depois os tiristores, e por último as condutâncias.
- e) Por vezes torna-se necessário fazer a equivalência entre fontes de tensão e fonte de corrente para poder determinar uma árvore para o circuito.
- f) Cada elemento que configura um enlace determina um percurso fechado fundamental, onde o sentido de referência do percurso fechado fundamental é o sentido dado ao elemento de enlace.

5.2.7 Exemplo de Montagem da Matriz F

O circuito mostrado na Figura 5.5 serve de exemplo de como obter a matriz F (matriz percurso fechado fundamental).

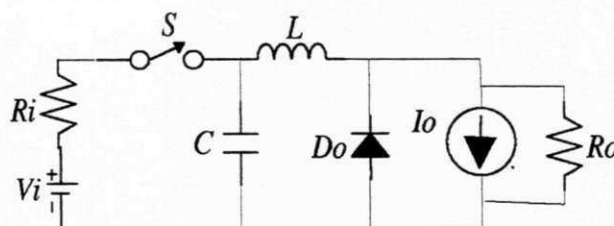


Figura 5.5: Exemplo de conversor

A Figura 5.6 mostra a disposição do circuito de acordo com o procedimento que deve ser feito para se obter a matriz F .

A árvore escolhida é mostrado em negrito. Cada elemento de enlace define um percurso fechado fundamental, desta forma tem-se três percursos fechados fundamentais. O sentido escolhido para percorrer cada percurso corresponde ao sentido dado ao

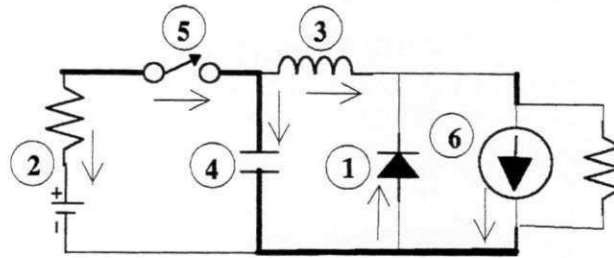


Figura 5.6: Configuração do conversor (árvore, elementos de enlace, elementos de árvore, e sentido dos elementos)

elemento de enlace correspondente. Os sentidos escolhidos definem a composição da matriz F . Os elementos da matriz F assumem os seguintes valores:

$F(i,j) = 1$ se o sentido do elemento da árvore coincide com o sentido do elemento de enlace que define o percurso fechado fundamental.

$F(i,j) = -1$ se o sentido não coincide com o sentido do elemento de enlace.

$F(i,j) = 0$ se o elemento não pertence ao percurso fechado fundamental.

Segundo estas regras obtém-se a seguinte matriz F para o circuito da Figura 5.6:

$$F = \begin{bmatrix} 0 & 0 & 1 \\ -1 & -1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

5.2.8 Fluxograma do Programa

O fluxograma do programa SICON2 é ilustrado na Figura 5.7. Na parte inicial do programa, o usuário define a estrutura e os parâmetros do circuito a ser simulado. Ou seja, a matriz F , os valores e o número de capacitores, indutores, resistores (indicando quais são semicondutores) e de fontes presentes no circuito, assim como a sequência de gatilhamento dos semicondutores. Inicialmente, os semicondutores são inicializados como abertos.

Definidos os parâmetros iniciais do circuito, o programa inicia uma sequência automática de cálculos. O período de amostragem é calculado, e as equações de estado discretas são determinadas. A solução destas equações fornece as variáveis de estado no instante T_e à frente $(k+1)$. São calculados os valores das variáveis de saída correspondente as tensões e correntes nos diversos elementos passivos. O estado do semicondutor é testado, e se não houver nenhuma mudança de estado nos semicondutores, o sistema

de equações anterior é novamente utilizado. Se houver mudança de estado dos semicondutores, passa a ser realizada a interpolação mencionada anteriormente, e assim calculado um novo período de amostragem e um novo sistema de equações de estado. Esta sequência se repete até o tempo final de simulação.

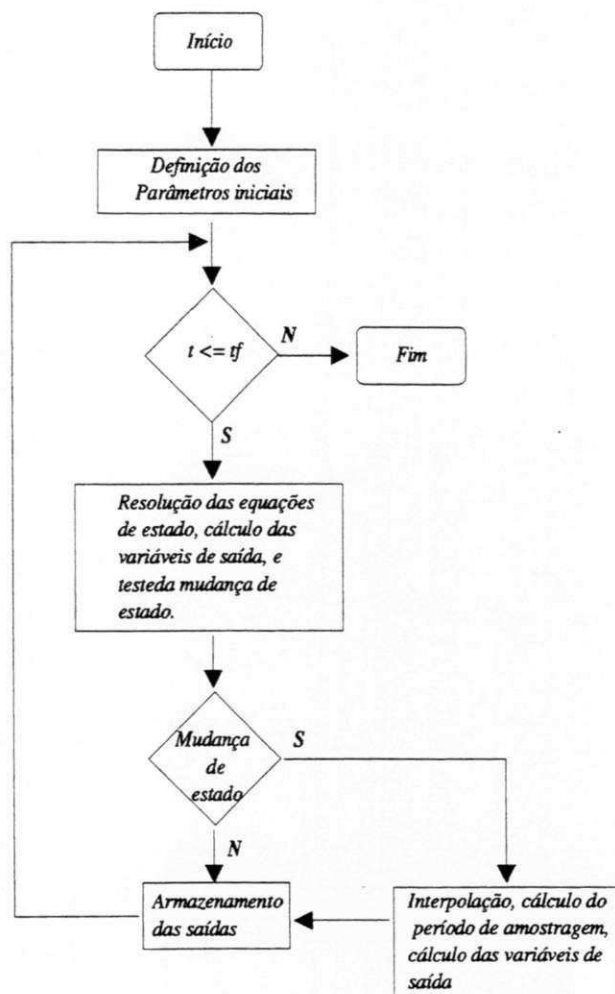


Figura 5.7: Fluxograma do programa SICON2

5.3 Resultados de Simulação

5.3.1 Conversor QRC Com Comutação a Corrente Nula (QRC-ZCS)

Para reduzir as perdas por chaveamento dos conversores PWM convencionais quando trabalham em altas frequências, surgiram inicialmente os conversores QRC's (Quasi Resonant Converter). A grande vantagem desses conversores é o fato de apresentarem comutação não-dissipativa, uma vez que são feitas sob tensão nula (ZVS - Zero Voltage

Switching) ou sob corrente nula (ZCS - Zero Current Switching) tornando-os mais eficientes em altas frequências.

O conversor QRC com comutação a corrente nula a ser simulado com o programa SICON2 é mostrado na Figura 5.8. Os dados relativos ao circuito são os seguintes:

$$V_i = 100 \text{ V}$$

$$R_i = 0.0001 \ \Omega$$

$$L = 600 \ \mu\text{H}$$

$$C = 5 \ \mu\text{F}$$

$$R_o = 10 \ \Omega$$

$$L_o = 10 \ \text{mH}$$

$$f_s = 500 \ \text{Hz}$$

Uma análise completa acerca do funcionamento do conversor é feita em [35].

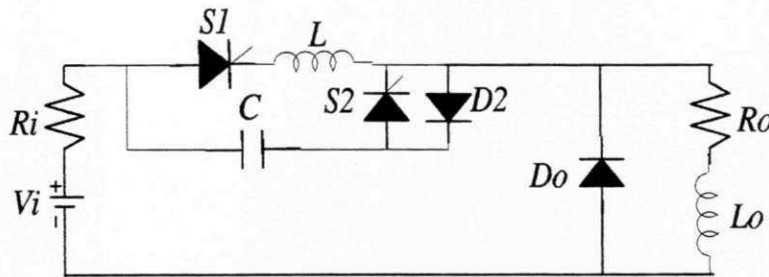


Figura 5.8: Conversor QRC-ZCS

A estrutura do circuito para a entrada de dados para a simulação é mostrada na Figura 5.9.

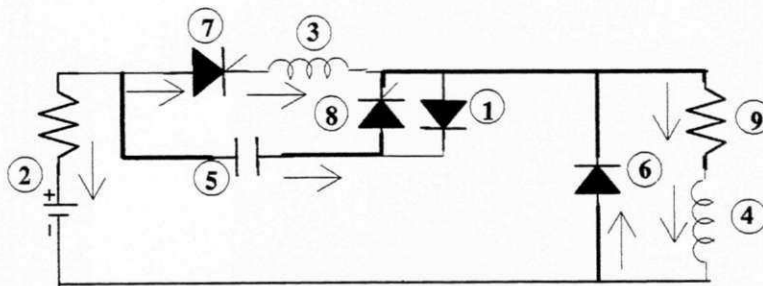


Figura 5.9: Disposição dos componentes para o conversor QRC-ZCS

A matriz F para o conversor em estudo, obtida através da Figura 5.9, é a seguinte:

$$F = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ -1 & 1 & 0 & -1 & 0 \\ -1 & 0 & 1 & -1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Os resultados obtidos em simulação são mostrados na Figura 5.10. S_2 é disparado no início e S_1 é disparado no meio do período de chaveamento.

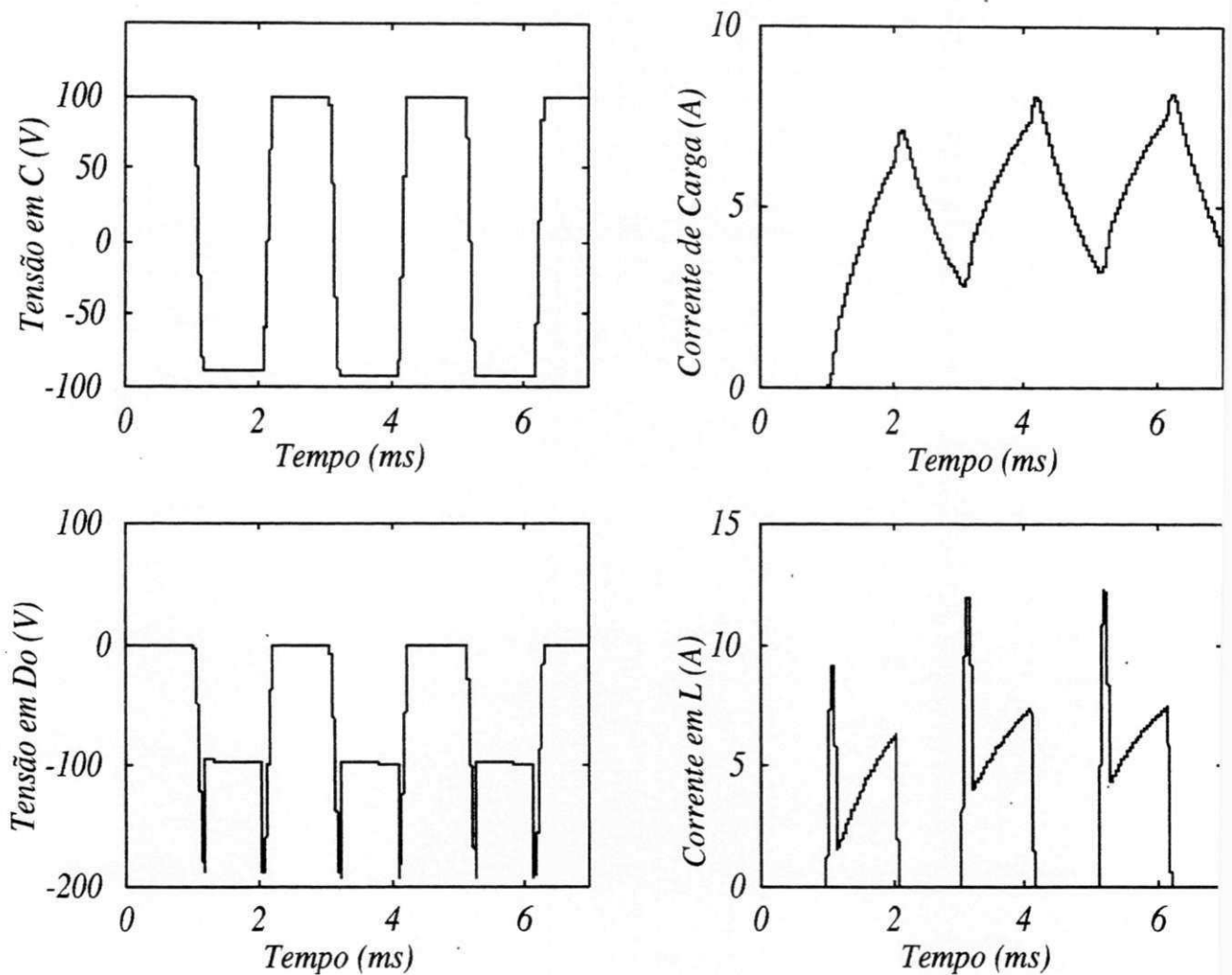


Figura 5.10: Resultado para o conversor QRC-ZCS.

5.3.2 Conversores CC/CC PWM Soft-Single-Switched (SSS)

Uma evolução aos conversores QRC's foram os conversores Quase-Ressonantes-PWM. Estes apresentam as vantagens dos conversores QRC's e operam em uma frequência de chaveamento fixa, eliminando os problemas de controle, mantendo as características não dissipativas. Entretanto tais conversores, a exemplo dos QRC's, possuem limitações de potência bem marcantes.

Os conversores desta subseção apresentam uma única chave ativa e possuem as seguintes vantagens:

- Podem operar de um modo não dissipativo em uma larga faixa de carga

- Possuem uma alta eficiência com cargas leves também
- Podem operar com uma baixa taxa de conversão de tensão
- Podem operar com características PWM em uma vasta faixa de frequência

Conversor Buck PWM-SSS

A Figura 5.11 mostra o conversor Buck PWM-SSS[36]. A descrição completa do funcionamento do conversor é descrita em 8 estágios, comentados a seguir:

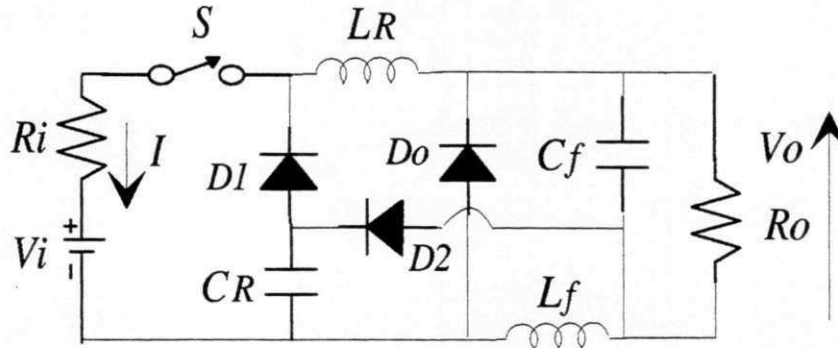


Figura 5.11: Conversor Buck PWM SSS

Estágio 1 $[t_0, t_1]$: O primeiro estágio começa quando a chave S é ligada. Neste estágio, a corrente através do indutor L_R aumenta linearmente até tornar-se igual a corrente de carga no tempo t_1 .

Estágio 2 $[t_1, t_2]$: O estágio ressonante tem início em t_1 . Neste intervalo, a corrente I_{L_R} cresce enquanto que a tensão V_{C_R} sai de $(-V_o)$ para $(V_i - V_o)$, depois decresce até a tensão do capacitor C_R se tornar V_i . Quando o diodo D_1 conduz no tempo t_2 , a tensão no capacitor C_R será grampeada, finalizando o período ressonante.

Estágio 3 $[t_2, t_3]$: Neste intervalo de tempo, a corrente através do indutor L_R decai linearmente até ser igual a corrente de carga. Durante este período, a tensão sobre o indutor L_R é V_o , e a tensão sobre o capacitor C_R permanece igual a V_i .

Estágio 4 $[t_3, t_4]$: Durante este estágio, somente a chave S e o indutor L_R estão conduzindo, e a transferência da energia da fonte para a carga ocorre em um tempo definido pela razão cíclica.

Estágio 5 $[t_4, t_5]$: Este estágio começa quando a chave S é aberta. Neste intervalo, o capacitor C_R descarrega linearmente até atingir zero de tensão, desta forma, o diodo D_2 passa a conduzir.

Estágio 6 $[t_5, t_6]$: Neste intervalo de tempo, existe uma ressonância entre o capacitor C_R e o indutor L_R . Durante este estágio, o capacitor C_R se carrega com uma tensão negativa enquanto que a corrente sobre o indutor decai. Isto acontece até a tensão

V_{CR} atingir $(-V_o)$ no tempo t_6 . Neste momento existe a possibilidade da corrente no indutor já ter atingido zero, esta situação pode acontecer se a corrente de carga for menor que $V_o(C_R/L_R)^2$. Esta não é uma condição necessária para o funcionamento do conversor, uma vez que ele opera com corrente de carga independente da corrente ressonante.

Estágio 7 [t_6, t_7]: Quando o diodo D_1 passar a conduzir, a corrente no indutor decrescerá linearmente até atingir zero pela ação da tensão de carga. Quando a corrente no capacitor C_R atingir zero, a chave S poderá ser fechada num modo ZCS.

Estágio 8 [t_7, t_8]: Neste estágio, somente o diodo D_o conduz a corrente de carga. A duração deste estágio é definida pelo índice de modulação da chave.

De acordo com os princípios de funcionamento descritos acima, pode-se construir as formas de onda teóricas mostradas na Figura 5.12.

Os valores dos elementos do conversor, para efeito de simulação, serão os seguintes:

$$V_i = 48 \text{ V}$$

$$R_i = 0.00001 \ \Omega$$

$$L_R = 27.8 \ \mu\text{H}$$

$$C_R = 10 \text{ nF}$$

$$R_o = 8 \ \Omega$$

$$L_f = 200 \ \mu\text{H}$$

$$C_f = 100 \ \mu\text{F}$$

$$f_s = 100 \text{ KHz}$$

A estrutura do circuito para a entrada de dados para a simulação com o programa SICON2 é mostrada na Figura 5.13.

A matriz F para o conversor Buck PWM-SSS, obtida através da Figura 5.13, é a seguinte:

$$F = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Algumas das curvas obtidas em simulação são mostradas na Figura 5.14.

Para comprovar os resultados simulados e teóricos, tem-se alguns resultados experimentais, mostrados na Figura 5.15, obtidos com as seguintes especificações:

$$V_i = 48\text{V}$$

$$S = \text{IRF640}$$

$$D_{01,2} = \text{U1620}$$

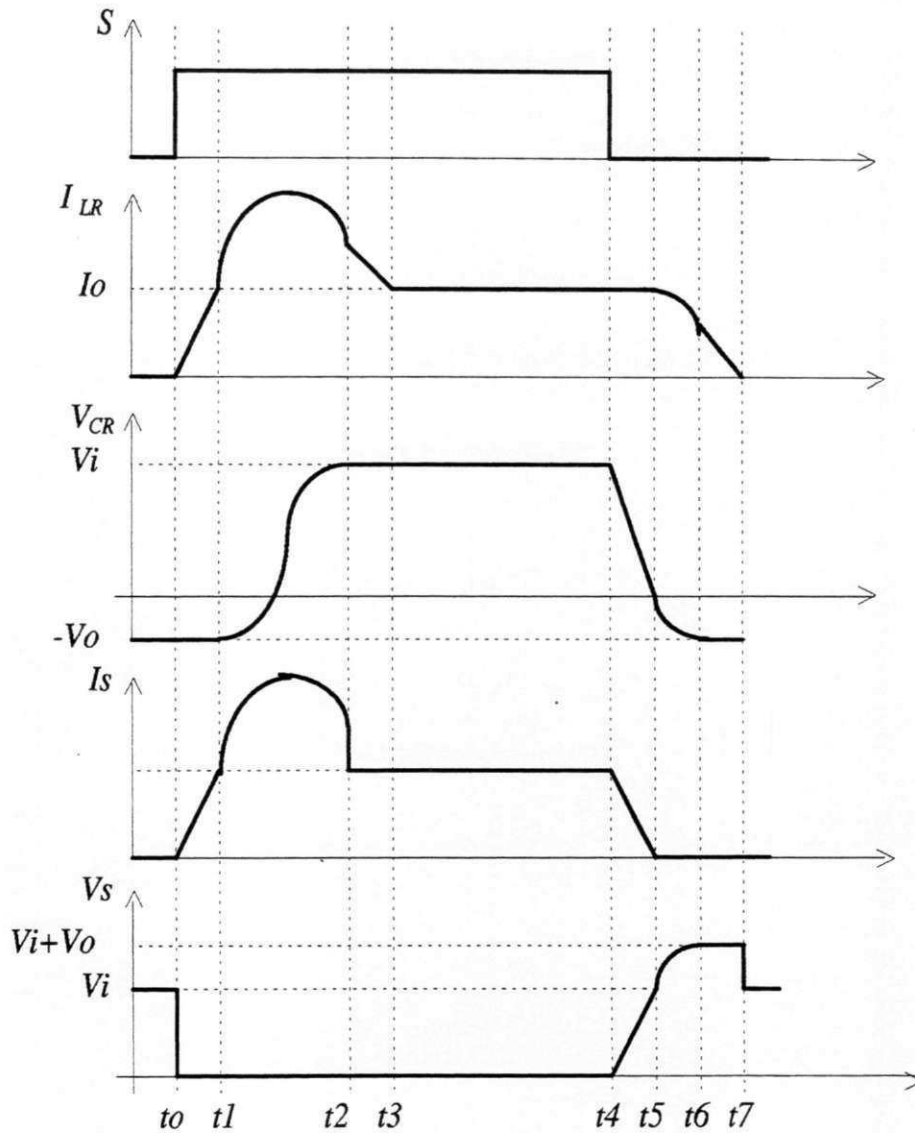


Figura 5.12: Formas de onda teóricas para o Conversor Buck PWM SSS

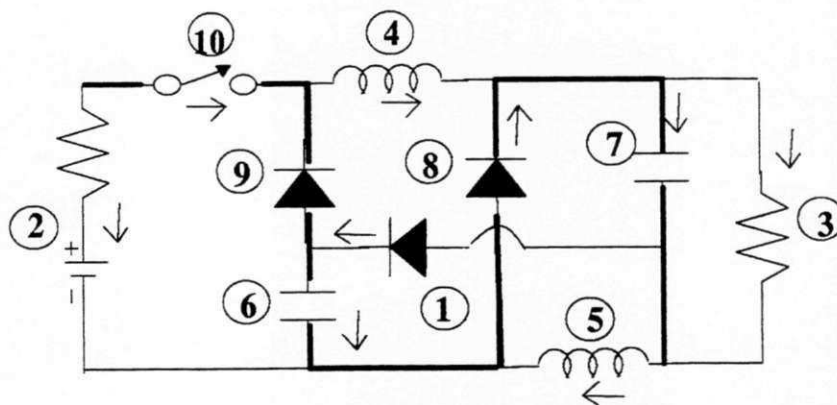


Figura 5.13: Disposição dos elementos no conversor Buck PWM SSS

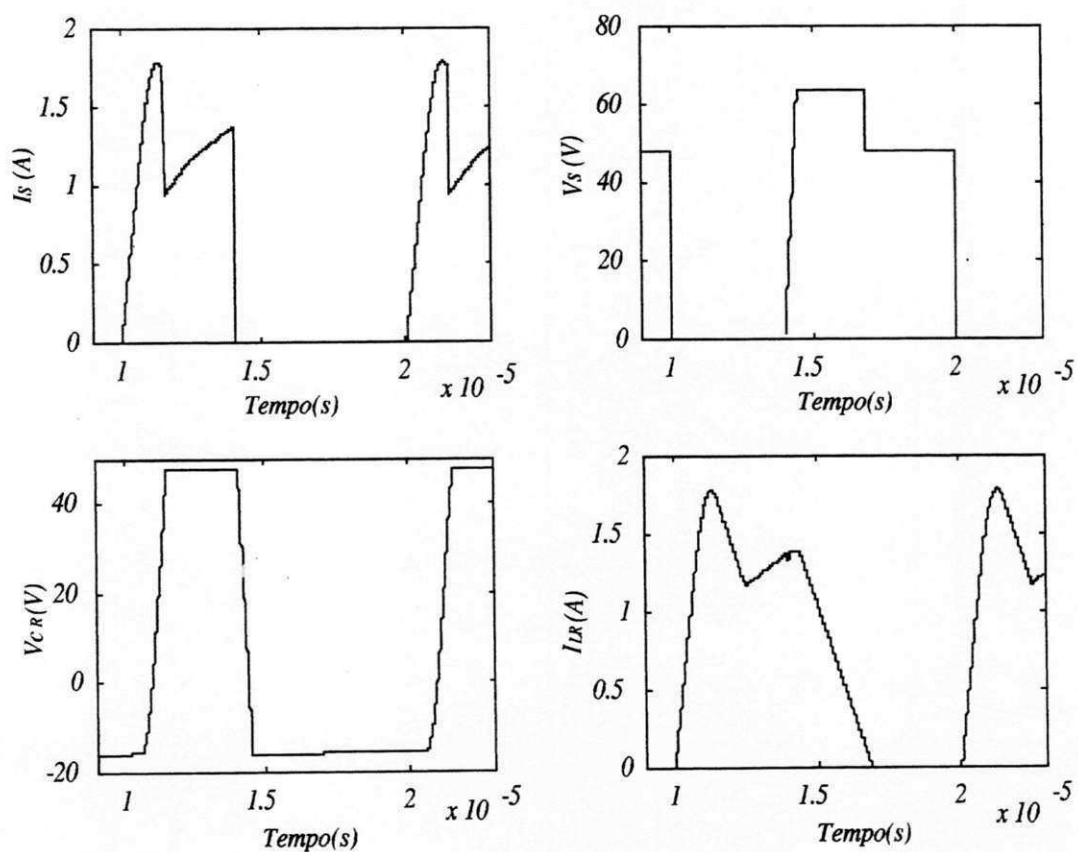


Figura 5.14: Resultados obtidos para o conversor Buck PWM SSS

$$V_o = 16V$$

$$C_R = 102nF$$

$$C_f = 100\mu F$$

$$F_s = 100 \text{ KHz}$$

$$L_F = 200 \mu H$$

$$L_R = 27,8\mu H$$

$$I_o = 1,1A$$

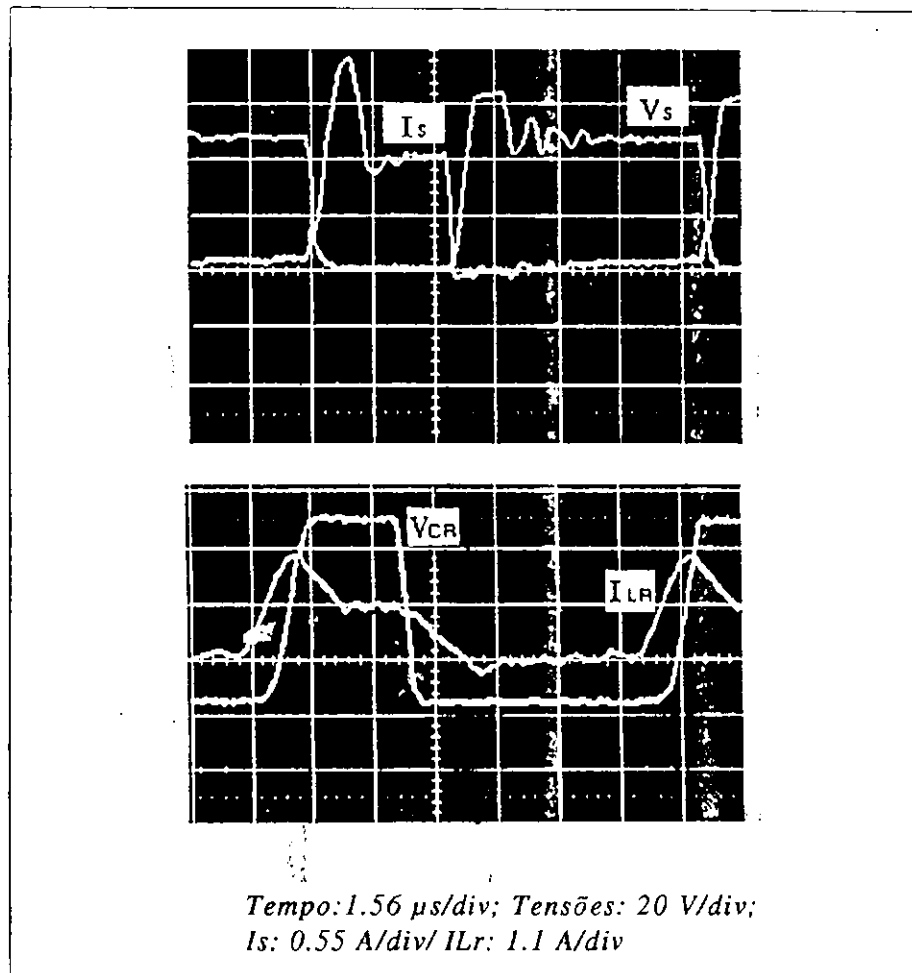


Figura 5.15: Formas de onda experimentais para o Conversor Buck PWM-SSS; Tempo: $1,56\mu S/div$, Tensões : $20 V/div$, I_s : $0,55 A/div$, I_{LR} : $1,1 A/div$

Os resultados experimentais obtidos, validaram o conversor Buck PWM-SSS. Verifica-se, também, a semelhança entre as curvas obtidas com a simulação, as curvas teóricas e as curvas experimentais. As curvas são traçadas de forma a permitir uma melhor comparação entre as diversas curvas (tensão em S (V_s), corrente em S (I_s), tensão em C_R (V_{CR}), e a corrente em L_R (I_{LR})).

Conversor Quadrático PWM-SSS

A Figura 5.16 mostra o Conversor Quadrático PWM-SSS [37] [36][38]. O seu ciclo de operação é dividido em 10 estágios, descritos a seguir:

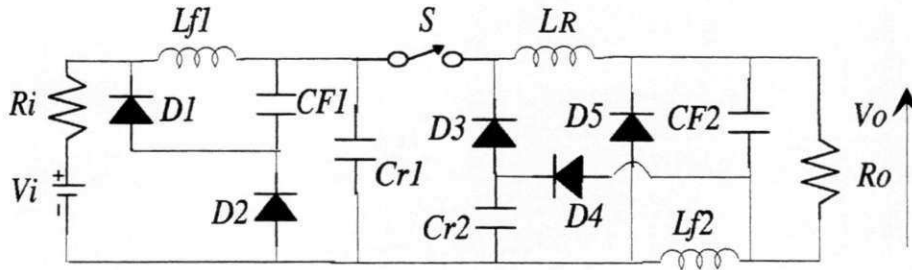


Figura 5.16: Conversor Quadrático PWM-SSS

Estágio 1 $[t_0, t_1]$: O primeiro estágio começa quando a chave S é fechada de um modo ZCS. Neste estágio, a corrente através do indutor ressonante L_R cresce linearmente até se tornar igual à corrente no indutor de filtro 1 (I_{LF1}) no tempo t_1 .

Estágio 2 $[t_1, t_2]$: Em t_1 o primeiro estágio ressonante começa. Neste intervalo, a corrente I_{LR} cresce enquanto V_{CR1} sai de $(V_G + V_{CF1})$ para V_{CF1} . Quando o diodo D_2 conduz no tempo t_2 , a tensão no capacitor C_{R1} é grampeada, terminando este período ressonante.

Estágio 3 $[t_2, t_3]$: Neste intervalo, a corrente no indutor L_R cresce linearmente até se tornar igual à corrente no indutor L_{F2} .

Estágio 4 $[t_3, t_4]$: Em t_3 , o segundo estágio ressonante começa. Neste intervalo, a corrente I_{LR} cresce enquanto a tensão V_{CR2} vai de $(-V_{CF2})$ para $(V_{CF1} - V_{CF2})$, depois decresce até a tensão do capacitor C_{R2} se tornar V_{CF1} . Quando o diodo D_3 conduzir em t_4 , a tensão do capacitor C_{R2} será grampeada, terminando este intervalo ressonante.

Estágio 5 $[t_4, t_5]$: Neste intervalo, a corrente no indutor L_R decresce linearmente até se tornar igual à corrente do indutor L_{F2} . Durante este período, a tensão no indutor L_R é V_{CF2} e no capacitor C_{R2} permanece V_{CF1} .

Estágio 6 $[t_5, t_6]$: Durante este estágio a chave S e os indutores L_R , L_{F1} e L_{F2} estão conduzindo, e a transferência de energia da fonte para a carga ocorre em um tempo definido pela razão cíclica.

Estágio 7 $[t_6, t_7]$: Este estágio começa quando a chave S é aberta. Isto acontece de um modo não dissipativo pois V_{CR1} é igual à V_{CR2} . Neste intervalo, o capacitor C_{R2} descarrega linearmente até atingir zero de tensão, assim, o diodo D_5 passa a conduzir. O capacitor ressonante C_{R1} carrega-se linearmente até atingir $(V_G + V_{CF1})$, com isto o diodo D_1 passa a conduzir.

Estágio 8 [t7,t8]: Neste intervalo, existe uma ressonância entre o capacitor C_{R2} e o indutor L_R . Durante este estágio, o capacitor C_{R2} se carrega com uma tensão negativa enquanto a corrente através do indutor decai. Isto acontece até a tensão $V_{C_{R2}}$ atingir $(-V_{CF2})$ no tempo t8. Neste momento existe a possibilidade da corrente no indutor já ter atingido zero, esta situação pode ocorrer se a corrente de carga for menor que $V_{CF2} (C_{R2}/L_R)^2$. Esta não é uma condição necessária para o funcionamento do conversor, uma vez que ele opera com corrente de carga independente da corrente ressonante. Neste período a tensão no capacitor C_{R1} permanece $(V_G + V_{CF1})$, até o próximo ciclo de operação começar.

Estágio 9 [t8, t9]: Quando o diodo $D4$ passar a conduzir, a corrente no indutor L_R decresce linearmente até atingir zero pela ação da tensão de carga. Quando a corrente no indutor L_R atingir zero, a chave S pode ser fechada de um modo ZCS.

Estágio 10 [t9,t10]: Neste estágio, apenas o diodo $D5$ conduz a corrente do indutor L_{F2} . A duração deste intervalo é definida pela modulação da chave. No final deste intervalo, a chave S é fechada e o próximo ciclo de operação começa.

De acordo com os princípios de funcionamento descritos acima, pode-se construir as formas de onda teóricas mostradas na Figura 5.17.

Os valores dos elementos do conversor, para efeito de simulação, são os seguintes:

$$V_i = 100 \text{ V}$$

$$R_i = 0.00001 \ \Omega$$

$$L_R = 2 \ \mu\text{H}$$

$$L_{f1} = 200 \ \mu\text{H}$$

$$L_{f2} = 200 \ \mu\text{H}$$

$$C_{R1} = 7.5 \ \text{nF}$$

$$C_{R2} = 7.5 \ \text{nF}$$

$$C_{F1} = 220 \ \mu\text{F}$$

$$C_{F2} = 220 \ \mu\text{F}$$

$$R_o = 2 \ \Omega$$

$$f_s = 200 \ \text{Khz}$$

A estrutura do circuito para a entrada de dados para a simulação com o programa SICON2 é mostrada na Figura 5.18.

A matriz F para o conversor Quadrático PWM-SSS, obtida através da Figura 5.18, é a seguinte:

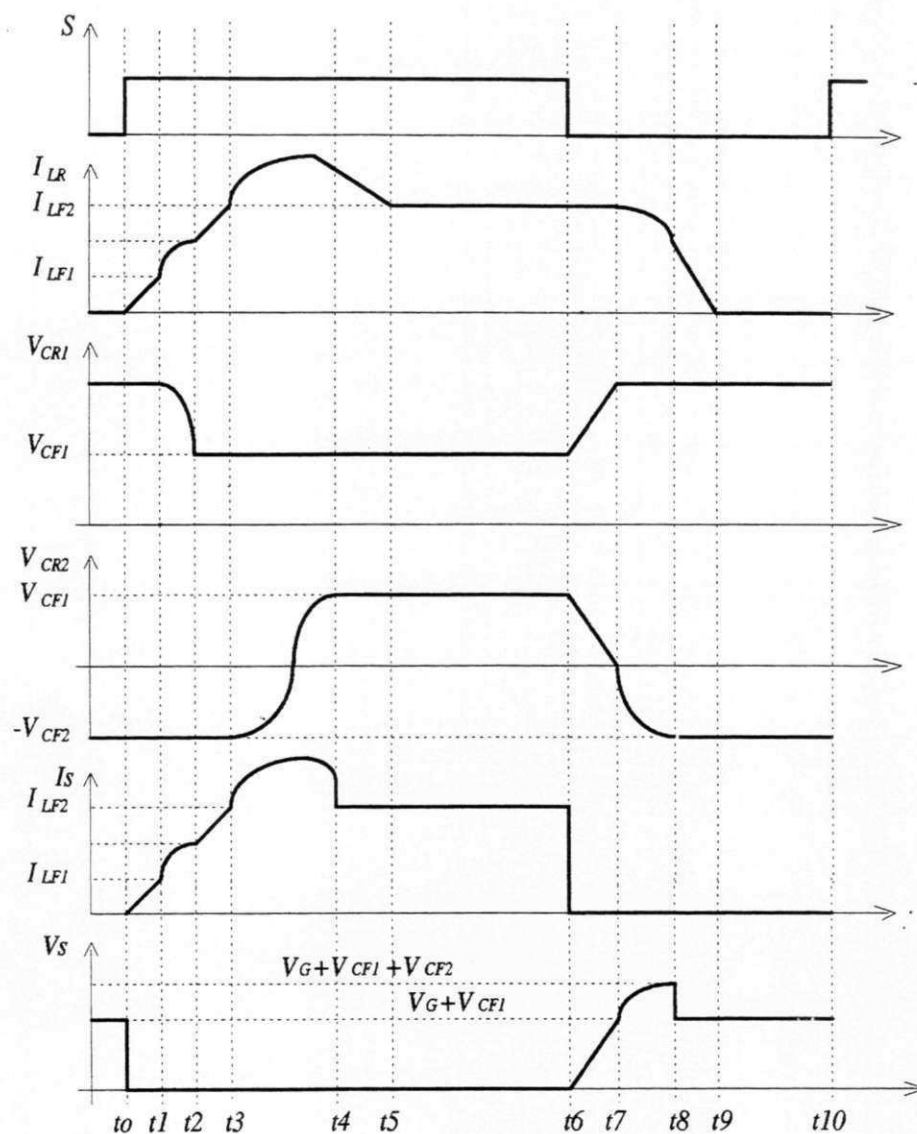


Figura 5.17: Formas de onda teóricas para o Conversor Quadrático PWM SSS

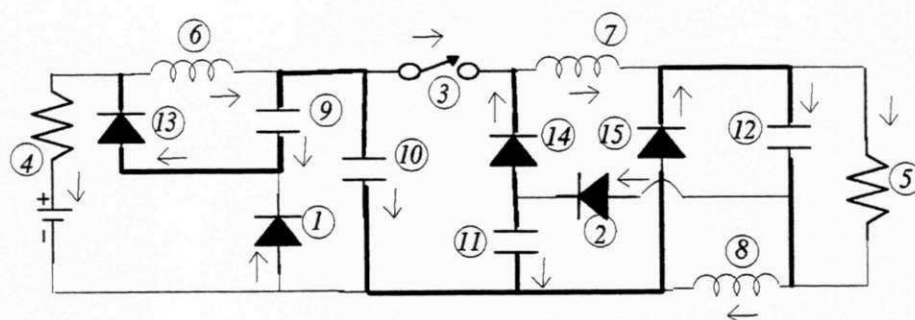


Figura 5.18: Disposição dos elementos no conversor Quadrático PWM- SSS

$$F = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Algumas das curvas obtidas em simulação são mostradas na Figura 5.19.

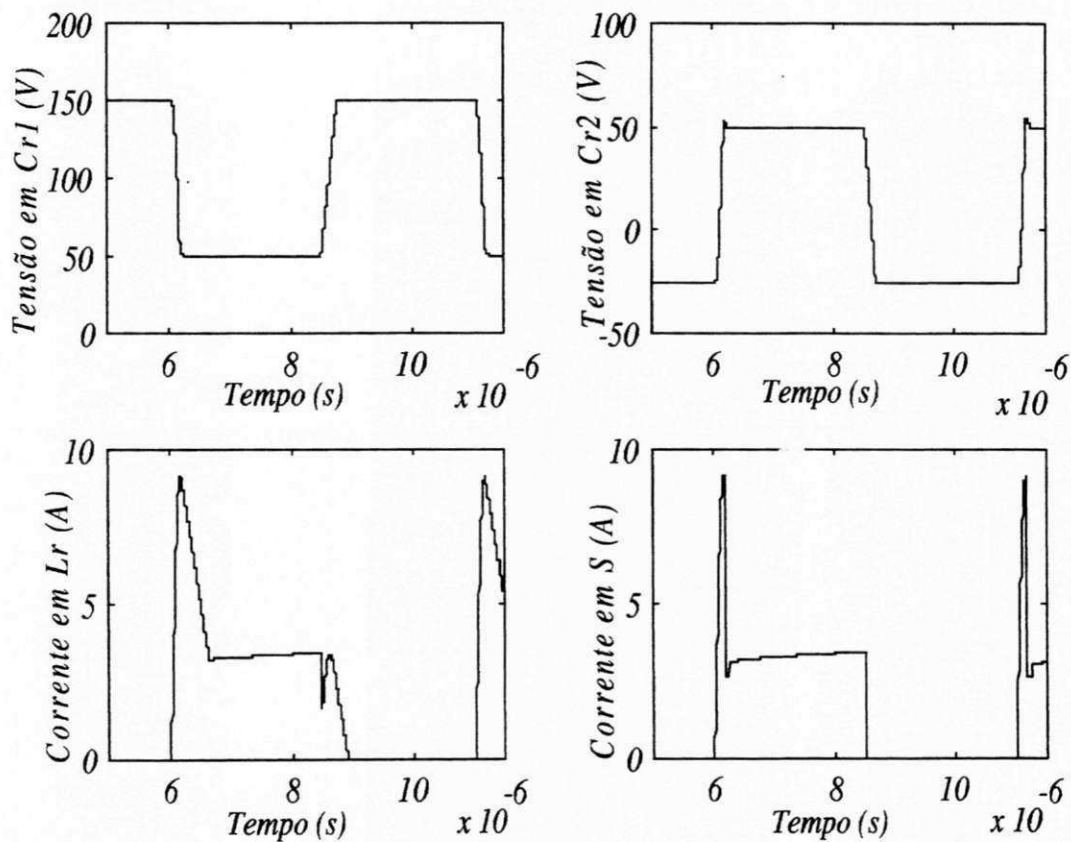


Figura 5.19: Resultados obtidos para o conversor Quadrático PWM SSS

Para comprovar os resultados simulados e teóricos, tem-se na Figura 5.20 alguns resultados experimentais, obtidos com as seguintes especificações:

- $V_G = 100V$
- $V_{CF2} = 25V$
- $L_R = 2\mu H$
- $F_s = 200kHz$

$$L_{f1} = 200 \mu H$$

$$L_{f2} = 200 \mu H$$

$$CR1 = 7.5 \text{ nF}$$

$$CR2 = 7.5 \text{ nF}$$

$$CF1 = 220 \mu F$$

$$CF2 = 220 \mu F$$

$$I_o = 4A$$

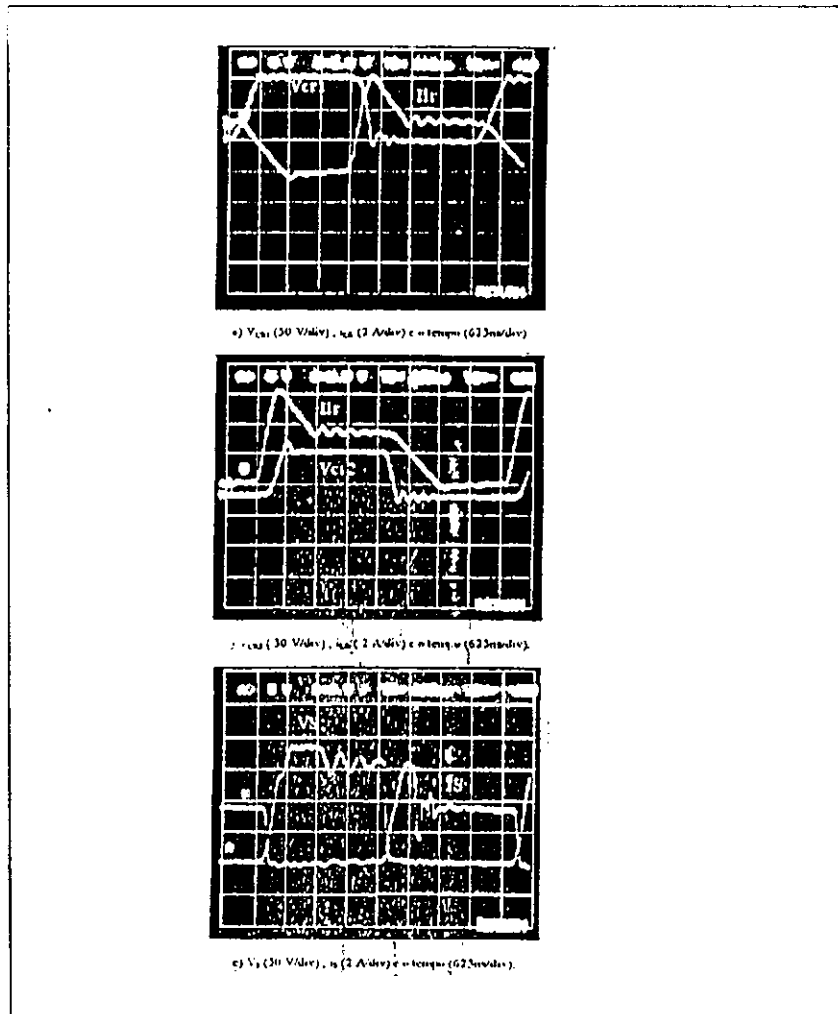


Figura 5.20: Formas de onda experimentais para o Conversor Quadrático PWM SSS

Observa-se, também neste caso, a semelhança entre as curvas obtidas com a simulação, as curvas teóricas e as curvas experimentais. As curvas são traçadas de forma a permitir uma melhor comparação entre as diversas curvas (tensão em Cr1, tensão em Cr2, corrente em S e a corrente em D1).

5.3.3 Inversor Monofásico em Ponte

O circuito básico para um inversor monofásico em ponte é mostrado na Figura 5.21

O bloqueio de $S1$ é feito pelo complementar $S4$. $S1$ e $S2$ são disparados por um pulso contínuo por 180° da tensão de saída do inversor. $S3$ e $S4$ são disparados 180° da tensão de saída após os disparos de $S1$ e $S2$. O funcionamento completo do circuito é descrito em [39]. Algumas curvas teóricas são mostradas na Figura 5.22. A simulação foi realizada para uma carga indutiva. Os valores para os componentes do inversor são os seguintes:

$$V_i = 100 \text{ V}$$

$$R_i = 0.00001 \ \Omega$$

$$C = 1 \text{ pF}$$

$$L = 100 \text{ mH}$$

$$R = 10 \ \Omega$$

$$f_s = 60 \text{ Hz}$$

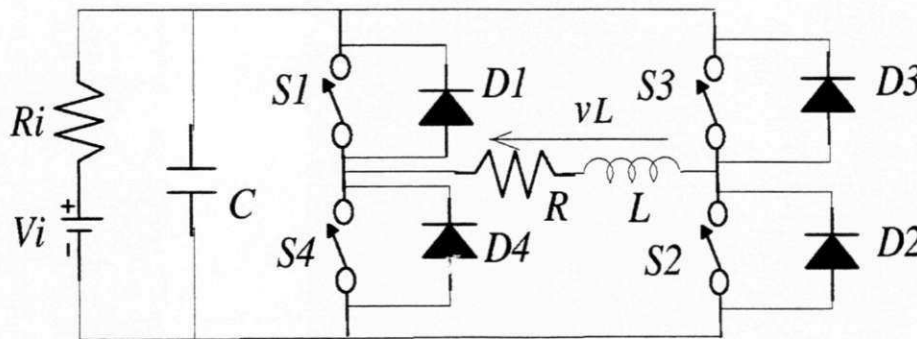


Figura 5.21: Inversor Monofásico em Ponte

O inversor descrito na Figura 5.21 é desenhado de forma a enfatizar os componentes para entrada de dados para a simulação com o programa SICON2. Este desenho é mostrado na Figura 5.23

A matriz F para o inversor monofásico em ponte, obtida através da Figura 5.23, é a seguinte:

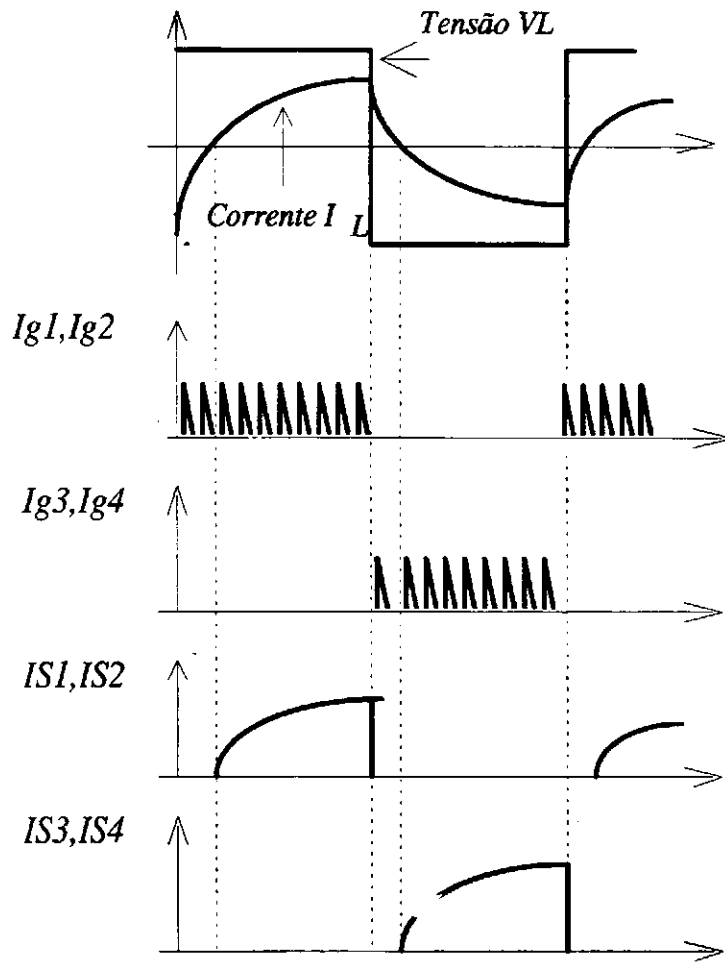


Figura 5.22: Algumas curvas teóricas para o inversor monofásico em ponte

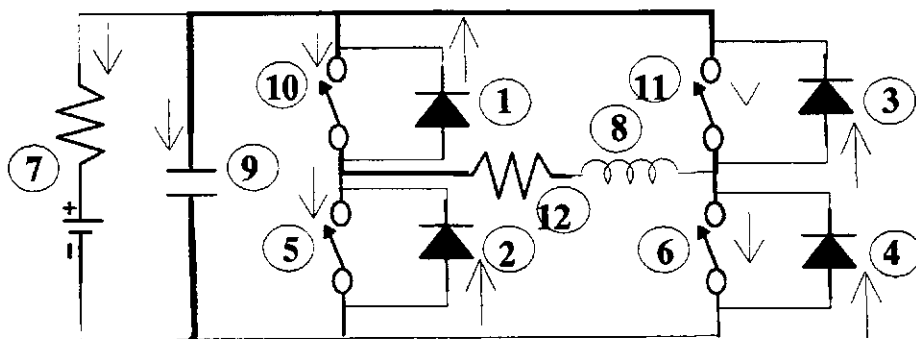


Figura 5.23: Disposição dos elementos do inversor monofásico em ponte ,

$$F = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 1 \end{bmatrix}$$

Resultados obtidos através da simulação são mostrados na Figura 5.24.

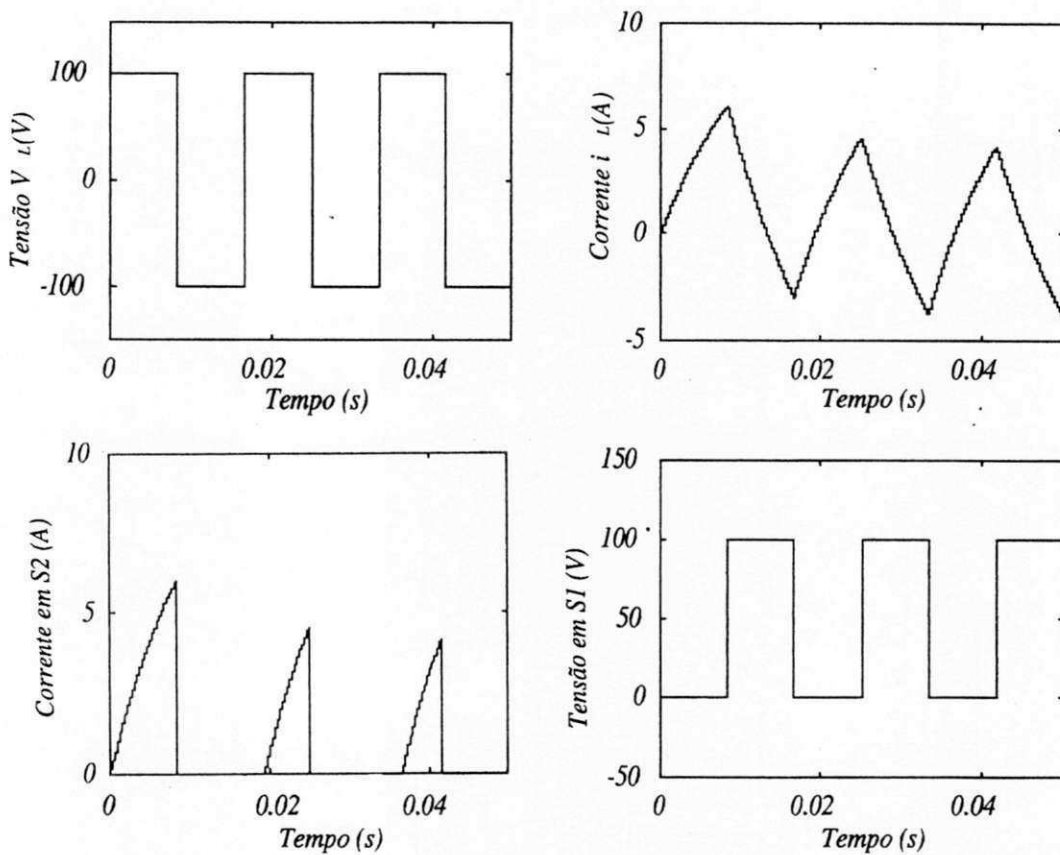


Figura 5.24: Resultados obtidos para o inversor monofásico em ponte

As curvas obtidas na Figura 5.24 refletem o comportamento teórico do circuito, ou seja, as diversas etapas de funcionamento do mesmo, descritas na Figura 5.22.

5.3.4 Retificador Monofásico em Ponte

O circuito retificador em ponte é por demais conhecido face a sua grande utilização em fontes de alimentação CC de baixa potência. As três possíveis configurações para este

tipo de conversor são: caso não-controlado, totalmente controlado e semi-controlado. O retificador empregado para a simulação é exatamente o caso controlado, ou seja, o apresentado na Figura 5.25, onde as chaves utilizadas são SCR's. A condução não acontece até que os tiristores sejam disparados e, para haver circulação de corrente, $S1$ e $S2$ têm de ser disparados juntos, assim como $S3$ e $S4$ no próximo semiciclo. A análise completa do funcionamento do circuito é mostrada em [39]. Os valores para os elementos do circuito da Figura 5.25 são os seguintes:

$$v_i = 190 \text{ V}$$

$$R_i = 0.00001 \Omega$$

$$R = 10 \Omega$$

$$L = 0.1 \text{ H}$$

$$C = 1 \text{ pF}$$

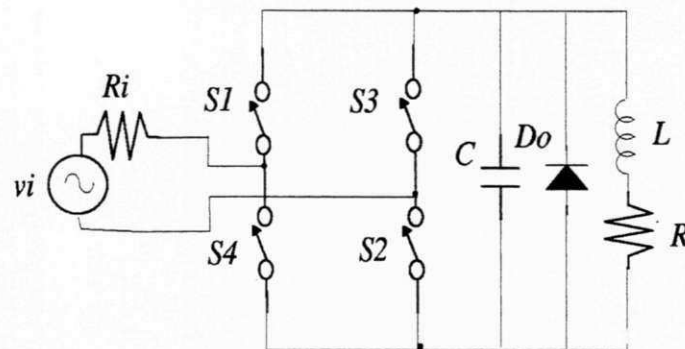


Figura 5.25: Retificador Monofásico em Ponte

O retificador monofásico em ponte desenhado de forma a enfatizar os componentes para entrada de dados para a simulação com o programa SICON2. Este desenho é mostrado na Figura 5.26.

A matriz F para o retificador monofásico em ponte é a seguinte:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}$$

Na Figura 5.27 são apresentados alguns resultados obtidos na simulação do retificador monofásico em ponte.

As curvas obtidas na Figura 5.27 refletem o comportamento teórico do circuito, ou seja, as diversas etapas de funcionamento do mesmo.

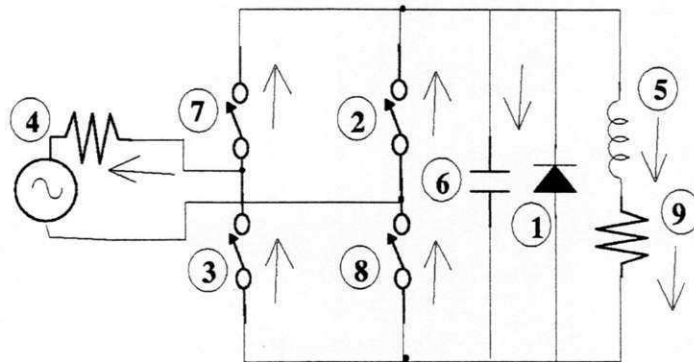


Figura 5.26: Disposição dos elementos do retificador monofásico em ponte

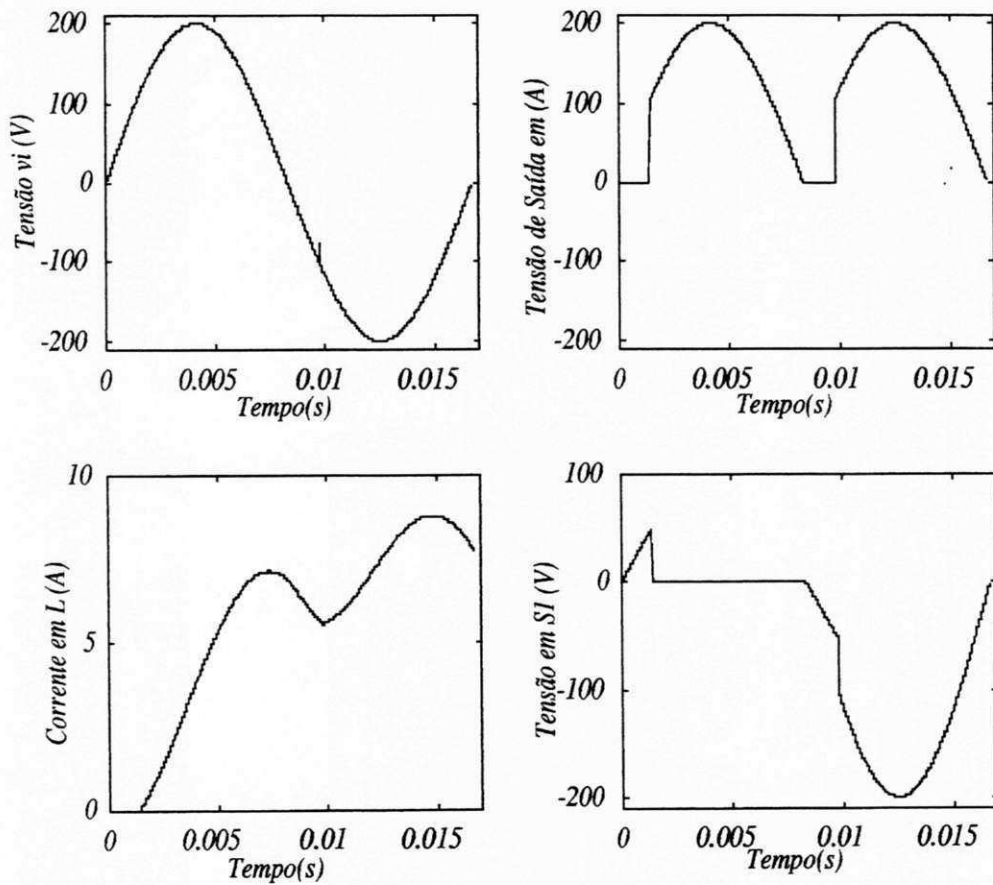


Figura 5.27: Resultados obtidos para o retificador monofásico em ponte

5.4 Conclusão

Neste capítulo foi apresentado o programa SICON2, um programa para simulação de conversores estáticos. As características básicas do programa SICON2 são: topologia fixa, representação dos semicondutores por resistores com resistências grandes (abertos) ou pequenas (fechados), representação do circuito através de equações de estado discretas, passo variável, interpolação linear na mudança de estado dos semicondutores, necessidade de pouco conhecimento sobre o funcionamento do conversor. O programa constitui-se num "toolbox" para simulação de conversores no ambiente MATLAB, o que lhe dá características tais como: recursos gráficos, codificação simples e uma grande variedade de ferramentas matemáticas.

Exemplos de aplicação na simulação de conversores foram apresentados. Várias curvas teóricas foram traçadas, assim como alguns resultados experimentais foram mostrados. Verifica-se que os resultados obtidos com o programa SICON2, refletem de forma precisa o comportamento desejado do circuito.

Capítulo 6

Análise Comparativa

6.1 Introdução

Neste capítulo são feitas comparações de resultados obtidos com os programas SICON1, SICON2, SACSO e o PSPICE.

O objetivo é estabelecer comparações através das curvas obtidas com tais simuladores, assim como tecer alguns comentários sobre o funcionamento dos mesmos. Resultados obtidos com outros simuladores, como é o caso do PSPICE, e o SACSO, validam ainda mais os programas SICON1 e SICON2.

Dois circuitos são simulados usando os programas SICON1, SICON2, SACSO e PSPICE, como forma de comparação entre estes programas.

6.2 Conversor CC-CC

O circuito ressonante da Figura 6.1 representa o primeiro circuito a ser simulado. O seu ciclo de operação é dividido em 3 estágios.

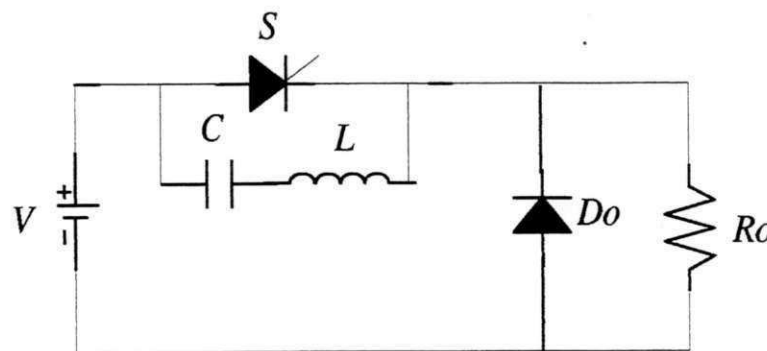


Figura 6.1: Conversor CC-CC ressonante

Estágio 1 ($t_0 < t < t_1$): Neste intervalo o SCR está bloqueado. Tem-se um ciclo ressonante entre L, C, e R_o . Este intervalo estará concluído quando $i_L = 0$ e $v_C = V$.

Estágio 2 ($t_1 < t < t_2$): Neste intervalo o SCR será ativado. O capacitor começará a descarregar através de uma corrente senoidal dada por:

$$i_c = v \sqrt{\frac{C}{L}} \text{sen} \frac{1}{\sqrt{LC}} t$$

Esta expressão continua válida até que o SCR deixe de conduzir em t_2 .

Estágio 3 ($t_2 < t < t_3$): com o SCR bloqueado a corrente de carga passa a circular por C, L, e a carga.

Na Figura 6.2 temos algumas curvas teóricas do circuito da Figura 6.1.

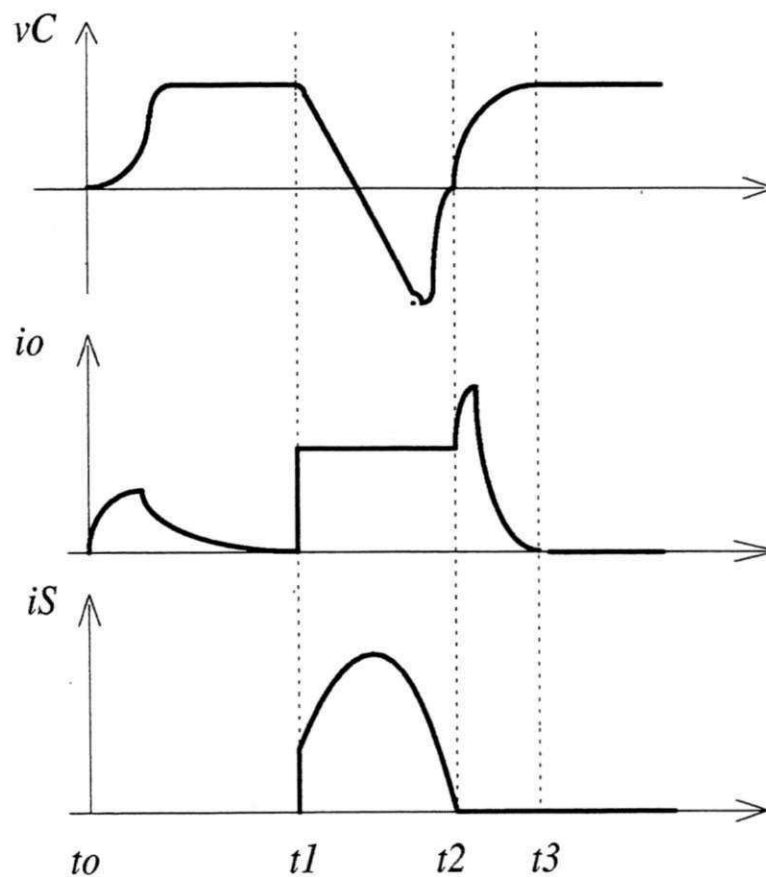


Figura 6.2: Formas de onda teóricas para o Conversor CC-CC ressonante

Os parâmetros do circuito a serem usados para as simulações são os seguintes:

$$V = 120V$$

$$R_o = 20\Omega$$

$$C = 20\mu F$$

$$L = 2mH$$

$$f_s = 200 \text{ Hz}$$

6.2.1 Resultados Obtidos Com o Programa SICON2

Na Figura 6.3 temos resultados de simulação obtidos através do programa SICON2.

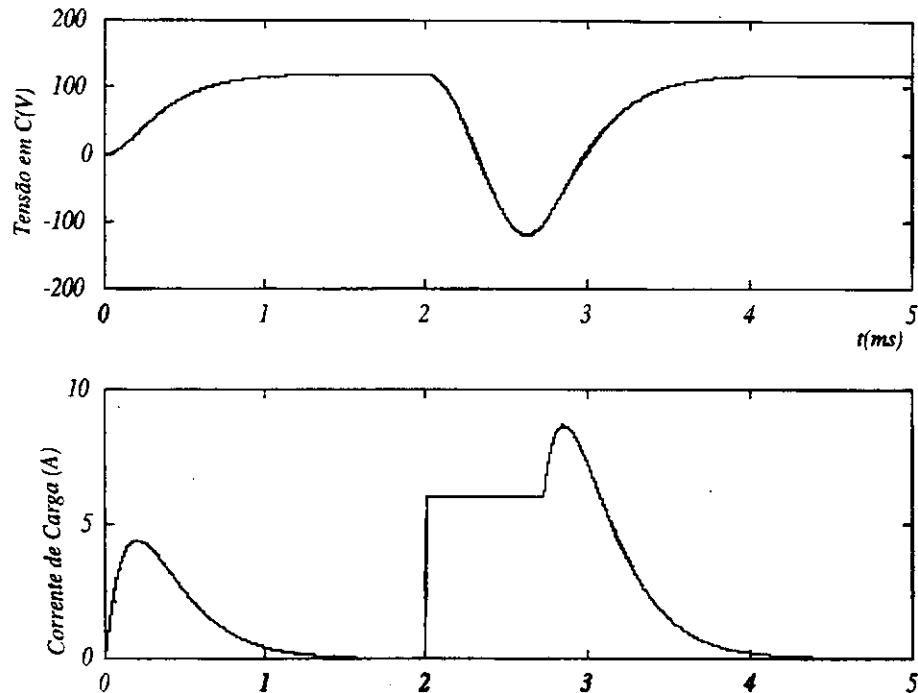


Figura 6.3: Resultado para o conversor CC-CC obtidos com o programa SICON2.

6.2.2 Resultados Obtidos Com o Programa SACSO

Na Figura 6.4 temos alguns resultados de simulação obtidos com o programa SACSO.

6.2.3 Resultados Obtidos Com o Programa PSPICE

Na Figura 6.5 temos alguns resultados obtidos com o programa PSPICE.

Verifica-se que as curvas obtidas com o SICON2, o SACSO e o PSPICE são bastante semelhantes. E todos os resultados refletem as etapas de funcionamento do circuito, descritas na Figura 6.1.

6.3 Retificador Monofásico

O segundo circuito a ser simulado corresponde a um retificador monofásico com um carga RL, representado na Figura 6.6.

Em $t = 0$, a tensão da fonte torna-se positiva e o diodo começa a conduzir. Sabe-se, no entanto, que uma indutância não permite variações bruscas de corrente e assim, se

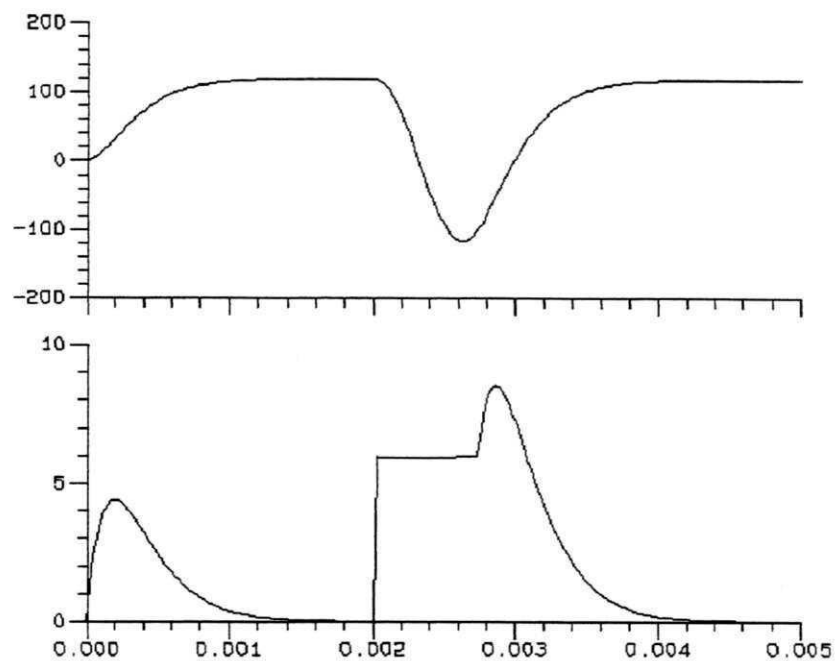


Figura 6.4: Resultado para o conversor CC-CC obtidos com o programa SACSO;(a) tensão no capacitor,(b) corrente na carga

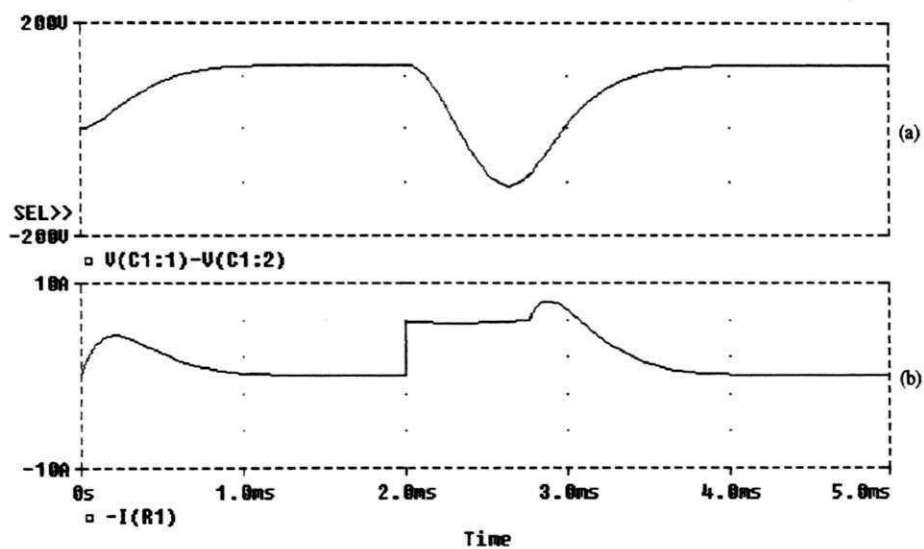


Figura 6.5: Resultado para o conversor CC-CC obtidos com o programa PSPICE;(a) corrente na carga,(b) tensão no capacitor

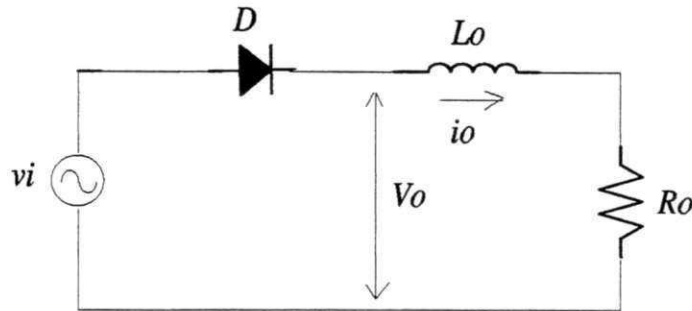


Figura 6.6: Retificador monofásico com carga RL

o diodo estava anteriormente cortado, a corrente na indutância logo após o início da condução é zero. A corrente que aparecerá no indutor estará atrasada em relação a tensão de um ângulo Φ dado por :

$$\Phi = \text{arc tag} \frac{\omega L}{R}$$

É interessante notar que a tensão V_o apresenta um trecho com valores negativos. Isto ocorre em virtude da corrente estar atrasada, e assim, quando a tensão se anula em $\omega t = \pi$, a corrente ainda possui um valor positivo. Neste ponto, como a corrente está diminuindo, V_L é negativa e portanto a indutância fornece a tensão necessária para manter o diodo em condução.

O valor β é determinado a partir da seguinte equação:

$$\text{sen}(\beta - \Phi) + \text{sen}\Phi e^{-\frac{R\beta}{\omega L}} = 0$$

Na Figura 6.7 temos algumas curvas teóricas para o retificador monofásico analisado.

6.3.1 Resultados Obtidos Com o Programa SICON1

Na Figura 6.8 temos resultados de simulação obtidos através do programa SICON1.

6.3.2 Resultados Obtidos Com o Programa SACSO

Na Figura 6.9 temos alguns resultados de simulação obtidos com o programa SACSO.

6.3.3 Resultados Obtidos Com o Programa PSPICE

Na Figura 6.10 temos alguns resultados obtidos com o programa PSPICE.

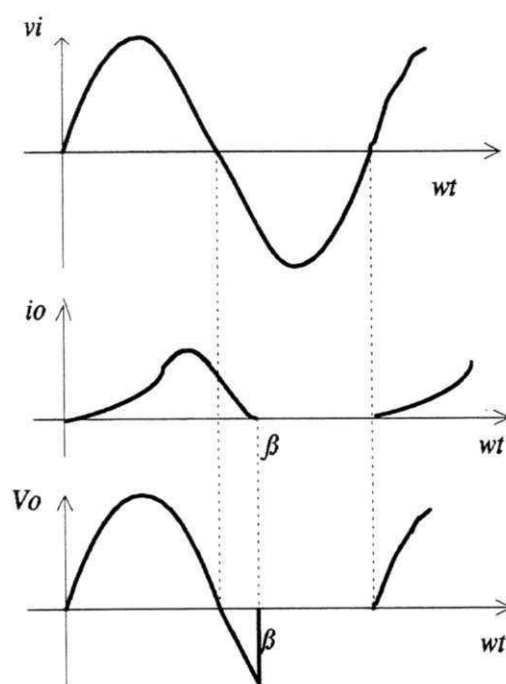


Figura 6.7: Formas de onda teóricas para o Retificador Monofásico com Carga RL

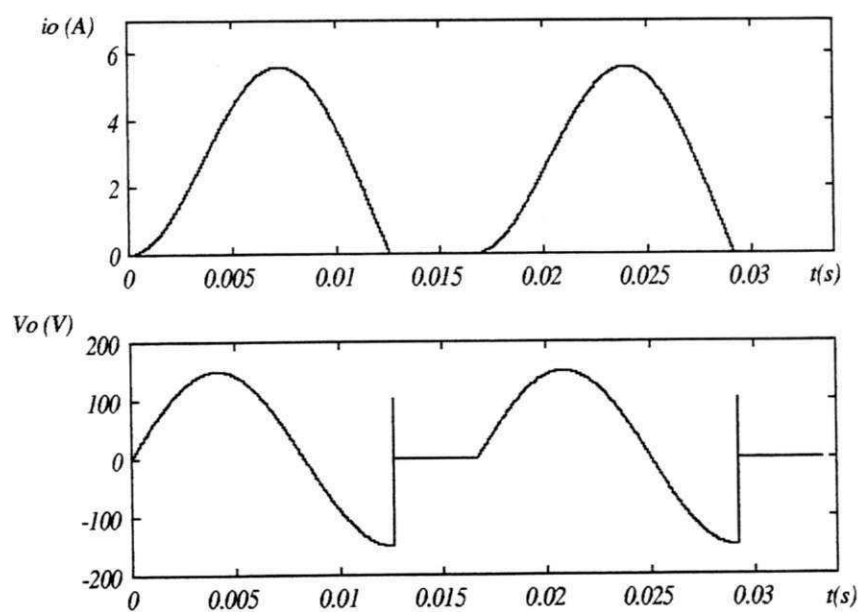


Figura 6.8: Resultado para o Retificador Monofásico obtidos com o programa SICON1

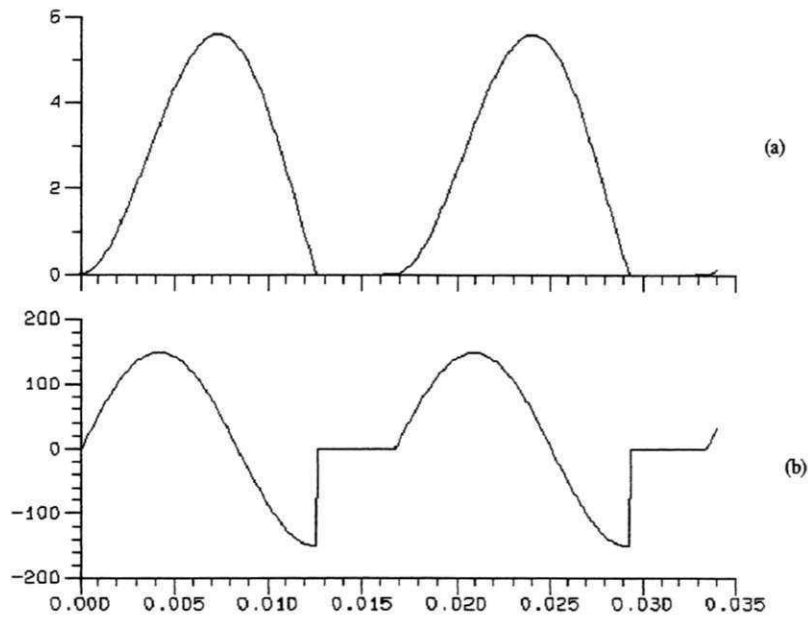


Figura 6.9: Resultado para o Retificador Monofásico obtidos com o programa SACSO;(a)corrente de carga,(b)tensão de carga

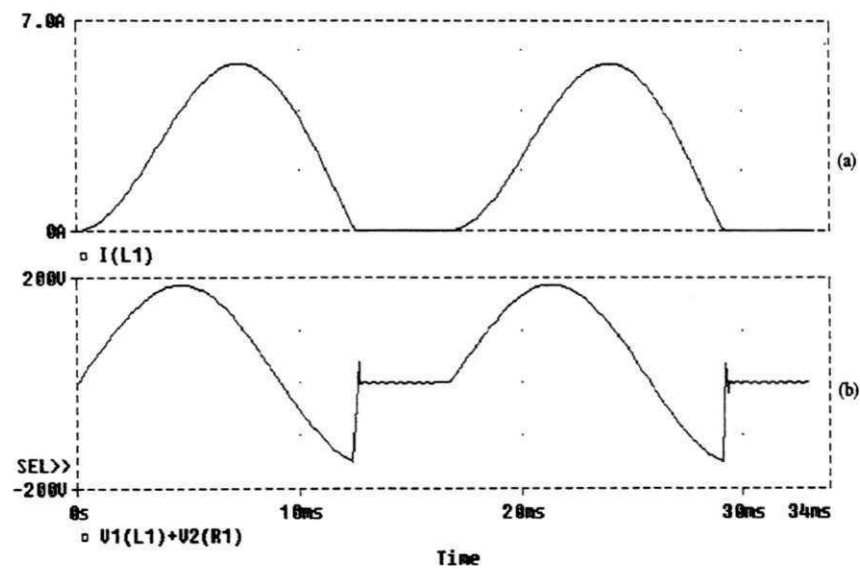


Figura 6.10: Resultado para o Retificador Monofásico obtidos com o programa PS-PICE;(a)corrente de carga,(b)tensão de carga

Verifica-se, também neste caso, que as curvas obtidas com o SICON1, o SACSO e o PSPICE são bastante semelhantes. E todos os resultados refletem as etapas de funcionamento do circuito, descritas na Figura 6.7.

6.4 Conclusões

Neste capítulo foram apresentados resultados de simulação obtidos com os programas SICON1, SICON2, SACSO e PSPICE. Os resultados obtidos com tais programas são bastantes semelhantes. As diferenças pequenas entre as formas de onda dos circuitos simulados são provavelmente devido às diferentes técnicas de simulação e diferentes modelamentos de chaves e diodos. Os programas SICON1 e SICON2, escritos no MATLAB, apresentam tempo de execução maior que o SACSO e o PSPICE, devido ao fato de que o MATLAB não compila os programas escritos na sua linguagem de programação. No entanto, o fácil manuseio do programas SICON1 e SICON2, principalmente o SICON1, torna-os poderosos "toolboxes" em Eletrônica de Potência. Os programas desenvolvidos, se comparados ao SACSO, possuem, dentre outras vantagens, o desenvolvimento dentro do ambiente Windows. O programa SICON1, se comparado ao PSPICE, possui a vantagem da simples interface com o usuário. O programa SICON2, não possui uma interface tão amigável com o usuário, no entanto possui a vantagem, se comparado com o SACSO, de o usuário não ter a preocupação com o período de amostragem.

Capítulo 7

Conclusão Geral

Este trabalho teve por objetivo o desenvolvimento de programas de simulação de conversores estáticos de energia. Os programas são "toolboxes" desenvolvidos dentro do ambiente do MATLAB. Os programas desenvolvidos permitem o estudo dos vários tipos de conversores encontrados na área de Eletrônica de Potência.

Os programas desenvolvidos foram o SICON1 e o SICON2. Os dois programas foram descritos no capítulo 4 e capítulo 5, respectivamente. A diferença fundamental entre os dois "toolboxes" consiste no tipo de tratamento matemático usado para modelar os conversores. Enquanto o SICON1 utiliza equações algébricas levantadas a partir de uma análise de nós para redes lineares invariantes, O SICON2 considera equações de estado no modelamento do conversor.

No capítulo 3, foi apresentado o tratamento matemático usado no desenvolvimento dos dois "toolboxes". Observa-se a simplicidade do tratamento envolvendo a análise de nós de redes lineares invariantes, com equações simples de serem obtidas. Na abordagem envolvendo equações de estado observa-se um tratamento um pouco mais complexo, envolvendo um tratamento matricial que necessita de bastante cuidado.

No capítulo 4, foi feita toda a análise a respeito do programa SICON1. As equações usadas pelo programa são simples, no entanto, para que elas possam ser algébricas de faz necessário adotar uma representação especial para os capacitores e indutores, pertencentes ao conversor. Foram analisadas três formas distintas de modelamento destes componentes. Em função da simplicidade de implementação adotou-se a aproximação (FE) para o modelamento dos capacitores e dos indutores. Constata-se, no entanto, que para produzir um modelo mais próximo do real necessita-se trabalhar com um período o menor possível. Duas concepções de topologia são tratadas. A primeira consiste numa simulação do conversor através de topologia fixa, onde o dispositivo semiconductor é tratado como um resistor que assume resistência pequena ou grande,

dependendo dos estados fechado ou aberto, respectivamente. A segunda consiste em adotar uma topologia variável, onde o dispositivo semiconductor é retirado do circuito quando em estado de bloqueio, e possuindo uma resistência pequena quando em estado de condução, este tipo concepção para o dispositivo semiconductor se aproxima mais do caso ideal, além de evitar dispersões nas constantes de tempo do circuito. Os resultados obtidos tem ambas as topologias são bastante satisfatórios. No entanto, para um mesmo período de amostragem a simulação do conversor com uma topologia variável possui resultados um pouco mais precisos, porém o tempo de execução torna-se maior, pois a subrotina para a simulação por topologia variável torna-se um pouco mais complexa, já que a retirada do elemento do circuito provoca uma total mudança nas dimensões das matrizes.

No capítulo 5, foi feito o estudo do programa SICON2. O tratamento do programa SICON2 é um pouco mais complexo em relação ao SICON1. O desenvolvimento do programa em termos de equações de estado permite representar os indutores e capacitores em seu estado natural. As equações são descritas de tal forma a admitir que não existam percursos fechados contendo unicamente capacitores, nem cortes puramente indutivo. Isto restringe a classe dos circuitos, todavia uma abordagem mais geral resultaria em equações muito mais complexas. Nas equações se faz necessário a presença de pelo menos um capacitor e um indutor. A interface com o usuário não é tão amigável se compararmos com o programa SICON1, já que se faz necessário entender como montar a matriz F . Um dos aspectos mais importante do programa SICON2 é a possibilidade de se trabalhar com um período de amostragem variável, onde este período é escolhido em função dos autovalores da matriz dinâmica. Os resultados obtidos com o programa SICON2 são bastante satisfatórios.

No capítulo 6, procurou-se validar ainda mais os programas SICON1 e SICON2, através de comparações com os programas SACSO e PSPICE. Algumas características do SICON1 e do SICON2 os torna mais vantajosos, em termo de uso, do que o SACSO e o PSPICE.

O ambiente MATLAB possui várias vantagens citadas anteriormente, no entanto tem-se problema com relação ao tempo de execução das subrotinas, já que as mesmas são interpretadas linha a linha. A utilização de linguagens compiladas, como a linguagem "C", permite um menor tempo de execução do programa.

7.1 Sugestões Para Trabalhos Futuros

Apesar de estarem prontos para serem utilizados pelo usuários de Eletrônica de Potência, os programas SICON1 e SICON2, podem ainda receberem modificações e otimizações em suas estruturas. Uma melhora na interface dos programas, é um exemplo de otimização que pode ser feita. A seguir são apresentadas algumas sugestões para trabalhos futuros.

- Desenvolvimento de versões dos programas em linguagem "C";
- Melhorar a interface dos programas SICON1 e SICON2;
- Implementações de outros possíveis modelos para os dispositivos semicondutores dentro da estrutura dos programas;
 - Estudar possibilidades de tentar implementar um período de amostragem variável para o programa SICON1;
 - Desenvolver estruturas de controle para adaptar-se com os dois programas desenvolvidos.

Capítulo 8

LISTAGEM DOS PROGRAMAS SICON1 e SICON2

8.1 PROGRAMA SICON1

```
\% toolbox sicon1.m
\% UNIVERSIDADE FEDERAL DA PARAIBA
\% DEPARTAMENTO DE ENGENHARIA ELETRICA
\%
\% PROGRAMA PARA SIMULACAO DE CONVERSORES ESTATICOS
\%
home
clear
\%
t=0;teta=0;h=1;g=0;v=1;a=0;c=1;tr=0;ind=0;cap=0;
diodo=0;vari=0;fte=0;fco=0;Dio=1;SCR=1;t1=0;
circuito=menu('TIPO DE CIRCUITO','conversor CC-CC','conversor
CA-CC','conversor CC-CA','conversor CA-CA');
tipoext=menu('Tipo de Topologia','Fixa','Vari\{a} vel');
elementos=input('entre com a quantidade de elementos ==>$ ');
elementos1=elementos;
nos=input('entre com a quantidade de n\{o}s ==>$ ');
nos1=input('entre com o n\{o} de refer\{e}ncia == $>$ ');
Vs1=zeros(elementos,1); \% vetor fonte de tens\{a}o dos bra\c {c}os \%
Vs=zeros(elementos,1);
Js=zeros(elementos,1); \% vetor fonte de corrente dos bra\c {c}os \%
Js1=zeros(elementos,1);
V=zeros(elementos,1);
J=zeros(elementos,1);
E=zeros((nos-1),1);
for i=1:elementos
home
k=menu(sprintf('Elementonumero%
\{d',i),'resistor','capacitor','indutor','diodo','SCR','Transistor','fonte
de tens\{a}o','fonte de corrente');
if k==1
close
```

```

home
R(h)=input('entre com o valor da resist\^{e}ncia ==>$ ');
G(h)=(R(h))\wedge$(-1);
Yb1(h)=G(h);
n inicial=input('no de origem ==>$ ');
n final=input('no final ==>$ ');
el(n inicial,i)=1;
el(n final,i)=-1;
h=h+1;
end
if k==2
close
home
cap=cap+1;
C(cap)=input('entre com o valor do capacitor ==>$ ');
vocap(cap)=input('entre com a tens\^{a}o inicial ==>$ ');
Vs(h,1)=vocap(cap);
Pcap(cap)=h;
n inicial=input('no de origem ==>$ ');
n final=input('no final ==>$ ');
el(n inicial,i)=1;
el(n final,i)=-1;
h=h+1;
end
if k==3
close
home
ind=ind+1;
L(ind)=input('entre com o valor da indutancia ==>$ ');
ioind(ind)=input('entre com a corrente inicial ==>$ ');
Js(h,1)=ioind(ind);
Pind(ind)=h;
n inicial=input('no de origem ==>$ ');
n final=input('no final ==>$ ');
el(n inicial,i)=1;
el(n final,i)=-1;
h=h+1;
end
if k==4
close
home
diodo=diodo+1;
Pdiodo(diodo)=h;
n inicial=input('no de origem ==>$ ');
nidiodo(diodo)=n inicial;
n final=input('no final ==>$ ');
nfdiodo(diodo)=n final;
el(n inicial,i)=1;
el(n final,i)=-1;
h=h+1;
end
if k==5
close
home
a=a+1;

```

```

if circuito==1
tdis(a)=input('entre com o tempo do primeiro puldo de disparo =>');
freqcha(a)=input('entre com a frequencia de chaveamento =>');
durscr(a)=input('entre com a dura\c {c}\{a}o do pulso de disparo => ');
end
if circuito==2
ang(a)=input('entre com o valor do angulo de disparo => ');
end
if circuito==3
tdis(a)=input('entre com o tempo do primeiro puldo de disparo =>');
freqcha(a)=input('entre com a frequencia de chaveamento =>');
durscr(a)=input('entre com a dura\c {c}\{a}o do pulso de disparo => ');
end
if circuito==4
ang(a)=input('entre com o valor do angulo de disparo => ');
end
Pscr(a)=h;
ninicial=input('no de origem => ');
niscr(a)=ninicial;
nfinal=input('no final => ');
nfscr(a)=nfinal;
el(ninicial,i)=1;
el(nfinal,i)=-1;
h=h+1;
end
if k==6
close
home
tr=tr+1;
if circuito==1
tdis(tr)=input('entre com o tempo do primeiro pulso de disparo =>');
freqcha(tr)=input('entre com a frequencia de chaveamento =>');
durtr(tr)=input('entre com a dura\c {c}\{a}o do pulso de disparo => ');
end
if circuito==2
ang(tr)=input('entre com o valor do angulo de disparo => ');
end
if circuito==3
tdis(tr)=input('entre com o tempo do primeiro pulso de disparo =>');
freqcha(tr)=input('entre com a frequencia de chaveamento =>');
durtr(tr)=input('entre com a dura\c {c}\{a}o do pulso de disparo => ');
end
if circuito==4
ang(tr)=input('entre com o valor do angulo de disparo => ');
end
Ptr(tr)=h;
ninicial=input('no de origem => ');
nitr(tr)=ninicial;
nfinal=input('no final => ');
nftr(tr)=nfinal;
el(ninicial,i)=1;
el(nfinal,i)=-1;
h=h+1;
end
if k==7

```

```

if circuito==1
Vs1(h)=input('entre com o valor da tensao =>$ ');
fte=fte+1;
Pfte(fte)=h;
ninicial=input('no de origem =>$ ');
nfinal=input('no final =>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=0.0001;
G(h)=10000;
Yb1(h)=G(h);
v=v+1;
h=h+1;
end
if circuito==2
Vs1(h)=input('entre com o valor da tensao =>$ ');
fte=fte+1;
Pfte(fte)=h;
freq(fte)=input('entre com o valor da frequencia =>$ ');
def(fte) = input('entre com o valor do defasamento =>$ ');
ninicial=input('no de origem =>$ ');
nfinal=input('no final =>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=0.00001;
G(h)=100000;
Yb1(h)=G(h);
v=v+1;
h=h+1;
end
if circuito==3
Vs1(h)=input('entre com o valor da tensao =>$ ');
fte=fte+1;
Pfte(fte)=h;
ninicial=input('no de origem =>$ ');
nfinal=input('no final =>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=0.0001;
G(h)=10000;
Yb1(h)=G(h);
v=v+1;
h=h+1;
end
if circuito==4
Vs1(h)=input('entre com o valor da tensao =>$ ');
fte=fte+1;
Pfte(fte)=h;
freq(fte)=input('entre com o valor da frequencia =>$ ');
def(fte) = input('entre com o valor do defasamento =>$ ');
ninicial=input('no de origem =>$ ');
nfinal=input('no final =>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=0.00001;

```

```

G(h)=100000;
Yb1(h)=G(h);
v=v+1;
h=h+1;
end
end
if k==8
if circuito==1
Js1(h)=input('entre com o valor da corrente =>>$ ');
fco=fco+1;
Pfco(fco)=h;
ninicial=input('no de origem =>>$ ');
nfinal=input('no final =>>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=10000;
G(h)=0.0001;
Yb1(h)=G(h)
c=c+1;
h=h+1;
end
if circuito==2
Js1(h)=input('entre com o valor da corrente =>>$ ');
fco=fco+1;
Pfco(fco)=h;
deffc(fco)=input('entre com o valor do defasamento');
freqco(fco)=input('entre com o valor da frecuencia =>>$ ');
ninicial=input('no de origem =>>$ ');
nfinal=input('no final =>>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=100000;
G(h)=0.00001;
Yb1(h)=G(h)
c=c+1;
h=h+1;
end
g=g+1;
if circuito==3
Js1(h)=input('entre com o valor da corrente =>>$ ');
fco=fco+1;
Pfco(fco)=h;
ninicial=input('no de origem =>>$ ');
nfinal=input('no final =>>$ ');
el(ninicial,i)=1;
el(nfinal,i)=-1;
R(h)=10000;
G(h)=0.0001;
Yb1(h)=G(h)
c=c+1;
h=h+1;
end
if circuito==4
Js1(h)=input('entre com o valor da corrente =>>$ ');
fco=fco+1;

```



```
Pfco(fco)=h;
deffc(fco)=input('entre com o valor do defasamento');
freqco(fco)=input('entre com o valor da frecuencia =>$ ');
ninizial=input('no de origem =>$ ');
nfinal=input('no final =>$ ');
el(ninizial,i)=1;
el(nfinal,i)=-1;
R(h)=100000;
G(h)=0.00001;
Yb1(h)=G(h)
c=c+1;
h=h+1;
end
g=g+1;
end
end
for i=1:nos
if i\sim$=nos1
for j=1:elementos
A(i,j)=el(i,j);
end
end
end
if tipoext==1
if circuito==1
Conv1
end
if circuito==2
Conv2
end
if circuito==3
Conv1
end
if circuito==4
Conv2
end
end
if tipoext==2
if circuito==1
Conver6
end
if circuito==2
Conver2
end
if circuito==3
Conver1
end
if circuito==4
Conver2
end
end
end
clc;
```

8.2 Subrotina para a Conversores CC/CC e CC/CA (topologia fixa)

```

\% Conv1.m
\% subrotina do programa SICON1 para a simula\c {c}\^{a}o de conversores
\%CC/CC e conversores CC/CA
\% op\c {c}\^{a}o por topologia fixa
\%
te=input('entre com per\{i}odo de amostragem ==>$ ');
tf=input('entre com tempo final de simula\c {c}\^{a}o ==>$ ');
p=0;
gat=0;
mod=0;
for i=1:cap
Yb1(Pcap(i))=(te/C(i))\wedge(-1);
end
for i=1:ind
Yb1(Pind(i))=te/L(i);
end
Yb=diag(Yb1);
t1=cputime;
while t\<=tf
p=p+1;
\% TESTE PARA OS SCRs \%
for d=1:a
if t\>=tdis(d)
gat=1;
end
if t\>=tdis(d)+ durscr(d)
gat=0;
tdis(d)=tdis(d)+1/freqcha(d);
end
if niscr(d)==nos1
E(niscr(d))=0;
end
if nfscr(d)==nos1
E(nfscr(d))=0;
end
if (E(niscr(d))-E(nfscr(d)))\>=0
if gat==1
Yb(Pscr(d),Pscr(d))=100000;
end
end
if (E(niscr(d))-E(nfscr(d)))\<=0
Yb(Pscr(d),Pscr(d))=0.00001;
end
end
\% TESTE PARA OS TRANSISTORES \%
for g=1:tr
if t\>=tdis(g)
gat=1;
end
if t\>=tdis(g)+durtr(g)

```

```

gat=0;
tdis(g)=tdis(g)+1/freqcha(g);
end
var6=0;
if nitr(g)==nos1
E(nitr(g))=0;
end
if nftr(g)==nos1
E(nftr(g))=0;
end
if gat==1
if (E(nitr(g))-E(nftr(g)))>=$=0
Yb(Ptr(g),Ptr(g))=100000;
end
end
if gat==0
Yb(Ptr(g),Ptr(g))=0.00001;
end
if (E(nitr(g))-E(nftr(g)))<=$0
Yb(Ptr(g),Ptr(g))=0.00001;
end
end
\% TESTE PARA OS DIODOS \%
for b=1:diodo
var6=0;
if nfdiodo(b)==nos1
E(nfdiodo(b))=0;
end
if nidiodo(b)==nos1
E(nidiodo(b))=0;
end
if (E(nidiodo(b))-E(nfdiodo(b)))>=$=0
Yb(Pdiodo(b),Pdiodo(b))=100000;
end
if (E(nidiodo(b))-E(nfdiodo(b)))<=$0
Yb(Pdiodo(b),Pdiodo(b))=0.00001;
end
end
end
t=t+te;
Yn=A*Yb*A';
tempo(p)=t;
Is=(A*Yb*Vs)-(A*Js); \% vetor fonte de corrente dos n\{o}s \%
E=(inv(Yn))*Is; \% vetor tens\{o}es dos n\{o}s \%
E1=real(E);
E2=E;
V=A'*E1; \% calculo das tens\{o}es dos bra\c {c}os \%
J=Yb*V - Yb*Vs + Js; \% calculo das correntes dos bra\c {c}os \%
\%
\% armazenamento das vari\{a}veis de interesse \%
for i=1:elementos
Tensao(p,i)=V(i);
Corrente(p,i)=J(i);
end
Dio=1;
SCR=1;

```

```

if fte>$0
Vs(Pfte(fte),1)=Vs1(Pfte(fte),1);
end
if fco>$0
Js(Pfco(fco),1)=Js1(Pfco(fco),1);
end
if cap>=$1
for i=1:cap
Vs(Pcap(i),1)=V(Pcap(i));
end
end
if ind>=$1
for i=1:ind
Vs(Pind(i),1)=-1*(J(Pind(i))*L(i)/te);
end
end
end
t1=cputime-t1;
for i=1:elementos
plot(tempo,Tensao(:,i))
title(sprintf('Tens\~{a}o no elemento \%d', i))
pause
close
plot(tempo,Corrente(:,i))
title(sprintf('Corrente no elemento \%d', i))
pause
close
end
home

```

8.3 Subrotina para a Conversores CA/CC e CA/CA (topologia fixa)

```

\% Conv2.m
\% subrotina do programa SICON2 para conversores CA/CC e CA/CA
\% Uso de topologia fixa
\%
te=(1/freq(1))/1500;
tf=2*(1/freq(1));
p=0;
for i=1:cap
Yb1(Pcap(i))=(te/C(i))$^{\wedge}$(-1);
end
for i=1:ind
Yb1(Pind(i))=te/L(i);
end
Yb=diag(Yb1);
fator1=0;
t1=cputime;
while t<=$tf
if teta>=$360
fator1=1;

```

```

end
p=p+1;
E(nos1)=0;
\% TESTE PARA OS SCRS \%
for d=1:a
if (E(niscr(d))-E(nfscr(d)))>$0
if teta$>$(ang(d)+fator1*360);
Yb(Pscr(d),Pscr(d))=10000000;
end
else
Yb(Pscr(d),Pscr(d))=0.0000001;
end
if teta<$(ang(d)+fator1*360);
Yb(Pscr(d),Pscr(d))=0.0000001;
end
end
\% TESTE PARA OS DIODOS \%
for b=1:diodo
if (E(nidiado(b))-E(nfdiado(b)))>$0
Yb(Pdiado(b),Pdiado(b))=10000000;
else
Yb(Pdiado(b),Pdiado(b))=0.0000001;
end
end
tempo(p)=t;
t=t+te;
teta=(360*freq(1)*t);
Yn=A*Yb*A';
Is=A*(Yb*Vs-Js); \% vetor fonte de corrente dos n\{o} s \%
E=(inv(Yn))*Is; \% vetor tens\~{o}es dos n\{o}s \%
E1=real(E);
V=A'*E1; \% calculo das tens\~{o}es dos bra\c{c}os \%
J=Yb*(V-Vs) + Js; \% calculo das correntes dos bra\c{c}os \%
\%
\% armazenamento das vari\{a}veis de interesse \%
Tensao(p,1:elementos)=V';
Corrente(p,1:elementos)=J';
if fte$>=1
for i=1:fte
Vs(Pfte(i),1)=Vs1(Pfte(i),1)*sin(2*pi*freq(i)*t-def(i));
end
end
if fco$>=1
for i=1:fco
Js(Pfco(i),1)=Js1(Pfco(i),1)*sin(2*pi*freq(i)*t-deffc(i));
end
end
if cap$>=1
for i=1:cap
Vs(Pcap(cap),1)=V(Pcap(cap));
end
end
if ind$>=1
for i=1:ind
Vs(Pind(ind),1)=-1*(J(Pind(ind))*L(ind)/te);

```

```

end
end
end
t1=cputime-t1;
for i=1:elementos
plot(tempo,Tensao(:,i))
title(sprintf('Tens\~{a}o no elemento \%d', i))
pause
close
plot(tempo,Corrente(:,i))
title(sprintf('Corrente no elemento \%d', i))
pause
close
end
home

```

8.4 Subrotina para a Conversores CC/CC (topologia variável)

```

\% Conver1.m
\% subrotina do programa SICON1.m para conversores CC/CC e CC/CA
\% Uso de estrutura variavel
\%
te=input('entre com per\~{i}odo de amostragem ==>$ ');
tf=input('entre com tempo final de simula\c {c}\~{a}o ==>$ ');
p=0;
for i=1:cap
Yb1(Pcap(i))=(te/C(i))\~{\wedge}$(-1);
end
for i=1:ind
Yb1(Pind(i))=te/L(i);
end
fator1=0;
mod=0;
gat=zeros(a+tr,1);
elementos1=2;
comando=zeros(a+tr,1);
aux=[1:elementos];
t1=cputime;
while t$<=$tf
p=p+1;
int=1;
E(nos1)=0;
elementos1=2;
clear Posint;
\% TESTE PARA OS SCRs \%
for d=1:a
if t$>=$tdis(d)
gat(d)=1;
end
if t$>=$tdis(d)+durscr(d)
gat(d)=0;

```

```

tdis(d)=tdis(d)+1/freqcha(d);
end
if (E(niscr(d))-E(nfscr(d)))>=$=0
if gat(d)==1
comando(d)=1;
end
end
if comando(d)==1
Yb1(Pscr(d))=100000;
end
if (E(niscr(d))-E(nfscr(d)))<$=0
comando(d)=0;
end
if comando(d)==0
Posint(int)=Pscr(d);
Tensao(p,Pscr(d))=(E(niscr(d))-E(nfscr(d)));
int=int+1;
elementos1=1;
end
end
\% TESTE PARA OS TRANSISTORES \%
for g=1:tr
if t$>=$=tdis(g)
gat(g)=1;
end
if t$>=$=tdis(g)+durtr(g)
gat(g)=0;
tdis(g)=tdis(g)+1/freqcha(g);
end
if (E(nitr(g))-E(nftr(g)))<$=0
comp1=1;
end
if gat(g)==0
comp2=1;
end
if gat(g)==1;
if (E(nitr(g))-E(nftr(d)))>=$=0
Yb1(Ptr(g))=100000;
elementos1=2;
end
end
if (comp1|$comp2)==1
Posint(int)=Ptr(g);
Tensao(p,Ptr(g))=(E(nitr(g))-E(nftr(g)));
int=int+1;
elementos1=1;
end
end
\% TESTE PARA OS DIODOS \%
for b=1:diodo
if (E(nidiodo(b))-E(nfdiodo(b)))>=$=0
Yb1(Pdiodo(b))=100000;
end
if (E(nidiodo(b))-E(nfdiodo(b)))<$=0
Posint(int)=Pdiodo(b);

```

```

Tensao(p,Pdiodo(b))=(E(nidiado(b))-E(nfdiado(b)));
int=int+1;
elementos1=1;
end
end
if elementos1$sim$=1;
fix1=aux;
Yb=diag(Yb1);
Yn=A*Yb*A';
Is=A*(Yb*Vs-Js); \% vetor fonte de corrente dos n\'}{o}s \%
E=(inv(Yn))*Is; \% vetor tens\~{o}es dos n\'}{o}s \%
E1=real(E);
V=A'*E1; \% calculo das tens\~{o}es dos bra\c {c}os \%
J=Yb*(V-Vs) + Js; \% calculo das correntes dos bra\c {c}os \%
end
if elementos1==1
indices=zeros(1,elementos);
indices(1,Posint)=Posint;
fix1=find(aux$sim$=indices);
B=A(:,fix1);
C=diag(Yb1(fix1));
D=Vs(fix1);
F=Js(fix1);
Yn=B*C*B';
Is=B*(C*D-F); \% vetor fonte de corrente dos n\'}{o}s \%
E=(inv(Yn))*Is; \% vetor tens\~{o}es dos n\'}{o}s \%
E1=real(E);
V=B'*E1; \% calculo das tens\~{o}es dos bra\c {c}os \%
J=C*(V-D) + F; \% calculo das correntes dos bra\c {c}os \%
end
tempo(p)=t;
t=t+te;
\%armazenamento das vari\'}{a}veis de interesse
Tensao(p,fix1)=V';
Corrente(p,fix1)=J';
if fte$>$=1
for i=1:fte
Vs(Pfte(i),1)=Vs1(Pfte(i),1);
end
end
if fco$>$=1
for i=1:fco
Js(Pfco(i),1)=Js1(Pfco(i),1);
end
end
if cap$>$=1
for i=1:cap
Vs(Pcap(i))=V(Pcap(i));
end
end
if ind$>$=1
for i=1:ind
Vs(Pind(i))=-1*(J(Pind(i))*L(i)/te);
end
end

```



```

end
t1=cputime-t1;
for i=1:elementos
plot(tempo,Tensao(:,i))
title(sprintf('Tens\~{a} o no elemento \%d', i))
pause
close
plot(tempo,Corrente(:,i))
title(sprintf('Corrente no elemento \%d', i))
pause
close
end
home

```

8.5 Subrotina para a Conversores CA/CC e CA/CA (topologia variável)

```

\% Conver2.m
\% subrotina do programa SICON1 para conversores CA/CC e CA/CA
\% Uso de topologia variavel
\%
te=(1/freq(1))/2000;
tf=2*(1/freq(1));
p=0;
for i=1:cap
Yb1(Pcap(i))=(te/C(i))$^{\wedge}$(-1);
end
for i=1:ind
Yb1(Pind(i))=te/L(i);
end
fator1=0;
aux=[1:elementos];
t=0;
t1=cputime;
while t<=$tf
if teta$>=$360
fator1=1;
end
p=p+1;
int=1;
elementos1=2;
E(nos1)=0;
\% TESTE PARA OS SCRs \%
for d=1:a
if (E(niscr(d))-E(nfscr(d)))$>$0
if teta$>$(ang(d)+fator1*360);
Yb1(Pscr(d))=10000000;
else
Posint(int)=Pscr(d);
Tensao(p,Pscr(d))=(E(niscr(d))-E(nfscr(d)));
int=int+1;
elementos1=1;

```

```

end
else
Posint(int)=Pscr(d);
Tensao(p,Pscr(d))=(E(niscr(d))-E(nfscr(d)));
int=int+1;
elementos1=1;
end
end
\% TESTE PARA OS TRANSISTORES \%
for g=1:tr
if teta$>$(ang(g)+fatori*360);
if (E(nitr(g))-E(nftr(g)))$>$0
Yb1(Ptr(g))=10000000;
else
Posint(int)=Ptr(g);
Tensao(p,Ptr(g))=(E(nitr(g))-E(nftr(g)));
int=int+1;
elementos1=1;
end
Posint(int)=Ptr(g);
Tensao(p,Ptr(g))=(E(nitr(g))-E(nftr(g)));
int=int+1;
elementos1=1;
end
end
\% TESTE PARA OS DIODOS \%
for b=1:diodo
if (E(nidiodo(b))-E(nfdiodo(b)))$>=$0
Yb1(Pdiodo(b))=100000;
else
Posint(int)=Pdiodo(b);
Tensao(p,Pdiodo(b))=(E(nidiodo(b))-E(nfdiodo(b)));
int=int+1;
elementos1=1;
end
end
if elementos1$\sim$=1;
fix1=aux;
Yb=diag(Yb1);
Yn=A*Yb*A';
Is=A*(Yb*Vs-Js); \% vetor fonte de corrente dos n\{o}s \%
E=(inv(Yn))*Is; \% vetor tens\{o}es dos n\{o}s \%
E1=real(E);
V=A'*E1; \% calculo das tens\{o}es dos bra\c{c}os \%
J=Yb*(V-Vs) + Js; \% calculo das correntes dos bra\c{c}os \%
else
indices=zeros(1,elementos);
indices(1,Posint)=Posint;
fix1=find(aux$\sim$=indices);
B=A(:,fix1);
C=diag(Yb1(fix1));
D=Vs(fix1);
F=Js(fix1);
Yn=B*C*B';
Is=B*(C*D-F); \% vetor fonte de corrente dos n\{o}s \%

```

```

E=(inv(Yn))*Is; \% vetor tens\~{o}es dos n\~{o}s \%
E1=real(E);
V=B'*E1; \% calculo das tens\~{o}es dos bra\c {c}os \%
J=C*(V-D) + F; \% calculo das correntes dos bra\c {c}os \%
end
tempo(p)=t;
t=t+te;
teta=(360*freq(1)*t);
\%armazenamento das vari\~{a}veis de interesse
Tensao(p,fix1)=V';
Corrente(p,fix1)=J';
if fte>=$=1
for i=1:fte
Vs(Pfte(i),1)=Vs1(Pfte(i),1)*sin(2*pi*freq(i)*t-def(i));
end
end
if fco>=$=1
for i=1:fco
Js(Pfco(i),1)=Js1(Pfco(i),1)*sin(2*pi*freq(i)*t-deffc(i));
end
end
if cap>=$=1
for i=1:cap
Vs(Pcap(i),1)=V(Pcap(i));
end
end
if ind>=$=1
for i=1:ind
Vs(Pind(i))=-1*(J(Pind(i))*L(i)/te);
end
end
end
t1=cputime-t1;
for i=1:elementos
plot(tempo,Tensao(:,i))
title(sprintf('Tens\~{a}o no elemento \%d', i))
pause
close
plot(tempo,Corrente(:,i))
title(sprintf('Corrente no elemento \%d', i))
pause
close
end
home

```

8.6 Programa SICON2

```

\% sicon2.m
\% TOOLBOX PARA SIMULACAO DE CONVERSORES ESTATICOS
\%
\% UNIVERSIDADE FEDERAL DA PARAIBA
\% CENTRO DE CIENCIAS E TECNOLOGIA
\% DEPARTAMENTO DE ENGENHARIA ELETRICA

```

```

\% LEIAM
\%
\%-----
\%Parte 1: Entrada da Estrutura e Dados do Conversor
\% -----
clear;clc;
circuito=menu('\&TIPO DE CIRCUITO','conversor CC-CC','conversor
CA-CC','conversor CC-CA','conversor CA-CA');clc;
nt = input('entre com a quantidade de n\{o} s ==>$ ');clc;
br = input('entre com a quantidade de bra\{c} os ==>$ ');clc;
nc = input('entre com o n\{u} mero de capacitores em \{a} rvore ==>$
');clc;
ll = input('entre com o n\{u} mero de indut\{a} ncias em enlace ==>$
');clc;
ld = input('entre com o n\{u} mero de diodos em enlace ==>$ ');clc;
ls = input('entre com o n\{u} mero de chaves em enlace ==>$ ');clc;
lrr = input('entre com o n\{u} mero de resist\{e} ncias em enlace ==>$
');clc;
nd = input('entre com o n\{u} mero de diodos em \{a} rvore ==>$ ');clc;
ns = input('entre com o n\{u} mero de chaves em \{a} rvore ==>$ ');clc;
ngg = input('entre com o n\{u} mero de condut\{a} ncias em \{a} rvore ==>$
> $ ');clc;
\%
F=input('MATRIZ F ==>$ '); \% submatriz percurso fechado fundamental
clc;
R1=[]; R2=[]; R3=[]; R4=[]; R5=[]; R6=[];
ro=100000; \% resistencia grande
rd=0.00001; \% resistencia pequena
gs=1/ro; \% condutancia pequena
gl=1/rd; \% alta condutancia
\%
rgg=ro;
rpp=rd;
er(1:ld+ls,1)=zeros(1:ld+ls)';
ig(1:nd+ns,1)=zeros(1:nd+ns)';
R1(1:ld,1)=rgg*ones(1:ld)';
R2(1:ls,1)=rgg*ones(1:ls)';
Rru = [R1;R2]; \% submatriz diodo-chave (enlace)
for i=1:lrr
S=sprintf(' valor da resist\{e} ncia numero \%d \{c\}\{c\}\{c\}',i, ',', ',', '$>$
');
R3(i,1)=input(S);
er(ld+ls+i,1)=input('entre com o valor da fonte de tens\{a} o
correspondente ==>$ ');
Amplitude(ld+ls+i,1)=er(ld+ls+i,1);
if er(ld+ls+i,1)$\sim$=0
tensao=menu('tipo de tens\{a} o ', 'CONT\{I} NUA', 'ALTERNADA');clc;
if tensao==2
def1=input('defasagem da tens\{a} o (em radianos) ==>$ ');
def(ld+ls+i,1)=def1;
end
end
end
Rrr = [R3]; \% submatriz resistencia (enlace)
R4(1:nd,1)=gs*ones(1:nd)';

```

```

R5(1:ns,1)=gs*ones(1:ns)';
Ggu = [R4;R5]; \% submatriz diodo-chave (arvore)
for i=1:ngg
S=sprintf(' valor da condut\^{a} ncia n\^{u} mero \%d \%c\%c\%c',i, ',' ,',' ,'%
$>$' );
R6(i,1)=input(S);
ig(nd+ns+i,1)=input('entre com o valor da fonte de corrente correspondente ==%
$>$ ');
end
Ggg = [R6]; \% submatriz condutancia
for i=1:ll
S=sprintf(' valor da indut\^{a} ncia n\^{u} mero \%d \%c\%c\%c',i, ',' ,',' ,'%
>$' );
L1(i,1)=input(S);
el(i,1)=input('fonte de tens\^{a} o sobre o indutor ==>$ ');
xkl1(i,1)=input('corrente inicial ==>$ ');
end
for i=1:nc
S=sprintf(' valor do capacit\^{a} ncia n\^{u} mero \%d \%c\%c\%c',i,
',' ,',' ,'$>$' );
C1(i,1)=input(S);
ic(i,1)=input('fonte de corrente do capacitor ==>$ ');
xkc1(i,1)=input('tens\^{a} o inicial ==>$ ');
end
for i=1:ls
Tfs=sprintf('frequ\^{e} ncia de chaveamento da chave de enlace n. \%d
\%c\%c\%c',i, ',' ,',' ,'$>$' );
fsls(i)=input(Tfs);
if circuito==1
tils(i)=input('instante inicial para chaveamento ==>$ ');
duracao(i)=input('dura\{c} \^{a} o do pulso de chaveamento ==>$ ');
end
if circuito==2
angls(i)=input('angulo de disparo ==>$ ');
duracao(i)=pi/20;
end
if circuito==3
tils(i)=input('instante inicial para chaveamento ==>$ ');
duracao(i)=input('dura\{c} \^{a} o do pulso de chaveamento ==>$ ');
end
if circuito==4
angls(i)=input('angulo de disparo ==>$ ');
duracao(i)=pi/20;
end
end
for i=1:ns
Tfn=sprintf('frequ\^{e} ncia de chaveamento da chave de \^{a} rvore n. \%d
\%c\%c\%c',i, ',' ,',' ,'$>$' );
fsns(i)=input(Tfn);
if circuito==1
tins(i)=input('instante inicial para chaveamento ==>$ ');
duracao(i)=input('dura\{c} \^{a} o do pulso de chaveamento ==>$ ');
end
if circuito==2
angns(i)=input('angulo de disparo ==>$ ');

```

```

duracao(i)=pi/20;
end
if circuito==3
tins(i)=input('instante inicial para chaveamento ==>$ ');
duracao(i)=input('dura\c{c} \^{a} o do pulso de chaveamento ==>$ ');
end
if circuito==4
angns(i)=input('angulo de disparo ==>$ ');
duracao(i)=pi/20;
end
end
\%
\%
\%
ne=nt-1; \%no. de ramos em tree;
li=br-ne; \%no. de links
lr=ld+ls+lrr; \%no. de resistencias,diodos e chaves em link
ng=nd+ns+ngg; \%no. de condutancias,diodos e chaves em tree
\%
\% SUBMATRIZES DA MATRIZ F \%
Frc=F(1:lr,1:nc);
Flc=F(lr+1:lr+ll,1:nc);
Frg=F(1:lr,nc+1:nc+ng);
Flg=F(lr+1:lr+ll,nc+1:nc+ng);
\%
\%parametros indutivos
L=diag([L1]); \%matriz diagonal indutancia
\%
\%parametros capacitivos
C=diag([C1]); \%matriz diagonal capacitancia
\%
xk1=[xkc1; xkl1];
\%
\%
\%-----
\%Parte 2: Simulacao do Conversor
\%-----
\%matrizes constantes
CL=[C zeros(nc,ll); zeros(ll,nc) L];
CLi=inv(CL);
\%
\% VALORES INICIAIS DAS VARIAVEIS
\%
jjr1=zeros(lr,1); \%correntes nos diodos, chaves e res. em link
vvg1=zeros(ng,1); \%tensoes nos diodos, chaves e cond. em tree
gvs1=zeros(ns,1); \%sinal de gatilho em chaves em tree
gjs1=zeros(ls,1); \%sinal de gatilho em chaves em link
gvs=gvs1;
gjs=gjs1;
\%
if circuito==1
tipo1
end
if circuito==2
tipo2

```

```

end
if circuito==3
tipo3
end
if circuito==4
tipo4
end
\%

```

8.7 Subrotina para circuitos CC/CC

```

\% tipo1.m
\% ESTRUTURA PARA SIMULACAO DE CONVERSORES CC/CC
\%
\%
\%PARAMETROS INICIAIS DE TEMPO
tt=0;
kmod=0;
tx0=0;
\%
\%
clg
chaves
\% TERMOS VARIAVEIS
Rru0=Rru;
Rr=diag([Rru; Rrr]);
Ggu0=Ggu;
Gg=diag([Ggu;Ggg]);
Gr=inv(Rr);
Rg=inv(Gg);
\%
\% CHAMADA DA FUNCTION
\%chamada da function fsc2.m
[Ad,Bd,Ac,Bc,Ri,Gi,Rg,Gr,te]=fsc2(Rr,Gg,CLi,Frc,Frg,Flc,Flg,nc,ll);
te;
\%uk=[ic; ig; er; el] \%vetor fontes de entrada
\%-----
\% simulacao do sistema
\%
k=0;
tfinal=input('entre com o tempo final de simula\c {c}\^{-a}o ==>$ ');
while tt <=$ tfinal
uk=[ic; ig; er; el];
k=k+1;
xk=xk1;
xk1=Ad*xk+Bd*uk;
xkc1=xk1(1:nc,:);
xkl1=xk1(nc+1:nc+ll,:);
\%calculo das variaveis de saida
jjr=jjr1;
vvg=vvg1;
jjr1=Ri*(-Frg*Rg*(Flg'*xkl1-ig)-Frc*xkc1-er);
vvg1=-Gi*(Frg'*Gr*(Frc*xkc1+er)-Flg'*xkl1+ig);
\%tempo atual

```

```

tt=tt+te;
\%
\% Simulacao do comportamento das chave
\%
\%
if q==1
scr
end
if q==2
transistor
end
if g==1
sdiodo
end
if g==2
ndiodo
end
if kmod > $ 0
tx0;
tx0=tx0/kmod;
[Add,Bdd] = c2d(Ac,Bc,tx0); \%
xk1=Add*xk+Bdd*uk; \%
xkc1=xk1(1:nc,:);
xkl1=xk1(nc+1:nc+ll,:);
\% calculo das variaveis de saida
jjr1=Ri*(-Frg*Rg*(Flg'*xkl1-ig)-Frc*xkc1-er); \%
vvg1=-Gi*(Frg'*Gr*(Frc*xkc1+er)-Flg'*xkl1+ig); \%
tt=tt-(te-tx0); \%
tx0=0;
kmod=0;
Rr;
Gg;
k;
Rr=diag([Rru; Rrr]);
Gg=diag([Ggu; Ggg]);
[Ad,Bd,Ac,Bc,Ri,Gi,Rg,Gr,te]=fsc2(Rr,Gg,CLi,Frc,Frg,Flc,Flg,nc,ll);
te;
end
\% DISPARO DAS CHAVES \%
for i=1:ls
if tt>$=tils(i)
gjs1(i,1)=1;
end
if tt>$tils(i)+duracao(i)
gjs1(i,1)=0;
tils(i)=tils(i)+1/fsls(i);
end
gjs(i,1)=gjs1(i,1);
end
for i=1:ns
if tt>$=tins(i)
gvs1(i,1)=1;
end
if tt>$tins(i)+duracao(i)
gvs1(i,1)=0;

```



```

tins(i)=tins(i)+1/fsns(i)+duracao(i);
end
gvs(i,1)=gvs1(i,1);
end
\%armazenamento das saidas
st(k) = tt;
for kk=1:nc
sxc(kk,k) = xkc1(kk);
end
for kk=1:ll
sxl(kk,k) = xkl1(kk);
end
for kk=1:ng
svg(kk,k) = vvg1(kk);
end
for kk=1:lr
sjr(kk,k) = jjr1(kk);
end
end
for i=1:nc
plot(st,sxc(i:i,:))
title(sprintf('Tens\~{a}o no capacitor \%d', i))
pause
close
end
for i=1:ll
plot(st,sxl(i:i,:))
title(sprintf('Corrente no capacitor \%d', i))
pause
close
end
for i=1:ng
plot(st,svg(i:i,:))
title(sprintf('Tens\~{a}o na condut\~{a}ncia \%d', i))
pause
close
end
for i=1:lr
plot(st,sjr(i:i,:))
title(sprintf('Corrente da resist\~{e}ncia \%d', i))
pause
close
end
\%

```

8.8 Subrotina para a conversão CA/CC

```

\% tipo2.m
\% ESTRUTURA PARA SIMULACAO DE CONVERSORES CA/CC
\%
\%
\%PARAMETROS INICIAIS DE TEMPO
tt=0;

```

```

kmod=0;
tx0=0;
\%
clg
\% TERMOS VARIÁVEIS
Rru0=Rru;
Rr=diag([Rru; Rrr]);
Ggu0=Ggu;
Gg=diag([Ggu;Ggg]);
Gr=inv(Rr);
Rg=inv(Gg);
\%
\% CHAMADA DA FUNCTION
\%chamada da function fsc2.m
[Ad,Bd,Ac,Bc,Ri,Gi,Rg,Gr,te]=fsc2(Rr,Gg,CLi,Frc,Frg,Flc,Flg,nc,ll);
te;
\%uk=[ic; ig; er; el] \%vetor fontes de entrada
\%-----
\% simulacao do sistema
\%
k=0;
w=2*pi*60;
teta =0;
if ls$\sim$=0
q=menu('TIPO DE TIRISTOR', 'SCR', ' Transistor');
end
if ns$\sim$=0
q=menu('TIPO DE TIRISTOR', 'SCR', ' Transistor');
end
tfinal=input('entre com o tempo final de simula\c{c} \^{a} o ==>$ ');
while tt<=$=tfinal
for i=1:lrr
er(ld+ls+i,1)=Amplitude(ls+ld+i,1)*sin(teta-def(ls+ld+i,1))
end
uk=[ic; ig; er; el];
k=k+1;
xk=xk1;
xk1=Ad*xk+Bd*uk;
xkc1=xk1(1:nc,:);
xkl1=xk1(nc+1:nc+ll,:);
\%calculo das variaveis de saida
jjr=jjr1;
vvg=vvg1;
jjr1=Ri*(-Frg*Rg*(Flg'*xkl1-ig)-Frc*xkc1-er);
vvg1=-Gi*(Frg'*Gr*(Frc*xkc1+er)-Flg'*xkl1+ig);
\%tempo atual
tt=tt+te;
teta=teta+w*te;
\%
\% Simulacao do comportamento das chave
\%
\%
if q==1
scr
end

```

```

if q==2
transistor
end
if ld$\sim$=0
sdiodo
end
if nd$\sim$=0
sdiodo
end
if kmod >= 0
tx0;
tx0=tx0/kmod;
[Add,Bdd] = c2d(Ac,Bc,tx0); \%
xk1=Add*xk+Bdd*uk; \%
xkc1=xk1(1:nc,:);
xkl1=xk1(nc+1:nc+ll,:);
\%calculo das variaveis de saida
jjr1=Ri*(-Frg*Rg*(Flg'*xkl1-ig)-Frc*xkc1-er); \%
vvg1=-Gi*(Frg'*Gr*(Frc*xkc1+er)-Flg'*xkl1+ig); \%
tt=tt-(te-tx0); \%
tx0=0;
teta=teta-(w*te-w*tx0);
kmod=0;
Rr=diag([Rru; Rrr]);
Gg=diag([Ggu;Ggg]);
[Ad,Bd,Ac,Bc,Ri,Gi,Rg,Gr,te]=fsc2(Rr,Gg,CLi,Frc,Frg,Flc,Flg,nc,ll);
te;
end
\% DISPARO DAS CHAVES \%
for i=1:ls
if teta>=angls(i)
gjs1(i,1)=1;
end
if teta>=angls(i)+duracao(i)
gjs1(i,1)=0;
angls(i)=angls(i)+2*pi;
end
gjs(i,1)=gjs1(i,1);
end
for i=1:ns
if teta>=angns(i)
gvs1(i,1)=1;
end
if teta>=angns(i)+duracao(i)
gvs1(i,1)=0;
angns(i)=angns(i)+2*pi;
end
gvs(i,1)=gvs1(i,1);
end
\%armazenamento das saidas
st(k) = tt;
sxc(1:nc,k) = xkc1(1:nc);
sxl(1:ll,k) = xkl1(1:ll);
svg(1:ng,k) = vvg1(1:ng);
sjr(1:lr,k) = jjr1(1:lr);

```

```

end
for i=1:nc
plot(st, sxc(i:i,:))
title(sprintf('Tens\~{a}o no capacitor \%d', i))
pause
close
end
for i=1:ll
plot(st, sxl(i:i,:))
title(sprintf('Corrente no indutor \%d', i))
pause
close
end
for i=1:ng
plot(st, svg(i:i,:))
title(sprintf('Tens\~{a}o na condut\~{a}ncia \%d', i))
pause
close
end
for i=1:lr
plot(st, sjr(i:i,:))
title(sprintf('Tens\~{a}o na resist\~{e}ncia \%d', i))
pause
close
end
\%
\%

```

8.9 Function fsc2 (Cálculo do Período de Amostragem)

```

function[Ad,Bd,Ac,Bc,Ri,Gi,Rg,Gr,te]=fsc2(Rr,Gg,CLi,Frc,Frg,Flc,Flg,nc,ll);
\% Function fsc2.m (18/11/97)
\%
Gr=inv(Rr)
Rg=inv(Gg)
Rt=Rr+Frg*Rg*Frg'
Gt=Gg+Frg'*Gr*Frg
Ri=inv(Rt);
Gi=inv(Gt);
Ht=Flc'-Frc'*Ri*Frg*Rg*Flg';
Zt=Flg*Gi*Flg';
Yt=Frc'*Ri*Frc;
B12=-Frc'*Ri*Frg*Rg;
B13=Frc'*Ri;
B22=-Flg*Gi;
B23=-Flg*Gi*Frg'*Gr;
Bt1=[ones(nc) B12 B13 zeros(nc,ll)];
Bt2=[zeros(ll,nc) B22 B23 ones(ll)];
Bt=[Bt1; Bt2]
Ac1=[-Yt Ht];
Ac2=-[Ht' Zt];

```

```
Aco=[Ac1; Ac2];
Ac=CLi*Aco;
Bc=-CLi*Bt;
\% calculo das matrizes discretizadas
egac=eig(Ac);
fd=(nc+1);
grau=1;
for mn=1:fd
imeg(mn)=abs(1*imag(egac(mn)));
tmac(mn)=abs(1/real(egac(mn)));
end
for sy=1:fd
for sd=1:fd
if imeg(sy)$>=$imeg(sd)
vv=imeg(sy);
if tmac(sy)$>=$tmac(sd)
tre=tmac(sy);
end
end
end
end
if vv$>$0
tei=6/vv/30;
end
tgg=1;
for mn=1:fd
tgg=tgg*sqrt(tmac(mn));
end
tgg1=tgg/4;
if tei$<$=tgg1
te=tei;
else
te=tgg1;
end
te=10e-6;
\%te=(1/60)/1500;
\%tf=2*(1/60);
\%
[Ad,Bd]=c2d(Ac,Bc,te)
```

Referências Bibliográficas

- [1] N. Mohan, T.M. Undeland, and W.P. Robbins. *Power Electronics: Converters, Applications and Design*. 2nd edition, John Wiley, New York, 1995.
- [2] B. Bose. Power electronics-recent advances and future perspectives. In *IEEE Ind. Elec. Conf. Proc.*, pages 14–15, 1993.
- [3] G.A. Franz. Multilevel simulation tools for power converters. *IEEE-Applied Power Conference*, pages 629–633, March 11-16 1990.
- [4] N. Mohan, W.P. Robbins, Undeland, R. Nilsen, and O. Mo. Simulation of power electronic and motion control systems-an overview. *IEEE Trans. on Education*, 82(8):1287–1302, August 1994.
- [5] V.J. Thottuvelil. Challenges in computer-based analysis/simulation and design of power electronic circuit. In *IEEE Workshop on Computer in Power Electronics*, pages 1–8, 1988.
- [6] J.G. Kassakian. Simulating of power electronic systems-an approach. *Proc. IEEE*, 67:1428–1439, October 1979.
- [7] W.J. McCalla. *Fundamentals of Computer-Aided Circuit Simulation*. Kluwer Academic Publishers, 1988.
- [8] D. Hart. Circuit simulation as an aid in teaching the principles of power electronics. *IEEE Trans. on Education*, 36(1):10–16, Feb 1993.
- [9] *PSPICE manual, versão 4.05*. MicroSim Corporation, 1995.
- [10] H. Foch, C. Reboulet, and J. Schonek. A general digital computer simulation program for thyristor static converters (programme sacso): Application examples. In *2nd IFAC Symposium on Control in Power Electronics and Electrical Drives*, pages 1–5, 1997.

- [11] D.C. Martins. Programa de simulação numerica aplicado a conversores estáticos de potência. In *IEEE*, pages 691–695, 1991.
- [12] *MATLAB*. The MathWorks, 1994.
- [13] D. Martins. Simulação sequencial sem o conhecimento a "priori"-aplicação e modelagem de transistores de potencia. In *7 Congresso Brasileiro de Automatica*, pages 418–423, 1988.
- [14] R.A. Rohrer. *Circuit Theory: An introduction to state variable approach*. McGraw-Hill, New York, 1970.
- [15] Simulation and cad in power electronics: Circuit software. *Power Electronics (France)*, 26:60–66, 1991.
- [16] F. Oms. Script simulator: from converter to semiconductor. *EPE'89*, 1:207–212, 1989.
- [17] V. Rajagopalan. User-friendly dedicated power electronic converter simulator. In *1990 IEEE Workshop on Computers in Power Electronics*, pages 183–204, Aug,1990.
- [18] C.C. Liu, J. Hsieh, G.H.K. Chang, J.M. Bocek, and Y.T. Hsiao. A fast-decoupled method for time-domain simulation of power converters. *IEEE Transaction on Power Electronics*, (1):37–45, January 1993.
- [19] C.J. Hsiao, R.B. Riddley, and F.C. Lee. The simulation of switching converters using the new version cosmir program. In *Proc. 1989 VPEC Seminar*, pages 121–130, 1989.
- [20] R. Tymersky. A fast time-domain simulator for power electronics systems. In *IEEE APEC'93*, pages 477–483, 1993.
- [21] W. Long. Emtp: a powerful tool for analyzing power system transients. In *IEEE Computer Applications in Power*, pages 36–41, July 1990.
- [22] J.A. Martinez-Velasco, G.A. Capolina, and H. Henao. Atp: A suitable tool for the simulation of power converters and drivers. In *1992 IEEE Workshop on Computer in Power Electronics*, pages 87–99, 1992.
- [23] G.A. Capolino and H. Henao. Design and simulation of power electronics converters control circuits using emtp. In *1990 IEEE Workshop on Computer in, Power Electronics*, pages 47–61, Aug 1990.

- [24] J.D. Lavers, H. Jin, and R.W. Cheung. Analysis of power electronics circuits with feedback control: a general approach. In *IEE PROC*, pages Processo N 3698/97-9, 1990.
- [25] C.W. Ho. The modified nodal approach to network analysis. *IEEE Transaction on Circuits and Systems*, (6):504-509, June 1975.
- [26] *SABER user's Manual, Release 2.3*. Analogy Inc., 1989.
- [27] S.A. Sudha, A. Chandrasehkar, and V. Rajagopalan. New approach to switch modelling in the analysis of power electronic systems. *IEE Procs-B*, 140(2):115-123, Mar 1993.
- [28] C. A. Desoer e E. S. Kuh. *Teoria Básica de Circuitos*. Guanabara Dois, 1983.
- [29] P.O. Lauritzen. Power semiconductor device models for use in circuit simulators. In *Conf. Rec. of IEEE Industry Applications Society*, pages 1559-1563, 1990.
- [30] D. Schroder. Modeling and cae for power electronic topologies. In *Proc. IECON*, pages 626-636, 1993.
- [31] O. Apeldoorn, L. Schulting, and H.-Ch. Skudelny. In *Power Eletronics Specialists Conf. Rec.*, pages 1074-1081, 1992.
- [32] L. T. Pillage, R. A. Rohrer, and C. Visweswariah. *Electronic Circuit and System Simulation Methods*. McGraw-Hill, Inc, 1994.
- [33] T. Oliveira, C.B. Jacobina, and E.R.C Silva. Automatic simulation of static converters. In *EPE'91*, pages 691-695, 1991.
- [34] M. Nakahara. A fast computer algorithm for switching converters. *IEEE Trans. on PE*, 12(1):180-186, March 11-16 1997.
- [35] E. R. C Silva. *Conversores CC-CC (apostila da disciplina Eletrônica de Potência do curso de graduação em Engenharia Elétrica da UFPB)*. Campina Grande, Brasil, 1992.
- [36] M. Vilela, V. Farias, and E.G. Silva. A buck pwm soft-switching converter using a single active swicth. In *IEEJ-IPEC'95 record*, pages 956-961, 1995.
- [37] M. Vilela, E. A. A. Coelho, J. Vieira, L.C. Freitas, and V. Farias. Pwm soft-switching converters using a single active swicth. In *IEEE-APEC'96 record*, pages 299-304, 1995.

- [38] L. C. Freitas, N. P. Filho, and V. J. Farias. A novel family of dc-dc pwm converters using self-ressonant principle. In *IEEE-PESC'94 record*, pages 1385–1391, 1994.
- [39] C.W. Lander. *Eletrônica Industrial-Teoria e Aplicação*. MAKRON Books, 1996.