

SISTEMA EXPERIMENTAL FOTOVOLTAICO DE GERAÇÃO DE ENERGIA
ELÉTRICA OPERANDO EM PARALELO COM A REDE ELÉTRICA CA

Dissertação de Mestrado



Universidade Federal de Juiz de Fora
Faculdade de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Autor: *Luis Claudio Gambôa Lopes*

Orientador: *Prof. Pedro Gomes Barbosa*

JUIZ DE FORA, MG - BRASIL
AGOSTO 2006

SISTEMA EXPERIMENTAL FOTOVOLTAICO DE GERAÇÃO DE ENERGIA
ELÉTRICA OPERANDO EM PARALELO COM A REDE ELÉTRICA CA

LUIS CLAUDIO GAMBÔA LOPES

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA FACULDADE DE ENGENHARIA DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. Pedro Gomes Barbosa, D.Sc.
(*Orientador*)

Prof. Márcio de Pinho Vinagre, Dr.Eng.

Prof. André Luís Marques Marcato, D.Sc.

Prof. Luis Guilherme Barbosa Rolim, Dr.-Ing.

JUIZ DE FORA, MG - BRASIL
AGOSTO DE 2006

*"Eu sou a luz do mundo; quem me segue não andará em trevas,
mas terá a luz da vida."*

João-8,12

Agradecimentos

Gostaria de agradecer, primeiramente, a Deus, por tudo.

À minha família, pelo apoio irrestrito e pela confiança que sempre depositaram em mim, fundamentais para minha formação.

Ao meu orientador e amigo, Prof. Pedro Gomes Barbosa, pela colaboração para realização desse trabalho e pelo sempre incentivo ao estudo e a pesquisa .

Aos professores da área de Instrumentação & Controle do Curso de Mestrado em Engenharia Elétrica da Universidade Federal de Juiz de Fora que de forma direta e indireta contribuíram para fortalecer minha formação científica.

Aos amigos do Laboratório de Sistemas Eletrônicos e do Curso de Mestrado em Engenharia Elétrica da UFJF com os quais eu convivi durante todo esse período de pesquisa.

À Universidade Federal de Juiz de Fora, à Faculdade de Engenharia e ao CNPq pelo suporte financeiro e pelas ferramentas necessárias ao desenvolvimento deste trabalho.

Resumo da Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFJF como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia Elétrica (M.S.)

SISTEMA EXPERIMENTAL FOTOVOLTAICO DE GERAÇÃO DE ENERGIA ELÉTRICA OPERANDO EM PARALELO COM A REDE ELÉTRICA CA

Luis Claudio Gambôa Lopes
Agosto / 2006

Orientador: Prof. Pedro Gomes Barbosa
Área de concentração: Instrumentação e Controle

Esta dissertação descreve as etapas de desenvolvimento de um sistema experimental de geração de energia elétrica de 30 kW baseado em painéis solares fotovoltaicos montado no Campus da Universidade Federal de Juiz de Fora (UFJF). São apresentados detalhes do sistema fotovoltaico como: tipo e técnicas de chaveamento dos conversores estáticos CC-CC e CC-CA, modelos digitais das partes do sistema, algoritmos de controle, circuitos de condicionamento, entre outros. A energia CC proveniente dos painéis fotovoltaicos é processada por conversores CC-CC *boost*. Esses conversores são controlados para rastrear o ponto de máxima potência (MPPT - “*Maximum Power Point Tracking*”) dos painéis fotovoltaicos. Dois inversores fonte de tensão (VSI - “*Voltage Source Inverter*”) conectados em série através de transformadores injetam a energia processada pelos conversores CC-CC na rede elétrica CA. Os conversores VSIs utilizam uma técnica de chaveamento multipulso com eliminação seletiva de harmônicos. Todos os algoritmos de controle dos conversores CC-CC e CC-CA são implementados em processadores de sinais digitais (DSP - “*Digital Signal Processor*”). Um sistema supervisor foi desenvolvido para monitorar as variáveis e ajustar a operação do sistema como compensador estático ou como sistema de geração disperso. Resultados de simulações digitais obtidos com o programa ATP/EMTP (*Alternative version of Electromagnetic Transient Program*) e com um programa emulador de DSPs são usados para validar as estratégias de controle e de chaveamento propostas. Finalmente resultados experimentais demonstram o funcionamento do sistema e validam o trabalho.

Abstract of the Dissertation presented to the Program of Electrical Engineering of UFJF as a partial fulfillment of the requirements for Master of Electrical Engineer Degree (M.S.)

EXPERIMENTAL PHOTOVOLTAIC SYSTEM OF ELECTRIC ENERGY
GENERATION SHUNT CONNECTED TO THE AC NETWORK

Luis Claudio Gambôa Lopes
August / 2006

Advisor: Prof. Pedro Gomes Barbosa
Concentration area: Instrumentation and Control

This dissertation describes the development steps of an experimental photovoltaic system of electric energy generation of 30 kW installed on the Federal University of Juiz de Fora Campus (UFJF). Details of the photovoltaic system are presented such as: CC-CC and CC-CA converters topologies and switching strategy, digital models, control algorithms, conditioning circuits, among others. The DC energy from the photovoltaic panels is processed by DC-DC type *boost* converters. These converters are controlled to track the maximum power point (MPPT) of the photovoltaic panels. Two voltage source inverters (VSI), series connected through two transformers, inject the energy processed by the *boost* converters into the AC network. The VSI's output voltages use multipulse technique with a selective harmonic elimination. All control algorithms of the DC-DC and DC-AC converters are designed and implemented in digital signal processors (DSP). A supervisory system is designed and developed to monitor the system variables and to control the operation of the system as static synchronous compensator or as dispersed generation system. Simulation results, obtained with ATP/EMTP (*Alternative version of Electromagnetic Transient Program*) and with a DSP *emulator* program, are used to validate the proposed control and switching strategies. Finally experimental results demonstrate the operation of the photovoltaic generation system and are used to validate this work.

Sumário

1	Introdução	1
1.1	Identificação do Problema	2
1.2	Motivação	4
1.3	Objetivos	5
1.4	Trabalhos Publicados	5
1.5	Sumário	6
2	Modelagem e Simulação Digital das Partes do Sistema de Geração Fotovoltaico	8
2.1	Introdução	8
2.2	Considerações Iniciais	9
2.2.1	Modelo Digital do Painel Solar Fotovoltaico	10
2.2.2	Simulação do Modelo do Painel	11
2.3	Conversores Estáticos CC-CC	13
2.3.1	Rastreamento do Ponto de Máxima Potência (MPPT)	14
2.3.2	Simulações do Conversor CC-CC com MPPT	17
2.4	Conversores Estáticos CC-CA	21

2.4.1	Técnicas de Chaveamento dos Conversores Estáticos	22
2.5	Controlador dos Conversores CC-CA	24
2.5.1	Simulação dos Conversores Estáticos	26
2.5.2	Simulação dos Conversores Estáticos com Modulação PWM	32
2.6	Conclusões Parciais	38
3	Circuito de Sincronismo (PLL)	39
3.1	Introdução	39
3.2	Modelo do PLL Baseado nas Potências Instantâneas	39
3.2.1	Análise do Funcionamento do PLL	41
3.3	Modelo Linearizado do q -PLL	42
3.3.1	Comportamento Dinâmico do q -PLL	44
3.3.2	Comportamento Estático	45
3.4	Modelo Discreto do q -PLL	46
3.4.1	Comportamento Dinâmico	47
3.4.2	Comportamento Estático	47
3.4.3	Comparação dos Modelos do q -PLL Digital e Contínuo	48
3.4.4	Problemas com a Implementação do q -PLL	50
3.5	Simulação do q -PLL	51
3.6	Conclusões Parciais	51
4	Implementação dos Controladores do Sistema de Geração Fotovoltaico	53
4.1	Introdução	53

4.2	Utilização do DSP	53
4.3	<i>Hardware</i> do <i>TMS320F243</i>	54
4.4	Rede <i>CAN</i>	55
4.5	<i>Emulador</i> do DSP <i>TMS320F243</i> da Texas Instruments	57
4.6	Simulações das Partes do Sistema de Geração Fotovoltaicos com o <i>Emulador</i>	59
4.6.1	Painel Solar Fotovoltaico	59
4.6.2	Conversores Estáticos CC-CC e MPPT	61
4.6.3	Circuito de Sincronismo (PLL)	65
4.6.4	Conversores Estáticos CC-CA	65
4.7	<i>Hardware</i>	72
4.7.1	Implementação do Controlador do Conversor CC-CC	72
4.7.2	Implementação do Controlador do Conversor CC-CA	73
4.8	Conclusões Parciais	82
5	Programa Supervisório do Sistema de Geração Fotovoltaico	83
5.1	Introdução	83
5.2	Gerenciamento de Falhas dos Conversores CC-CC	83
5.3	Sistema Supervisório dos Conversores CC-CC	84
5.4	Gerenciamento de Falhas dos Conversores CC-CA	85
5.5	Sistema Supervisório dos Conversores CC-CA	87
5.5.1	Mapa de Teclas e Funções	90
5.5.2	Arquivo de Configuração	90
5.6	Conclusões Parciais	91

6	Resultados Experimentais	92
6.1	Introdução	92
6.2	Conversores CC-CC	95
6.3	Conversores CC-CA	95
6.3.1	Operação Injetando Potência Ativa na Rede CA	98
6.3.2	Operação como Compensador Estático Síncrono (STATCOM)	100
6.3.3	Operação Injetando Potência Ativa e Compensando Potência Reativa nos Terminais do Sistema de Geração	105
6.3.4	Simulação com Valores Reais	105
6.4	Conversores CC-CA com Modulação PWM	107
6.5	Rotinas dos DSPs	113
6.6	Conclusões Parciais	116
7	Conclusões e Trabalhos Futuros	117
7.1	Conclusões Finais	117
7.2	Trabalhos Futuros	118
A	Princípio de Funcionamento do Sistema de Geração Fotovoltaico	122
B	Listagem dos <i>Models</i> Usados no ATP/EMTP para Simulação	124
B.1	Modelos para Acionamento do Conversor CC-CC	124
B.1.1	diodo_pv.mod	124
B.1.2	mppt.mod	124
B.2	Modelos para Acionamento do Conversor CC-CA	125

B.2.1	pll.mod	125
B.2.2	disparo.mod	126
B.2.3	controle.mod	127
C	Listagem do Algoritmo de Controle para o TMS320F243	129
C.1	Arquivos de Cabeçalho	129
C.1.1	f243.h	129
C.1.2	can.h	130
C.2	Mppt	130
C.2.1	_mppt.asm	130
C.2.2	mppt.c	133
C.3	Comando	135
C.3.1	_comando.asm	135
C.3.2	comando.c	139
C.4	Disparo	142
C.4.1	_disparo.asm	142

Lista de Figuras

1.1	Diagrama de Blocos Funcional de um SGD Baseado em Células Fotovoltaicas.	3
2.1	Diagrama de Blocos Simplificado do Acoplamento dos Conversores.	9
2.2	Modelo do Painel Fotovoltaico.	10
2.3	Esquema de Simulação do Painel no ATP.	12
2.4	Corrente x Tensão nos Terminais do Painel Fotovoltaico.	12
2.5	Potência x Tensão nos Terminais do Painel Fotovoltaico.	13
2.6	Topologia Básica do Conversor Boost.	13
2.7	Curvas Típicas de Painéis Fotovoltaicos.	15
2.8	Método MPPT.	16
2.9	Algoritmo de Rastreamento de Máxima Potência.	17
2.10	Esquema de Simulação no ATP	18
2.11	Tensão de Entrada e Saída do Conversor <i>Boost</i>	18
2.12	Corrente de Entrada e Saída do Conversor <i>Boost</i>	19
2.13	Potência de Entrada do Conversor <i>Boost</i>	19
2.14	Potência de Saída do Conversor <i>boost</i>	20
2.15	Potência x Tensão do Conversor <i>Boost</i>	20

2.16	Detalhes do Rastreamento: (a) Ponto Superior; (b) Ponto Inferior.	21
2.17	Topologia Básica de um VSI Trifásico.	22
2.18	Esquema de Conexão dos VSIs Trifásicos.	23
2.19	Técnica de Chaveamento do VSI com Eliminação Seletiva Harmônica.	24
2.20	Diagrama de Blocos do Controle do Conversor CC-CA.	25
2.21	Esquema de Simulação da Planta no ATP.	27
2.22	Tensão de Linha na Saída de um Conversor CC-CA sem Carga.	28
2.23	Tensão de Linha na Saída do Acoplamentos dos Dois Conversores CC-CA sem Carga.	29
2.24	Potência Imaginária Instantânea Injetada no Sistema CA.	29
2.25	Potência Ativa Instantânea Proveniente do Painel Fotovoltaico e Potência Ativa Instantânea Injetada no Sistema CA.	30
2.26	Potências Ativa e Imaginária Instantâneas nos Terminais dos Conversores CC-CA.	30
2.27	Tensão na Fase “a” na Saída dos Conversores CC-CA.	31
2.28	Corrente na Fase “a”.	31
2.29	Tensão no Elo CC.	32
2.30	Técnica de Chaveamento PWM do VSI	33
2.31	Tensão de Linha na de um Saída do Conversor CC-CA sem Carga.	34
2.32	Tensão de Linha na Saída do Acoplamentos dos Dois Conversores CC-CA sem Carga.	35
2.33	Potência Imaginária Instantânea Injetada na Rede CA.	35
2.34	Potência Ativa Proveniente do Painel Fotovoltaico e Potência Ativa Instan- tânea Injetada na Rede CA.	36
2.35	Potências Ativa e Imaginária Instantâneas nos Terminais dos Conversores.	36

2.36	Tensão na Fase “a” na Saída do Conversor CC-CA.	37
2.37	Corrente na Fase “a”.	37
2.38	Tensão no Elo CC.	38
3.1	Diagramas dos Circuitos de Detecção de Frequência: (a) p -PLL e (b) q -PLL.	40
3.2	Vetor Espacial de Tensão e Corrente.	41
3.3	Diagrama de Blocos do q -PLL Linearizado.	43
3.4	Lugar das Raízes do Modelo do q -PLL Linearizado.	45
3.5	Lugar das Raízes do Modelo Discreto	47
3.6	Resposta ao Degrau dos Modelos Linear e Digital do q -PLL.	49
3.7	Detalhe da Resposta ao Degrau dos Modelos Linear e Digital do q -PLL.	49
3.8	Tensão de Fase e Ângulo de Fase Detectado pelo q -PLL Digital.	50
3.9	Esquema de Simulação no ATP	51
3.10	Simulação do q -PLL no ATP	51
4.1	Plataforma DSP Desenvolvida para Implementação dos Controles do Sistema.	54
4.2	Diagrama de Ligação da Rede CAN.	56
4.3	Interface de Depuração do <i>Emulador</i>	58
4.4	Esquema de Simulação do Painel Fotovoltaico.	60
4.5	Característica Corrente x Tensão nos Terminais do Painel Solar Fotovoltaico.	60
4.6	Característica Potência x Tensão nos Terminais do Painel Solar Fotovoltaico.	61
4.7	Esquema de Simulação do Conversor CC-CC com MPPT Usando o <i>Emulador</i>	61
4.8	Tensões de Entrada e de Saída no Conversor: (a) Simulador; (b) <i>Emulador</i>	62
4.9	Potência de Entrada Rastreada Pelo Conversor: (a) Simulador; (b) <i>Emulador</i>	63

4.10	Potência de Saída Fornecida Pelo Conversor: (a) Simulador; (b) <i>Emulador</i> .	64
4.11	Esquema de Simulação do PLL.	65
4.12	Tensão da Fase 'a' e Ângulo Detectado.	65
4.13	Esquema de Simulação do Conversor CC-CA Usando o <i>Emulador</i>	67
4.14	Potências nos Terminais dos Conversores CC-CA: (a) Simulador; (b) <i>Emulador</i>	68
4.15	Tensão v_a : (a) Simulador; (b) <i>Emulador</i>	69
4.16	Corrente i_a : (a) Simulador; (b) <i>Emulador</i>	70
4.17	Tensão do Elo CC: (a) Simulador; (b) <i>Emulador</i>	71
4.18	Circuito de Medição da Tensão do Elo CC.	72
4.19	Foto do Circuito de Medição de Tensão do Elo CC.	73
4.20	Circuito de Medição da Tensão e Corrente do Painel.	74
4.21	Foto do Circuito de Medição da Tensão e Corrente do Painel.	74
4.22	Circuito de condicionamento das Tensões Trifásicas do Sistema.	75
4.23	Detalhe dos Circuitos de Condicionamento das Tensões Trifásicas do Sistema.	76
4.24	Circuito de Proteção do Conversor A/D do DSP de Disparo.	76
4.25	Foto do Circuito de Proteção do Conversor A/D do DSP de Disparo.	77
4.26	Circuito de Disparo dos Conversores CC-CA.	78
4.27	Foto do Circuito de Disparo dos Conversores CC-CA.	78
4.28	Circuito de Comunicação da Rede CAN.	79
4.29	Foto do Circuito de Comunicação da Rede CAN (componentes).	79
4.30	Circuito de Condicionamento das Correntes Trifásicas do Sistema Elétrico.	80

4.31	Foto do Circuito de Condicionamento das Correntes Trifásicas do Sistema Elétrico.	80
4.32	Circuito de Proteção do Conversor A/D do DSP de Controle.	81
4.33	Foto do Circuito de Proteção do Conversor A/D do DSP de Controle.	81
4.34	Diagrama de Ligação da Rede CAN.	82
5.1	Relatório de Saída do Conversor <i>Boost</i>	84
5.2	Diagrama de Controle	86
5.3	Tela da Supervisão das Potências	88
5.4	Tela da Supervisão das Frequências.	88
5.5	Tela de Supervisão das Tensões e Corrente	89
5.6	Tela do Supervisório com Janela de Erros Ativa.	89
6.1	Foto com os Arranjos dos Painéis Solares Fotovoltaicos.	93
6.2	Foto de um Conversor CC-CC <i>Boost</i> e Circuitos Condicionadores.	93
6.3	Foto do Conversor VSI No. 1 e Transformadores de Conexão.	94
6.4	Foto do Conversor VSI No. 2 e Transformadores de Conexão.	94
6.5	Foto dos DSPs de Controle e de Disparo dos Conversores VSIs.	95
6.6	Tensão de Saída (superior) e Tensão de Entrada (inferior).	96
6.7	Tensão na Chave (superior) e Corrente no Indutor (inferior) [escala: $1mV/A$].	96
6.8	Tensão de 3 Níveis na Saída de um Conversor em Aberto.	97
6.9	Tensão de 5 Níveis na Saída dos Conversores em Aberto.	97
6.10	Tensão de Fase da Rede CA.	98
6.11	Espectro de Frequência da Tensão de Saída.	98

6.12	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] do Conversor Flutuando.	99
6.13	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] Injetando Potência Ativa sem MPPT.	99
6.14	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] Injetando Potência Ativa com MPPT.	100
6.15	Tela de Potências Ativa e Reativa Instantâneas Obtidas do Programa Supervisório Quando o Sistema Fotovoltaico Injeta Potência Ativa na Rede. . .	101
6.16	Tela dos Valores Coletivos das Tensões e Correntes e Valor Médio da Tensão CC Obtida do Programa Supervisório para o Sistema Fotovoltaico Injetando Potência Ativa na Rede.	101
6.17	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] com o Sistema Sintetizando 20 kVAR Indutivo.	102
6.18	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] com o Sistema Sintetizando 20 kVAR Capacitivo.	102
6.19	Tela de Potências do Supervisório Sintetizando Potência Reativa.	103
6.20	Tela de Tensões do Supervisório Sintetizando Potência Reativa.	104
6.21	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] com o Sistema Sintetizando 35 kVAR Indutivo.	104
6.22	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] com o Sistema Sintetizando 35 kVAR Capacitivo.	105
6.23	Tela de Potências do Supervisório Injetando Potência Ativa e Reativa.	106
6.24	Tensão na Fase “a”.	107
6.25	Corrente na Fase “a”.	107
6.26	Tensão PWM de 3 Níveis na Saída de um Conversor em Aberto.	108
6.27	Tensão PWM de 5 Níveis na Saída dos Conversores em Aberto.	109
6.28	Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1m V/A] do Conversor Flutuando.	109

6.29	Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1m V/A] Injetando Potência Ativa com MPPT.	110
6.30	Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1m V/A] com o Sistema Sintetizando 20 kVAR Indutivo.	111
6.31	Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1m V/A] com o Sistema Sintetizando 20 kVAR Capacitivo.	111
6.32	Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1m V/A] com o Sistema Sintetizando 35 kVAR Indutivo.	112
6.33	Tensão (superior) e Corrente (inferior) [escala: 1m V/A] [escala: 1m V/A] com o Sistema Sintetizando 35 kVAR Capacitivo.	112
6.34	Tempo de Execução da Rotina “MPPT”.	113
6.35	Tempo de Execução da Rotina “controle”.	114
6.36	Tempo de Execução da Rotina “disparo”.	115
6.37	Tempo de Execução da Rotina “disparo pwm”.	115
A.1	(a) Diagrama Unifilar do Sistema Fotovoltaico; (b) Diagrama Fasorial.	123

Lista de Tabelas

2.1	Valores do Modelo Digital do Painel Fotovoltaico.	11
2.2	Valores do Esquema de Simulação da Planta no ATP	26
3.1	Erro Estático do Modelo q -PLL Linearizado	46
3.2	Erro Estático do Modelo q -PLL Discreto	48
4.1	Características Básicas do <i>DSP TMS320F243</i> da <i>Texas Instruments</i>	55
4.2	Módulos Usados do DSP TMS320F243	55
4.3	Características do <i>Emulador</i>	57
4.4	Comandos do <i>Emulador</i>	58
5.1	Configuração da Porta Serial	84
5.2	Controle Gerais do Supervisório	90
5.3	Controle da Potência Reativa de Referência	90
5.4	Controle dos Conversores	90
5.5	vmode - Modo de resolução de Vídeo	91
5.6	vfull - Modo de Apresentação do Programa	91
5.7	gridx: Intervalos entre os Eixos dos Gráficos	91

6.1	Valores do Esquema de Simulação da Planta no ATP	106
-----	--	-----

Capítulo 1

Introdução

A sempre crescente demanda de energia elétrica nos países industrializados associada à escassez dos recursos naturais tem encorajado os cientistas e engenheiros em todo o mundo a buscarem formas alternativas, tais como a eólica, a solar, as marés, as células a combustíveis, etc., para a geração de energia elétrica.

Dentro deste cenário, painéis solares fotovoltaicos têm sido propostos e utilizados como Sistemas de Geração Dispersa e Distribuída (SGD) de energia elétrica. Esses sistemas de geração têm sido pesquisados e instalados com potências que vão desde algumas dezenas até centenas de quilo *Watts* (kW) (Dugan, Key e Ball, 2006).

Com a queda atual dos preços dos painéis fotovoltaicos por *Watt-pico* (Wp) gerado, baixo impacto ambiental das instalações (Sayigh, 2003), confiabilidade e facilidades de instalação, a energia solar fotovoltaica vem se tornando uma boa alternativa para complementar a geração do sistema de energia elétrica. Os equipamentos necessários para o condicionamento da energia gerada por sistemas fotovoltaicos usados atualmente são mais confiáveis, compactos e mais baratos, sendo mais um atrativo para o uso desses sistemas.

Quanto a capacidade de geração, foi sugerida por (Castañer e Silvestre, 2002) a classificação dos sistemas fotovoltaicos conectados à rede elétrica como:

- Pequeno porte - capacidade instalada de 1 a 10 kWp . Geralmente, encontrados em aplicações residenciais: telhados de casas, escolas ou outras pequenas construções;
- Médio Porte - capacidade de 10 kWp a algumas centenas de kWp . Aplicações típicas são: sistemas fotovoltaicos integrados em prédios e indústrias;

- Grande Porte - Potência de 500 kWp a MWp , em sistemas centralizados. Usualmente, esses sistemas são operados por empresas do setor elétrico.

Os sistemas de geração fotovoltaicos podem ser: *isolados* quando são a única fonte de energia e geralmente utilizam armazenamento de energia em acumuladores, *híbridos* quando são utilizados juntos a outras fontes de energia e *interligados ao sistema elétrico*. Os sistemas fotovoltaicos interligados ao sistema elétrico não necessitam de armazenamento de energia, sendo neste caso toda geração de energia injetada na rede elétrica.

O condicionamento da energia fotovoltaica de sistemas interligados a rede elétrica podem ser compostos de variadas topologias e configurações de conversores estáticos, operando com chaveamento de alta ou baixa frequência. As topologias podem ser divididas em quatro grupos:

1. Único estágio inversor não isolado
2. Único estágio inversor isolado
3. Múltiplos estágios não isolados de conversão
4. Múltiplos estágios isolados de conversão

Em (Rodrigues, 2004) é apresentada uma detalhada revisão bibliográfica mostrando diversas configurações e topologias possíveis de conversores estáticos para serem usadas na conexão de sistemas fotovoltaicos à rede CA.

Este trabalho descreve as etapas de implementação de um sistema experimental de controle de de geração de energia elétrica de um sistema baseado em painéis solares fotovoltaicos de porte médio (30 kWp) interligado ao sistema elétrico montado no Campus da Universidade Federal de Juiz de Fora.

1.1 Identificação do Problema

Devido à natureza unidirecional (CC) da energia elétrica gerada nos painéis solares fotovoltaicos é necessário o uso de conversores estáticos para condicionar a energia elétrica antes desta poder ser usada para alimentar cargas residenciais, comerciais e industriais em corrente alternada.

Uma configuração possível para o sistema de geração de energia elétrica fotovoltaico consiste em um número considerável de painéis solares fotovoltaicos conectados formando grupos (“*arrays*”) e conectados a vários conversores CC-CC cujos terminais de saída são ligados a um barramento CC comum. Então, a energia elétrica que chega ao barramento CC do arranjo de conversores, proveniente dos painéis solares fotovoltaicos, é processada pelos conversores CC-CA e injetada no sistema de distribuição. Esses conversores CC-CA são responsáveis pela interface entre o sistema CC e o sistema de distribuição de energia elétrica (Barbosa, Rolim, Watanabe e Hanitsch, 1998). A Figura 1.1 mostra um diagrama de blocos da topologia utilizada neste trabalho.

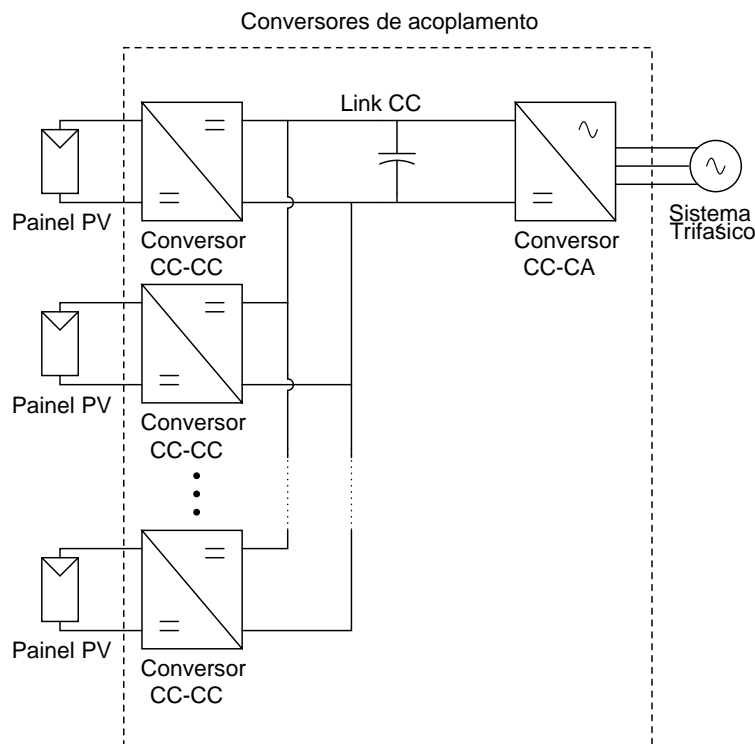


Figura 1.1: Diagrama de Blocos Funcional de um SGD Baseado em Células Fotovoltaicas.

Todos os conversores mostrados na Figura 1.1 são baseados em chaves semicondutoras autocomutadas como por exemplo: IGBTs (*Insulated Gate Bipolar Transistors*), MOSFETs (*Metal Oxide Semiconductor - Field Effect Transistor*), GTOs (*Gate Turn-off Thyristors*). Esses interruptores de potência permitem o controle tanto do instante de início de condução quanto do instante de corte, através de sinais elétricos aplicados em seus terminais de “*gate*”.

Os conversores CC-CA autocomutados podem ser classificados como inversores fonte de tensão (VSIs - “*Voltage Source Inverters*”) ou inversores fonte de corrente (CSIs - “*Current Source Inverters*”) (Mohan, Undeland e Robbins, 1995). Essa divisão é con-

seqüência do tipo da fonte unidirecional conectada nos terminais CC do conversor. Contudo, independentemente do tipo da fonte CC, os conversores CC-CA podem operar como fontes de tensão ou corrente independentes, controlando os fluxos de potência ativa e reativa em seus terminais, respectivamente.

Contudo, o controle dos conversores CC-CC e CC-CA mostrados na Figura 1.1 requer um sistema robusto e flexível. A utilização de Processadores Digitais de Sinais (DSP - “*Digital Signal Processors*”) em aplicações onde existe necessidade de um tempo de resposta pequeno, agrega facilidades no projeto e flexibilidade para a implementação dos algoritmos de controle dos conversores.

1.2 Motivação

Os avanços tecnológicos das últimas décadas foram responsáveis por grandes transformações nos meios de produção utilizados nas sociedades industrializadas. A cada ano são desenvolvidos mais e mais equipamentos que utilizam energia elétrica para realizar as tarefas para as quais são projetados.

Nesse sentido, o aproveitamento da energia elétrica gerada em painéis solares fotovoltaicos é uma forma interessante de atender rapidamente a sempre crescente demanda deste tipo de energia. A injeção da energia elétrica convertida diretamente no sistema elétrico tem um grande apelo econômico visto que não é necessário o uso de acumuladores (baterias), que são equipamentos de custo elevado e requerem manutenção periódica, isso torna o sistema mais barato e flexível.

Conversores estáticos de energia baseados em chaves semicondutoras de potência, controlados por microcontroladores (μC) ou microprocessadores (μP), são encontrados como estágio de entrada de uma grande variedade de equipamentos eletro-eletrônicos em aplicações residenciais, comerciais e industriais.

Processadores digitais proporcionam versatilidade e simplicidade em controle de equipamentos elétricos além de simplificar a tarefa de implementação dos sistemas de controle. Em aplicações de controle onde se deseja operação em “*tempo real*” a utilização de processadores digitais de sinais tem sido cada vez mais difundida, substituindo os microcontroladores.

Essa utilização é em grande parte devido à redução dos custos dos processadores

DSPs observado nas últimas décadas. Aliada à redução dos custos, tem-se as facilidades e a flexibilidade alcançadas com o projeto e implementação de sistemas de controle totalmente digitais.

1.3 Objetivos

No desenvolvimento do sistema experimental de geração de energia elétrica baseado em painéis solares fotovoltaicos os seguintes objetivos foram seguidos:

- i.* Estudo dos tipos de conversores estáticos mais usuais para essa função fabricados no Brasil e no exterior;
- ii.* Estudo das técnicas de controle para conversão da energia elétrica CC em CA;
- iii.* Proposta de uma arquitetura para o sistema de geração de energia elétrica baseado em painéis solares fotovoltaicos;
- iv.* Desenvolvimento de um algoritmo de controle inteligente baseado em DSPs para controlar a tensão e a corrente de saída dos conversores CC-CC (MPPT) e CC-CA do sistema de geração;
- v.* Desenvolvimento de um circuito de sincronismo para permitir a conexão em paralelo do sistema de geração, formado por painéis solares, com a rede de distribuição;
- vi.* Desenvolvimento de um emulador do DSP para simulação dos algoritmos e teste do sistema.
- vii.* Implementação em DSP do circuito de sincronismo proposto;
- viii.* Desenvolvimento e montagem de circuitos de condicionamento de sinais para o controle dos conversores estáticos;
- ix.* Desenvolvimento de um sistema de controle de falhas;
- x.* Desenvolvimento de um sistema de supervisão de operação;

1.4 Trabalhos Publicados

Como resultado dos estudos realizados durante esse período os seguintes trabalhos foram publicados:

- [1] CARLETTI, Ricardo Lima; LOPES, Luis Claudio Gambôa; BARBOSA, Pedro Gomes. "Active and reactive power control scheme for a grid-connected photovoltaic generation system based on VSI with selective harmonic elimination". Proceedings of 8th Brazilian Power Electronics Conference, Recife, Brasil, 2005, pp. 129–134.
- [2] LOPES, Luis Claudio Gambôa; BARBOSA, Pedro Gomes. "Modelagem e implementação de um circuito de sincronismo digital, usando o DSPTMS320F243 da Texas Instruments". Revista Princípiã (ISBN: 15182983), Editora da UFJF, Juiz de Fora - MG, V. 9, 2004, pp. 71-83.
- [3] CARLETTI, Ricardo Lima; LOPES, Luis Claudio Gambôa; BARBOSA, Pedro Gomes. "A dispersed generation system based on photovoltaic cells: converter configuration and switching strategy". Proceedings of 7th Brazilian Power Electronics Conference, Fortaleza, Brasil, 2003, pp. 404-409.
- [4] LOPES, Luis Claudio Gambôa; CARLETTI, Ricardo Lima; BARBOSA, Pedro Gomes. "Implementation of a digital and a deadbeat PLL circuits based on instantaneous powers theory with a DSP TMS320F243". Proceedings of 7th Brazilian Power Electronics Conference, Fortaleza, Brasil, 2003, pp. 180-185.
- [5] LOPES, Luis Claudio Gambôa; BARBOSA, Pedro Gomes. "Implementação de um circuito PLL digital e um deadbeat baseados na teoria das potências instantâneas com o DSP TMS320F243". Anais do I Seminário em Sistemas de Potência, Automação e Controle (Sempac), UFJF, Juiz de Fora, Brasil, 2003.

1.5 Sumário

No Capítulo 2 são descritas as partes do sistema de geração fotovoltaico, descrevendo os equipamentos e conversores utilizados, suas funções no sistema de geração e as técnicas utilizadas para o controle dos conversores. Resultados de simulações digitais no programa ATP/EMTP (*“Alternative version of Electromagnetic Transients Program”*) são usados para demonstrar a validade das técnicas de controle aplicadas ao sistema de geração fotovoltaico.

No Capítulo 3 são apresentadas as etapas de projeto e implementação de um circuito de sincronismo (PLL - *“Phase Locked Loop”*) necessário para sincronizar as tensões sintetizadas pelos conversores CC-CA com as tensões do sistema elétrico CA.

No Capítulo 4 é apresentada a implementação digital dos modelos e dos algorit-

mos de controle dos conversores CC-CC e CC-CA num programa emulador de DSP. O desenvolvimento e a utilização do programa emulador do DSP permite investigar o desempenho dos algoritmos de controle quando são utilizados processadores digitais cujas variáveis têm tamanho limitado, a aritmética é de ponto fixo (saturada), os tempos de conversão analógico digital é finito, ocorrem estouros nos registradores (“*overflow*”), etc. Serão ainda mostrados os circuitos de interface necessários para os DSP's.

No Capítulo 5 é apresentado um sistema de supervisão e operação do sistema de geração fotovoltaico sendo descritas suas características de operação e aquisição de dados. É implementado também um algoritmo de gerenciamento de falhas dos conversores CC-CC e CC-CA para bloquear a operação do sistema em condições de falhas.

No Capítulo 6 são apresentados alguns resultados experimentais do sistema de geração fotovoltaico operando interligado com a rede CA injetando potência ativa. São também mostrados resultados experimentais dos conversores CC-CA operando como compensador estático de energia reativa.

Finalmente no Capítulo 7 são apresentadas as conclusões deste trabalho e são propostos temas para futuros trabalhos.

Capítulo 2

Modelagem e Simulação Digital das Partes do Sistema de Geração Fotovoltaico

2.1 Introdução

Apesar de poderem ser usados tanto VSIs quanto CSIs para converter para CA a energia elétrica CC gerada nos painéis fotovoltaicos, observa-se uma tendência na utilização de conversores estáticos tipo fonte de tensão (VSI). Esta escolha é reforçada pelo fato de que, em aplicações de médias e altas potências, os conversores fonte de corrente (CSI) apresentam maiores perdas nas chaves semicondutoras.

Assim, como o objetivo principal desse trabalho é a implementação de um sistema de geração de energia elétrica de aproximadamente 30 kW, baseado em painéis solares fotovoltaicos, optou-se pela utilização de VSIs. A Figura 2.1 mostra o diagrama de blocos simplificado da topologia adotada onde pode-se observar os painéis solares, o conversor CC-CC, o capacitor do elo CC e o conversor CC-CA.

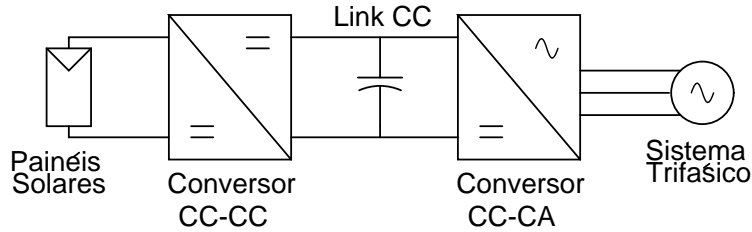


Figura 2.1: Diagrama de Blocos Simplificado do Acoplamento dos Conversores.

Na figura anterior, o elo CC isola os conversores CC-CC e CC-CA permitindo que os controles dos dois conversores sejam feitos de maneira desacoplada. O controlador do conversor CC-CA mantém a tensão do elo CC regulada em função da potência injetada no sistema trifásico. Já o controle do conversor CC-CC garante que a potência convertida nos painéis seja injetada no elo CC do sistema fotovoltaico. O conversor CC-CC tem a capacidade adicional de rastrear o ponto de máxima potência dos painéis solares fotovoltaicos (MPPT - “*Maximum Power Point Tracking*”).

Neste capítulo, o programa ATP/EMTP (*Alternative Transients Program/ Electromagnetic Transients Program*) (ATP, 2005) será adotado como ferramenta de simulação para testar a eficiência dos algoritmos desenvolvidos para o controle dos conversores CC-CC e CC-CA apresentados neste capítulo.

2.2 Considerações Iniciais

O elo CC do sistema de geração disperso mostrado na Figura 2.1 tem uma característica unidirecional em tensão em consequência da escolha do conversor VSI para conectar o sistema de geração fotovoltaico ao sistema elétrico CA. Assim optou-se que o conversor CC-CC deveria ter uma característica de fonte de corrente como estágio de entrada.

Devido à potência instalada definiu-se que o conversor CC-CA deveria ser isolado galvanicamente da rede CA através de transformadores. A necessidade de evitar o fluxo reverso de energia para os painéis fotovoltaicos levou à escolha do conversor *boost* (Mohan, Undeland e Robbins, 1995). Os conversores *boost* CC-CC são fundamentais para adequar os níveis de tensão e de corrente gerados nos painéis fotovoltaicos ao nível da tensão do elo CC da instalação. Eles têm a vantagem adicional de possuir um diodo conectado entre os terminais CC de entrada e de saída, não permitindo que os painéis fotovoltaicos sejam

alimentados com uma corrente reversa oriunda dos conversores CC-CA.

2.2.1 Modelo Digital do Painel Solar Fotovoltaico

Foi adotado nesta etapa um modelo simplificado para o painel solar fotovoltaico baseado em (Jiang, Huang, Hsio e Chen, 2005) e (Koutroulis, Kalaitzakis e Voulgaris, 2001). A Figura 2.2 mostra o circuito equivalente do painel solar fotovoltaico onde I_f é uma fonte de corrente que emula a insolação sobre o painel, sendo as características terminais do painel obtidas de (2.1).

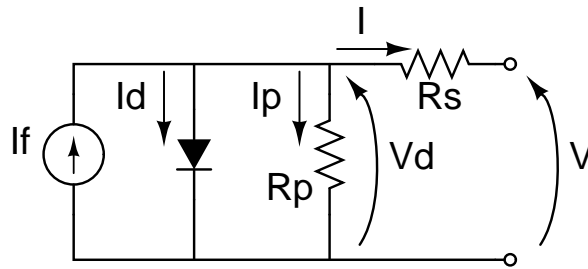


Figura 2.2: Modelo do Painel Fotovoltaico.

$$\begin{cases} I = I_f - I_d - I_p \\ V_d = V + I.R_s \end{cases} \quad (2.1)$$

onde V e I são a tensão e a corrente de saída do painel solar fotovoltaico em (V) e (A), respectivamente; R_p e R_s são resistências paralelo e série do painel, respectivamente.

A corrente do diodo I_d é definida por:

$$I_d = I_s \left(e^{\frac{V_d}{V_T}} - 1 \right) \quad (2.2)$$

onde I_s é a corrente de saturação reversa do painel em (A) sendo a tensão V_T dada como

se segue:

$$V_T = \frac{nkT_k}{q} \quad (2.3)$$

onde T_k é a temperatura ambiente em *Kelvin* (K), k é a constante de Boltzmann ($1,38 \times 10^{-23} J/K$), q é a carga do elétron ($1,6 \times 10^{-19} C$) e n um parâmetro empírico dependente das dimensões da junção.

2.2.2 Simulação do Modelo do Painel

Os resultados das simulações digitais que serão apresentados nesta seção foram obtidos considerando os valores fornecidos na Tabela 2.1 para o painel solar fotovoltaico. Essas escolhas tiveram como objetivo principal aproximar a curva característica do sistema digital com a do painel fotovoltaico real. Nessas simulações foi assumido que a temperatura do painel ficou constante. Essa hipótese permite usar um valor constante para a tensão V_T . A variação da energia gerada pelo painel é conseguida variando somente o valor da fonte I_f de zero a um valor máximo que simula o nível de radiação solar sobre o painel.

Tabela 2.1: Valores do Modelo Digital do Painel Fotovoltaico.

Elemento	Valor
R_p	5k Ω
R_s	0,01 Ω
V_t	3,6872V
I_s	42,56 μ A
$I_{f_{máx}}$	3,87A

Na Figura 2.3 é mostrado o esquema de simulação do painel usado no ATP para obter suas curvas características. Nesse esquema a fonte de 3,78 A representa a corrente I_f do modelo mostrado na Figura 2.2, a fonte IPV representa a corrente I_d pelo diodo e a fonte VP representa a tensão nos terminais do painel fotovoltaico. As equações da característica não linear do painel solar fotovoltaico foi implementada usando o ambiente *Models* a partir de (2.1) e (2.2).

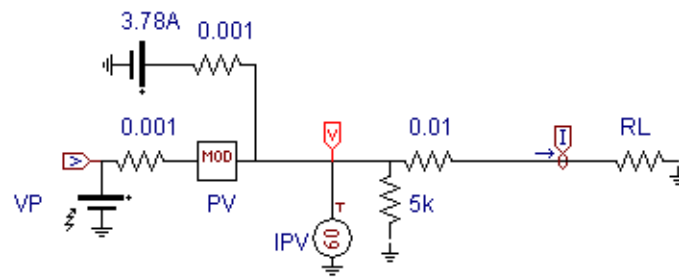


Figura 2.3: Esquema de Simulação do Pannel no ATP.

Nas Figuras 2.4 e 2.5 são mostradas as curvas características de corrente e potência em relação a tensão do modelo do pannel utilizado nas simulações do ATP.

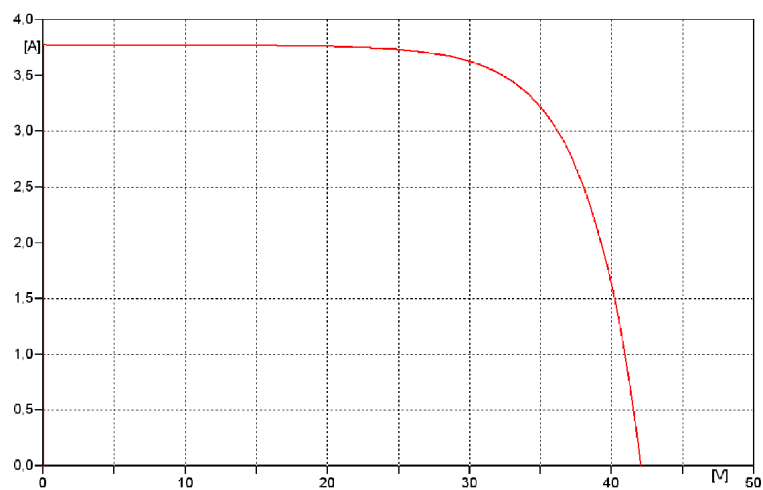


Figura 2.4: Corrente x Tensão nos Terminais do Pannel Fotovoltaico.

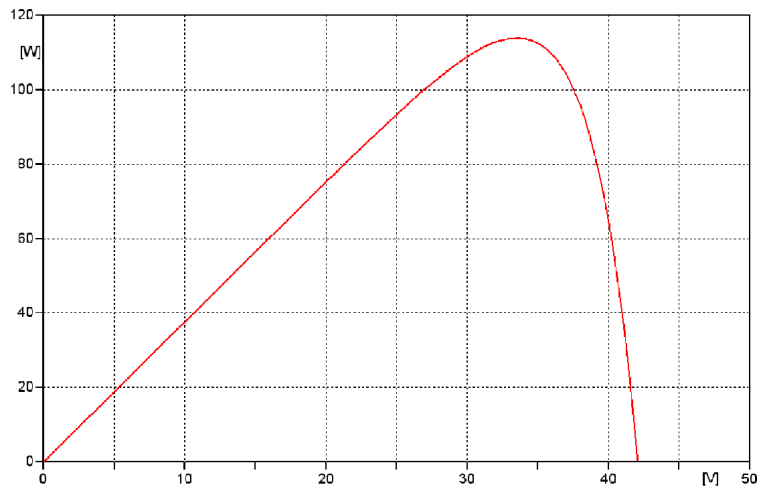


Figura 2.5: Potência x Tensão nos Terminais do Painel Fotovoltaico.

2.3 Conversores Estáticos CC-CC

Como explicado anteriormente, o conversor CC-CC utilizado para alimentar o elo CC da instalação é do tipo *boost*. Devido à indutância normalmente conectada nos terminais de entrada dos conversores *boost* os painéis fotovoltaicos se comportam como uma fonte de corrente. Desse modo o conversor tem como variável de entrada a corrente drenada dos painéis fotovoltaicos e como variável de saída a tensão do elo CC.

Devido ao diodo conectado em série no circuito de saída, o fluxo de potência no conversor *boost* é unidirecional, isto é, sentido fonte-carga. A Figura 2.6 mostra a topologia básica do conversor boost utilizando IGBT com diodo conectado em anti-paralelo. Os conversores boost têm como função elevar a tensão de entrada, no sistema proposto, a tensão de saída será sempre fixada pelo conversor CC-CA (elo CC), logo só a tensão de entrada do conversor poderá variar com a insolação do painel fotovoltaico.

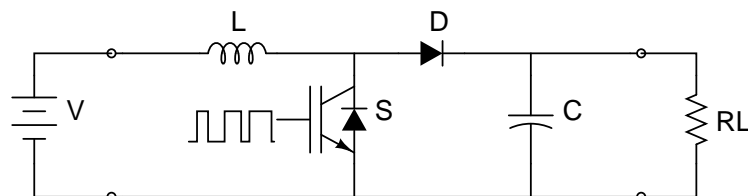


Figura 2.6: Topologia Básica do Conversor Boost.

Considerando que os conversores CC-CC funciona no modo de condução contínua (CCM - “*Continuous Conduction Mode*”), tem-se que a corrente no indutor L de entrada do conversor é sempre maior que zero e o comportamento do conversor é praticamente linear (Mohan, Undeland e Robbins, 1995). Devido às perdas associadas aos elementos conversor, a razão cíclica dos conversores (D - “*duty cycle*”) foi limitado em 45 %, isso se deve ao fato de que para altos valores de razão cíclica se tem uma baixa utilização das chaves e os efeitos parasitas das perdas são relevantes fazendo com que as relações ideais do modo de condução contínua não sejam mais válidas (Mohan, Undeland e Robbins, 1995).

Com objetivo de aumentar a eficiência do sistema foi implementado uma técnica de rastreamento de máxima potência (MPPT - “*Maximum Power Point Tracker*”) para controlar os conversores CC-CC. Desse modo o controlador varia a razão cíclica do conversor para controlar a corrente drenada e a tensão nos terminais do painel para seguir o ponto de máxima potência da curva característica.

2.3.1 Rastreamento do Ponto de Máxima Potência (MPPT)

O algoritmo MPPT adotado foi baseado no método da perturbação e observação (P&O) que pode ser facilmente implementado em um microcontrolador (Koutroulis, Kalitakis e Voulgaris, 2001) e (Santos e Antunes, 2003). O algoritmo é muito simples e consiste em medir a potência de entrada, aplicar uma perturbação e observar o comportamento do sistema. Apesar do método P&O ser de fácil implementação e apresentar bons resultados (Hohm e Ropp, 2003), ele pode apresentar problemas no rastreamento de máxima potência, no caso de um sombreamento parcial dos painéis acontecem distorções nas curvas características, aparecendo máximos locais de potência e o método pode convergir para um máximo local ao invés do máximo global.

A Figura 2.7 mostra as variações típicas observadas nas curvas da corrente e da potência de um painel fotovoltaico em função de sua tensão terminal. As variações observadas nessas figuras dependem do nível de insolação. Os pontos de máxima potência são os joelhos das curvas potência x tensão. O conversor CC-CC deve ser então capaz de operar nesses pontos para extrair a máxima potência dos painéis.

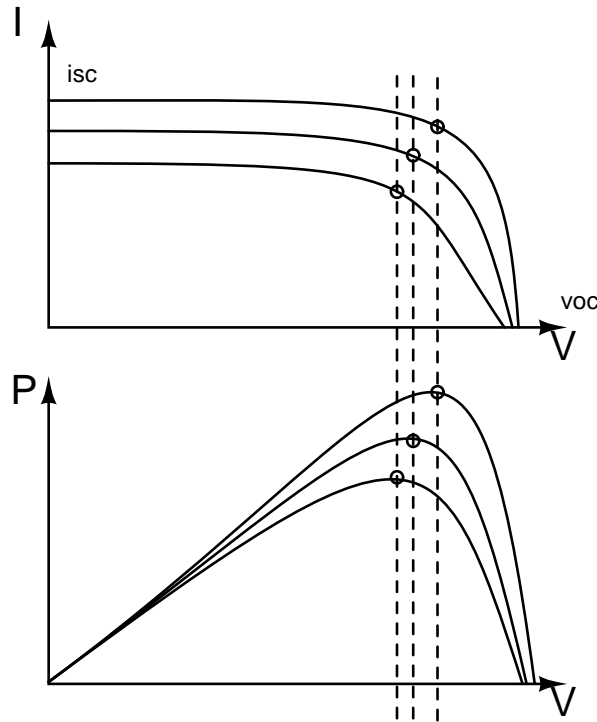


Figura 2.7: Curvas Típicas de Painéis Fotovoltaicos.

Assim o método da perturbação e observação consiste em se aplicar um aumento ou diminuição na razão cíclica (D) do conversor e verificar se a potência de entrada aumentou ou diminuiu. A Figura 2.8 mostra uma representação gráfica desse método. Partindo do ponto P_1 aplica-se uma perturbação Δp variando a razão cíclica do conversor (D). Observa-se então que a potência de entrada do conversor aumenta. Procede-se a aplicação de sucessivas perturbações no mesmo sentido enquanto a potência observada crescer. Quando o sistema passa pelo ponto de máxima potência, a potência convertida pelo painel começa diminuir (ponto P_2). Nesse instante muda-se a direção da perturbação Δp aplicada, reduzindo a razão cíclica do conversor CC-CC. Tem-se então conversor operando entre os dois pontos adjacentes ao ponto de máxima potência $P_{m\acute{a}x}$.

O método P&O será mais preciso quanto menor for a perturbação aplicada, Contudo neste caso a convergência será mais lenta, podendo ser compensada pelo aumento da frequência de amostragem, o que requer maior esforço computacional. Sua utilização para rastreamento de potência de painéis fotovoltaicos não requer uma resposta dinâmica muito rápida devido a característica de variações muito lentas de potência nesses sistemas, o algoritmo pode operar com uma frequência de amostragem baixa e com pequenas perturbações (resposta dinâmica lenta) sem problemas.

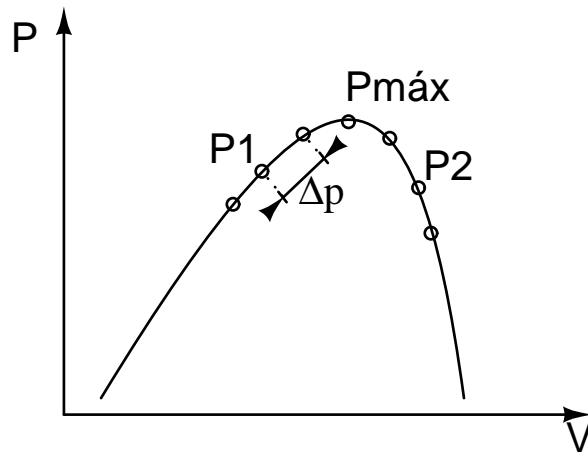


Figura 2.8: Método MPPT.

A Figura 2.9 mostra o fluxograma do algoritmo de rastreamento de máxima potência (MPPT) utilizado para o controle dos conversores CC-CC, onde α é o incremento da razão cíclica que ocasiona uma perturbação Δp , o valor de α é definido na inicialização e define a velocidade de convergência do algoritmo.

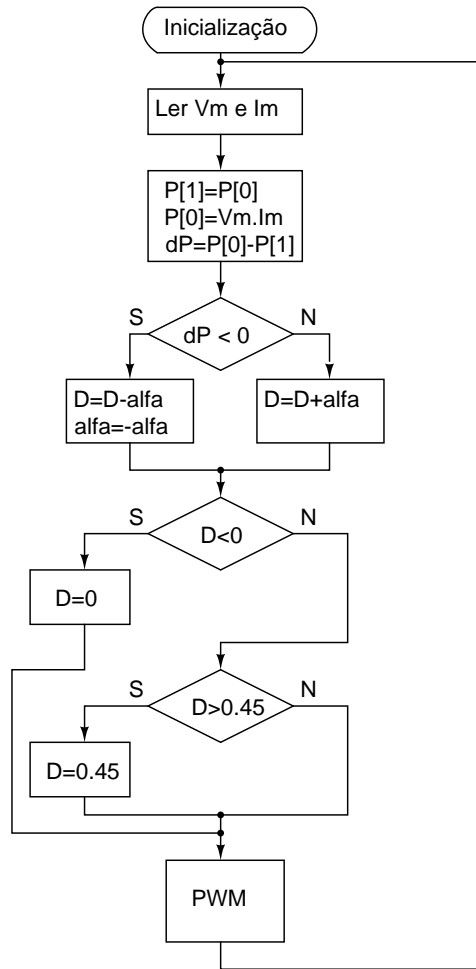


Figura 2.9: Algoritmo de Rastreamento de Máxima Potência.

2.3.2 Simulações do Conversor CC-CC com MPPT

As simulações foram feitas considerando o conversor CC-CC no modo de condução contínua. As relações das tensões e correntes médias de entrada e de saída do conversor podem ser escritas como:

$$\begin{cases} \frac{V_o}{V_i} = \frac{1}{1-D} \\ \frac{I_o}{I_i} = 1-D \end{cases} \quad (2.4)$$

Na Figura 2.10 pode ser visto o esquema utilizado para a simulação do conversor CC-CC no programa ATP/EMTP, na esquerda pode ser observado o *model PV* que

simula o comportamento de três painéis fotovoltaicos em série, no centro o *model MPPT* que controla o conversor CC-CC *boost* e na direita uma fonte DC que simula o elo CC.

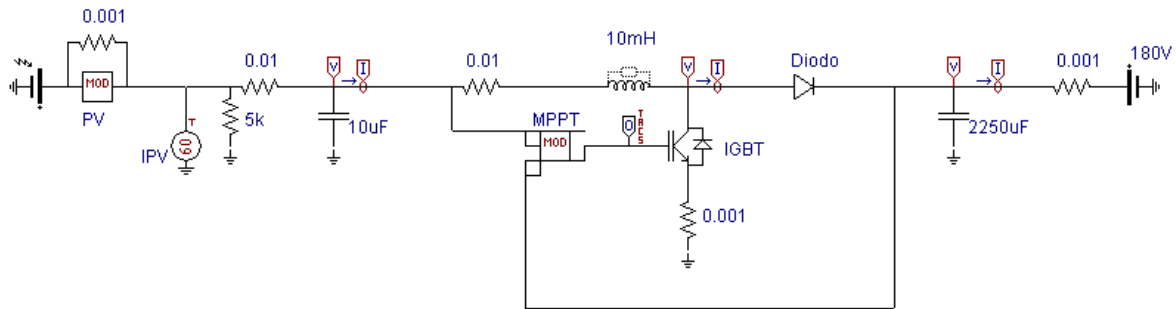


Figura 2.10: Esquema de Simulação no ATP

A Figura 2.11 mostra a tensão de saída do conversor CC-CC e a tensão de entrada. No período compreendido entre 0 e 0,2 segundos ocorre o processo de inicialização das variáveis. Em $t = 0,5 s$ a insolação é reduzida instantaneamente e ocorre uma redução na potência injetada pelo painel no elo CC. A redução instantânea permite avaliar o tempo de resposta do algoritmo, embora na prática as variações de potência dos painéis fotovoltaicas não sejam instantâneas e o tempo de resposta não seja crítico. A fonte CC ideal de 180 V conectada nos terminais de saída do conversor *boost* foi usada para manter a tensão do elo CC constante. Esse procedimento não implicará em erros já que no sistema completo o controlador dos conversores CC-CA irá manter essa tensão fixa num valor pré-determinado de projeto.

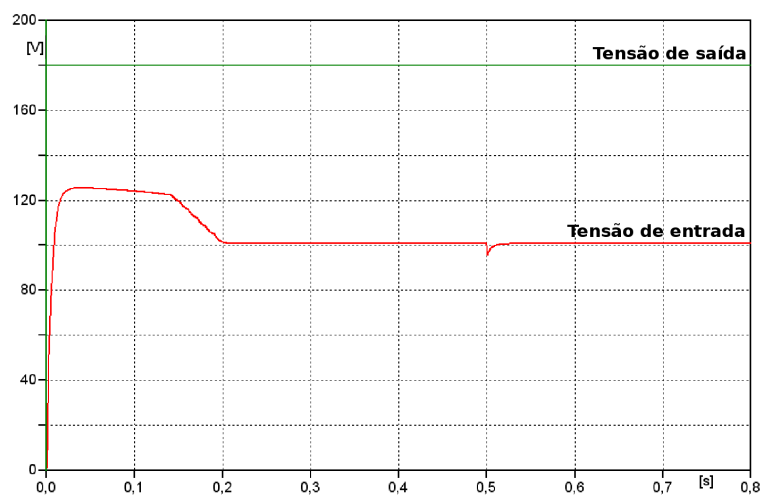


Figura 2.11: Tensão de Entrada e Saída do Conversor *Boost*.

Na Figura 2.12 pode ser observada a variação de corrente nos terminais de entrada e de saída do conversor CC-CC. Tanto a corrente de entrada quanto a corrente de saída sofrem reduções em suas amplitudes a partir de $t = 0,5$ s devido a redução da potência convertida pelos painéis solares fotovoltaicos.

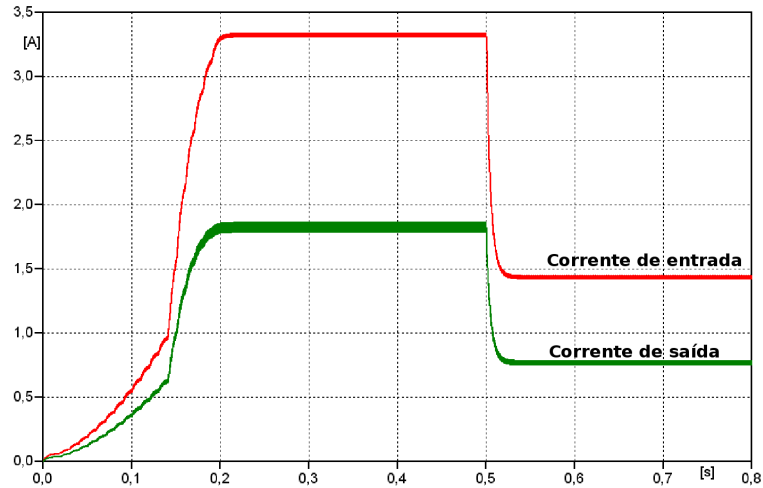


Figura 2.12: Corrente de Entrada e Saída do Conversor *Boost*.

Nas Figuras 2.13 e 2.14 são mostradas as potências de entrada e saída do conversor CC-CC, respectivamente. Essas potências são idênticas visto que as chaves semicondutoras do conversor *boost* são ideais. Essas curvas foram obtidas considerando o fator 100 % e 50 % para a insolação do painel.

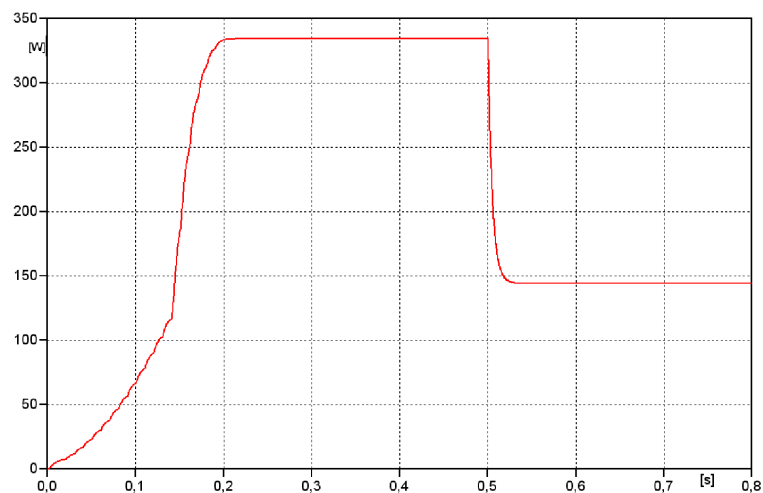


Figura 2.13: Potência de Entrada do Conversor *Boost*

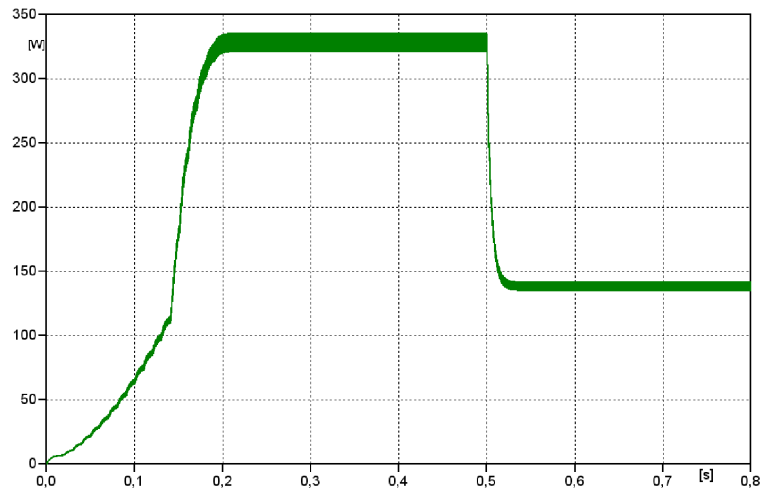


Figura 2.14: Potência de Saída do Conversor *boost*

A Figura 2.15 mostra a curva de potência x tensão para o painel. Nessa figura é possível observar a trajetória do ponto de operação devido ao rastreamento do ponto de máxima potência pelo algoritmo MPPT. As Figuras 2.16 (a) e (b) mostram detalhes das trajetórias de rastreamento dos dois pontos de máxima potência na curva Potência x Tensão, para 100 % e 50 % de insolação, respectivamente. Note que o algoritmo MPPT implementado fica sempre buscando o ponto de máxima potência.

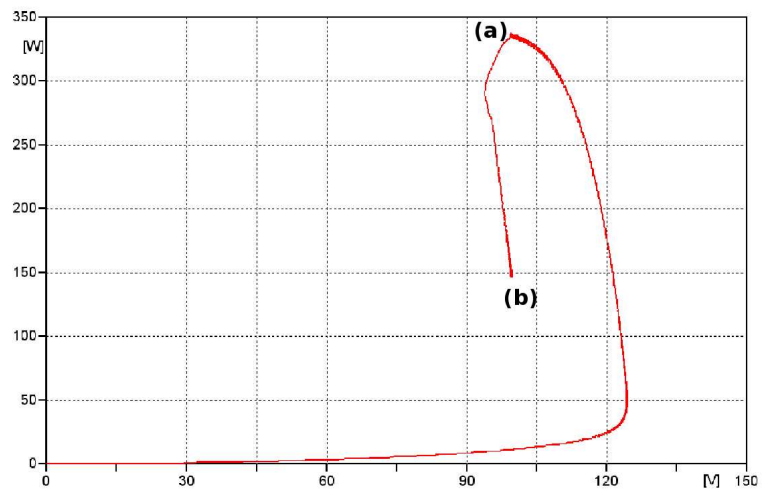


Figura 2.15: Potência x Tensão do Conversor *Boost*

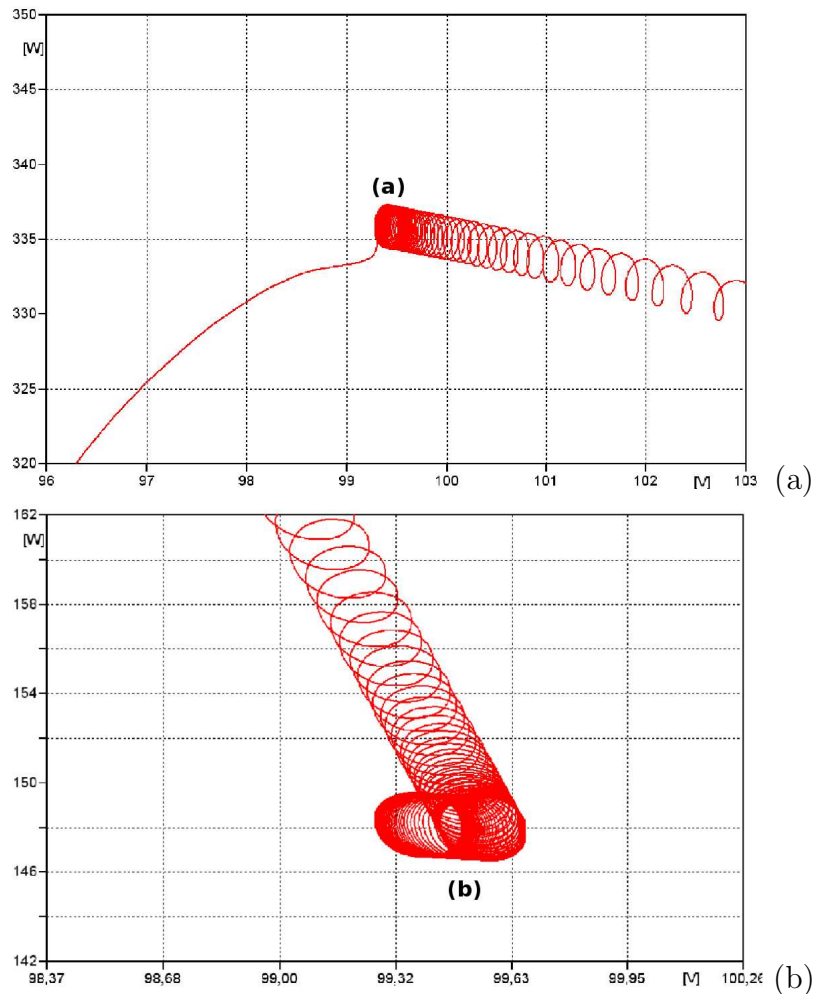


Figura 2.16: Detalhes do Rastreamento: (a) Ponto Superior; (b) Ponto Inferior.

2.4 Conversores Estáticos CC-CA

A Figura 2.17 mostra a topologia básica de um VSI trifásico com seis IGBTs e seis diodos conectados anti-paralelo. Esses diodos são necessários para permitir a condução de corrente reversa pelas chaves semicondutoras.

O conversor VSI mostrado na Figura 2.17, sem a fonte de tensão V_d , pode ser usado também como um compensador estático síncrono (STATCOM - “*STATic synchronous COMpensator*”) (Barbosa, 2000). Essa característica será usada nas próximas seções para projetar o controlador do conversor CC-CA com o objetivo de agregar ao sistema de geração fotovoltaico a capacidade de operação como compensador estático de energia reativa.

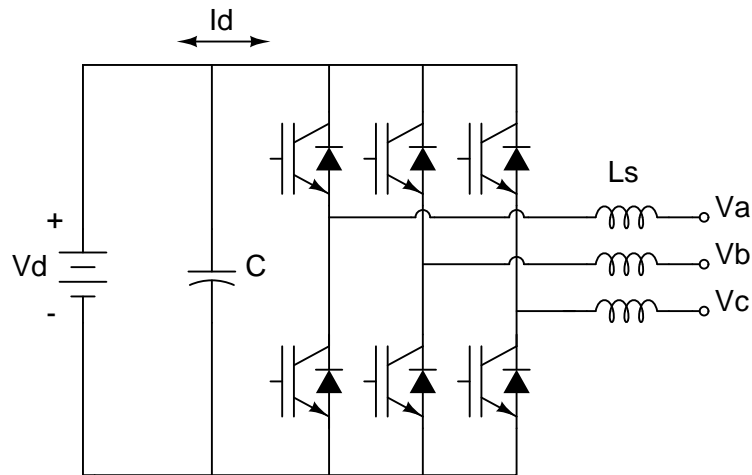


Figura 2.17: Topologia Básica de um VSI Trifásico.

2.4.1 Técnicas de Chaveamento dos Conversores Estáticos

As técnicas de chaveamento por modulação da largura de pulso (PWM - "Pulse Width Modulation") são largamente utilizadas em aplicações industriais de baixas e médias potências. Elas têm a vantagem de gerarem harmônicos em altas frequências que são facilmente eliminados com a utilização de pequenos filtros passivos (Mohan, Undeland e Robbins, 1995).

Contudo as aplicações em altas e extra-alta potências, nas quais as perdas por chaveamento são críticas, o chaveamento dos conversores estáticos é geralmente feito em onda retangular. Neste caso porém o conteúdo harmônico da tensão sintetizada é elevado já que os VSIs são chaveados em baixa frequência. Nessas aplicações, as técnicas de chaveamento multipulso têm sido preferidas, onde vários VSIs têm suas tensões de saída combinadas formando uma tensão com múltiplos níveis e com um baixo índice de distorção harmônica.

Em conseqüência da potência instalada do sistema de geração de energia elétrica fotovoltaico e considerando as recomendações propostas pela normas (ANSI/IEEE-519, 1992), foi adotada, *inicialmente*, uma estratégia intermediária para chavear os conversores VSIs do sistema de geração fotovoltaico.

A Figura 2.18 mostra o arranjo utilizado. Foram utilizados dois conversores VSIs trifásicos conectados em série através dos enrolamentos de dois transformadores Δ -Y. A conexão série dos VSIs reduz a potência dos conversores já que a tensão do sistema CA é

dividida entre os dois conversores CC-CA, além disso, a ligação Δ do lado dos conversores serve para minimizar os desequilíbrios gerados por erros de chaveamento. A estratégia adotada para o chaveamento dos VSIs combina um chaveamento multipulso com um chaveamento PWM de baixa frequência, porém com eliminação seletiva de harmônicos conforme mostrado na Figura 2.19 (Carletti, Lopes e Barbosa, 2003). São feitos recortes (“*choppers*”) na forma de onda de tensão de cada VSI com ângulos predeterminados para eliminar determinados harmônicos. Os ângulos dos recortes foram escolhidos para eliminar o 11°, 13° e 23° harmônicos presentes nas tensões de saída de cada VSI.

Neste trabalho os recortes não serão usados para controlar a componente fundamental de tensão sintetizada pelos conversores. Como será explicado mais tarde, o controle da componente fundamental da tensão será feito por uma malha externa que atua sobre o ângulo de defasagem das tensões sintetizadas pelos VSIs.

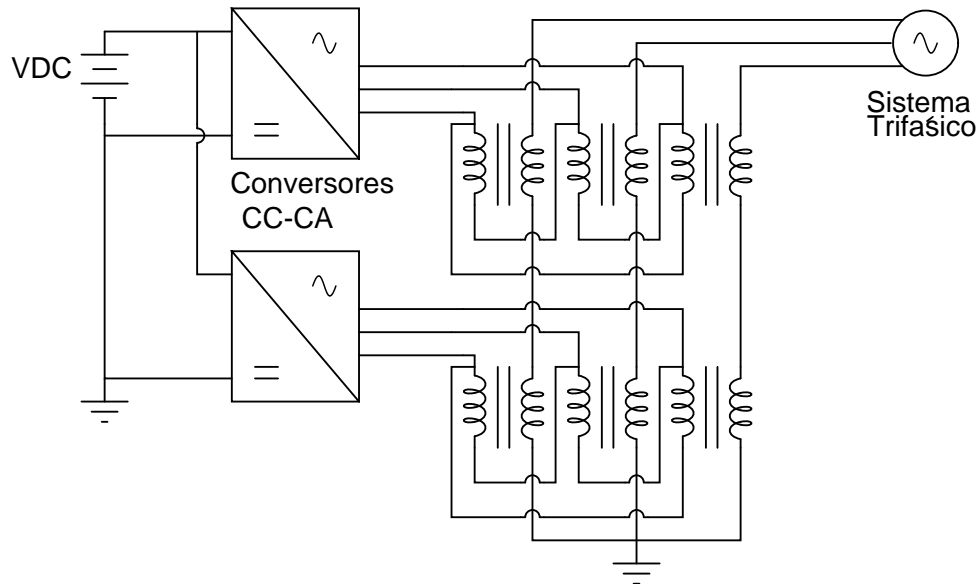


Figura 2.18: Esquema de Conexão dos VSIs Trifásicos.

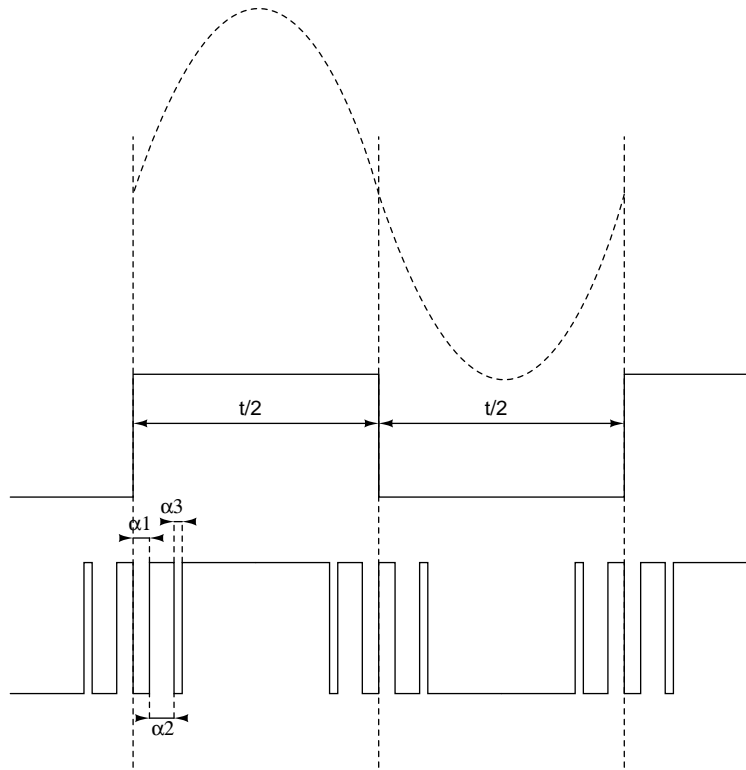


Figura 2.19: Técnica de Chaveamento do VSI com Eliminação Seletiva Harmônica.

Para otimizar o conteúdo harmônico da tensão de saída dos conversores da Figura 2.18 os pulsos enviados para as chaves semicondutoras dos dois VSIs são defasados de 30° entre si. Esta defasagem eletrônica entre os sinais de disparo é conseguida adiantando e atrasando os ângulos de disparo de cada fase dos dois VSIs de $+15^\circ$ e -15° , respectivamente.

2.5 Controlador dos Conversores CC-CA

O controlador dos conversores CC-CA do sistema de geração fotovoltaico é semelhante ao controle proposto para um STATCOM e está mostrado no diagrama de blocos da Figura 2.20(Barbosa, 2000). Ele é baseado nas teorias das potências real e imaginária instantâneas (Akagi, Kanazawa e Nabae, 1984) e (Watanabe, Stephan e Aredes, 1993) e seu funcionamento é descrito a seguir.

As tensões e correntes trifásicas instantâneas nos terminais dos VSI sofrem uma transformação de Clarke e são usadas para calcular uma potência imaginária instantânea. O erro entre a potência imaginária de referência e a potência medida nos terminais CA

do VSI alimenta um controlador PI cuja saída fornece um sinal que é proporcional ao ângulo de defasagem das tensões que os conversores devem sintetizar em seus terminais. O princípio de funcionamento do controlador está mostrado no Apêndice A. Maiores detalhes da técnica de controle podem ser obtidos em (Carletti, Lopes e Barbosa, 2003).

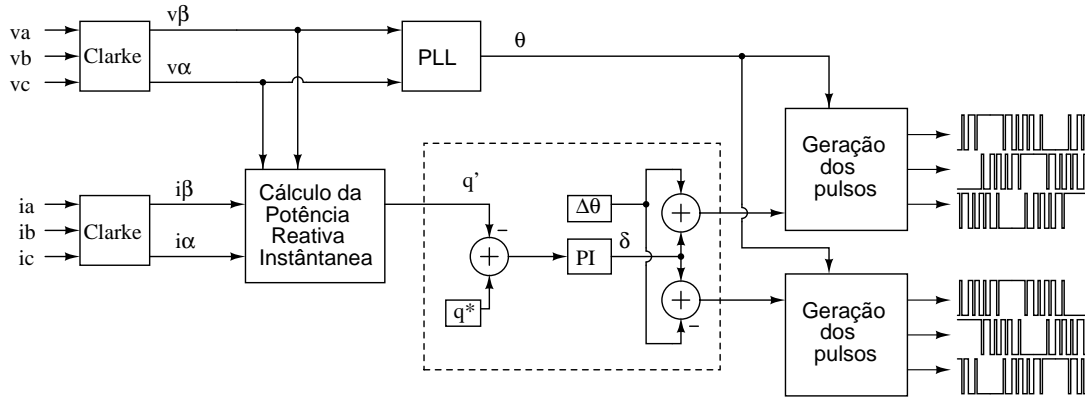


Figura 2.20: Diagrama de Blocos do Controle do Conversor CC-CA.

Quando uma quantidade de energia ativa proveniente dos painéis fotovoltaicos chega ao elo CC a tensão sobre o capacitor e as tensões trifásicas sintetizadas pelo VSI tendem a crescer. A variação da amplitude das tensões alternadas do VSI força a potência imaginária instantânea injetada ou absorvida na rede CA variar. O controlador então altera o ângulo de carga (δ) alterando a fase das tensões chaveadas nos terminais dos VSIs. Assim a potência ativa elo CC, proveniente dos painéis fotovoltaicos, é injetada no sistema trifásico CA e a potência imaginária instantânea nos terminais fica igual ao valor da potência imaginária de referência.

Caso seja definido um valor diferente de zero para a potência imaginária instantânea de referência, o conversor CC-CA, além da potência ativa proveniente dos painéis fotovoltaicos, injeta ou absorve uma potência imaginária instantânea. Essa característica é interessante para os períodos sem insolação em que os painéis não estão injetando potência ativa na rede CA. Nesses períodos o sistema de geração fotovoltaico poderá funcionar como um compensador estático de energia reativa ¹.

¹O valor médio da potência imaginária instantânea (q) nos terminais dos conversores CC-CA, devido às componentes fundamentais de sequência positiva das tensões e correntes trifásicas instantâneas, é numericamente idêntico à potência reativa convencional ($Q_{3\Phi}$).

2.5.1 Simulação dos Conversores Estáticos

Os algoritmos de controle dos conversores CC-CA do sistema de geração fotovoltaico foram implementados no ATP/EMTP (*Alternative Transients Program/ Electromagnetic Transients Program*). Como o objetivo básico é o desenvolvimento de controladores digitais para os conversores, fez-se a opção por utilizar a linguagem *Models* para programar os algoritmos de controle (ATP, 2005).

Na Figura 2.21 é mostrado o esquema usado para simulação dos conversores CC-CA e os valores utilizados na simulação são mostrados na Tabela 2.2. São usados três ambientes *Models*, dois para os blocos de disparo e um para o de controle de potência reativa instantânea. Cada um desses blocos *Models* corresponde a um DSP onde o algoritmo de controle é escrito em linguagem estruturada de programação. A escolha pela arquitetura de controle distribuída é baseada na limitação dos DSPs (TMS320F243 da *Texas Instruments*) utilizados para a implementação prática dos controladores dos dois conversores.

Os conversores CC-CA estão ligados por transformadores a uma linha a parâmetros concentrados, que por sua vez está conectada a um fonte trifásica ideal que representa a subestação do sistema CA. Os sinais de disparo dos conversores CC-CA devem ser sincronizados com a tensão do sistema CA. A tarefa de detecção do ângulo de fase das tensões alternadas é feita por um circuito PLL, que será discutido em detalhes no Capítulo 3.

Tabela 2.2: Valores do Esquema de Simulação da Planta no ATP

Dispositivo	Valor	Valor em PU
Potência instalada	30 kW	100%
Tensão de linha (pico)	180 V	100%
Resistência das chaves	$10\mu\Omega$	0,0003%
Resistência da Linha	$0,5 \Omega$	15,43%
Reatância da Linha	$0,754 \Omega$	23,27%
Reatância primária dos transformadores	$0,324 \Omega$	10%
Reatância secundária dos transformadores	$0,324 \Omega$	10%
Reatância do filtro capacitivo	$56,43 \Omega$	1741,87%

A Figura 2.22 mostra a tensão de linha na saída de um conversor CC-CA sem carga. Como explicado anteriormente, os ângulos dos recortes usados para a eliminação seletiva dos harmônicos são iguais a $4,4262^\circ$, $18,0406^\circ$ e $23,0919^\circ$, esses ângulos foram escolhidos para eliminar o 11° , 13° e 23° harmônicos presentes nas tensões de saída de cada VSI da configuração mostrada na Figura 2.18.

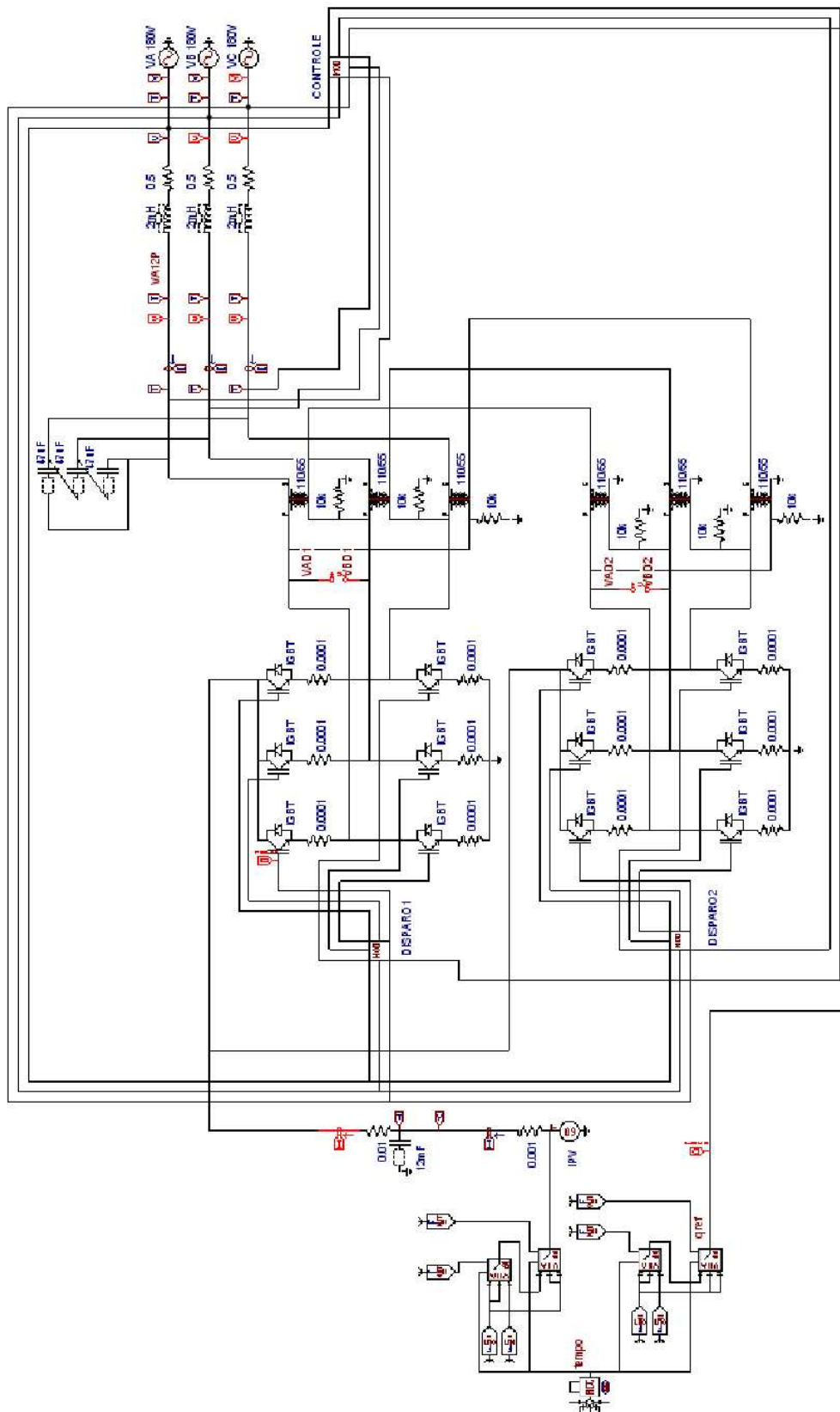


Figura 2.21: Esquema de Simulação da Planta no ATP.

A Figura 2.23 mostra a tensão de linha na saída dos transformadores de acoplamento dos dois conversores CC-CA sem carga. Observe que a tensão resultante possui 5 níveis distintos além dos recortes necessários para a eliminação seletiva de harmônicos.

Na Figura 2.24 é mostrado o funcionamento do sistema quando a referência do controle de potência imaginária é variada em degrau de 0 para $10kvai$ ² de amplitude entre 0,3 s e 0,6 s. Nesta simulação é possível verificar o funcionamento do sistema gerando apenas potência imaginária em seus terminais.

Na Figura 2.25 é possível observar o comportamento do sistema quando uma certa quantidade de potência ativa, proveniente dos painéis fotovoltaicos, é injetada no elo CC. O sistema opera inicialmente sem absorver/fornecer potência ativa em seus terminais, quando em $t = 0,9$ s a fonte CC que emula os painéis fotovoltaicos injeta 10 kW no elo CC. O controlador mostrado na Figura 2.20 detecta essa variação e passa a fornecer potência ativa nos terminais CA do VSI. Em $t = 1,2$ s a potência ativa injetada pelo painel é feita igual a zero novamente e o controlador do conversor segue a ação reduzindo a potência fornecida em seus terminais. O atraso observado na resposta dos dois sinais de potência é devido às constantes de tempo dos reguladores PI do controlador dos VSIs.

Na Figura 2.26 são mostradas as potências ativa e reativa instantâneas nos terminais CA do sistema de geração fotovoltaico. Note que quando o conversor está funcionando como compensador estático de reativo é consumida potência ativa do sistema CA para suprir as perdas (período compreendido entre 0,3 s a 0,6 s), essas perdas são a soma das perdas nos conversores, transformadores e na linha de transmissão.

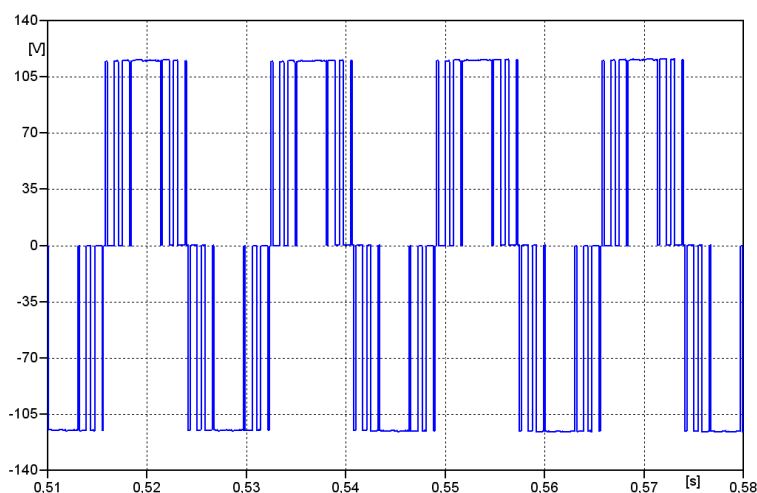


Figura 2.22: Tensão de Linha na Saída de um Conversor CC-CA sem Carga.

²A unidade da potência imaginária instantânea é definida como *vai* (Volt-Ampère imaginário).

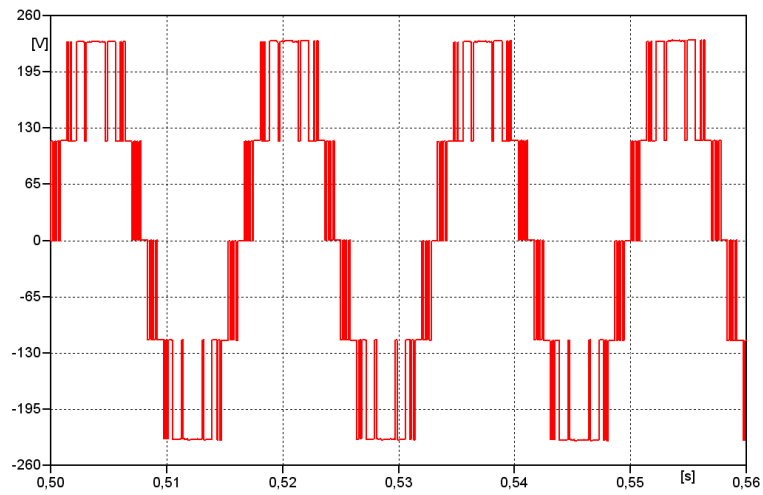


Figura 2.23: Tensão de Linha na Saída do Acoplamentos dos Dois Conversores CC-CA sem Carga.

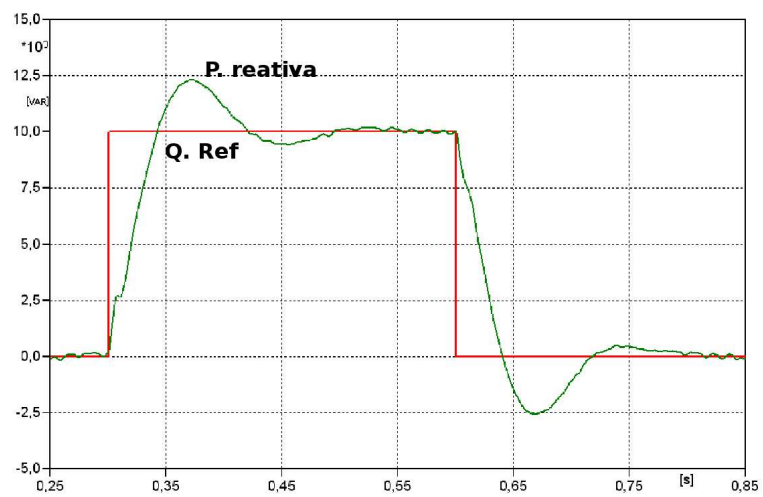


Figura 2.24: Potência Imaginária Instantânea Injetada no Sistema CA.

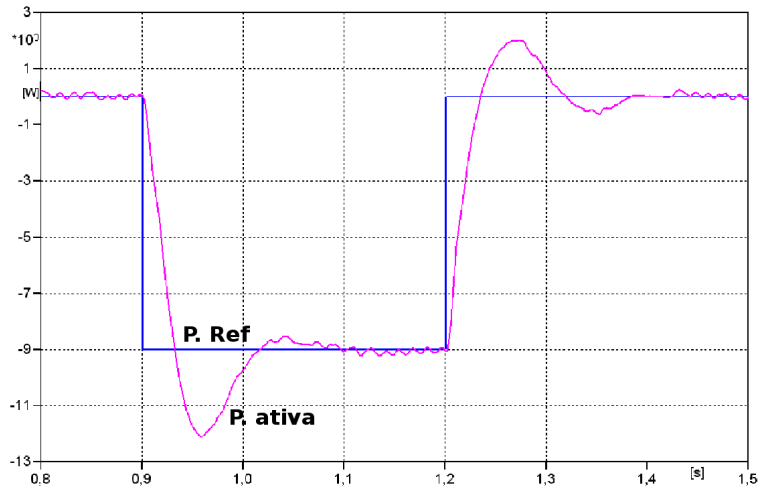


Figura 2.25: Potência Ativa Instantânea Proveniente do Painel Fotovoltaico e Potência Ativa Instantânea Injetada no Sistema CA.

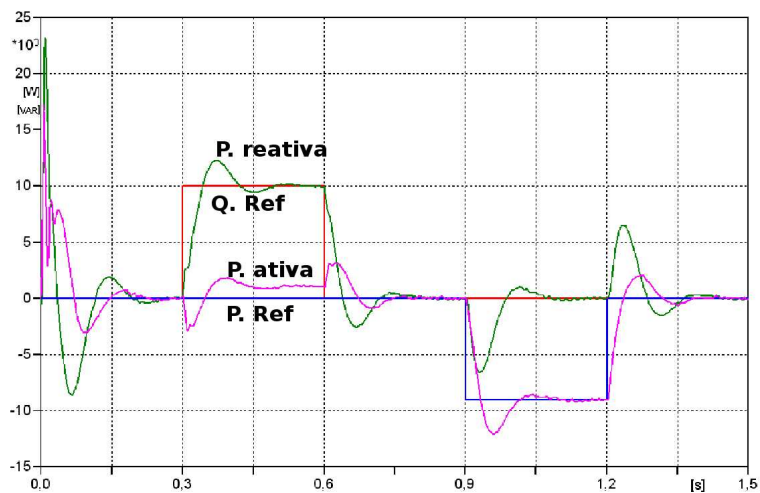


Figura 2.26: Potências Ativa e Imaginária Instantâneas nos Terminais dos Conversores CC-CA.

A Figura 2.27 mostra a tensão de linha da fase “a” na saída do transformador de acoplamento dos conversores CC-CA enquanto que a Figura 2.28 mostra o comportamento da corrente na fase “a”. Finalmente a Figura 2.29 mostra o comportamento da tensão do elo CC do conversor. Observe que as variações observadas nas três figuras estão relacionadas com os valores das potências ativa e reativa instantâneas nos terminais do conversor CC-CA, estando de acordo com o funcionamento esperado do controlador, que é explicado no Apêndice A.

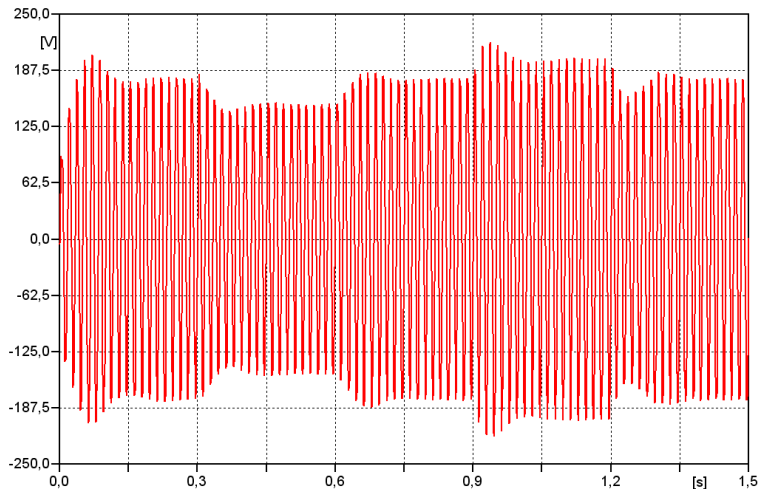


Figura 2.27: Tensão na Fase “a” na Saída dos Conversores CC-CA.

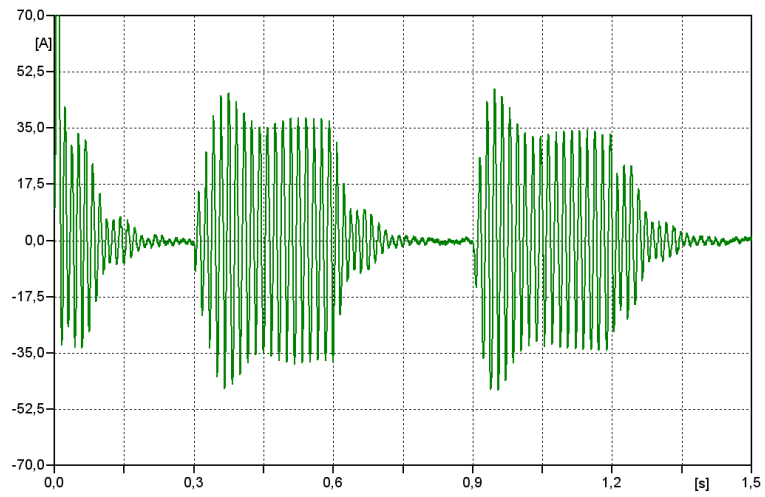


Figura 2.28: Corrente na Fase “a”.

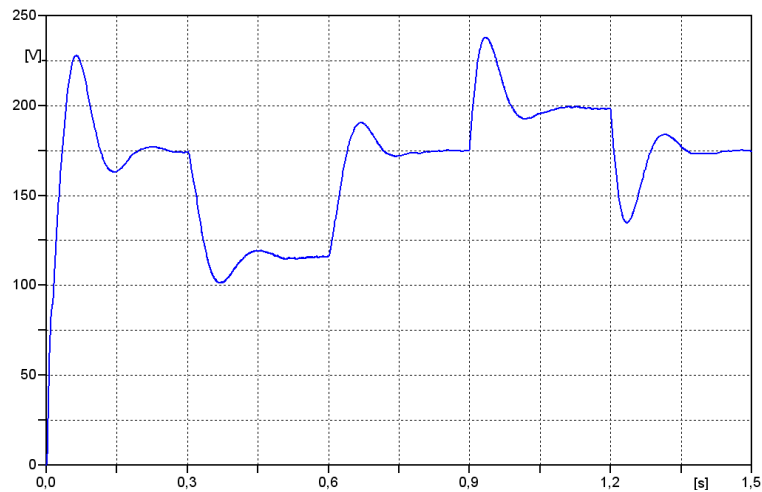


Figura 2.29: Tensão no Elo CC.

2.5.2 Simulação dos Conversores Estáticos com Modulação PWM

A modulação Seno- Δ -PWM ou PWM-senoidal é uma estratégia de chaveamento onde uma portadora triangular de frequência f_s é usada para modular a largura dos pulsos de chaveamento dos VSIs com o objetivo de obter tensões trifásicas com amplitude, frequência e fase controladas (Mohan, Undeland e Robbins, 1995).

Figura 2.30 mostra a estratégia de chaveamento PWM utilizada, onde se pode observar os instantes de condução e cortes dos IGBTs do conversor, que são determinados a partir da comparação entre a tensão de referência senoidal (v_{ref}^*), com uma frequência angular fundamental ω (rad/s), e a portadora triangular (v_{tri}), com uma frequência angular ω_s (rad/s).

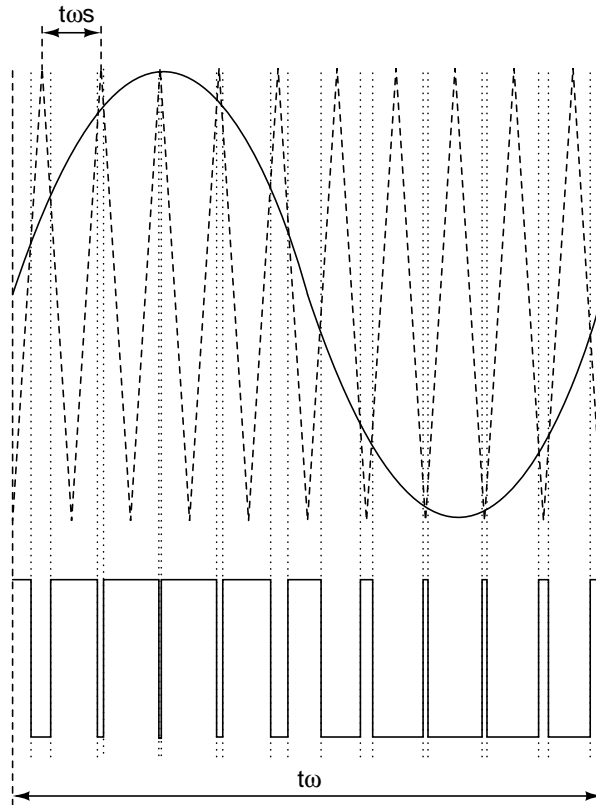


Figura 2.30: Técnica de Chaveamento PWM do VSI

O projeto inicial dos controladores dos conversores CC-CA do sistema de geração fotovoltaico foi modificado para incluir a estratégia de modulação PWM para podermos avaliar as diferenças e os resultados das técnicas de chaveamento adotadas. As rotinas dos blocos *Models* responsáveis pelos disparos das chaves autocomutadas foram reescritas conforme discutido anteriormente, já o bloco *Models* corresponde a malha de controle da potência imaginária instantânea permaneceu inalterado.

A Figura 2.31 mostra a tensão de linha na saída de um conversor CC-CA sem carga. Essa tensão é obtida a partir da modulação da tensão de referência senoidal com uma portadora triangular com uma frequência de 6 kHz. A Figura 2.32 mostra a tensão de linha na saída do acoplamentos dos transformadores dos dois conversores CC-CA sem carga..

Na Figura 2.33 é mostrado o funcionamento do sistema quando a referência do controle de potência imaginária é variada em degrau de 0 para 10k vai de amplitude entre 0,3 s e 0,6 s. Nesta simulação é possível verificar o funcionamento do sistema gerando apenas potência reativa em seus terminais.

Na Figura 2.34 é possível observar o comportamento do sistema quando uma certa quantidade de potência ativa, proveniente dos painéis fotovoltaicos, é injetada no elo CC. O sistema opera inicialmente sem absorver/fornecer potência ativa em seus terminais quando em $t = 0,9$ s a fonte CC que emula o comportamento dos painéis fotovoltaicos injeta 10 kW no elo CC. Em $t = 1,2$ s a potência ativa injetada pelo painel é feita igual a zero novamente e o controlador do conversor segue a ação reduzindo a potência fornecida em seus terminais.

Na Figura 2.35 são mostradas as potências ativa e imaginária instantâneas nos terminais CA do sistema de geração fotovoltaico. Note que quando o conversor está funcionando como compensador estático consome uma quantidade de potência ativa do sistema CA para suprir as perdas do mesmo (período compreendido entre 0,3 s a 0,6 s), como já explicado anteriormente.

A Figura 2.36 mostra a tensão de linha da fase “a” na saída do transformador de acoplamento dos conversores CC-CA enquanto que a Figura 2.37 mostra o comportamento da corrente na fase “a”. Finalmente a Figura 2.38 mostra o comportamento da tensão do elo CC do conversor. Observe que as variações observadas nas três figuras estão relacionadas com os valores das potências ativa e imaginária instantâneas nos terminais do conversor CC-CA.

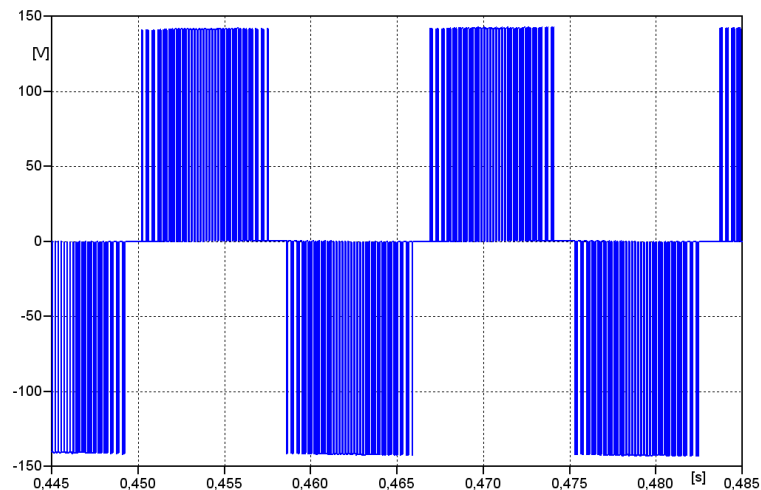


Figura 2.31: Tensão de Linha na de um Saída do Conversor CC-CA sem Carga.

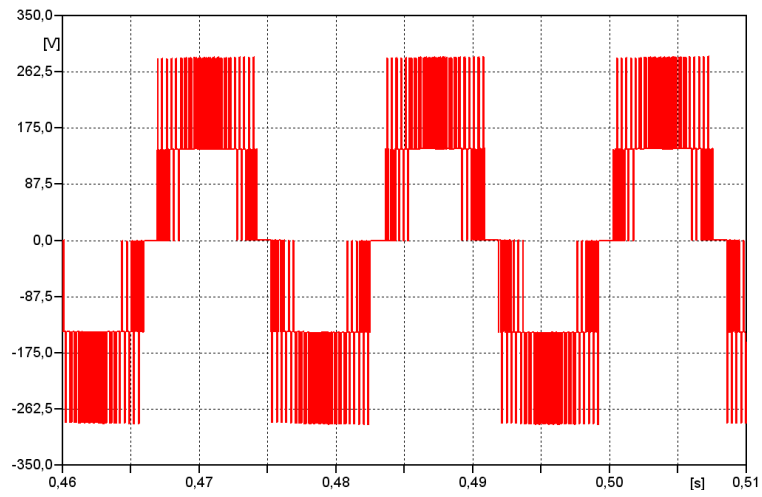


Figura 2.32: Tensão de Linha na Saída do Acoplamentos dos Dois Conversores CC-CA sem Carga.

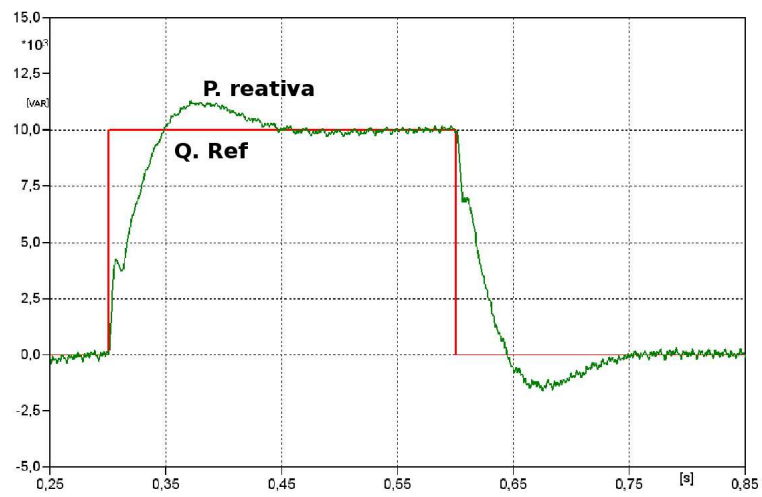


Figura 2.33: Potência Imaginária Instantânea Injetada na Rede CA.

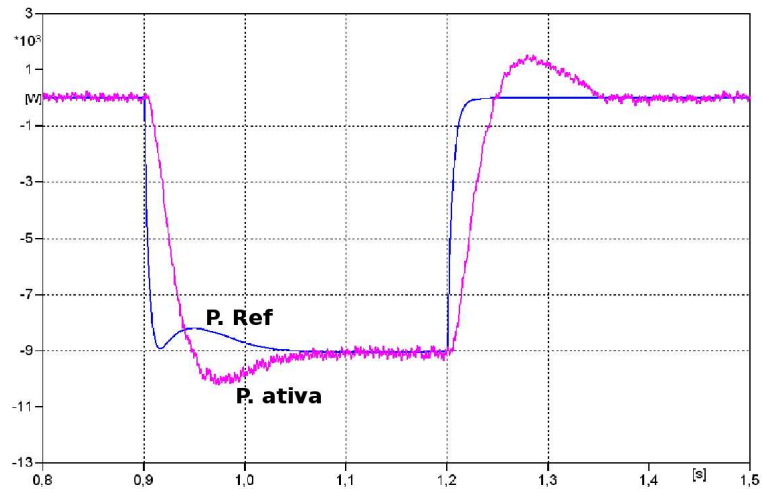


Figura 2.34: Potência Ativa Proveniente do Pannel Fotovoltaico e Potência Ativa Instantânea Injetada na Rede CA.

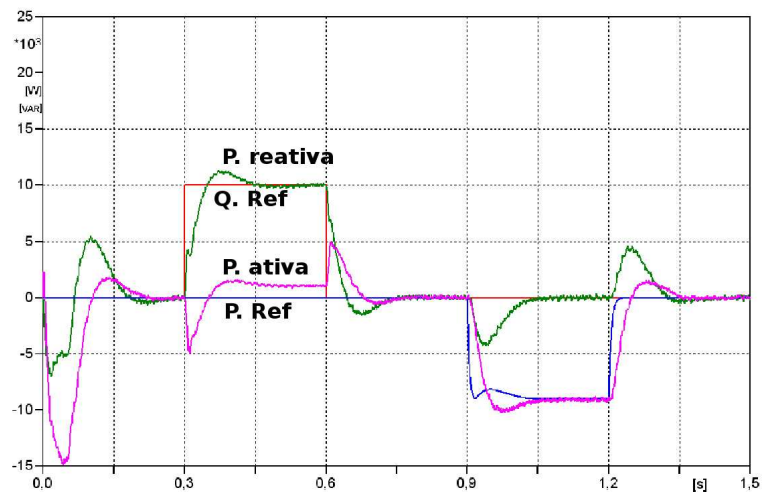


Figura 2.35: Potências Ativa e Imaginária Instantâneas nos Terminais dos Conversores.

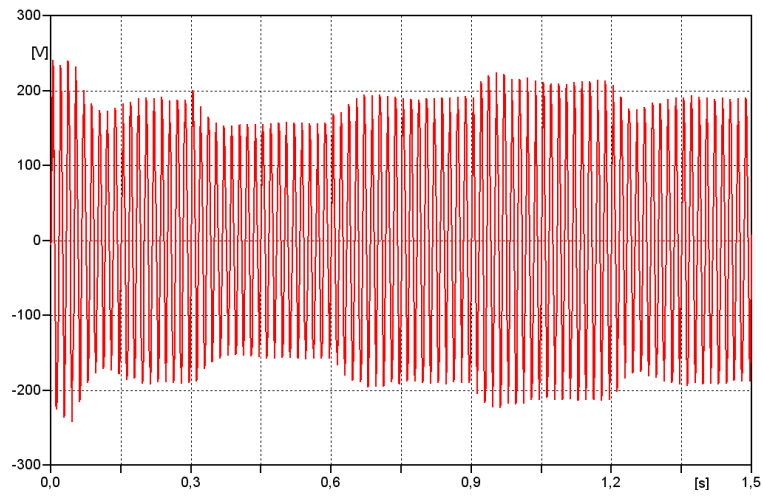


Figura 2.36: Tensão na Fase “a” na Saída do Conversor CC-CA.

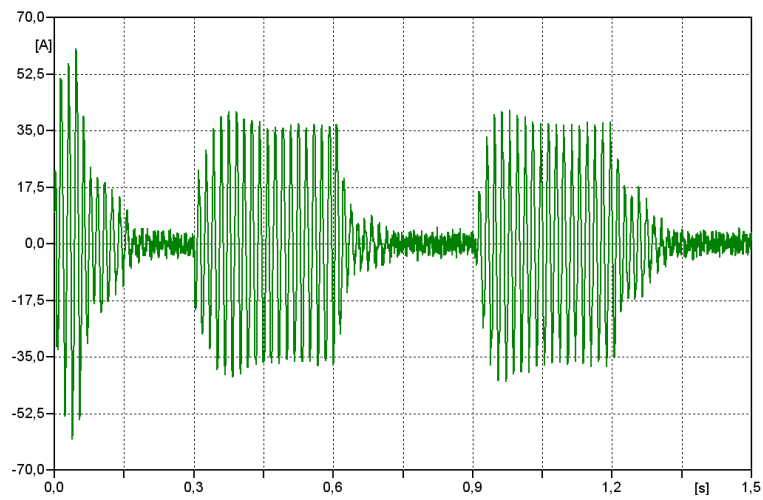


Figura 2.37: Corrente na Fase “a”.

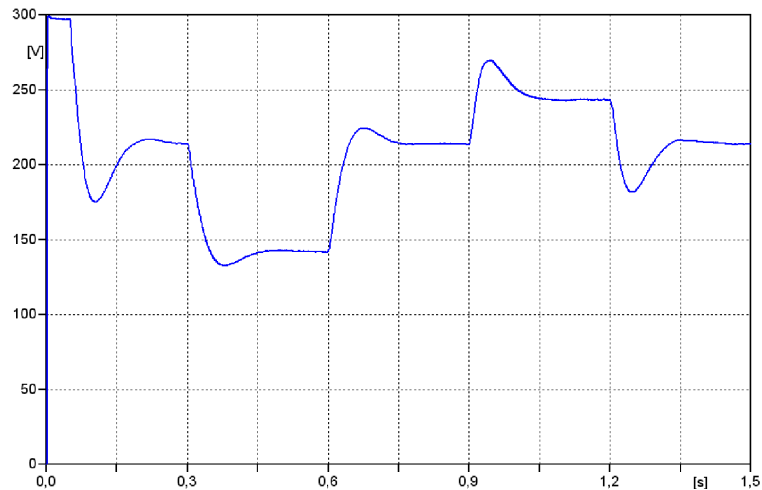


Figura 2.38: Tensão no Elo CC.

2.6 Conclusões Parciais

Este Capítulo descreveu todas as partes constituintes do sistema fotovoltaico de geração de energia elétrica. Foram apresentados os conversores estáticos e os algoritmos de controle utilizados. Todo o sistema foi simulado no programa de transitórios eletromagnéticos ATP/EMTP.

O desenvolvimento de um modelo digital aproximado para os painéis solares fotovoltaicos permitiu investigar a utilização de técnicas de rastreamento do ponto de máxima potência para controlar os conversores CC-CC.

Inicialmente o controle dos conversores CC-CA utilizou uma técnica de chaveamento multipulso com eliminação seletiva de harmônicos. Porém devido à concepção modular do sistema de controle, foi possível investigar o desempenho do sistema usando uma técnica de modulação PWM.

O resultados obtidos demonstraram que as topologias e as técnicas adotadas para o controle dos conversores CC-CC e CC-CA podem ser perfeitamente empregadas no controle do sistema de geração proposto.

Capítulo 3

Circuito de Sincronismo (PLL)

3.1 Introdução

No capítulo anterior foi mostrado que os pulsos de disparo dos conversores CC-CA devem ser sincronizados com as tensões alternadas do sistema CA para garantir correta operação dos conversores CC-CA. A falta de sincronismo entre a tensão sintetizada pelos VSIs e a tensão alternada do sistema CA pode provocar falhas e danos nos conversores. Para gerar o sinal de referência de sincronismo pode ser usado um circuito de sincronismo (PLL - “Phase Locked Loop”).

Neste capítulo serão apresentadas as etapas do projeto de um circuito de sincronismo implementado em software, que garanta bom desempenho e resposta rápida dos conversores CC-CA.

3.2 Modelo do PLL Baseado nas Potências Instantâneas

As Figura 3.1 (a) e (b) mostram os diagramas de blocos dos dois circuitos PLL estudados. Ambos os circuitos usam o conceito da potência real e imaginária instantâneas (Akagi, Kanazawa e Nabae, 1984). Nessas figuras as tensões instantâneas (v_α) e (v_β) são obtidas pela aplicação da transformada de Clarke (Watanabe, Stephan e Aredes, 1993)

num conjunto de tensões trifásicas instantâneas de um sistema elétrico como se segue:

$$\begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad (3.1)$$

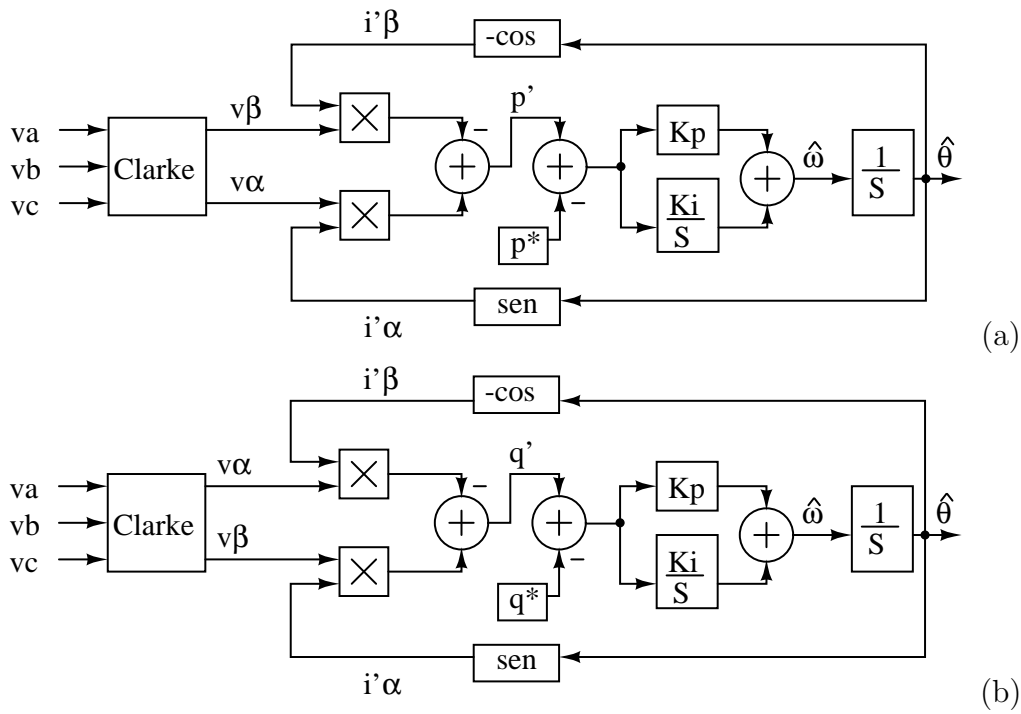


Figura 3.1: Diagramas dos Circuitos de Detecção de Frequência: (a) p -PLL e (b) q -PLL.

Uma transformação idêntica pode ser aplicada a um conjunto de correntes trifásicas para obter as correntes (i'_α) e (i'_β) mostrada na Figura 3.1 (a) e (b). Contudo, conforme vamos explicar mais adiante essas correntes instantâneas são fictícias, ou seja, não são reais e somente são usadas para detectar a frequência e a fase das tensões do sistema.

As componentes v_α , v_β , i'_α e i'_β podem ser interpretadas como as projeções instantâneas, sobre um sistema de coordenadas ortogonais (α - β) como mostrado na Figura 3.2, de um vetor espacial de tensão (\vec{v}) e um vetor espacial de corrente (\vec{i}) que giram com uma velocidade angular ω constante. Deve ser observada a sequência de fase do sistema, caso seja invertida, o vetor espacial irá girar em sentido inverso e o PLL não funcionará corretamente.

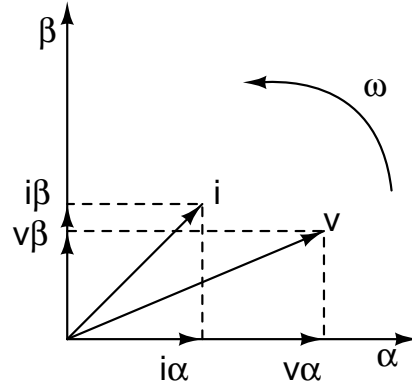


Figura 3.2: Vetor Espacial de Tensão e Corrente.

Desse modo as potências real $p'(W)$ e imaginária $q'(vai)$ instantâneas podem ser calculadas como se segue:

$$\begin{bmatrix} p' \\ q' \end{bmatrix} = \begin{bmatrix} v_\alpha & v_\beta \\ v_\beta & -v_\alpha \end{bmatrix} \begin{bmatrix} i'_\alpha \\ i'_\beta \end{bmatrix}, \quad (3.2)$$

onde vai é a unidade adotada para a potência imaginária instantânea e significa volt-ampere imaginário.

3.2.1 Análise do Funcionamento do PLL

Da Figura 3.1 (a) e (b), as potências real (p') ou imaginária (q') instantâneas calculadas são comparadas com os sinais das potências real (p^*) e imaginária (q^*) de referência, respectivamente. Da comparação desses sinais, é gerado um sinal de erro que é usado para alimentar um controlador PI que em sua saída fornece a frequência angular do sistema ($\hat{\omega}$). Então, a frequência angular detectada alimenta um bloco de integração usado para gerar o sinal de sincronismo $\hat{\theta} = \hat{\omega}t + \hat{\delta}$. Sendo que $\hat{\omega}$ representa a frequência detectada do sistema em rad/s e $\hat{\delta}$ representa a fase detectada do sistema em rad . Finalmente, o sinal de sincronismo detectado $\hat{\theta}$ é usado para gerar as correntes fictícias i'_α e i'_β que são os sinais de realimentação do circuito de sincronismo.

Pelo exposto a potência instantânea medida só é igual à potência de referência desejada se a velocidade angular ($\hat{\omega}$) do vetor espacial referente às correntes fictícias for igual à velocidade angular (ω) do vetor espacial referente a tensões trifásicas instantâneas. Em (Barbosa, 2000) é mostrado que quando as potências instantâneas de referência são

nulas, *i.e.*, $p^* = 0$ e $q^* = 0$, o ângulo de fase ($\hat{\delta}$) do vetor espacial das correntes fictícias é $-\pi/2 \text{ rad}$ ou $+\pi/2 \text{ rad}$ para o caso do p -PLL ou 0 rad ou $\pi \text{ rad}$ para o caso do q -PLL. Contudo é também mostrado nessa mesma referência que os ângulos $+\pi/2 \text{ rad}$ do p -PLL e $\pi \text{ rad}$ caso do q -PLL são soluções instáveis¹.

Como matematicamente não há diferenças entre os dois modelos de PLL, no entanto o nome de cada PLL tem os prefixos p ou q em consequência da potência real ou imaginária instantânea usada como sinal de realimentação respectivamente (Barbosa, 2000). Então, deste ponto para frente, vamos analisar somente o q -PLL.

3.3 Modelo Linearizado do q -PLL

O comportamento dinâmico do q -PLL é fortemente dependente dos ganhos do controlador PI mostrados nas Figuras 3.1 (a) e (b). Para o q -PLL, pode-se chegar a um modelo de pequenos sinais linearizado, a partir da expressão da potência imaginária instantânea que é, para um sistema de tensões trifásicas equilibradas, dada por:

$$q' = v_{\beta}i'_{\alpha} - v_{\alpha}i'_{\beta} = -\sqrt{3}V \text{sen}(\theta - \hat{\theta}), \quad (3.3)$$

onde, V é a tensão eficaz de fase do sistema em (*Volts*), $\theta = (\omega t + \phi)$ sendo que ω é a frequência angular em (*rad/s*) e ϕ é o ângulo de fase em (*rad*) da tensão de fase do sistema, respectivamente, $\hat{\theta} = (\hat{\omega} t + \hat{\phi})$ sendo que $\hat{\omega}$ é frequência angular em (*rad/s*) e $\hat{\phi}$ é ângulo de fase em (*rad*) detectados pelo PLL, respectivamente.

Expandindo (3.3) através da série de Taylor e desprezando os termos de derivada superior a um obtém-se o seguinte modelo linearizado para o q -PLL:

$$\begin{aligned} \Delta q' = & -\sqrt{3}V_0 \cos(\theta_0 - \hat{\theta}_0) \Delta \theta + \sqrt{3}V_0 \cos(\theta_0 - \hat{\theta}_0) \Delta \hat{\theta} - \\ & -\sqrt{3} \text{sen}(\theta_0 - \hat{\theta}_0) \Delta V, \end{aligned} \quad (3.4)$$

onde o subscrito (0) indica que as variáveis em questão estão em regime permanente e o

¹Apresentam pólos no semiplano direito do plano complexo (Ogata, 1997).

símbolo (Δ) indica pequenas variações das grandezas em torno do ponto de operação em regime permanente.

A Figura 3.3 mostra a representação por blocos do q -PLL obtido a partir de (3.4) cujo modelo matemático escrito no espaço de estados é dado por:

$$s \begin{bmatrix} \Delta \hat{\theta}(s) \\ \Delta \hat{\omega}(s) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k_i k_0 & -k_p k_0 \end{bmatrix} \begin{bmatrix} \Delta \hat{\theta}(s) \\ \Delta \hat{\omega}(s) \end{bmatrix} + \begin{bmatrix} 0 \\ (\frac{k_i k_0}{s}) + (k_p k_0) \end{bmatrix} \begin{bmatrix} \Delta \omega(s) \\ \Delta V(s) \end{bmatrix} \quad (3.5)$$

onde, $k_0 = \sqrt{3}V_0 \cos(\theta_0 - \hat{\theta}_0)$ e $k_1 = -\sqrt{3} \text{sen}(\theta_0 - \hat{\theta}_0)$.

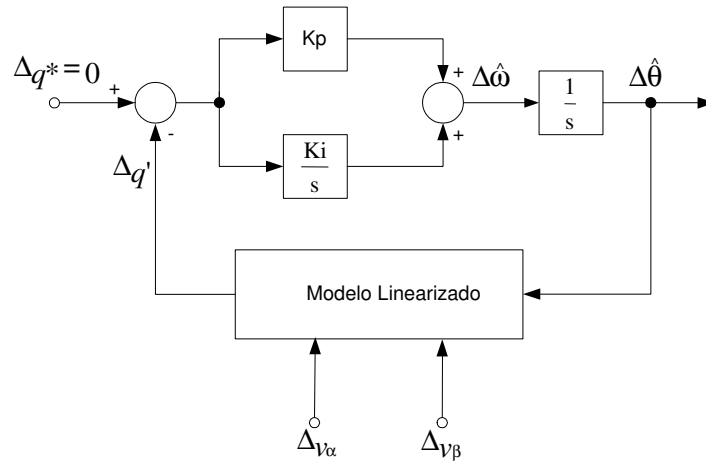


Figura 3.3: Diagrama de Blocos do q -PLL Linearizado.

Como mostrado em (Sasso, Sotelo, Ferreira, Watanabe, Aredes e Barbosa, 2002), as funções de transferência de malha fechada do q -PLL são dadas por:

$$\frac{\Delta \hat{\omega}}{\Delta V} = \frac{(-k_p k_1) s^2 + (-k_i k_1) s}{s^2 + (k_p k_0) s + (k_i k_0)} \quad (3.6)$$

e,

$$\frac{\Delta \hat{\omega}}{\Delta \omega} = \frac{(k_p k_0) s + (k_i k_0)}{s^2 + (k_p k_0) s + (k_i k_0)} \quad (3.7)$$

Em um ponto de operação, considerando $\theta_0 = \hat{\theta}_0$ e assumindo que $V_0 = 1 pu$, $k_0 = 1,732$ e $k_1 = 0$, tem-se que (3.6) e (3.7) podem ser reescritas como:

$$\frac{\Delta\hat{\omega}}{\Delta V} = 0 \quad (3.8)$$

e,

$$\frac{\Delta\hat{\omega}}{\Delta\omega} = \frac{(1,732k_p)s + (1,732k_i)}{s^2 + (1,732k_p)s + (1,732k_i)} \quad (3.9)$$

De (3.8) pode-se concluir que as variações de amplitude na tensão do sistema não afetam a frequência angular detectada ($\Delta\hat{\omega}$) pelo q -PLL e de (3.9) pode-se concluir que somente as variações da frequência angular do sistema ($\Delta\omega$) afetam a frequência angular detectada ($\Delta\hat{\omega}$) pelo q -PLL.

3.3.1 Comportamento Dinâmico do q -PLL

Para calcular os ganhos do controlador PI, foi levado em consideração o comportamento dinâmico do modelo do q -PLL, utilizando a forma canônica da função de transferência de malha fechada de segundo grau dada por:

$$F(s) = \frac{2\xi\omega_n s + \omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (3.10)$$

Comparando (3.7) e (3.10) tem-se que a frequência natural não amortecida do circuito em (rad/s) é dada por $\omega_n = \sqrt{k_0 k_i}$ enquanto que o fator de amortecimento é igual a $\xi = \left(\frac{k_p k_0}{2\omega_n}\right) = \left(\frac{k_p}{2}\right) \sqrt{\frac{k_0}{k_i}}$, como o tempo de ação integral é dado pela relação $T_i = \frac{k_p}{k_i}$, logo o fator de amortecimento pode ser encontrado por $\xi = \frac{\sqrt{k_p k_0 T_i}}{2}$.

Desse modo, arbitrando um valor para ξ e para T_i é possível determinar analiticamente o valor de k_p conforme mostrado a seguir:

$$k_p = \left(\frac{4 \xi^2}{k_0 T_i}\right) \quad (3.11)$$

De (3.11), escolhendo o coeficiente de amortecimento (ξ) da função de transferência de malha fechada e o tempo da ação integral (T_i) igual a 0,707106 e 0,006 s, respectivamente, os ganhos do controlador devem ser $k_p = 192,45 \text{ rad.s}^{-1}/\text{vai}$ e $k_i = 32075,01 \text{ rad}/\text{vai}$ (Ogata, 1997), esse valores foram escolhidos para o PLL possuir uma boa resposta dinâmica e boa imunidade aos ruídos. A escolha desses parâmetros é crucial para o bom funcionamento do PLL, quanto mais rápida a resposta dinâmica pior a imunidade aos ruídos e quanto melhor a imunidade aos ruídos mais lenta a resposta dinâmica, a melhor opção é escolher um meio termo entre as duas situações.

Esses ganhos também poderiam ser facilmente encontrados a partir do gráfico do lugar das raízes mostrado na Figura 3.4, obtido com o pacote de simulação matemática SciLab (*Scientific Laboratory*) (SCILAB, 2005).

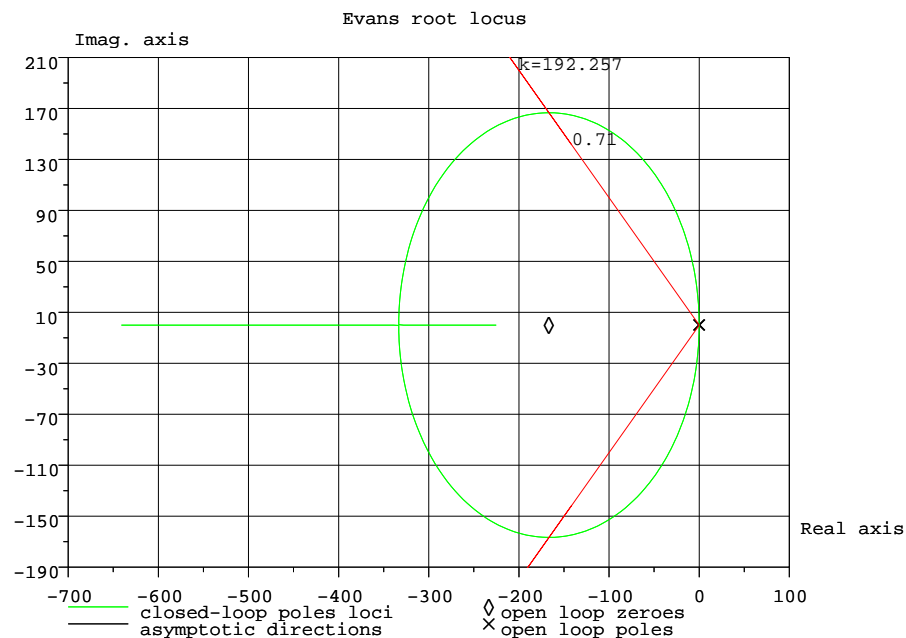


Figura 3.4: Lugar das Raízes do Modelo do q -PLL Linearizado.

3.3.2 Comportamento Estático

No caso do q -PLL, que foi modelado como um sistema de segunda ordem, seu comportamento estático pode ser avaliado a partir do cálculo do erro estático do circuito de sincronismo para entradas do tipo degrau, rampa e parábola respectivamente utilizando a função de transferência de malha aberta ($G(s)$) usada para obter (3.9):

$$G(s) = \frac{(1,732k_p)s + (1,732k_i)}{s^2} \quad (3.12)$$

Sendo que os resultados obtidos estão mostrados de forma resumida na tabela a seguir.

Tabela 3.1: Erro Estático do Modelo q -PLL Linearizado

Tipo de Entrada	Equação	Valores Calculados
U_{-1} (Degrau)	$\frac{1}{1+\lim_{s \rightarrow 0}(G(s))} = 0$	0
U_{-2} (Rampa)	$\frac{1}{\lim_{s \rightarrow 0}(sG(s))} = 0$	0
U_{-3} (Parábola)	$\frac{1}{\lim_{s \rightarrow 0}(s^2G(s))} = \frac{1}{k_i k_0} = \frac{1}{\omega_n^2}$	$1,8 \times 10^{-5}$

3.4 Modelo Discreto do q -PLL

Há vários métodos para converter um sistema contínuo já existente em um discreto ou digital, como: Transformação bilinear (Tustin), invariância ao impulso e invariância ao degrau, sendo este último o método adotado nesse trabalho. No entanto, a conversão do domínio s para o domínio z causa algumas distorções na localização dos pólos no plano complexo de tempo discreto (Ogata, 1994).

O método da invariância ao degrau que foi usado para converter (3.9) faz uso da resposta ao degrau do sistema contínuo. O processo de conversão consiste em expandir a resposta ao degrau em frações parciais e usando uma tabela de transformadas Z (Ogata, 1994) achar a representação discreta para cada termo. Então, multiplicando as respostas discretas pelo inverso da transformada z do degrau, $\left(\frac{1}{1-z^{-1}}\right)$, é possível determinar a função de transferência do sistema no domínio z .

Assim, (3.9) pode ser reescrita no domínio z como:

$$\frac{\Delta\hat{\omega}}{\Delta\omega} = \frac{A_1 z + A_2}{z^2 + (A_1 - 2)z + (A_2 + 1)} \quad (3.13)$$

onde,

$$A_1 = \left(\frac{k_0 T_s}{2}\right) [T_s k_i + 2k_p],$$

$$A_2 = \left(\frac{k_0 T_s}{2}\right) [T_s k_i - 2k_p],$$

e T_s é o período de amostragem do sistema em (s).

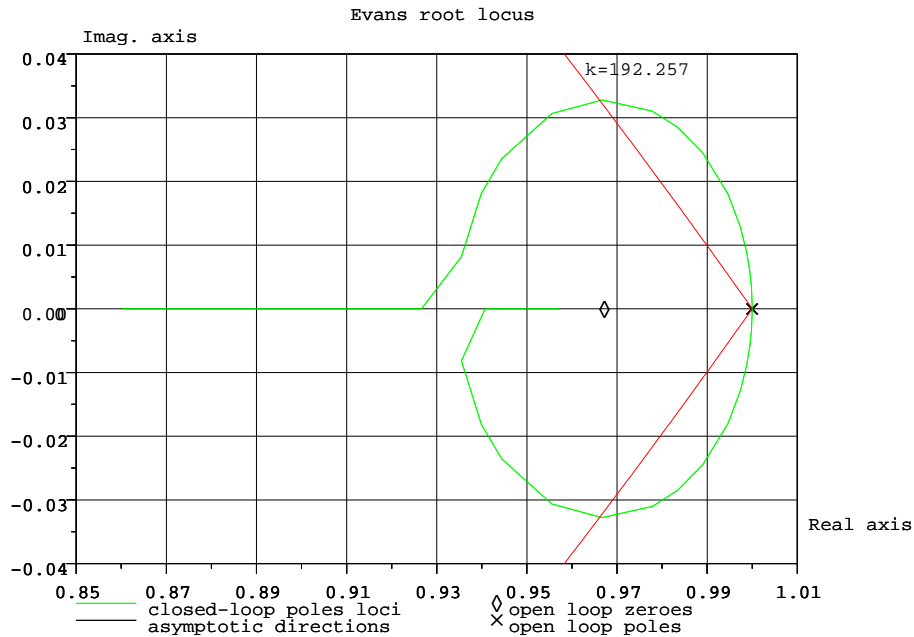


Figura 3.5: Lugar das Raízes do Modelo Discreto

3.4.1 Comportamento Dinâmico

Baseado em (3.13), se uma resposta similar à obtida para o modelo contínuo desenvolvido no capítulo anterior é desejada para o q -PLL digital, utilizando um período de amostragem de $T_s = 0,0002$ s estimado empiricamente, os ganhos que devem ser usados pode ser retirados do lugar das raízes mostrado na Figura 3.5, e são $k_p = 192,257 \text{ rad}\cdot\text{s}^{-1}/\text{vai}$ e $k_i = 32042,94 \text{ rad}/\text{vai}$. Substituindo os ganhos escolhidos em (3.13) obtém-se:

$$\frac{\Delta\hat{\omega}}{\Delta\omega} = \frac{0.06771z - 0.06549}{z^2 - 1.932z + 0.9345} \quad (3.14)$$

3.4.2 Comportamento Estático

A Tabela 3.2 mostra o comportamento do erro estático do modelo discreto do q -PLL para para entradas do tipo degrau, rampa e parábola, respectivamente, para a função

de transferência de malha aberta $G(s)$ que deu origem a (3.13).

$$G(s) = \frac{A_1 z + A_2}{(z - 1)^2} \quad (3.15)$$

Tabela 3.2: Erro Estático do Modelo q -PLL Discreto

Tipo de Entrada	Equação	Valores Calculados
U_{-1} (Degrau)	$\frac{1}{1 + \lim_{z \rightarrow 1} (G(z))} = 0$	0
U_{-2} (Rampa)	$\frac{1}{\lim_{z \rightarrow 1} \left(\frac{(z-1)}{T} G(z) \right)} = 0$	0
U_{-3} (Parábola)	$\frac{1}{\lim_{z \rightarrow 1} \left(\frac{(z-1)^2}{T^2} G(z) \right)} = \frac{1}{k_i k_0}$	$1,8 \times 10^{-5}$

Da comparação dos resultados apresentados na Tabela 3.1, para o modelo contínuo, com os resultados apresentados na Tabela 3.2, para o modelo discreto, pode-se concluir que os dois modelos possuem um comportamento estático bastante próximo. Essa característica é consequência dos ganhos k_0 e k_i do modelo discreto serem iguais aos do modelo contínuo devido ao período de amostragem T_s adotado para o primeiro.

3.4.3 Comparação dos Modelos do q -PLL Digital e Contínuo

A Figura 3.6 mostra a resposta ao degrau dos modelos do q -PLL no domínio contínuo e discreto dados por (3.9) e (3.14), respectivamente. A Figura 3.7 mostra em detalhe a resposta ao degrau dos modelos do q -PLL no domínio contínuo e discreto. Note que o comportamento dos dois modelos são muito próximos. A pequena diferença entre as duas respostas pode ser explicada pelo período de amostragem T_s finito adotado para o modelo discreto.

A Figura 3.8 mostra o resultado da simulação digital do q -PLL discreto quando a frequência angular da tensão de fase do sistema elétrico é variada em degrau de 60 Hz para 120 Hz quando $t = 0.033$ s. Note que a rampa que representa a frequência angular detectada do modelo tem uma dinâmica bastante rápida, demorando menos do que meio ciclo de 60 Hz para chegar ao regime permanente.

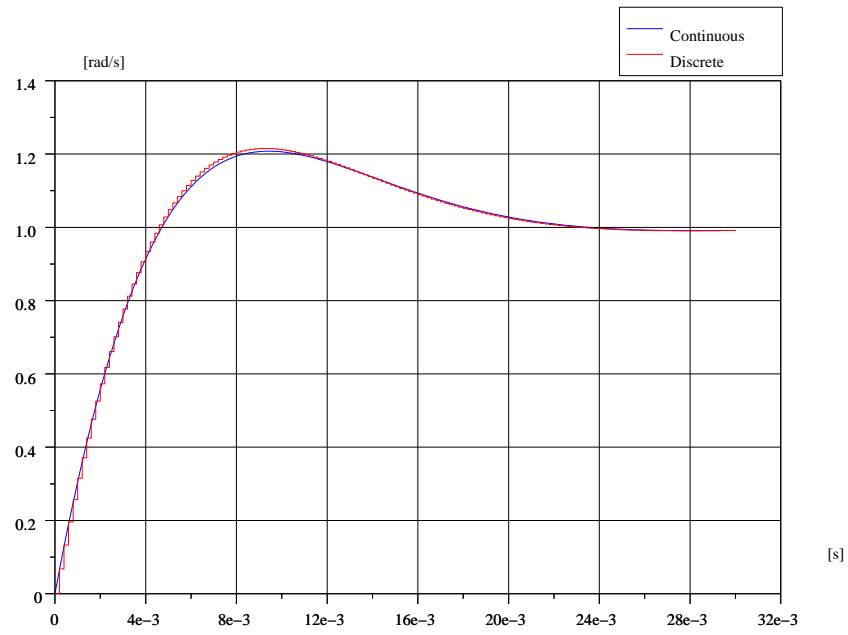


Figura 3.6: Resposta ao Degrau dos Modelos Linear e Digital do q -PLL.

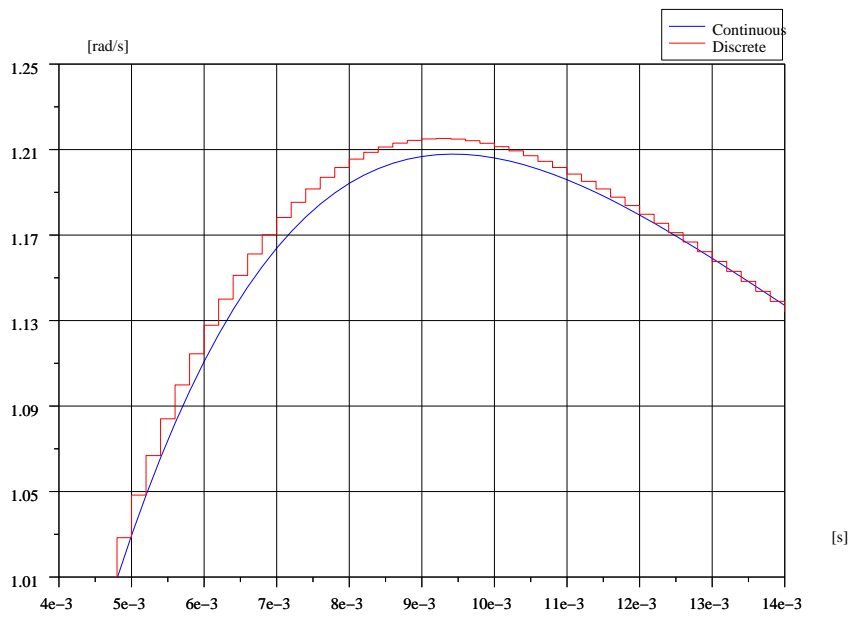


Figura 3.7: Detalhe da Resposta ao Degrau dos Modelos Linear e Digital do q -PLL.

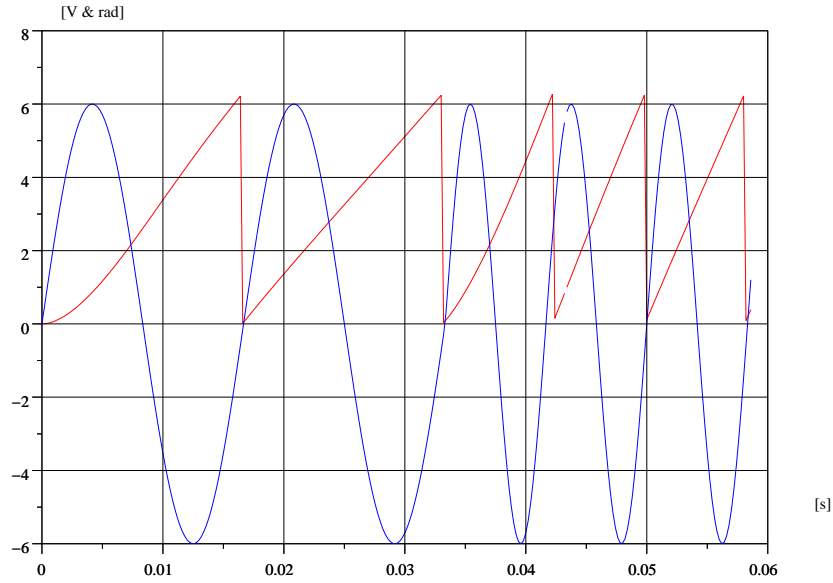


Figura 3.8: Tensão de Fase e Ângulo de Fase Detectado pelo q -PLL Digital.

3.4.4 Problemas com a Implementação do q -PLL

Na prática o modelo q -PLL apresenta problemas de funcionamento quando a tensão do sistema sofre variação ou mudança de valor de seu módulo, quando por exemplo ocorre a entrada ou saída de uma carga no sistema. Neste caso o q -PLL sai de seu ponto de operação, já que ele foi projetado para operar com uma tensão de 1 pu de amplitude em sua entrada.

Para resolver esse problema, neste trabalho, o sinal de erro do q -PLL foi normalizado, dividindo (3.3) pelo valor da tensão coletiva instantânea do sistema calculada por:

$$v_{\Sigma} = \sqrt{\frac{v_{\alpha}^2 + v_{\beta}^2}{3}} \quad (3.16)$$

Neste caso, considerando que as tensões instantâneas do sistema são senoidais, o valor de $v_{\Sigma} = V$ (pu). Esta normalização então garante uma tensão de entrada de 1 pu .

3.5 Simulação do q -PLL

Novamente neste capítulo foi utilizado o programa ATP/EMTP para validar o modelo discreto do q -PLL através de simulações digitais. O esquema utilizado para simulação está mostrado na Figura 3.9. O modelo digital para simulação no ATP está listado no apêndice B. Na Figura 3.10 é mostrada a resposta do modelo q -PLL.

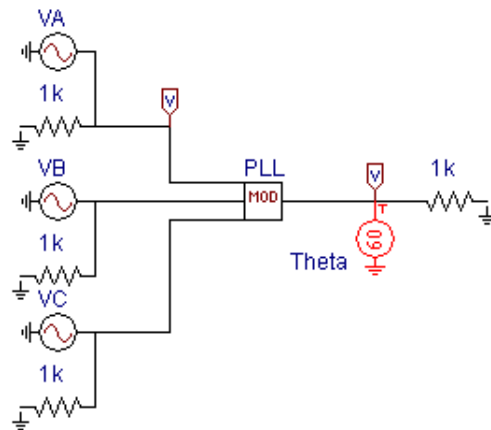


Figura 3.9: Esquema de Simulação no ATP

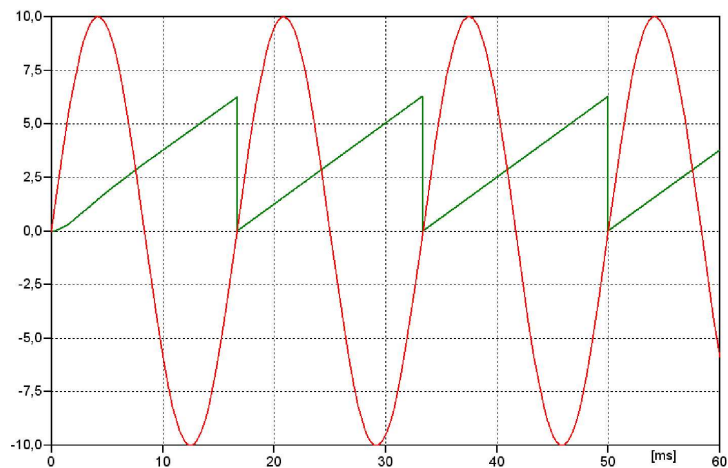


Figura 3.10: Simulação do q -PLL no ATP

3.6 Conclusões Parciais

Esta capítulo apresentou um modelo de um circuito de sincronismo (PLL) baseado na teoria das potências instantâneas. Foi usado um modelo de pequenos sinais para

projetar os ganhos do controlador PI do PLL analógico.

Em paralelo foi desenvolvido um modelo matemático discreto do circuito de sincronismo estudado. Foram usadas ferramentas de controle para projetar os ganhos dos controladores do PLL.

O circuito de sincronismo (PLL) apresentado apresentou bom desempenho. Os resultados obtidos possibilitam que ele seja usado para sincronizar os pulsos de disparo dos conversores estáticos do sistema fotovoltaico de geração de energia elétrica.

Capítulo 4

Implementação dos Controladores do Sistema de Geração Fotovoltaico

4.1 Introdução

Com objetivo de obter flexibilidade e rapidez na tarefa de implementar os controladores dos conversores CC-CC e CC-CA do sistema de geração fotovoltaico foi feita a opção pela utilização microprocessadores como plataforma de desenvolvimento.

Contudo, como o controle necessita de características de operação em “*tempo real*” a utilização de processadores de sinais digitais (DSP) foi adotada. Desse modo, todos os algoritmos de controle foram implementados em processadores DSP TMS320F243 da *Texas Instruments*.

4.2 Utilização do DSP

Processadores de sinais digitais (DSPs - *Digital Signal Processors*) podem ser divididos em duas famílias denominadas de processadores aritméticos de *ponto-fixo* e de *ponto-flutuante*. Na maioria dos casos, o uso de processadores de *ponto-flutuante* resulta em soluções técnicas melhores para controladores digitais. No entanto, se o custo é crítico, processadores de *ponto-fixo* podem ser uma opção barata se o desempenho do controlador digital não for drasticamente afetado (Instruments, 1992). Essas considerações nos levou a implementar o sistema de controle dos conversores em DSPs de *ponto-fixo* da Texas, o

TMS320F243. A Figura 4.1 mostra um *kit* de desenvolvimento DSK TMS320F243 (DSK - *Development Starter Kit*) totalmente projetado e montado nos laboratórios da UFJF e usado neste trabalho.

Além do custo inicial, existem vários outros problemas para se implementar controladores digitais em DSPs, como, por exemplo, os coeficientes de equações e ganhos dos controladores têm que ser representados em uma palavra de dados de tamanho limitado (*finito*), e ainda, os resultados de operações matemáticas têm que ser limitados em uma palavra de dados de tamanho finito e podem perder parte do resultado final. Esses problemas são conhecidos como erros de quantização e truncamento e podem causar oscilações e instabilidades.

Um outro problema é o estouro dos registradores (*overflow*). Operações matemáticas sucessivas, na maioria dos processadores, podem fazer o resultado de um cálculo ir do maior número positivo que pode ser representado para um número negativo (Instruments, 1992). Cada conversor CC-CA usa um DSP idêntico ao mostrado na Figura 4.1, os quais foram projetados e montados nos laboratórios de pesquisa da UFJF.

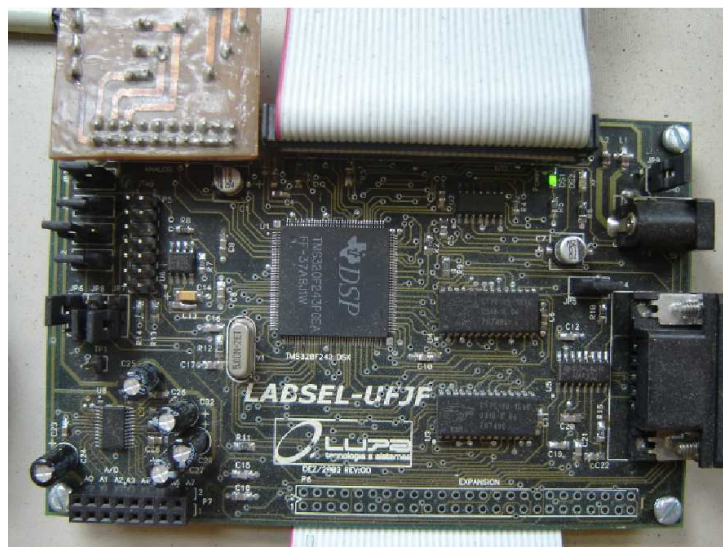


Figura 4.1: Plataforma DSP Desenvolvida para Implementação dos Controles do Sistema.

4.3 *Hardware do TMS320F243*

Na Tabela 4.1 são listadas as características do *DSP TMS320F243* utilizadas durante o projeto (Instruments, 1992):

Tabela 4.1: Características Básicas do *DSP TMS320F243* da *Texas Instruments*.

Nome do parâmetro	TMS320F243
MIPS ¹	20
Frequência (<i>MHz</i>)	20
RAM (Words)	544
Flash (Words)	8K
Boot loader available	Flash
Interface de memória externa	Sim
Canais PWM	8
10-bit A/D (Número de canais)	8
Tempo de conversão (<i>ns</i>)	900
Módulo CAN	Sim
Temporizadores (<i>Timers</i>)	2
Porta Serial	2

¹ *MIPS* - *Millions of Instructions Per Second*.

Na Tabela 4.2 estão listados os módulos internos do DSP usados na implementação digital do circuito de controle da Figura 2.20.

Tabela 4.2: Módulos Usados do DSP TMS320F243

Módulo interno	Função
Temporizador de uso geral	Geração da frequência de amostragem do controlador
Conversor A/D de 10 bits de resolução	Obtenção das amostras de tensão
Saídas Digitais	Saídas para disparos de chaves semicondutoras
Módulo CAN	Controle remoto do disparo das chaves
Canais PWM	Geração de disparo
Porta serial	Supervisão e controle dos conversores

Os canais do conversor A/D têm como entrada um sinal de 0 a 5 Volts, o que leva à necessidade de circuitos de condicionamento, visto que as tensões e correntes senoidais têm excursões positivas e negativas, o que inviabiliza a conexão direta aos DSPs. As saídas digitais são compatíveis com TTL, como os *drivers* dos conversores são alimentados com 15 V, é necessário condicionar todos sinais de tensão e corrente alternada.

4.4 Rede *CAN*

A rede *CAN* (*Control Area Network*) é um barramento serial desenvolvido para interligar dispositivos “inteligentes” como sensores e atuadores de um sistema (BOSCH,

2005), com capacidade multi-mestre, possibilitando que todos os nós da rede transmitam dados e vários nós possam requisitar o barramento simultaneamente, com características de transmissão em tempo real conforme padronizado pela ISO (*Internationally standardized by the international Standizaton Organization*) na norma ISO 11898 e cobre as duas primeiras camadas do modelo de referência ISO/OSI.

Nas redes *CAN* não há endereçamento do remetente nem do destinatário, mas ao invés, as mensagens são classificadas por prioridade. O transmissor envia a mensagem para todos os nós da rede *CAN* (*broadcasting*). Cada nó decide, baseado no identificador da mensagem recebida, se deve ou não processar a mensagem. Cada mensagem *CAN* transmite de 0 a 8 *bytes* de informações, maiores quantidades de dados podem ser transmitidas usando a segmentação dos dados. A taxa de transmissão máxima é de 1 *Mbit/s* em redes de até 40 *m*, para redes de até 500m pode se conseguir taxas de até 125 *kbit/s* e para transmissões acima de 1 *km* de distância uma taxa de 50 *kbit* pode ser conseguida. As redes *CAN* podem ser usadas em sistemas de comunicação embarcada com microcontroladores e também para sistemas de comunicação abertos de dispositivos inteligentes.

O barramento *CAN* serial, originalmente desenvolvido para uso em automóveis, é cada vez mais usado nas redes de campo industriais já que ambos os casos as características desejadas são: baixo custo, a habilidade de funcionar em ambientes com ruídos eletromagnéticos, alto grau de capacidade de tempo real e fácil uso.

Como o sistema de controle mostrado na Figura 2.21 para os conversores CC-CA é distribuído entre três blocos *Models* é necessária a comunicação em "*tempo real*" entre os DSPs responsáveis por realizar as tarefas de cada bloco. A interligação entre o DSP de controle e os DSPs de disparo usados no controle dos conversores CC-CA foi feita utilizando uma rede *CAN* cuja estrutura é mostrada na Figura 4.2.

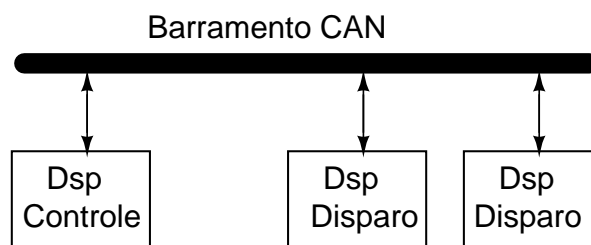


Figura 4.2: Diagrama de Ligação da Rede CAN.

4.5 *Emulador* do DSP *TMS320F243* da Texas Instruments

Para atender às características de desempenho desejada para o sistema de geração fotovoltaico os algoritmos de controle dos DSPs foram implementados em linguagem Assembly e linguagem C utilizando a plataforma de desenvolvimento *CodeComposer* da Texas Instruments (Apêndice C).

Porém antes dos testes finais de campo esses algoritmos foram testados com auxílio de um *Emulador* desenvolvido para o DSP *TMS320F243* da Texas Instruments.

Os DSPs da Texas Instruments têm como plataforma básica de desenvolvimento a ferramenta *CodeComposer*. Essa é uma ferramenta proprietária, sem flexibilidade e não integrada a nenhum *software* de simulação de circuitos. Com o objetivo de depurar os programas desenvolvidos, uma vez que o *CodeComposer* apresentava muitas limitações foi desenvolvido um programa *Emulador* para o DSP.

O programa *Emulador* desenvolvido foi escrito em linguagem C. Ele lê o arquivo objeto do programa no formato COFF, idêntico ao carregado no DSP, e emula o funcionamento do mesmo. Essa estratégia integra os programas do DSP ao simulador da planta da usina permitindo realizar simulações digitais da planta (painéis fotovoltaicos, conversores CC-CC e CC-CA e sistema CA) e dos circuitos digitais de controle (circuitos lógico de disparo, circuitos de sincronismo (PLL) e algoritmos de rastreamento de máxima potência (MPPT).

Desse modo, os controladores podem ser exaustivamente testados sem risco de danificar os equipamentos devido a erros nos códigos dos algoritmos de controle. Na Tabela 4.3 estão listadas as características que foram implementadas no *Emulador*.

Tabela 4.3: Características do *Emulador*.

Item	Emulação de evento
1	Interrupções
2	Timers
3	Conversores A/D
4	Porta serial
5	Rede CAN

Para facilitar a tarefa de testar os softwares escritos para os DSPs no programa *Emulador* foi desenvolvida uma interface de depuração (“*debug*”) onde podem ser vistos

- **dsp_stop**: Finaliza instância do objeto DSP e descarrega memória.

A função **dsp_start** inicializa um objeto DSP e carrega o programa na memória de programa, preparando o objeto DSP para utilização, a função **dsp_step** executa um ciclo de operação do programa carregado no objeto DSP, devendo ser chamada continuamente para a execução do programa, sendo as chamadas intercaladas com o programa de simulação para atualização das entradas e saídas e a função **dsp_stop** libera a memória alocada pelo objeto DSP e só deve ser chamada no final do programa onde o objeto DSP não for mais necessário. As entradas e saídas dos módulos emulados podem ser acessadas diretamente entre as chamadas da função **dsp_step**. As funções devem ser encapsuladas em um modelo compatível com o simulador de circuitos utilizado, onde as entradas serão do tipo nó de tensão ou corrente por uma fonte, e as saídas serão fontes de corrente ou tensão controladas pelo *Emulador*.

4.6 Simulações das Partes do Sistema de Geração Fotovoltaicos com o *Emulador*

As simulações dos controladores do sistema de geração fotovoltaico foram feitas seguindo os mesmos procedimentos adotados no Capítulo 2 para a simulação com o programa ATP. Como não foi possível a obtenção do código fonte do ATP para uso junto ao *Emulador*, o *Emulador* foi utilizado junto a um simulador de circuitos de modelos discretos de acompanhamento que utiliza o Método Nodal Modificado (MNM) junto com a integração bilinear(trapezoidal) para resolução das equações do circuito (Raghuram, 1989) e (Ogrodzki, 1994). Contudo, diferentemente do programa ATP onde os controles foram implementados no ambiente *Models*, os controles foram substituídos pelos algoritmos carregados no *Emulador* do DSP.

4.6.1 Painel Solar Fotovoltaico

O modelo do painel fotovoltaico usado foi testado conforme esquema mostrado na Figura 4.4.

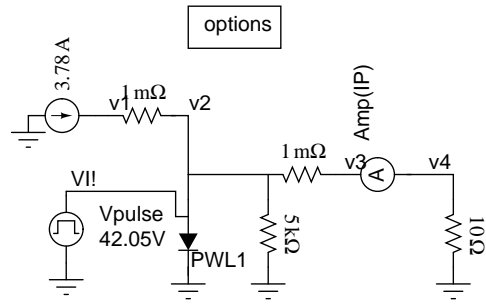


Figura 4.4: Esquema de Simulação do Painel Fotovoltaico.

Na Figura 4.5 está mostrada a curva característica corrente *versus* tensão enquanto que na Figura 4.6 é mostrada a relação entre potência e tensão nos terminais do painel solar fotovoltaico.

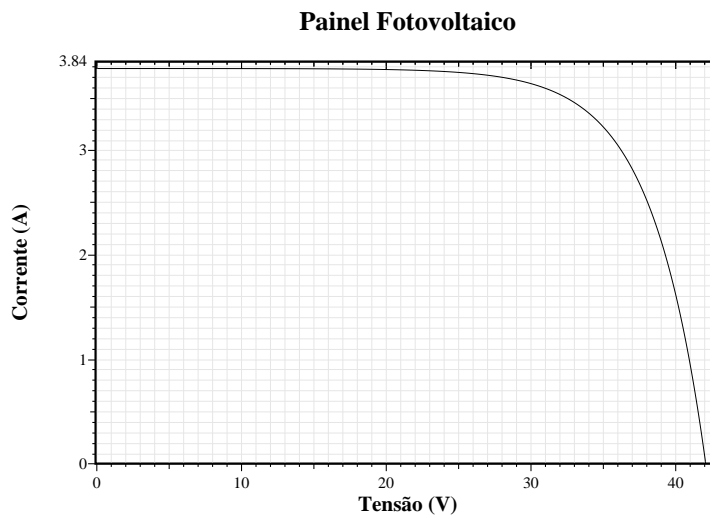


Figura 4.5: Característica Corrente x Tensão nos Terminais do Painel Solar Fotovoltaico.

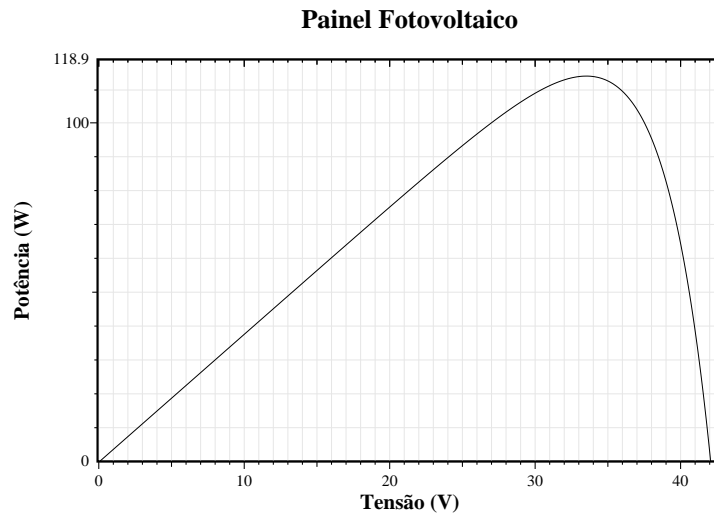


Figura 4.6: Característica Potência x Tensão nos Terminais do Painel Solar Fotovoltaico.

4.6.2 Conversores Estáticos CC-CC e MPPT

Nessa simulação foi utilizado o *Emulador* do TMS320F243 rodando o programa do DSP e o modelo do painel descrito na Seção 2.2.1. Foram usados três painéis conectados em série para o teste do algoritmo de MPPT. O esquema de simulação utilizando o *Emulador* é mostrado na Figura 4.7.

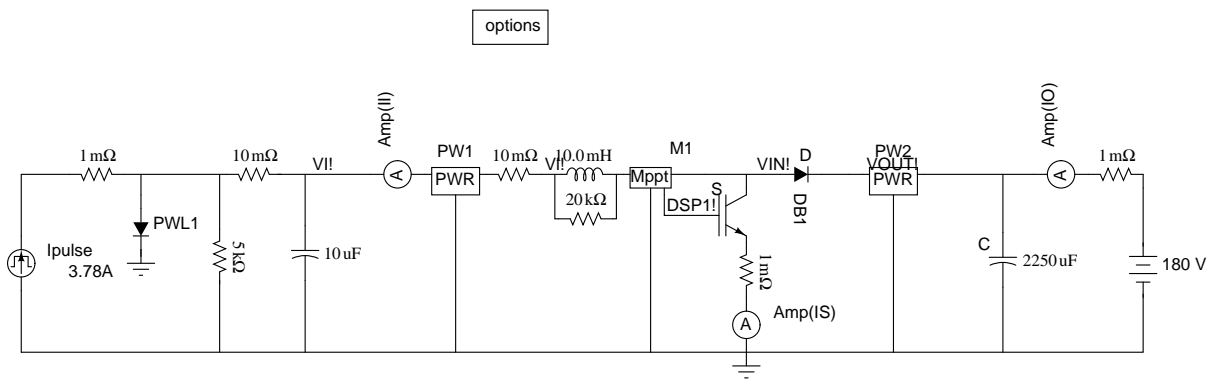


Figura 4.7: Esquema de Simulação do Conversor CC-CC com MPPT Usando o *Emulador*.

Nas Figuras 4.8 (a) e (b) são mostradas as tensões de entrada e de saída simuladas e emuladas, respectivamente. Pode-se notar pequenas diferenças entre os valores obtidos com o simulador e com *Emulador*, as quais podem ser explicadas pelos erros de quantização do conversor A/D e pela taxa de amostragem do DSP.

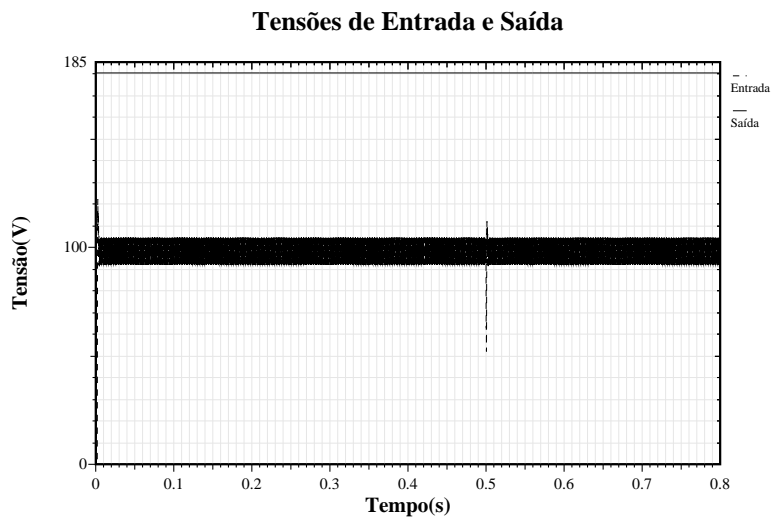
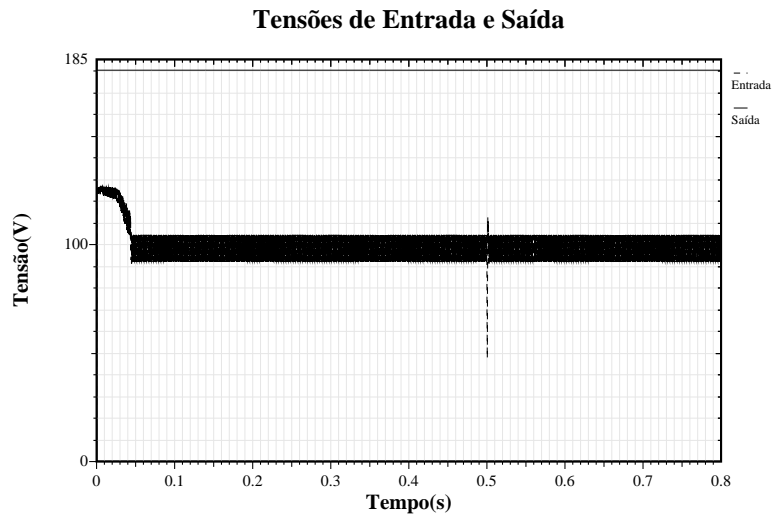
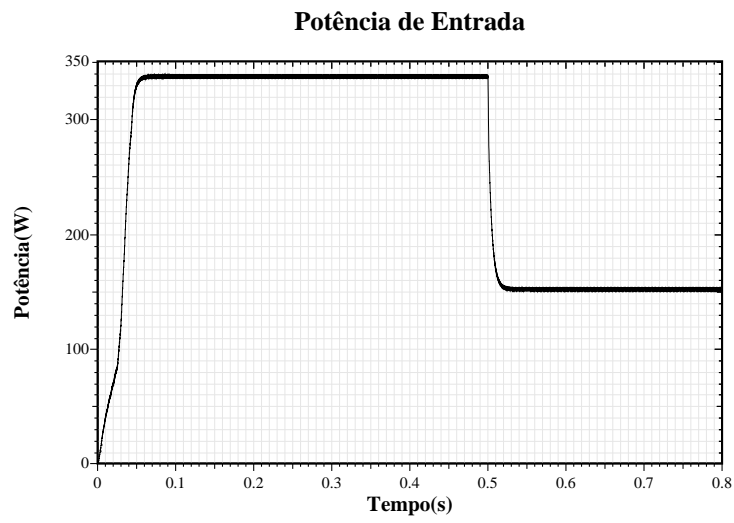
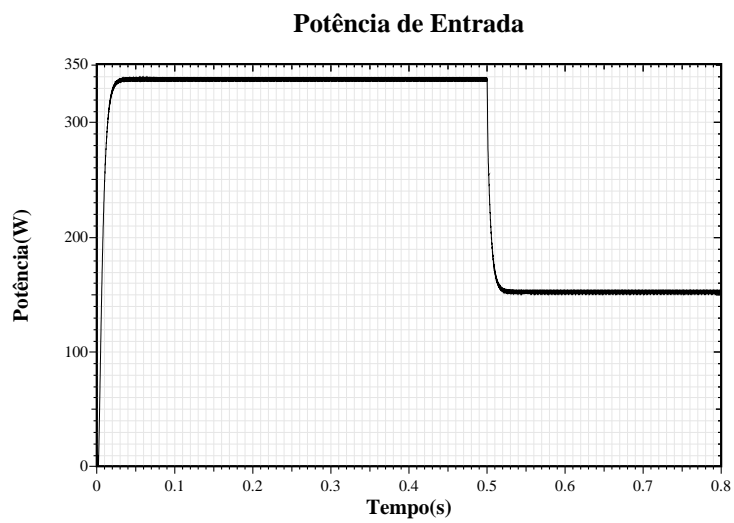


Figura 4.8: Tensões de Entrada e de Saída no Conversor: (a) Simulador; (b) *Emulador*.

Nas Figuras 4.9 (a) e (b) são mostradas as curvas da potência de entrada do conversor CC-CC obtidas com o algoritmo de MPPT. Nas Figuras 4.10 (a) e (b) são mostradas as curvas potência de saída fornecida pelo conversor CC-CC obtidas também com o simulador e com o *Emulador*, respectivamente.

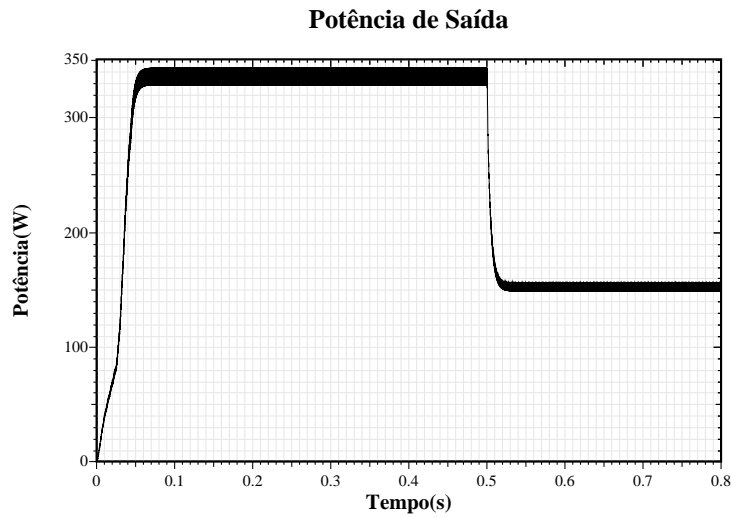


(a)

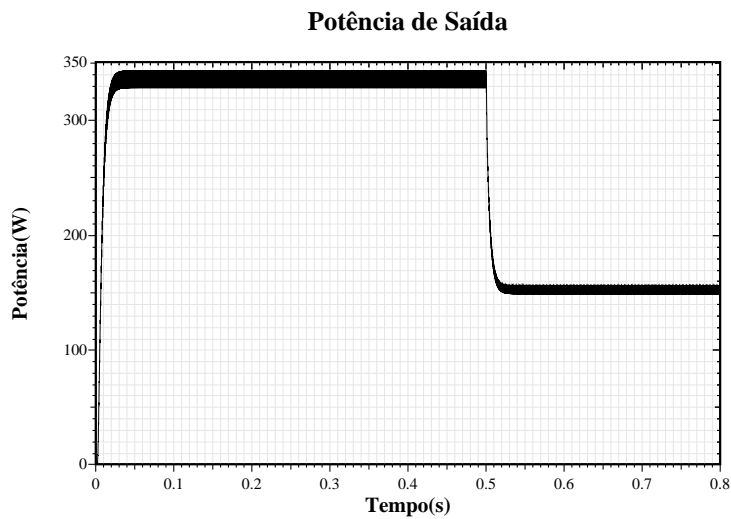


(b)

Figura 4.9: Potência de Entrada Rastreada Pelo Conversor: (a) Simulador; (b) *Emulador*.



(a)



(b)

Figura 4.10: Potência de Saída Fornecida Pelo Conversor: (a) Simulador; (b) *Emulador*.

Como pode ser observado nas Figuras 4.9 (a) e (b) e nas Figuras 4.10 (a) e (b) das potências de entrada e saída do conversor CC-CC apresentam um comportamento um pouco diferente. Essa característica é devida aos pequenos atrasos de conversão e da amostragem no *Emulador* do DSP. Essas pequenas diferenças já eram esperadas no processo de digitalização, e são devidas as limitações do processador de *ponto-fixa* adotado.

4.6.3 Circuito de Sincronismo (PLL)

O circuito de sincronismo discutido em detalhes no Capítulo 3 foi implementado no programa *Emulador* e testado de acordo com o esquema da Figura 4.11. O resultado da simulação está mostrado na Figura 4.12. Nessa figura pode-se observar o funcionamento do PLL.

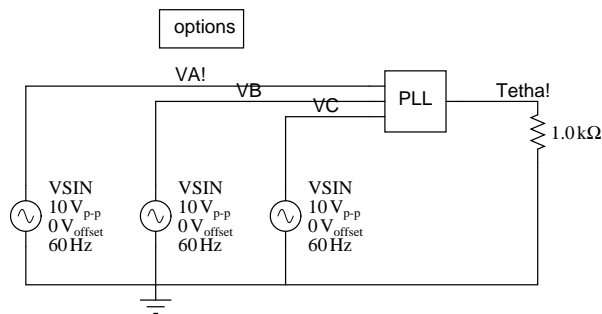


Figura 4.11: Esquema de Simulação do PLL.

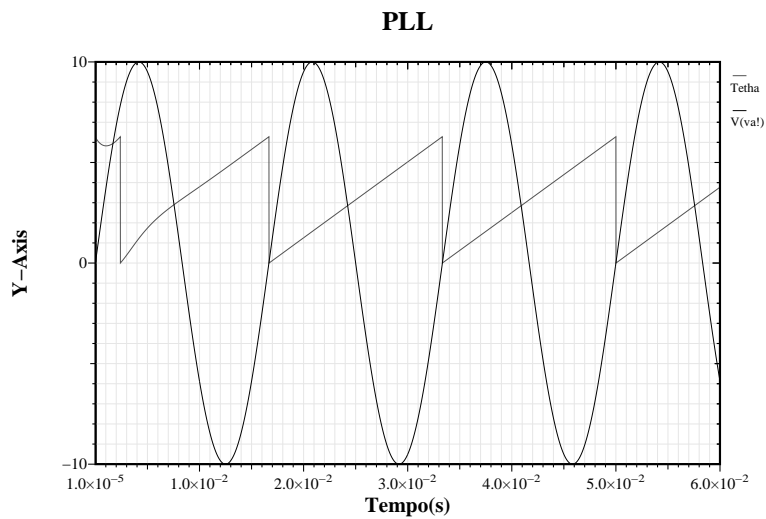


Figura 4.12: Tensão da Fase 'a' e Ângulo Detectado.

4.6.4 Conversores Estáticos CC-CA

O esquema utilizado na simulação dos conversores CC-CA utilizando o *Emulador* é mostrado na Figura 4.13. Neste caso observar-se que no esquema de simulação utilizando o *Emulador* não existe a conexão do módulo de controle com os módulos de

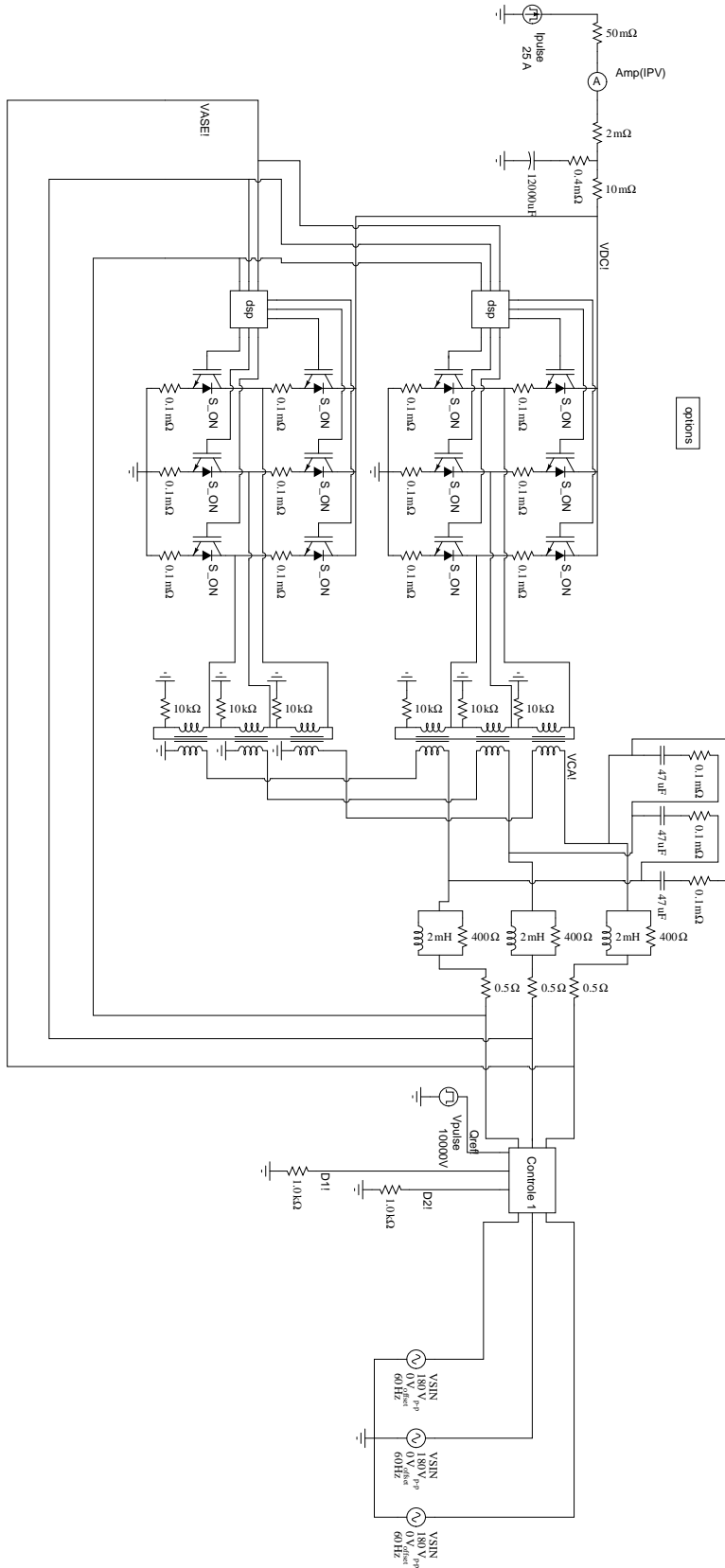
disparo, a comunicação entre os módulos é feita pela emulação da rede CAN pelo programa *Emulador*.

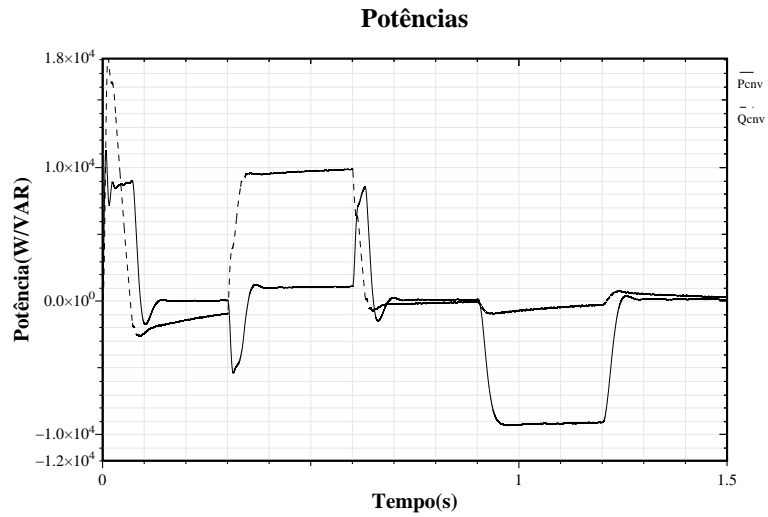
As Figuras 4.14 (a) e (b) mostram as potências ativa e imaginária instantâneas nos terminais dos conversores CC-CA. Os dois resultados apresentam diferenças em relação ao tempo de resposta do controle que é mais lento no *Emulador* em comparação com o resultado obtido com o simulador de circuitos devido à amostragem do controle implementado no DSP.

Nas Figuras 4.15 (a) e (b) são mostradas a tensão na fase “a” na saída do transformador de acoplamento dos conversores CC-CA obtidas com o simulador e com o programa *Emulador*, respectivamente. Note que a variação dessa tensão é diretamente proporcional à variação de potência dos conversores.

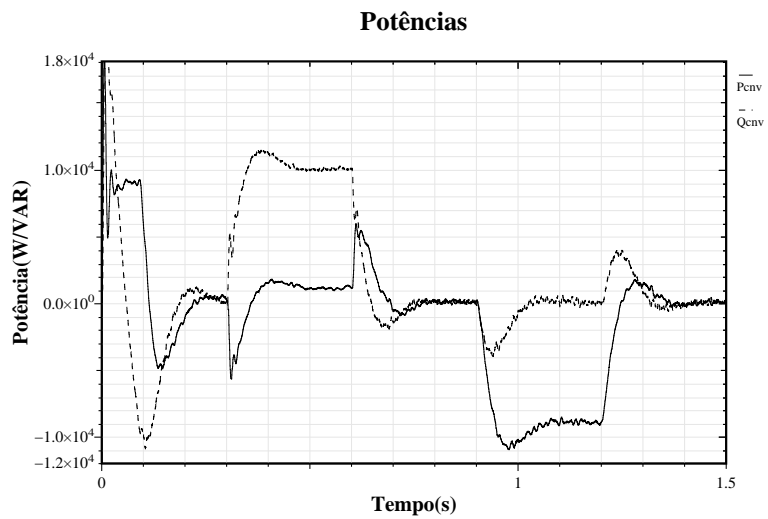
Nas Figuras 4.16(a) e (b) são mostradas as corrente na fase “a” enquanto que nas Figuras 4.17(a) e (b) são mostradas a tensão no elo CC dos conversores obtidas com o programa de simulação e o programa *Emulador*, respectivamente. Pode se verificar nas simulações que apesar das diferenças dos resultados devida as limitações do processador de *ponto-fixe* adotado e da digitalização dos algoritmos, o controle funciona como esperado.

Figura 4.13: Esquema de Simulação do Conversor CC-CA Usando o Emulador.



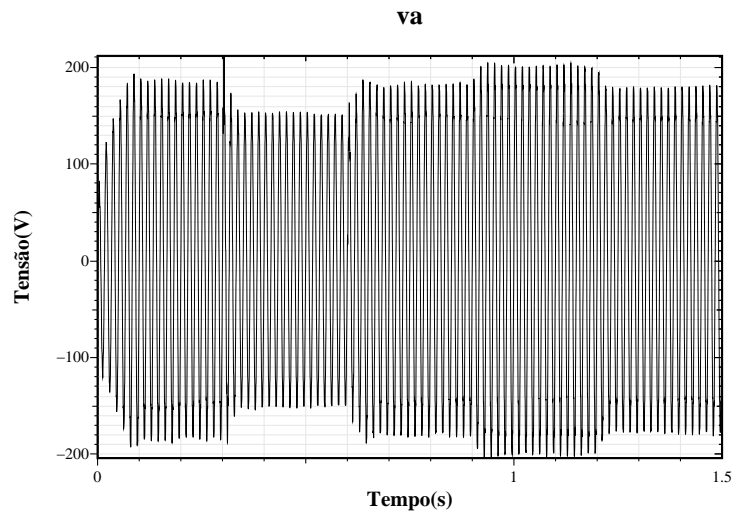


(a)

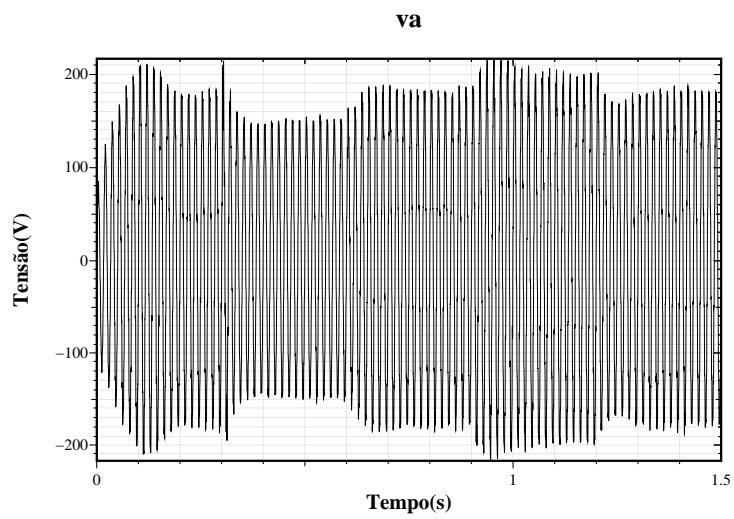


(b)

Figura 4.14: Potências nos Terminais dos Conversores CC-CA: (a) Simulador; (b) *Emulador*

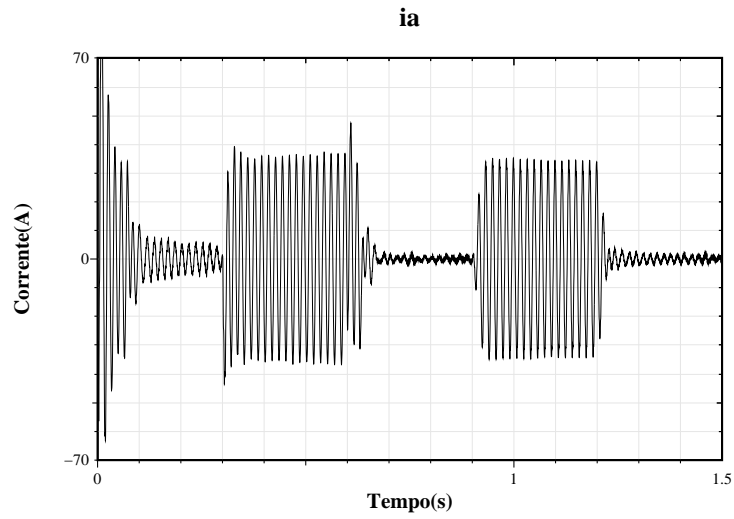


(a)

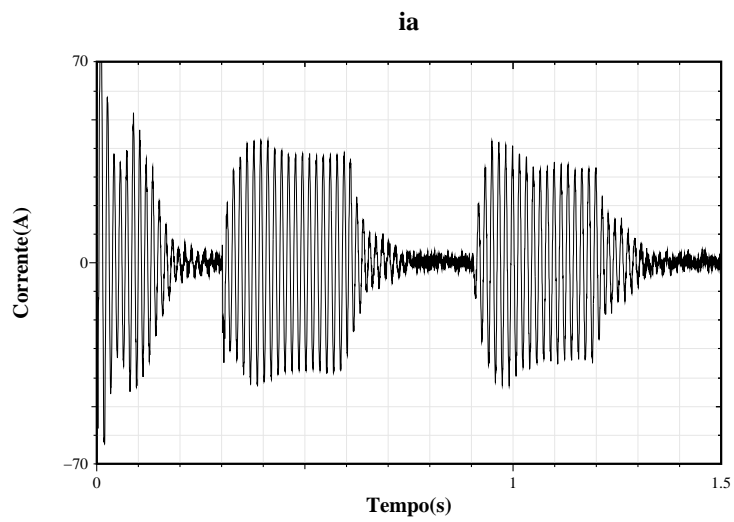


(b)

Figura 4.15: Tensão v_a : (a) Simulador; (b) *Emulador*.

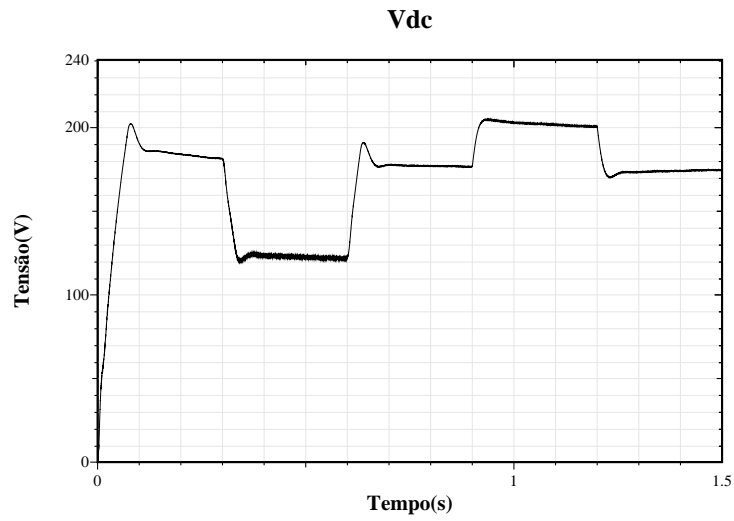


(a)

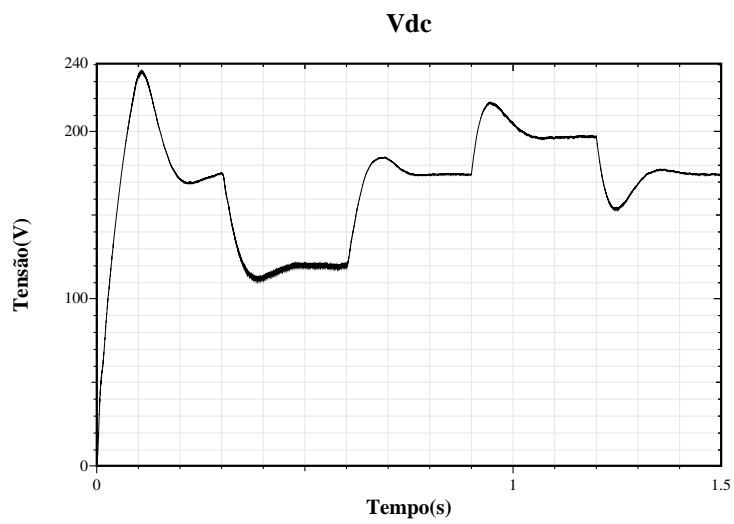


(b)

Figura 4.16: Corrente i_a : (a) Simulador; (b) *Emulador*.



(a)



(b)

Figura 4.17: Tensão do Elo CC: (a) Simulador; (b) *Emulador*.

4.7 Hardware

4.7.1 Implementação do Controlador do Conversor CC-CC

A implementação do algoritmo de controle do conversor CC-CC foi feita em um *DSP TMS320F243* onde foram utilizadas três entradas A/D de tensão e uma saída digital condicionada. O algoritmo implementado manda informações sobre as variáveis do sistema através de uma interface *RS232*.

Circuitos de Condicionamento CC-CC

O circuito de condicionamento consiste em 3 partes: (i) condicionamento da tensão e corrente de entrada, (ii) condicionamento do pulso de disparo de saída e (iii) alimentação do *driver* de controle do conversor CC-CC.

O esquema do circuito de medição de tensão do elo CC está mostrado na Figura 4.18 e sua foto na Figura 4.19. Ele é composto de um sensor de efeito Hall de tensão *LV 25-P* que é conectado ao barramento do sistema elétrico por uma resistência de 33 k Ω . Essa resistência garante um fundo de escala de 300 V. A saída do sensor *LV 25-P* é ligada a um amplificador operacional em configuração conversor de corrente em tensão e em seguida passa por um amplificador operacional em configuração somador inversor, onde é somado com uma tensão de referência CC (*offset*) usada para compensar a queda de tensão na saída do circuito devido ao diodo em série do circuito de proteção do DSP.

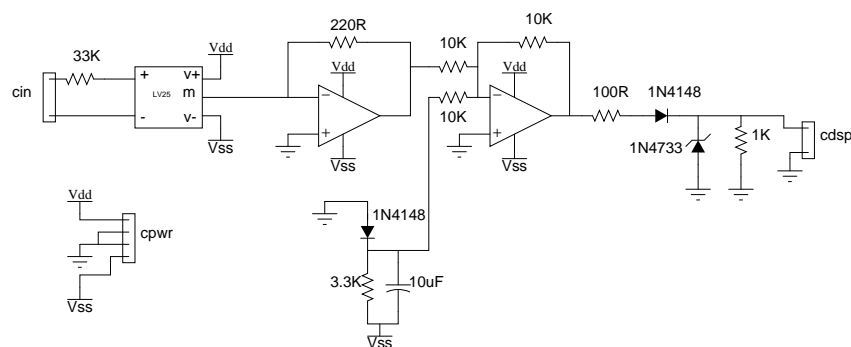


Figura 4.18: Circuito de Medição da Tensão do Elo CC.

O circuito de medição da tensão e da corrente do painel está na mostrado na Figura 4.20 e sua foto mostrada na Figura 4.21. Este circuito é ligado diretamente aos sensores de efeito Hall *LV 25-P*, de tensão, e *LA 55-P*, de corrente, conectados nos terminais de entrada do conversor CC-CC. Os sensores são ligados a um filtro *Bessel* passa-baixa, de

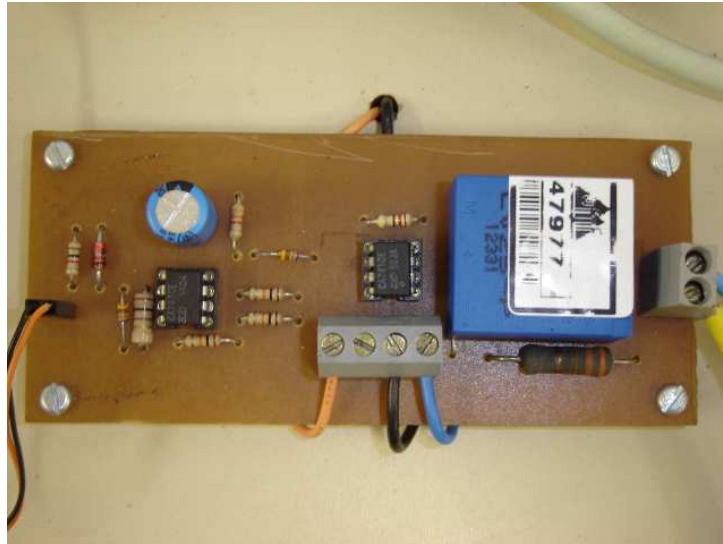


Figura 4.19: Foto do Circuito de Medição de Tensão do Elo CC.

segunda ordem, implementado com um amplificador operacional na configuração *Sallen-Key* modificado (Smith, 1999) e com uma frequência de corte de 1840 Hz . Os sinais de saída desses filtros passam por um amplificador inversor onde são ajustados os ganhos. Por fim os sinais de corrente e de tensão passam por um amplificador operacional em configuração somador inversor onde são somadas tensões de referência CC (*Offset*) para compensar a queda de tensão de saída devido ao circuito de proteção do DSP.

Na mesma placa ainda se encontram um *buffer* de coletor aberto *SN7406* usado para condicionar o sinal TTL de disparo do DSP aos 15 Volts necessários para acionar o “*driver*” *SKHI 20op* da Semikron. No circuito de saída do *buffer* tem-se resistores de *pullup* conectados no barramento de 15 Volts CC. Ainda tem-se nessa placa um circuito *buffer* como entrada do sinal de erro proveniente do *drive SKHI 20op* cuja saída tem um resistor de *pulldown* de $10\text{ k}\Omega$ que forma um divisor de tensão com resistor interno do *SKHI 20 op* de $22\text{ k}\Omega$ com objetivo de ajustar o sinal de erro do “*drive*” ao nível TTL. A alimentação do *SKHI 20op* também é feita por este circuito.

4.7.2 Implementação do Controlador do Conversor CC-CA

A implementação do algoritmo foi feita em três DSPs TMS320F243, um de controle e dois de geração dos sinais de disparo do inversor. No DSP de controle foram utilizadas sete entradas do conversor A/D de tensão. Já nos DSPs de disparo foram utilizadas três entradas do conversor A/D, seis saídas digitais e três entradas digitais (TTL) condicionadas.

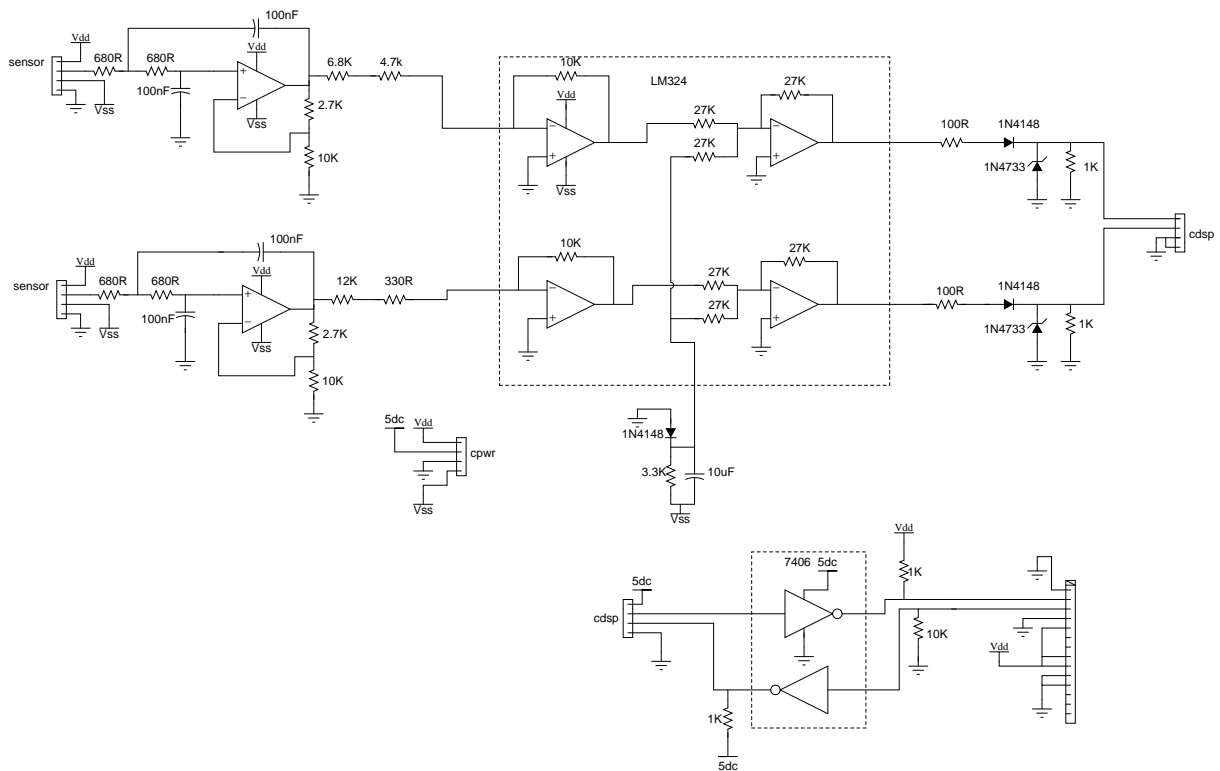


Figura 4.20: Circuito de Medição da Tensão e Corrente do Painel.

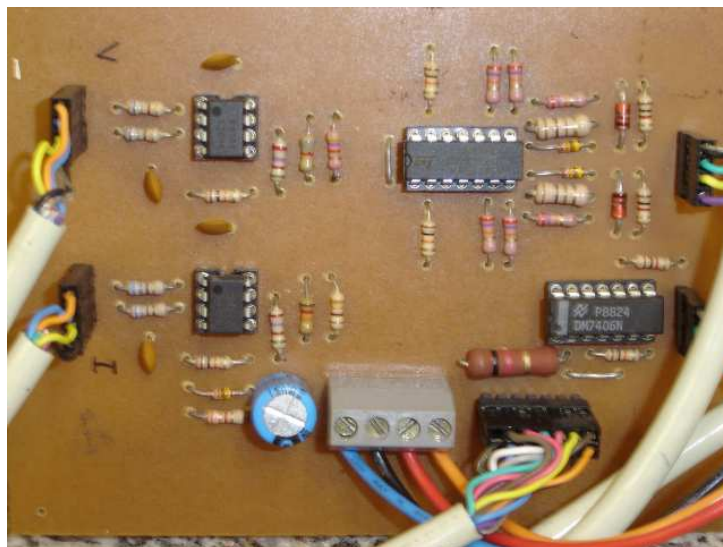


Figura 4.21: Foto do Circuito de Medição da Tensão e Corrente do Painel.

O algoritmo implementado envia e recebe informações sobre as variáveis do sistema através de uma interface *RS232*, configurada conforme a Tabela 5.1. A comunicação entre os três DSPs é feita através de uma rede CAN.

Circuitos de Condicionamento CC-CA

Os conversores CC-CC e CC-CA projetados para a utilização no sistema de geração disperso fotovoltaico são baseados em chaves semicondutoras IGBTs. Esses interruptores de potência permitem controlar tanto o instante de entrada em condução quanto o instante de corte através de sinais elétricos aplicados em seus terminais de *gate*.

Contudo para conseguir executar tal tarefa os sinais de disparo gerados pelo controlador implementado no circuito DSP mostrado na Figura 2.20 devem ser condicionados antes de serem enviados para os *drives* de disparo dos interruptores.

A Figura 4.22 mostra o circuito condicionador para a medição das tensões trifásicas instantâneas do sistema. Ele consiste de três circuitos idênticos, um por fase, composto de um sensor de efeito Hall de tensão *LV25-P* ligado a um amplificador operacional, na configuração de conversor de corrente-tensão, cuja saída alimenta um amplificador inversor somador que ajusta o sinal amostrado no nível exigido pelo conversor analógico digital do DSP *TMS320F243*. Esses sinais são enviados para os DSPs que controlam os sinais de disparo de cada conversor CC-CA. Nesses DSPs eles são usados para a determinação dos ângulos de chaveamento dos conversores estáticos. Dentro dos dois DSPs de disparo dos conversores CC-CA estão também implementados dois circuitos de sincronismo (PLL) independentes. A Figura 4.23 mostra o protótipo do circuito de condicionamento para a medição das tensões trifásicas instantâneas do sistema.

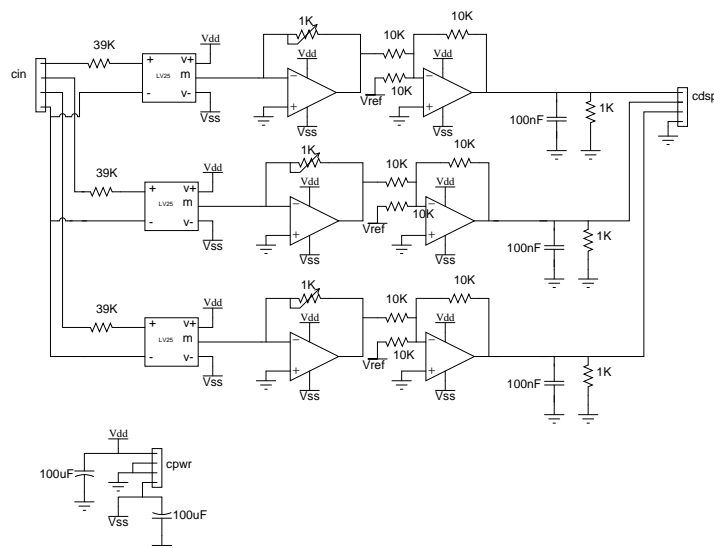


Figura 4.22: Circuito de condicionamento das Tensões Trifásicas do Sistema.

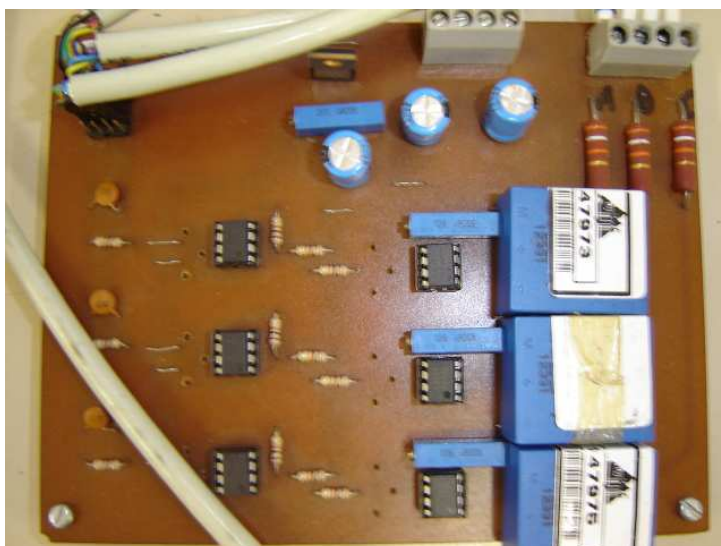


Figura 4.23: Detalhe dos Circuitos de Condicionamento das Tensões Trifásicas do Sistema.

A Figura 4.24 mostra o circuito de proteção contra tensões negativas e contra sobre-tensão na entrada dos conversores A/D interno dos DSPs de controle dos conversores. Esse circuito é necessário já que a entrada do DSP só suporta excursões da tensão de entrada na faixa compreendida entre 0 e $5 V_{CC}$. Qualquer tensão fora dessa faixa pode ocasionar queima do DSP.

A proteção contra tensões negativas é feita por um diodo $1N4148$ conectado em série com a porta de entrada. Já a proteção contra sobre-tensão por um diodo zener $1N4733$ de 5,1 Volts. Na Figura 4.25 está mostrada a foto do protótipo montado e usado nos testes.

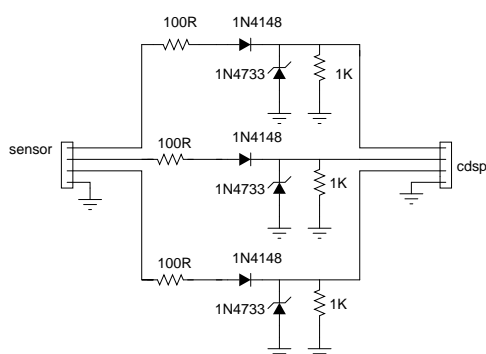


Figura 4.24: Circuito de Proteção do Conversor A/D do DSP de Disparo.

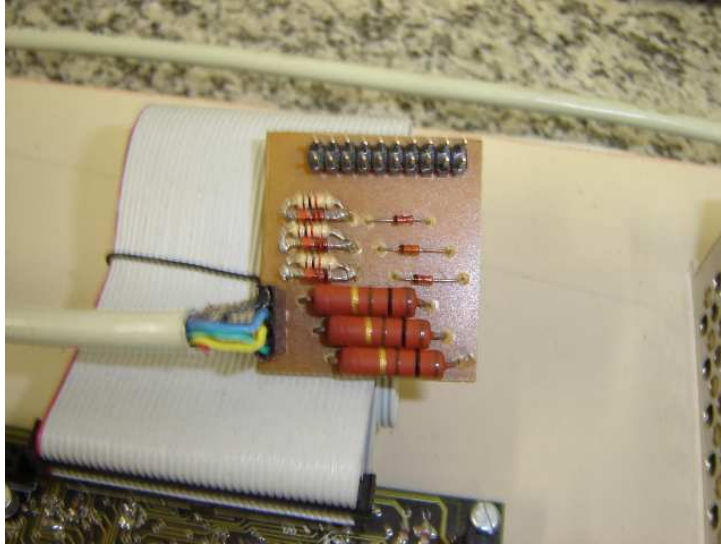


Figura 4.25: Foto do Circuito de Proteção do Conversor A/D do DSP de Disparo.

A Figura 4.26 mostra o circuito responsável por ajustar os níveis de tensão dos sinais digitais gerados pelos DSPs e usados para disparar e cortar os IGBTs dos conversores CC-CA. O circuito é composto por dois conjuntos de *buffers* TTL. O primeiro conjunto de *buffer* é do tipo inversor com coletor aberto *SN7406*. Ele permitir o acionamento do *driver SKHI 22A* da Semikron usado para acionar, com um sinal de $15 V_{CC}$, os IGBTs dos conversores estáticos CC-CA. Cada *buffer* têm um resistor de *pullup* de $1\text{ k}\Omega$ conectado em sua saída.

O segundo conjunto de *buffers* é do tipo inversor e é formado por um resistor de *pulldown* de $4,7\text{ k}\Omega$ e a resistência interna do *drive SKHI 20A* ($10\text{ k}\Omega$). Esse conjunto condiciona, através de um divisor de tensão, o sinal de erro dos *drives* ao um nível compatível TTL. A supervisão desses sinais permite também monitorar os erros enviados pelo *driver SKHI 22A* quando sobre-correntes e sobre-tensões ocorrem nos IGBTs do conversor.

Cada placa recebe os sinais de disparo enviados por um dos DSPs e os envia para o conversor correspondente. Nesses módulos estão montados um circuito *transceiver SN65HVD251* responsável pela interface dos sinais TTL do DSP e os sinais da rede CAN. A Figura 4.27 mostra o protótipo do circuito de condicionamento.

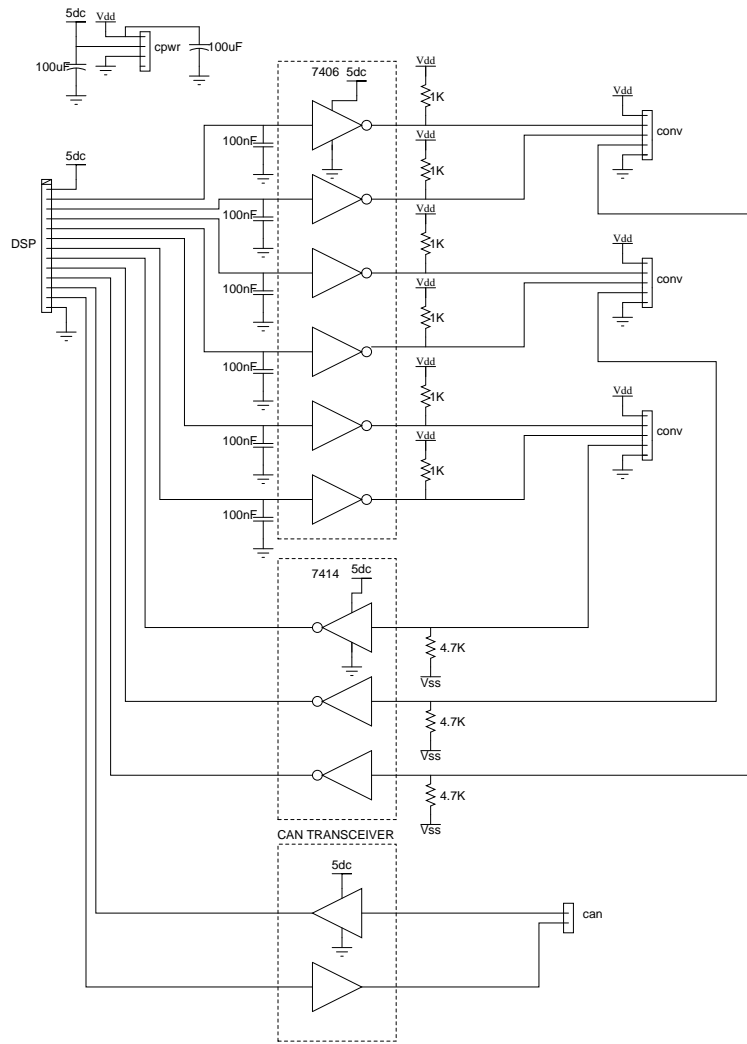


Figura 4.26: Circuito de Disparo dos Conversores CC-CA.

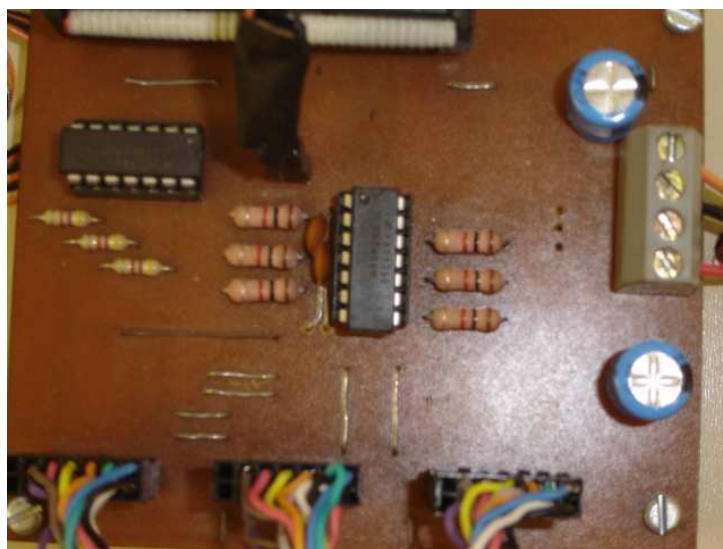


Figura 4.27: Foto do Circuito de Disparo dos Conversores CC-CA.

A Figura 4.28 mostra o detalhe do circuito responsável por conectar o DSP à rede CAN e alimentar o DSP de controle da malha. Como explicado anteriormente essa função é realizada pelo *transceiver* SN65HVD251 que faz interface dos sinais TTL do DSP e os sinais da rede CAN. Na Figura 4.29 pode-se observar a foto do protótipo.

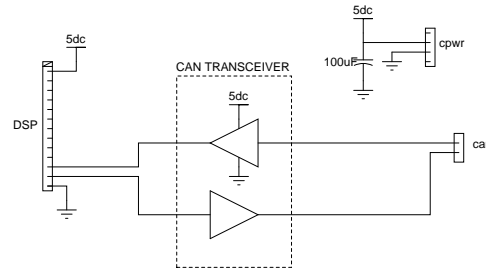


Figura 4.28: Circuito de Comunicação da Rede CAN.

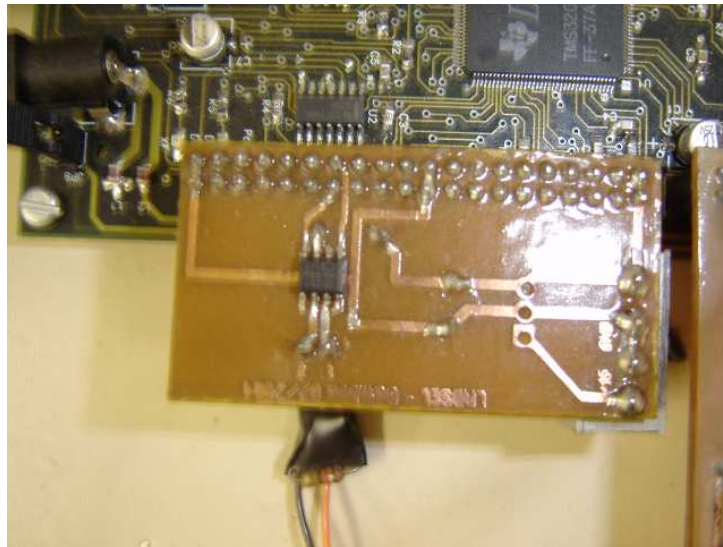


Figura 4.29: Foto do Circuito de Comunicação da Rede CAN (componentes).

A Figura 4.30 mostra o circuito condicionador de medição das correntes trifásicas instantâneas do sistema de geração fotovoltaica. Ele consiste de três circuitos idênticos, um para cada fase. Cada circuito é composto de um sensor de efeito Hall de corrente *LA 205-S* ligado em série com um amplificador operacional na configuração de seguidor de tensão cuja saída alimenta um amplificador inversor somador usado para ajustar o sinal amostrado ao nível requerido pelo conversor analógico digital do *DSP TMS320F243*.

Os sinais das correntes do sistema de geração fotovoltaico são então enviados para o DSP de controle juntamente com os sinais das tensões trifásicas. Dentro desse DSP é calculada a potência imaginária instantânea nos terminais dos conversores CC-CA. A Figura 4.31 mostra o protótipo do circuito de condicionamento para a medição das correntes trifásicas instantâneas do sistema de geração.

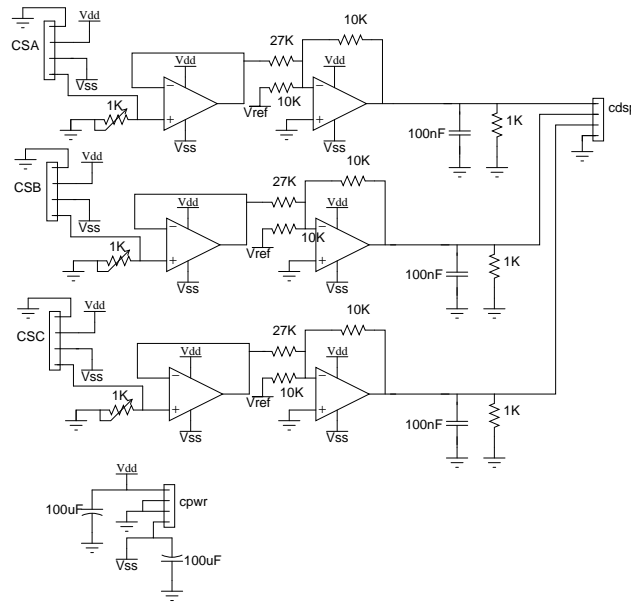


Figura 4.30: Circuito de Condicionamento das Correntes Trifásicas do Sistema Elétrico.

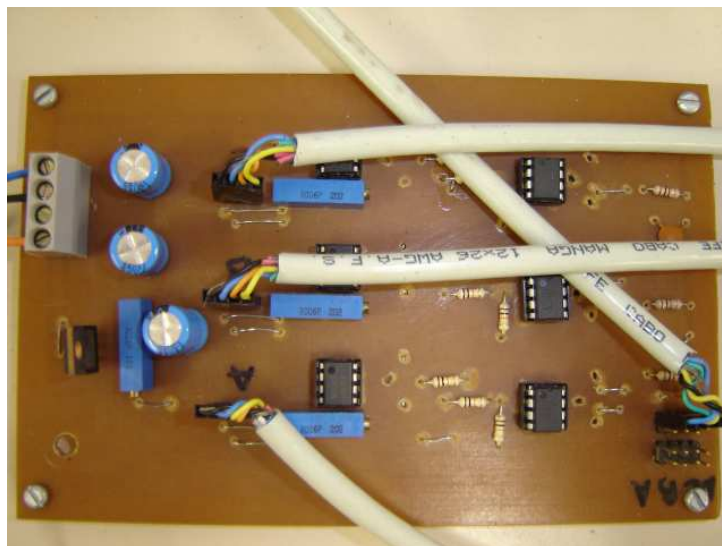


Figura 4.31: Foto do Circuito de Condicionamento das Correntes Trifásicas do Sistema Elétrico.

A Figura 4.32 mostra o esquema de proteção contra tensões negativas e contra sobre tensão na entrada dos conversores A/D do DSP de controle. O circuito de proteção é idêntico ao circuito da Figura 4.24 porém aqui são medidas três tensões alternadas, três correntes alternadas e uma tensão CC. A Figura 4.33 mostra a fotografia do protótipo do circuito de proteção.

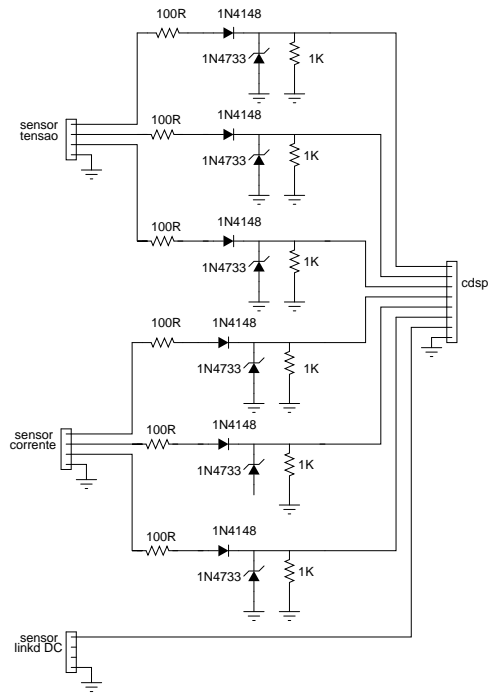


Figura 4.32: Circuito de Proteção do Conversor A/D do DSP de Controle.

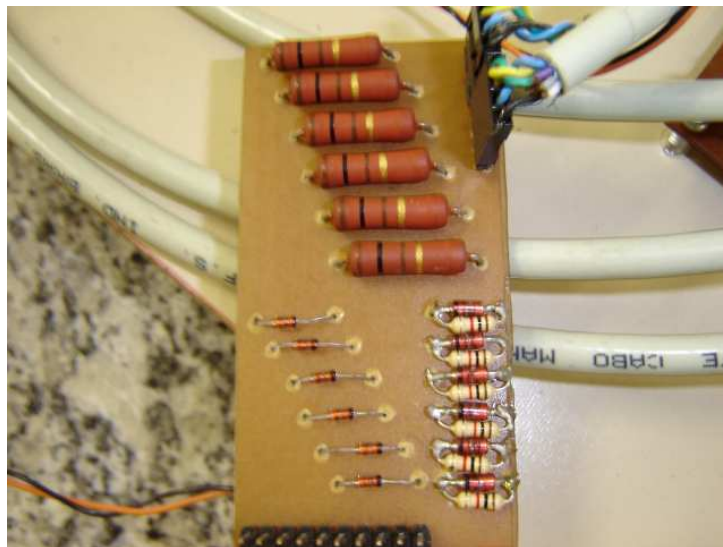


Figura 4.33: Foto do Circuito de Proteção do Conversor A/D do DSP de Controle.

O circuito de medição da tensão do elo CC utilizado pelos conversores CC-CA é idêntico ao utilizado pelos conversores CC-CC, e é descrito na Seção 4.7.1 e mostrado nas Figuras 4.18 e 4.19.

Rede CAN

A parte física da rede CAN consiste de um par de cabos e dois resistores de $120\ \Omega$ como terminadores em cada extremidade, a comunicação é feita pelo sinal diferencial gerado pelo *transceiver* smd *SN65HVD251*. Esse circuito garante o bom funcionamento e a imunidade aos ruídos de chaveamento gerados pelos conversores. O esquema de ligação utilizado está mostrado na Figura 4.34.

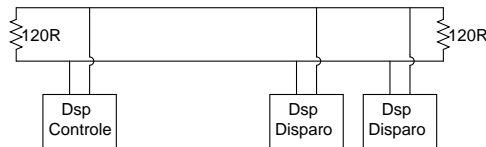


Figura 4.34: Diagrama de Ligação da Rede CAN.

4.8 Conclusões Parciais

Nesse capítulo foi descrito o processo de digitalização das técnicas de controle propostas no Capítulo 2 para o controle dos conversores CC-CC e CC-CA. Foi desenvolvido um *Emulador* do DSP que junto com um simulador de circuitos permitiu testar os algoritmos utilizados nos DSP's através de simulações.

As simulações dos circuitos com a emulação do DSP demonstraram que a implementação das técnicas de controle nos DSP's foram feitas de forma satisfatória com resultados muito próximos das simulações feitas com o ATP/EMTP no Capítulo 2. As diferenças entre os resultados do simulador utilizado e do ATP/EMTP se devem aos erros numéricos das simulações e pequenas diferenças nos modelos utilizados. As pequenas diferenças existentes entre controle simulado e o emulado já eram esperadas no processo de digitalização e não afetam significativamente o funcionamento do controle digitalizado, e são ocasionadas pelas limitações do processador de *ponto-fixa* adotado, atrasos na conversão dos conversores A/D e pela taxa de amostragem utilizada nos algoritmos discretizados.

Foram projetados os circuitos de interface entre os DSP's e os *drivers* dos conversores e os circuitos de condicionamento para as medições de grandezas elétricas do sistema.

Capítulo 5

Programa Supervisório do Sistema de Geração Fotovoltaico

5.1 Introdução

O sistema experimental fotovoltaico de geração de energia elétrica é monitorado remotamente, em *tempo real*, por um sistema supervisório desenvolvido em linguagem C. Esse sistema supervisório gerencia as falhas dos conversores, permite a alteração dos sinais de referência dos controladores além de emitir relatórios do sistema fotovoltaico.

A monitoração das tensões e das correntes do sistema fotovoltaico é feita pelos DSPs de controle e de disparo os quais se comunicam via barramento *CAN*. O sistema supervisório roda em um microcomputador remoto (*Host*) e se comunica com os DSPs através da porta serial.

5.2 Gerenciamento de Falhas dos Conversores CC-CC

Para se evitar que o valor da tensão de saída do conversor *boost* se eleve acima dos valores nominais dos semicondutores instalou-se um sensor de efeito Hall de tensão em seus terminais de saída. O sinal proveniente desse sensor é usado pelo sistema de gerenciamento para bloquear os disparo do conversor *boost* caso haja alguma falha que

impossibilite o conversor CC-CA de controlar a tensão do elo CC.

5.3 Sistema Supervisório dos Conversores CC-CC

A supervisão do sistema de controle é feita a partir da conexão de um cabo serial do computador remoto (*Host*) ao DSP com as configurações da tabela 5.1.

Tabela 5.1: Configuração da Porta Serial

Velocidade	9600bps
Bits de dados	8
Paridade	Nenhuma
Bits de parada	1

A saída do DSP fornece os valores da tensão, da corrente e da potência ativa nos terminais de saída do painel solar fotovoltaico. Além dessas variáveis elétrica é possível obter a razão cíclica (*dutycycle*) do conversor *boost* e a Tensão de saída. Na Figura 5.1 mostra um trecho com o relatório obtido com os dados aquisitados do painel solar fotovoltaico/conversos *boost*.

Vpv= 163,05V	Ipv= 020,88A	Vo= 204,21V	Duty= 033,86%	Pot= 03405,31W
Vpv= 162,61V	Ipv= 021,46A	Vo= 203,89V	Duty= 033,86%	Pot= 03491,26W
Vpv= 162,17V	Ipv= 022,28A	Vo= 203,25V	Duty= 034,26%	Pot= 03614,64W
Vpv= 163,49V	Ipv= 021,29A	Vo= 204,21V	Duty= 035,07%	Pot= 03481,44W
Vpv= 163,05V	Ipv= 021,58A	Vo= 204,21V	Duty= 033,06%	Pot= 03519,77W
Vpv= 163,05V	Ipv= 021,46A	Vo= 204,21V	Duty= 034,26%	Pot= 03500,69W
Vpv= 163,49V	Ipv= 021,23A	Vo= 204,85V	Duty= 032,06%	Pot= 03471,87W
Vpv= 163,49V	Ipv= 021,46A	Vo= 204,21V	Duty= 030,46%	Pot= 03510,13W
Vpv= 163,05V	Ipv= 021,29A	Vo= 204,21V	Duty= 030,46%	Pot= 03472,08W
Vpv= 163,05V	Ipv= 021,82A	Vo= 204,21V	Duty= 029,25%	Pot= 03557,93W
Vpv= 164,81V	Ipv= 020,00A	Vo= 205,82V	Duty= 029,25%	Pot= 03297,40W
Vpv= 163,93V	Ipv= 020,35A	Vo= 205,18V	Duty= 028,45%	Pot= 03337,35W
Vpv= 163,93V	Ipv= 020,35A	Vo= 204,85V	Duty= 028,45%	Pot= 03337,35W
Vpv= 163,93V	Ipv= 020,82A	Vo= 204,85V	Duty= 024,04%	Pot= 03414,07W
Vpv= 163,49V	Ipv= 021,35A	Vo= 204,21V	Duty= 024,24%	Pot= 03491,00W
Vpv= 163,05V	Ipv= 021,29A	Vo= 204,21V	Duty= 026,65%	Pot= 03472,08W
Vpv= 163,93V	Ipv= 021,06A	Vo= 204,85V	Duty= 027,85%	Pot= 03452,43W

Figura 5.1: Relatório de Saída do Conversor *Boost*.

5.4 Gerenciamento de Falhas dos Conversores CC-CA

O DSP responsável pela execução do algoritmo do sistema de controle executa em paralelo uma rotina de monitoração e de gerenciamento de falhas onde conversores CC-CA são bloqueados quando uma anomalia é detectada. O bloqueio dos pulsos de disparo dos conversores garante a integridade dos equipamentos e evita o fenômeno de *ilhamento* do sistema fotovoltaico (Carletti, 2005) e (ANSI/IEEE-929, 2000).

O *ilhamento* acontece quando a alimentação do sistema elétrico é interrompida pela concessionária por alguma falha detectada ou para execução de manutenção (corretiva ou programada) do sistema de distribuição (ANSI/IEEE-929, 2000). Nessa situação, os transformadores de distribuição e conseqüentemente o lado de alta-tensão da rede CA podem continuar energizados pelo sistema de geração fotovoltaico caso não seja desconectado. O funcionamento do sistema de proteção contra *ilhamento* é baseado na medição da frequência, tensões e correntes do sistema elétrico (John, Zhihong e Kolwalkar, 2004), na falta do sistema como referência, a malha de controle perde rapidamente a frequência de operação do sistema e os disparos dos conversores são bloqueados, como demonstrado em (Carletti, 2005), uma chave contatora na conexão do sistema PV com o sistema elétrico, alimentado pelo sistema elétrico, serve de isolamento, sem o sistema elétrico e com os disparos bloqueados a chave desarma e isola totalmente os sistemas.

A Figura 5.2 mostra a estrutura do sistema de monitoramento implementado. Existem três retângulos com linhas tracejadas representando os DSPs de controle e de sincronismo e de geração dos pulsos de disparo dos conversores. À esquerda tem-se representado o computador remoto (*host*) e a linha grossa, vertical, localizada entre os DSPs representa o barramento *CAN* pelo qual os DSPs se comunicam. O sistema de gerenciamento de falhas dos conversores CC-CA foi projetado para verificar os seguintes eventos:

1. Sobre tensão de alimentação
2. Sobre tensão de linha
3. Sub tensão de linha
4. Sobre corrente de linha
5. Frequência de operação máxima

6. Frequência de operação mínima

7. Erros nos *drivers* de disparo

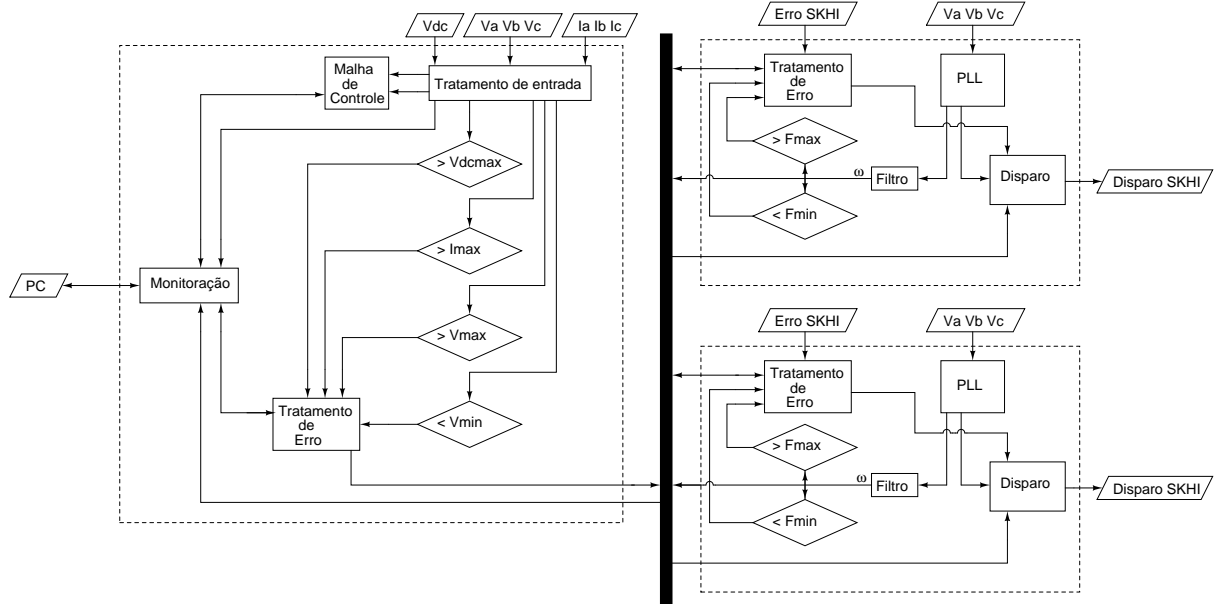


Figura 5.2: Diagrama de Controle

Os valores eficazes da tensão e da corrente de linha são estimados através do cálculo do valor coletivo da tensão instantânea (v_{Σ}) e da corrente instantânea (i_{Σ}) conforme mostrado a seguir:

$$\begin{cases} v_{\Sigma} = \sqrt{\frac{v_{\alpha}^2 + v_{\beta}^2}{3}} \\ i_{\Sigma} = \sqrt{\frac{i_{\alpha}^2 + i_{\beta}^2}{3}} \end{cases} \quad (5.1)$$

onde as tensões v_{α} e v_{β} são obtidas aplicando uma transformação de Clarke nas tensões trifásicas instantâneas, conforme mostrado em (3.1). De maneira semelhante as correntes i_{α} e i_{β} são obtidas aplicando a mesma uma transformação linear nas correntes de linha do sistema de geração fotovoltaico.

5.5 Sistema Supervisório dos Conversores CC-CA

Para monitoramento das grandezas do sistema de geração de energia elétrica fotovoltaico foi desenvolvido um sistema supervisório com o qual é possível observar em “*tempo real*” as grandezas elétricas descritas a seguir:

1. Potência Ativa trifásica
2. Potência Reativa trifásica
3. Valor coletivo da tensão de linha
4. Valor coletivo da corrente de linha
5. Tensão no elo CC
6. Frequência de linha

Além de receber informações sobre o *status* de operação dos conversores CC-CA:

1. Ângulos de disparo
2. Erros do sistema

Todas as grandezas monitoradas ainda são armazenadas em um arquivo para análise posterior. O sistema foi escrito em linguagem C, utilizando a biblioteca SDL (SDL, 2005), o que permite seu uso em vários sistemas operacionais.

Na Figura 5.3 é mostrada a tela do supervisório onde são desenhadas as formas de onda das potências instantâneas ativa e reativa nos terminais do sistema de geração fotovoltaico calculadas pelo DSP de controle. A Figura 5.4 mostra a tela do supervisório onde são disponibilizadas as frequências detectadas pelos circuitos de sincronismo (PLL) implementados nos dois DSPs de disparo. Na Figura 5.5 é mostrada a tela do supervisório onde são lidos os valores coletivos da tensão de fase da rede e da corrente de fase e a tensão do elo CC. A Figura 5.6 mostra a tela do supervisório com a janela de erros aberta, onde se podem ver os erros detectados pelos DSPs de controle e disparo.

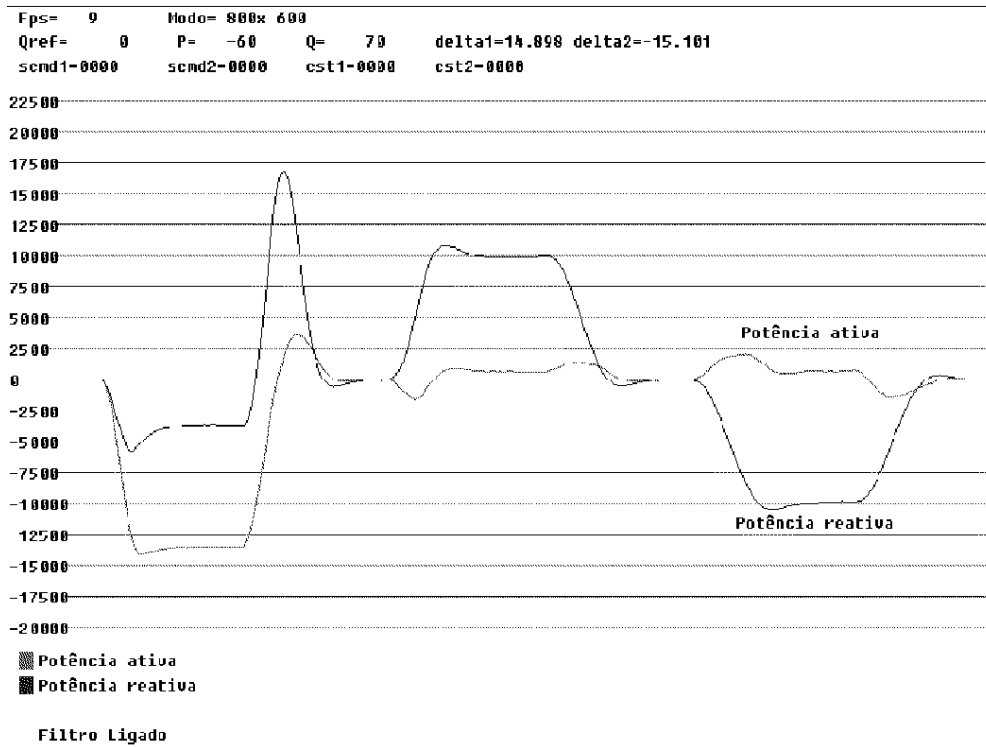


Figura 5.3: Tela da Supervisão das Potências

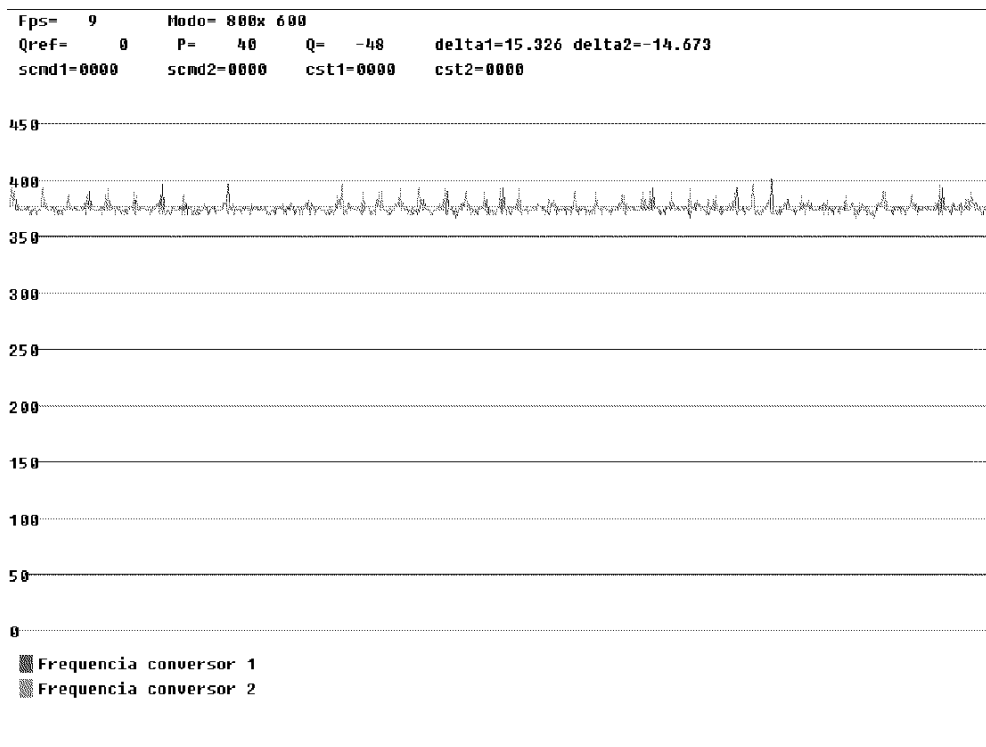


Figura 5.4: Tela da Supervisão das Frequências.

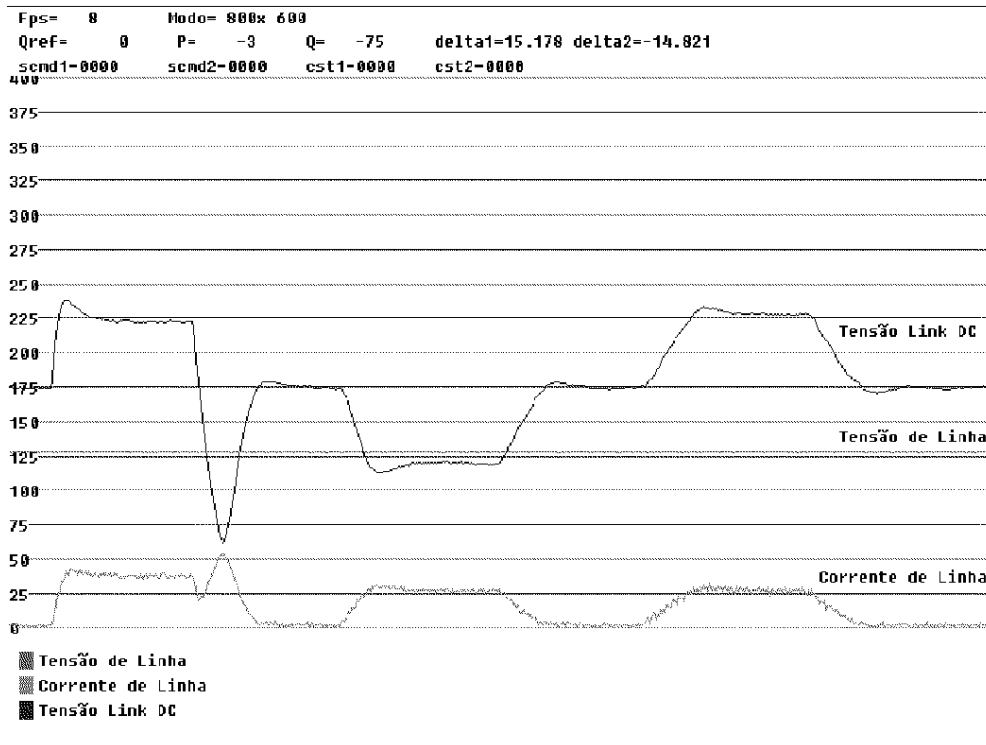


Figura 5.5: Tela de Supervisão das Tensões e Corrente

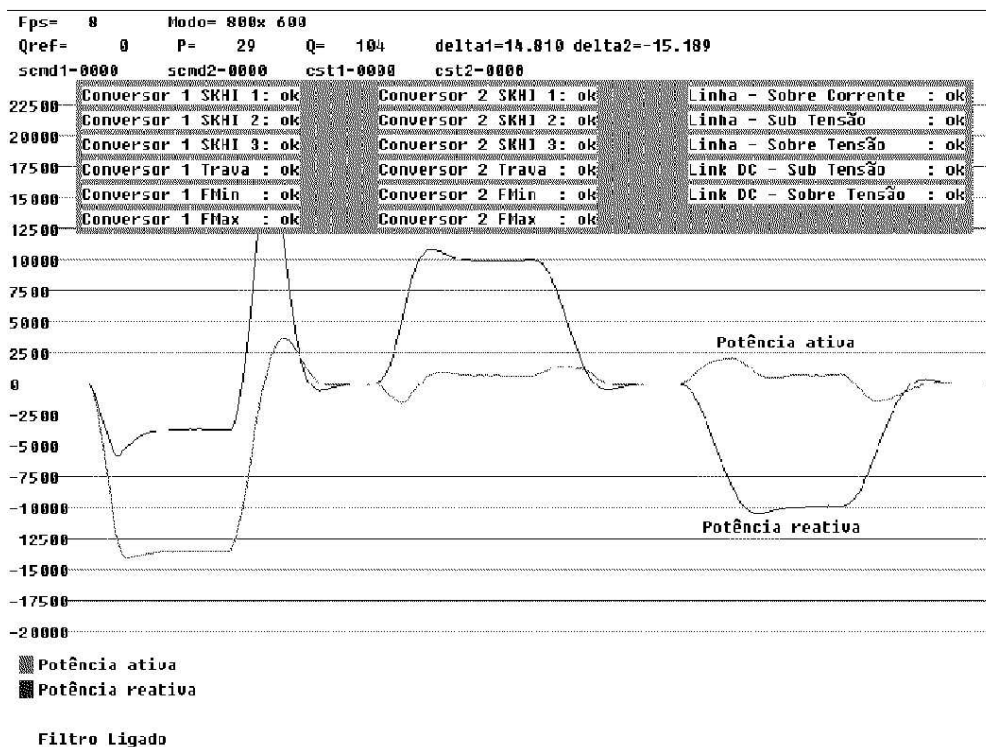


Figura 5.6: Tela do Supervisor com Janela de Erros Ativa.

5.5.1 Mapa de Teclas e Funções

Nas Tabela 5.2 é mostrado o mapa da teclas que o programa supervisorio utiliza para controle geral, na Tabela 5.3 é mostrado o mapa de teclas usados para o controle da referênciã da potência reativa e na Tabela 5.4 é mostrado o mapa de teclas do controle dos conversores CC-CA.

Tabela 5.2: Controle Gerais do Supervisorio

Tecla	Função
<1>	mostra tela de potências
<2>	mostra tela de Freqüências
<3>	mostra tela de Tensões e correntes
<4>	mostra tela de <i>log</i> de erros quando a sub-janela de erros está aberta
<e>	mostra sub-janela de erros
<f>	liga e desliga o filtro das potências
<F1>	aumenta resolução da tela
<F2>	diminui resolução da tela
<F12>	Salva imagem da tela <screen_XXX.pcx>
<ESC>	Fecha supervisorio

Tabela 5.3: Controle da Potência Reativa de Referênciã

Tecla	Função
<a>	Adiciona 1000 VAR a potência reativa de referênciã
<s>	Subtrai 1000 VAR a potência reativa de referênciã
<q>	Adiciona 100 VAR a potência reativa de referênciã
<w>	Subtrai 100 VAR a potência reativa de referênciã

Tabela 5.4: Controle dos Conversores

Tecla	Função
<l>	Trava conversor 1 ²
<k>	Destrava conversor 1 ²
<o>	Trava conversor 2 ²
<i>	Destrava conversor 2 ²

² O software do DSP trava e destrava os dois conversores simultaneamente.

5.5.2 Arquivo de Configuração

O arquivo de configuração *graph.cfg* permite configurar alguns parâmetros do supervisorio. O conteúdo do arquivo de configuração tem o seguinte formato:

```
vmode= 7
```

```
vfull= 0
grid1= 2500
grid2= 50
grid3= 25
```

Nas Tabela 5.5, Tabela 5.6 e Tabela 5.7 são fornecidas as descrições das variáveis e dos valores que podem ser usados em cada uma.

Tabela 5.5: vmode - Modo de resolução de Vídeo

Código	Resolução
1	320x240
2	400x300
3	512x384
4	640x480
5	800x600
6	960x720
7	1024x768

Tabela 5.6: vfull - Modo de Apresentação do Programa

Código	Apresentação
0	janela
1	tela cheia

As variáveis gridx são os intervalos entre os eixos dos gráficos das telas do supervisor, sua configuração é mostrada na tabela a seguir:

Tabela 5.7: gridx: Intervalos entre os Eixos dos Gráficos

Descrição	Aplicação
grid1	tamanho das divisões do gráfico de potência (-30000 à 30000)
grid2	tamanho das divisões do gráfico de frequência (0 à 550)
grid3	tamanho das divisões do gráfico de tensões e correntes (0 à 450)

5.6 Conclusões Parciais

Foi desenvolvido nesse capítulo o sistema de gerenciamento de falhas que protege os equipamentos de falhas preservando a integridade dos mesmos e um sistema supervisor que permite a visualização de variáveis do processo e controle do mesmo.

Capítulo 6

Resultados Experimentais

6.1 Introdução

Neste capítulo serão apresentados os resultados experimentais obtidos com a operação da planta de geração de energia elétrica fotovoltaica operando interligada com a rede CA. A operação dos conversores CC-CC é investigada para a operação com e sem o algoritmo de rastreamento de máxima potência (MPPT). Já os conversores CC-CA são controlados considerando inicialmente que os VSIs são chaveados usando a técnica multi-pulso com eliminação seletiva de harmônicos e depois são apresentados resultados usando modulação PWM de alta frequência.

As Figura 6.1 a Figura 6.5 mostram respectivamente os arranjos com os painéis solares fotovoltaicos, um dos conversores *boost* com seus circuitos de condicionamento e DSP de controle, os dois conversores VSI e seus transformadores de conexão e os DSPs de controle e de disparo do sistema de geração fotovoltaico.



Figura 6.1: Foto com os Arranjos dos Painéis Solares Fotovoltaicos.



Figura 6.2: Foto de um Conversor CC-CC *Boost* e Circuitos Condicionadores.



Figura 6.3: Foto do Conversor VSI No. 1 e Transformadores de Conexão.

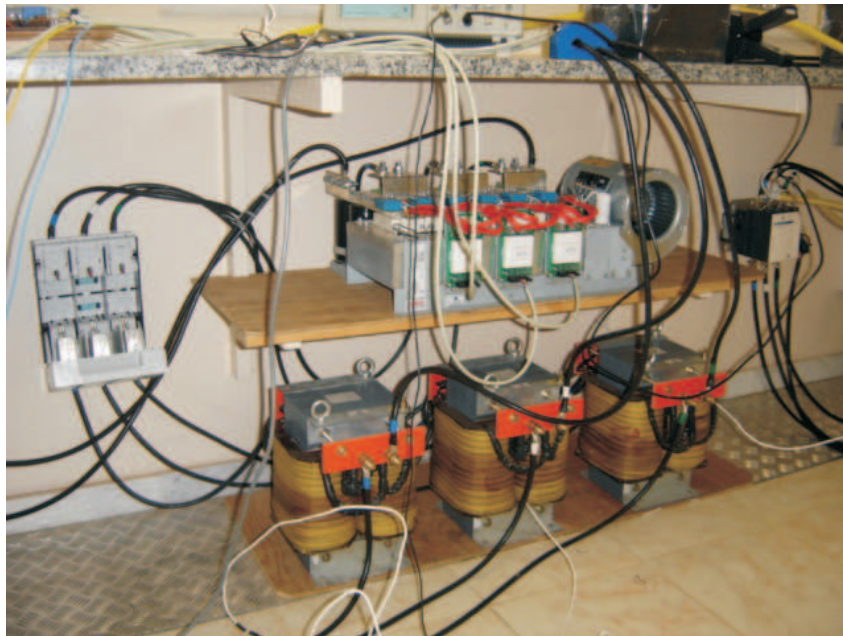


Figura 6.4: Foto do Conversor VSI No. 2 e Transformadores de Conexão.

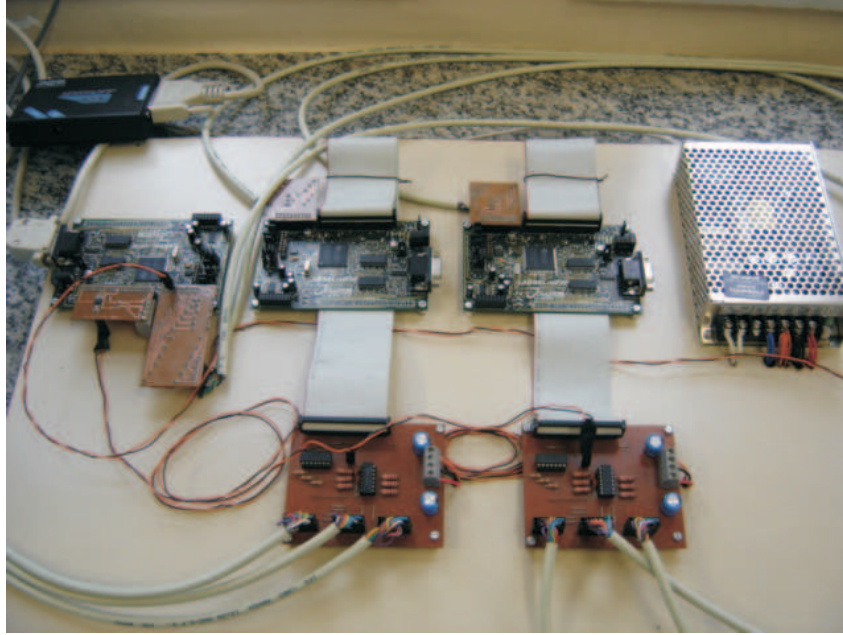


Figura 6.5: Foto dos DSPs de Controle e de Disparo dos Conversores VSIs.

6.2 Conversores CC-CC

A Figura 6.6 mostra as tensões de entrada e saída no conversor *boost*. Como esperado, para a correta operação dos conversores *boost*, a tensão de entrada é menor que a tensão de saída. Na Figura 6.7 é mostrada a tensão na chave semicondutora autocomutada e a corrente pelo indutor de entrada do conversor. Nessa figura pode-se observar que o valor da corrente é sempre maior que zero, caracterizando que o conversor CC-CC opera no modo de operação em condução contínua (CCM - “*Continuous Conduction Mode*”).

6.3 Conversores CC-CA

Na Figura 6.8 pode ser observado na forma de onda de tensão de fase com 3 níveis distintos. Conforme discutido no Capítulo 2, os ângulos dos recortes foram calculados para eliminar somente três componentes harmônicas (11° , 13° e 23°) nas tensões de saída dos conversor VSI. A Figura 6.9 mostra a tensão de fase com 5 níveis obtida a partir da conexão dos dois conversores CC-CA em série. As tensões mostradas nas Figuras 6.8 e 6.9 foram obtidas antes do sistema fotovoltaico ser conectado em paralelo com a rede CA.

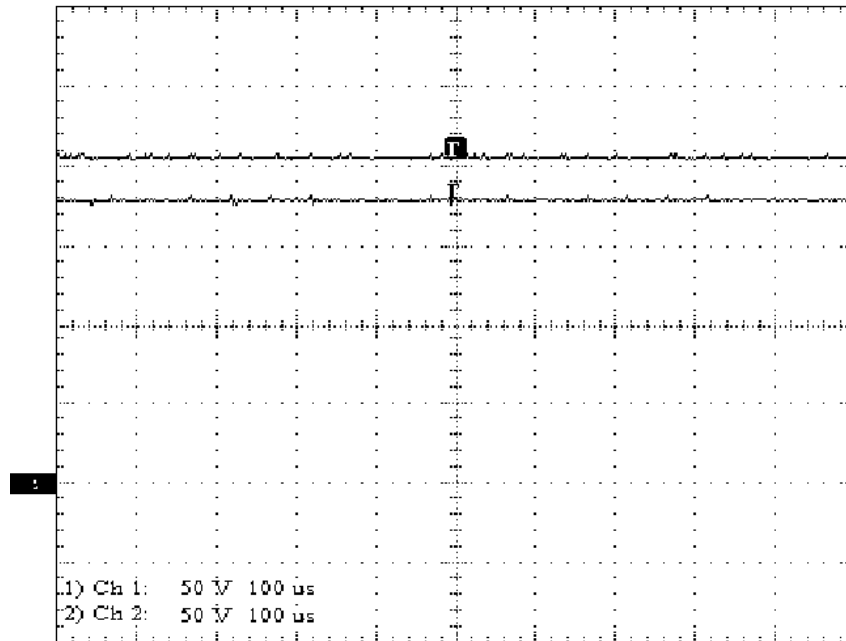


Figura 6.6: Tensão de Saída (superior) e Tensão de Entrada (inferior).

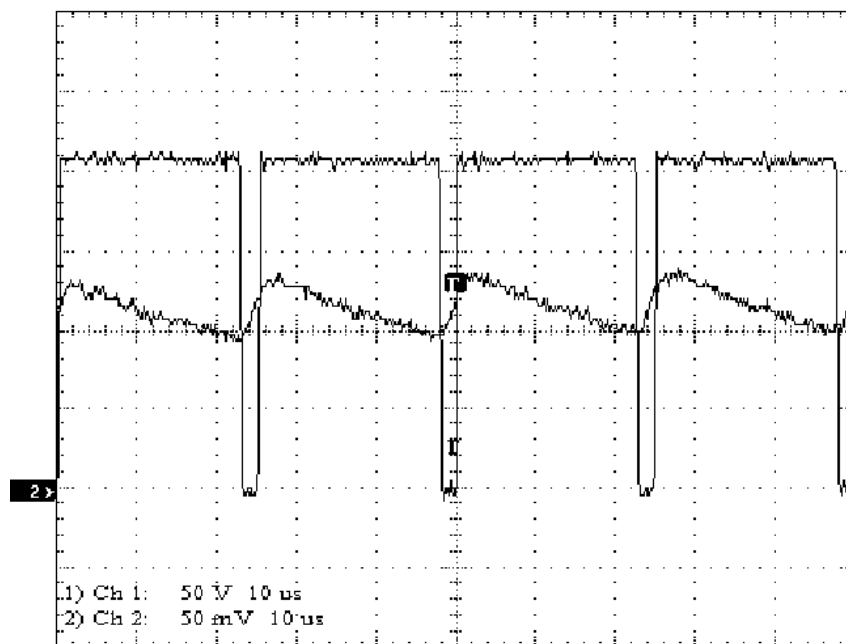


Figura 6.7: Tensão na Chave (superior) e Corrente no Indutor (inferior) [escala: 1mV/A].

Na Figura 6.10 é mostrada a forma de onda da tensão de fase da rede elétrica após a conexão dos conversores CC-CA. A Figura 6.11 mostra o espectro harmônico da tensão resultante. Observe que em consequência dos erros e atrasos no chaveamento dos VSIs o 11^o, 13^o e 23^o harmônicos da tensão sintetizada pelos inversores não são cancelados totalmente pela modulação seletiva. Contudo, a componente da tensão harmônica com maior amplitude observada é a 11^o, cujo valor é aproximadamente 4 %.

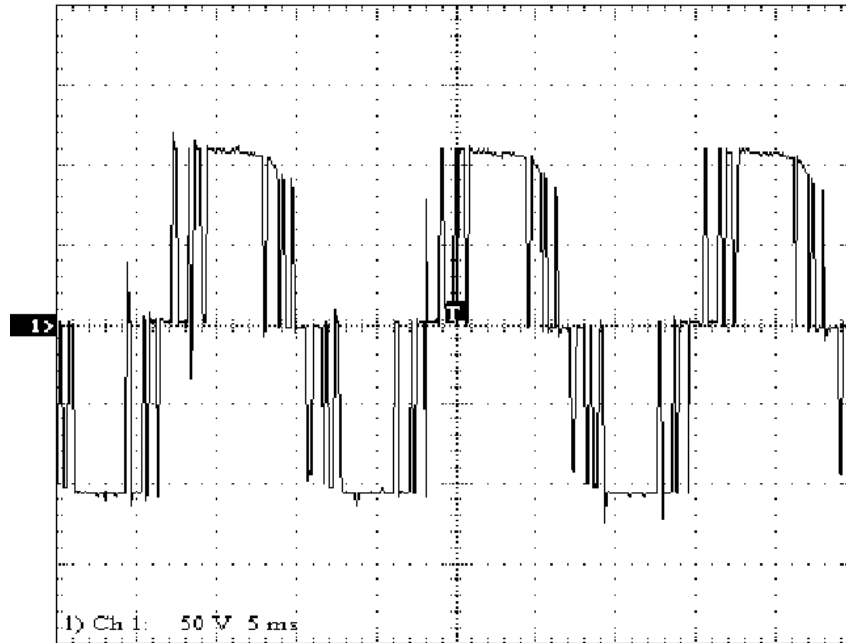


Figura 6.8: Tensão de 3 Níveis na Saída de um Conversor em Aberto.

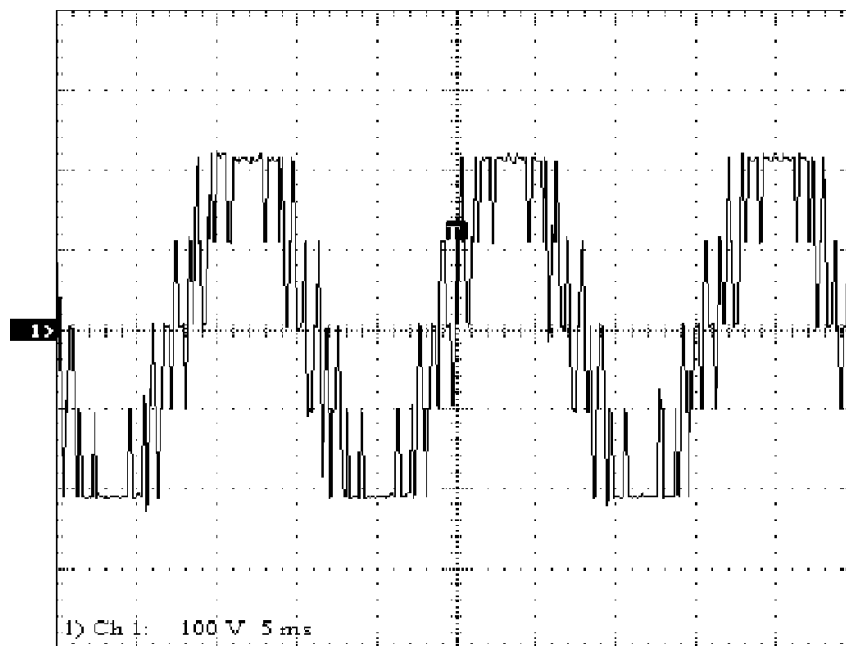


Figura 6.9: Tensão de 5 Níveis na Saída dos Conversores em Aberto.

A Figura 6.12 mostra a forma de onda da tensão e da corrente nos terminais dos conversores CC-CA quando o sistema está flutuando. Neste caso o sistema de geração fotovoltaico opera em paralelo com a rede CA e nenhuma potência ativa ou reativa flui em seus terminais. A forma de onda da corrente é bastante irregular e apresenta discrepâncias em relação aos resultados das simulações apresentados no Capítulo 2. Este comportamento pode ser explicado pelo fato de que as reatâncias de dispersão dos transformadores de conexão serem muito baixas, muito menores do que as adotadas no modelo digital. A

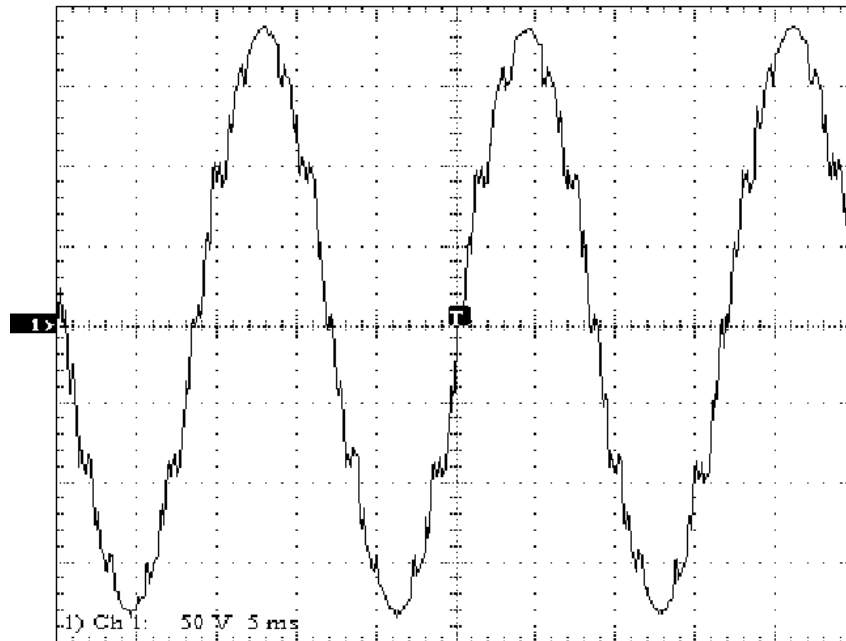


Figura 6.10: Tensão de Fase da Rede CA.

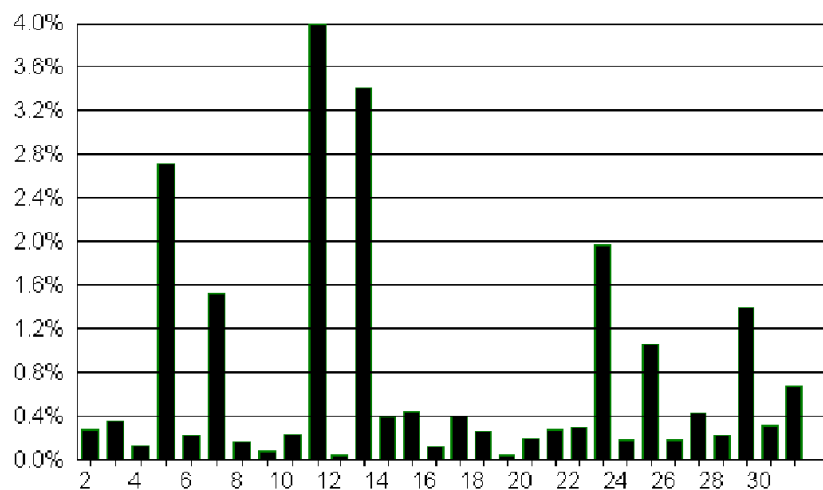


Figura 6.11: Espectro de Frequência da Tensão de Saída.

forma de onda das corrente pode ser melhorada se um pequeno filtro passivo for conectado entre os terminais dos transformadores de acoplamento.

6.3.1 Operação Injetando Potência Ativa na Rede CA

A Figura 6.13 mostra a tensão de fase e a corrente de linha quando o conversor CC-CA está injetando aproximadamente 6 kW de potência ativa no sistema elétrico com o algoritmo MPPT desativado. Na Figura 6.14 é mostrada a tensão de fase e a corrente de linha quando o conversor CC-CA está injetando aproximadamente $9,5 \text{ kW}$ de potência

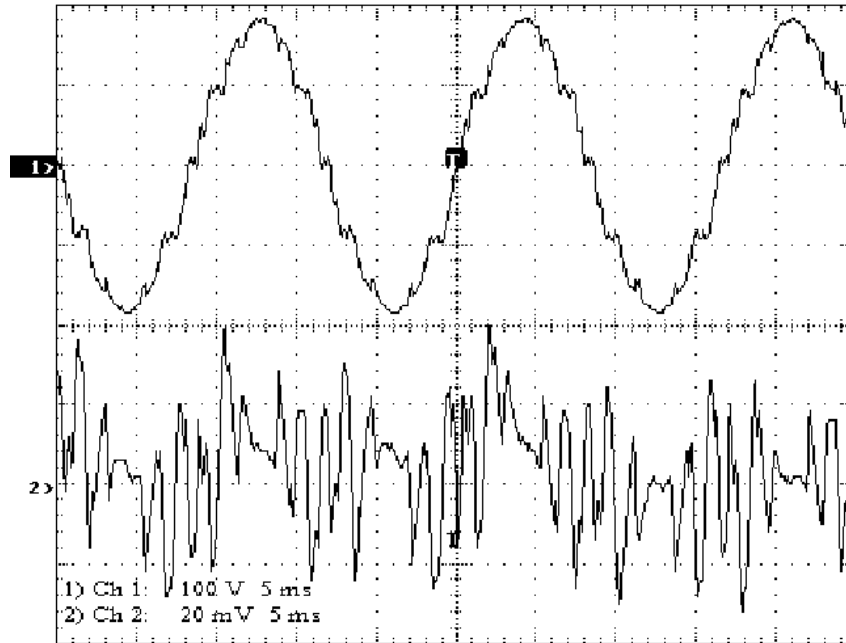


Figura 6.12: Tensão (superior) e Corrente (inferior) [escala: 1mV/A] do Conversor Flutuando.

ativa no sistema elétrico. A diferença entre os dois resultados pode ser explicada pelo fato dos conversores CC-CC, no segundo caso, estarem sendo controlados com o algoritmo de rastreamento de máxima potência (MPPT). A utilização da estratégia MPPT aumentou em aproximadamente 58 % a potência injetada no sistema CA.

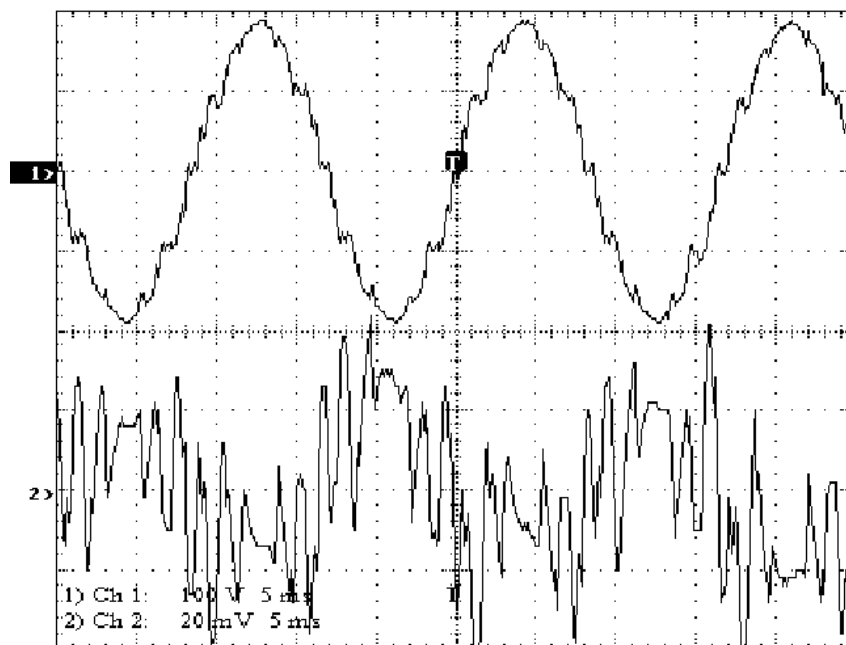


Figura 6.13: Tensão (superior) e Corrente (inferior) [escala: 1mV/A] Injetando Potência Ativa sem MPPT.

Na Figura 6.15 é mostrado o comportamento das potências ativa e reativa ins-

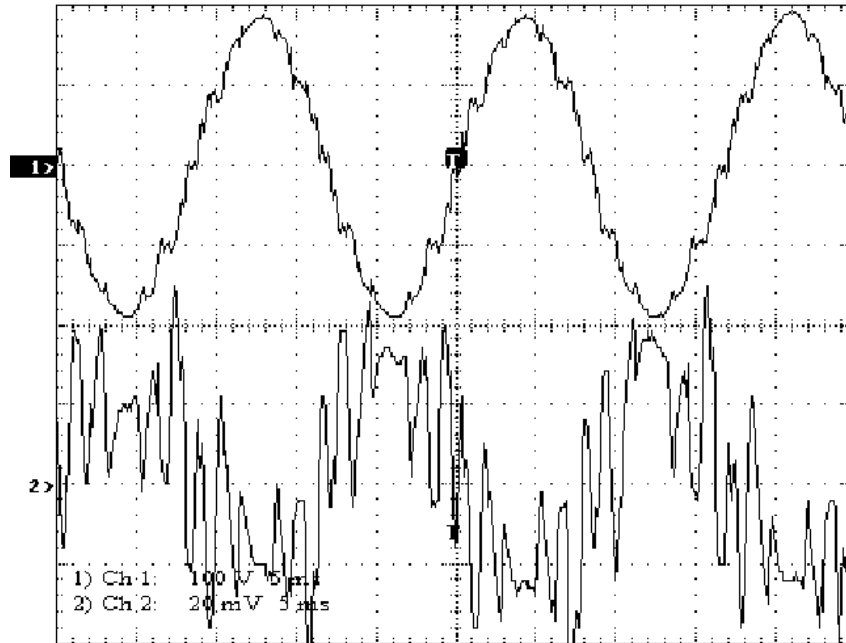


Figura 6.14: Tensão (superior) e Corrente (inferior) [escala: 1m V/A] Injetando Potência Ativa com MPPT.

tantâneas nos terminais do sistema de geração. Na Figura 6.16 são mostrados os valores coletivos instantâneos da tensão e da corrente de linha CA, além do valor médio da tensão do elo CC dos conversores. Essas figuras foram obtidas diretamente das telas do sistema supervisorio cujas características foram apresentadas e discutidas em detalhes no Capítulo 5. Nessas figuras pode-se observar todas as etapas de operação do sistema de geração fotovoltaico. Inicialmente o sistema está flutuando quando então passa a injetar aproximadamente 6 kW de potência ativa na rede CA. Durante esse período o algoritmo MPPT está desativado. Posteriormente, com o algoritmo MPPT ativo, o sistema fotovoltaico passa a injetar aproximadamente 9,5 kW de potência ativa na rede CA.

6.3.2 Operação como Compensador Estático Síncrono (STATCOM)

A Figura 6.17 mostra as formas de onda da tensão de fase e a corrente de linha quando os conversores CC-CA estão absorvendo 20 kVAR de potência reativa do sistema elétrico. Neste caso os conversores estáticos operam com uma característica indutiva, isto é, corrente atrasada de 90° ($\pi/2$ rad) em relação a tensão de fase.

A Figura 6.18 mostra as formas de onda da tensão de fase e da corrente de linha quando o conversor CC-CA está injetando 20 kVAR de potência reativa no sistema CA.

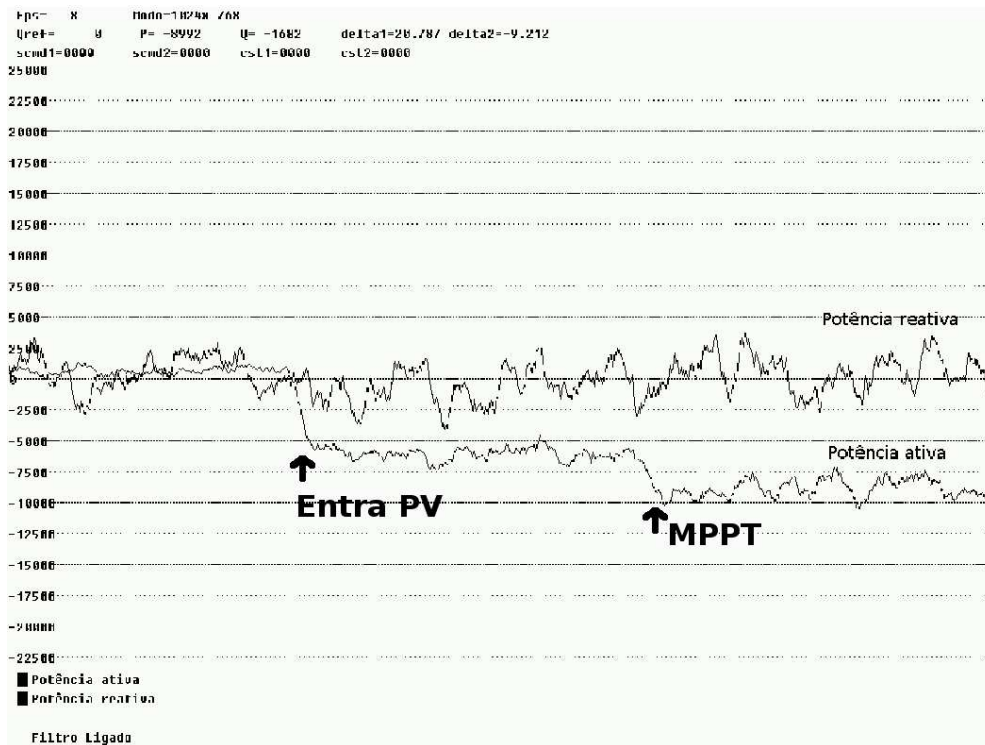


Figura 6.15: Tela de Potências Ativa e Reativa Instantâneas Obtidas do Programa Supervisório Quando o Sistema Fotovoltaico Injeta Potência Ativa na Rede.

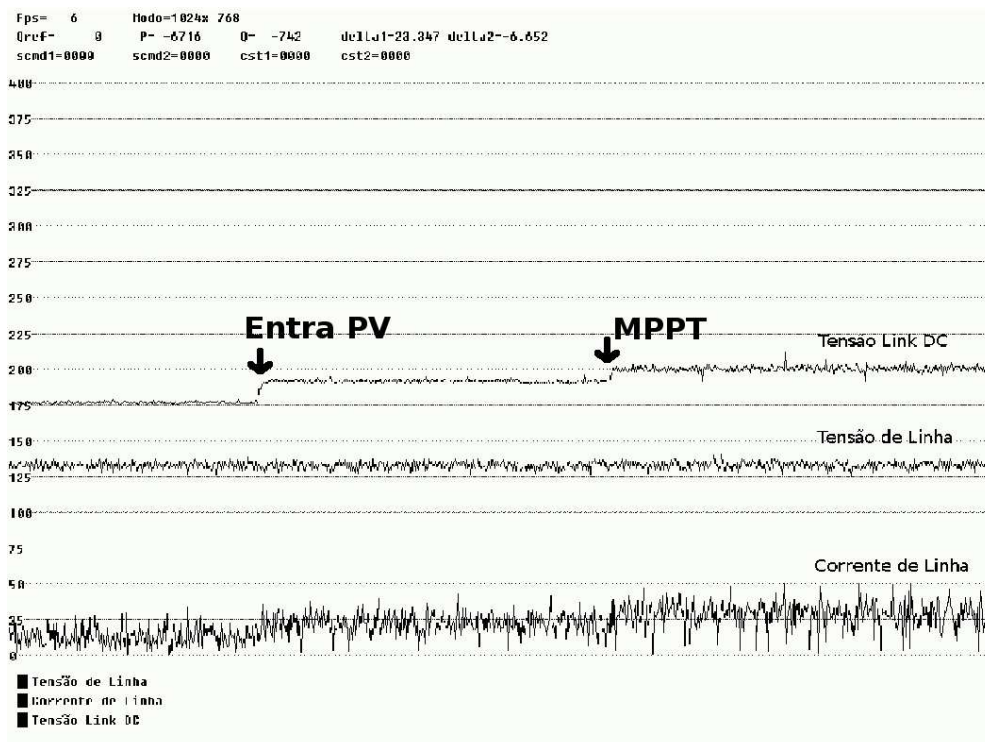


Figura 6.16: Tela dos Valores Coletivos das Tensões e Correntes e Valor Médio da Tensão CC Obtida do Programa Supervisório para o Sistema Fotovoltaico Injetando Potência Ativa na Rede.

Neste caso os conversores operam com uma característica capacitiva sintetizando uma corrente adiantada de 90° ($\pi/2 \text{ rad}$) em relação à tensão de fase. Pode-se observar nas duas figuras que as correntes apresentam distorções devido ao baixo valor de indutância de dispersão dos transformadores de conexão utilizados.

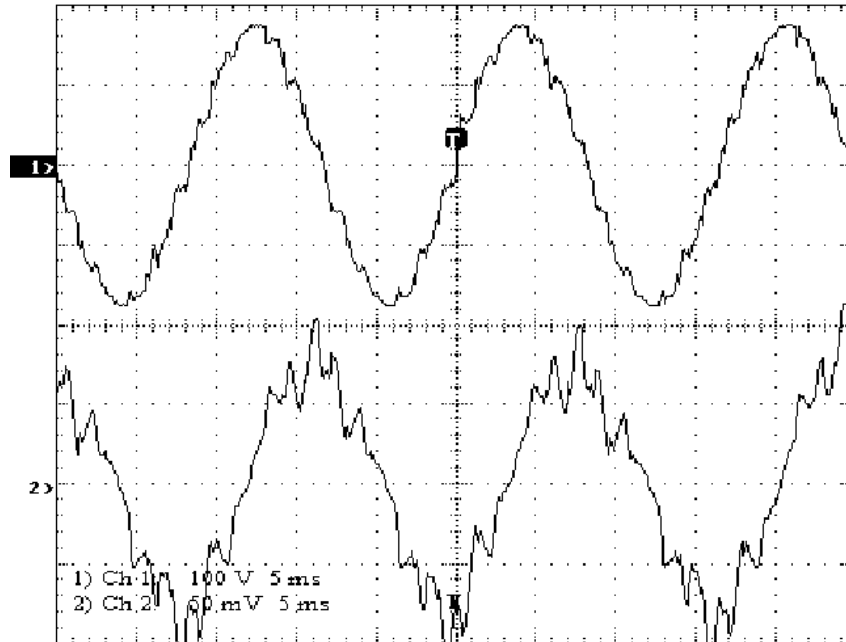


Figura 6.17: Tensão (superior) e Corrente (inferior) [escala: 1 mV/A] com o Sistema Sintetizando 20 kVAR Indutivo.

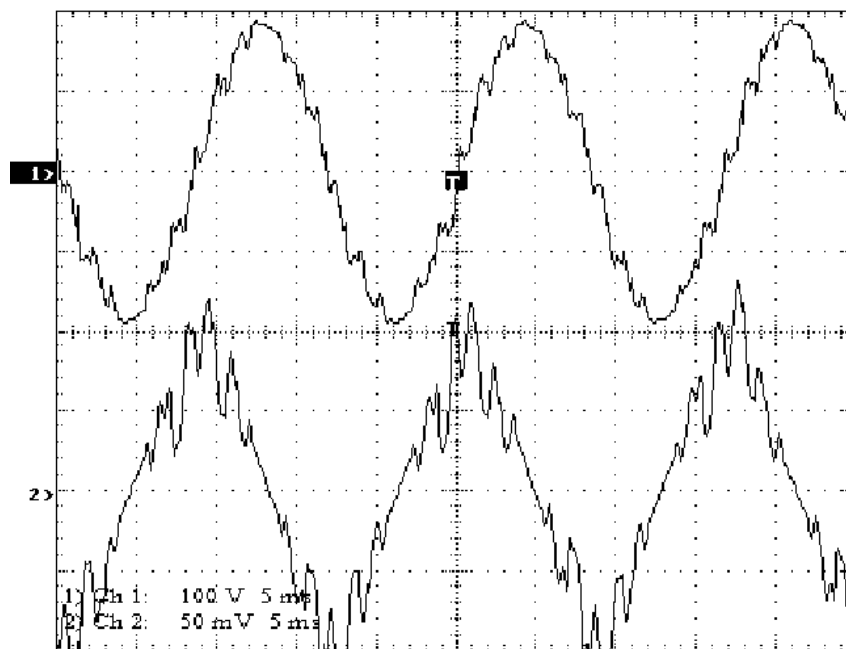


Figura 6.18: Tensão (superior) e Corrente (inferior) [escala: 1 mV/A] com o Sistema Sintetizando 20 kVAR Capacitivo.

As Figuras 6.19 e 6.20 mostram respectivamente as telas das potências ativa e reativa instantâneas nos terminais do sistema fotovoltaico e os valores coletivos das tensões

e correntes de linha além da tensão do elo CC dos conversores. Inicialmente o sistema está flutuando quando a potência reativa de referência do controlador dos conversores é variada rapidamente de 0 para +20 kVAR. A partir desse instante os conversores do sistema de geração passam a operar com uma característica indutiva absorvendo energia reativa da rede CA. Posteriormente o sinal de potência reativa de referência é variado rapidamente de +20 kVAR para -20 kVAR. A partir desse ponto os conversores passam a operar com uma característica capacitiva. Note que, como o método de chaveamento adotado para controlar os conversores não controla a componente fundamental das tensões sintetizadas, a tensão do elo CC dos conversores varia para que os conversores sintetizem a potência reativa desejada. Também pode-se observar que à medida que a potência reativa nos terminais dos conversores é diferente de zero, os VSIs absorvem potência ativa do sistema elétrico para compensar as perdas e manter o capacitor do elo CC carregado.

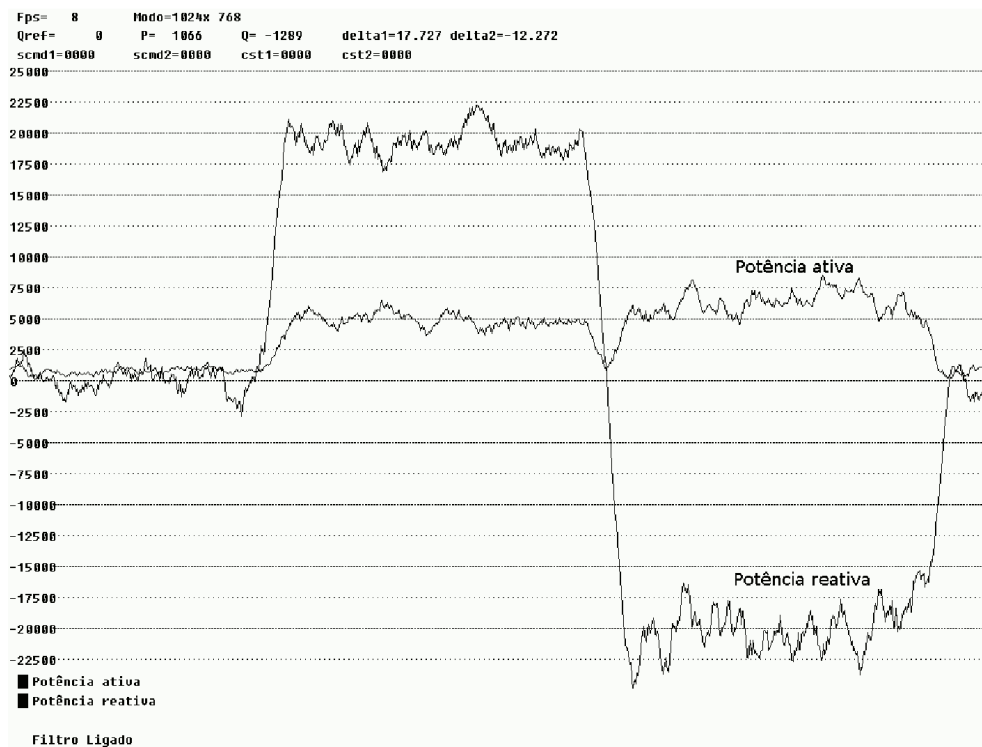


Figura 6.19: Tela de Potências do Supervisório Sintetizando Potência Reativa.

A Figura 6.21 mostra as forma de onda da tensão e da corrente de linha quando os conversores VSIs estão absorvendo 35 kVAR de potência reativa do sistema elétrico e a Figura 6.22 mostra as formas de onda da tensão e da corrente de linha para o caso em que os conversores injetam 35 kVAR de potência reativa. Note que a forma de onda da corrente de linha melhora quando uma maior potência reativa é absorvida ou gerada pelos conversores estáticos.

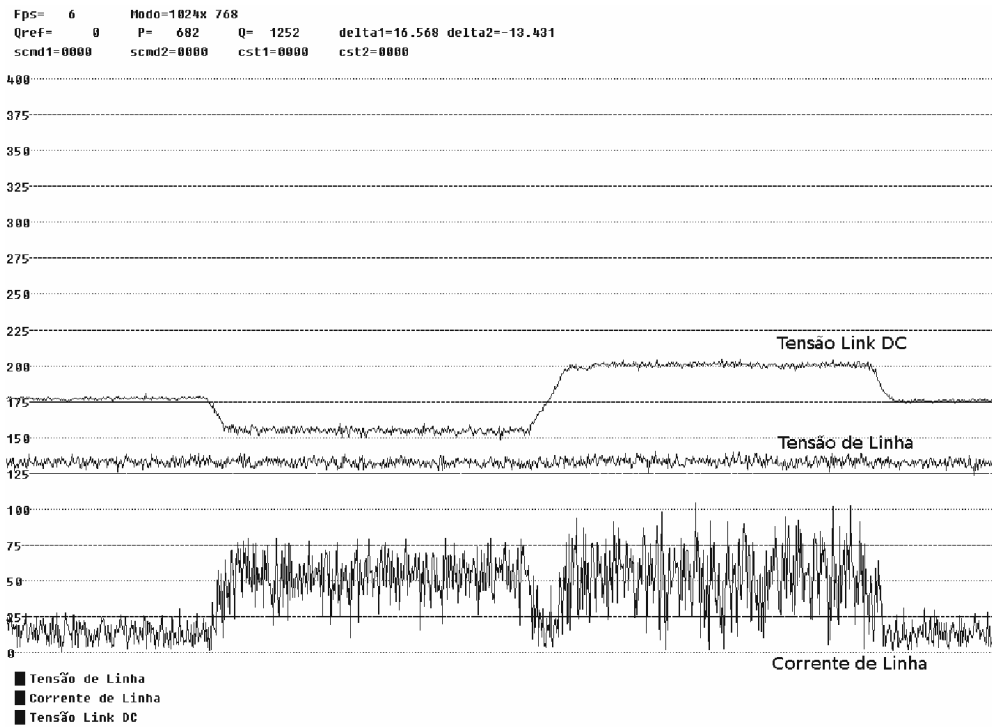


Figura 6.20: Tela de Tensões do Supervisor Síntetizando Potência Reativa.

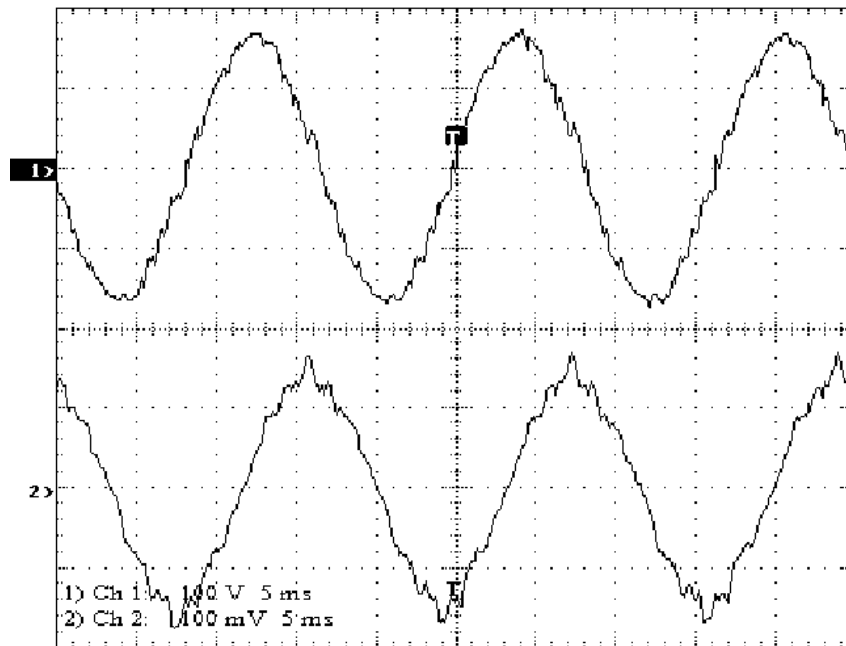


Figura 6.21: Tensão (superior) e Corrente (inferior) [escala: 1mV/A] com o Sistema Síntetizando 35 kVAR Indutivo.

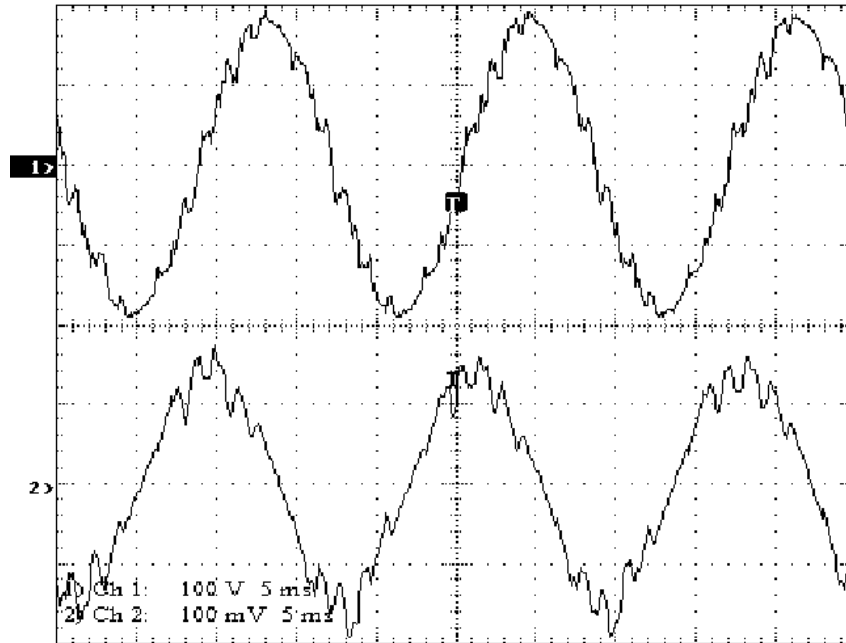


Figura 6.22: Tensão (superior) e Corrente (inferior) [escala: 1mV/A] com o Sistema Sintetizando 35 kVAR Capacitivo.

6.3.3 Operação Injetando Potência Ativa e Compensando Potência Reativa nos Terminais do Sistema de Geração

Nas Seções anteriores foram mostrados resultados experimentais do sistema de geração fotovoltaico operando injetando potência ativa ou injetando/absorvendo potência reativa em paralelo com a rede elétrica. A Figura 6.23 mostra a tela do sistema supervisor onde pode-se observar a capacidade dos conversores do sistema de geração fotovoltaico de injetar e/ou absorver simultaneamente tanto potência ativa quanto potência reativa em seus terminais.

6.3.4 Simulação com Valores Reais

Para comprovar que as diferenças entre as simulações e os resultados experimentais são realmente causadas pelos valores das indutâncias, que na prática são bem menores, as simulações do ATP utilizando o esquema da Figura 2.21 foram refeitas com os valores da Tabela 6.1, onde as indutâncias dos transformadores e da linha são dez vezes menores que as simuladas no Capítulo 2.

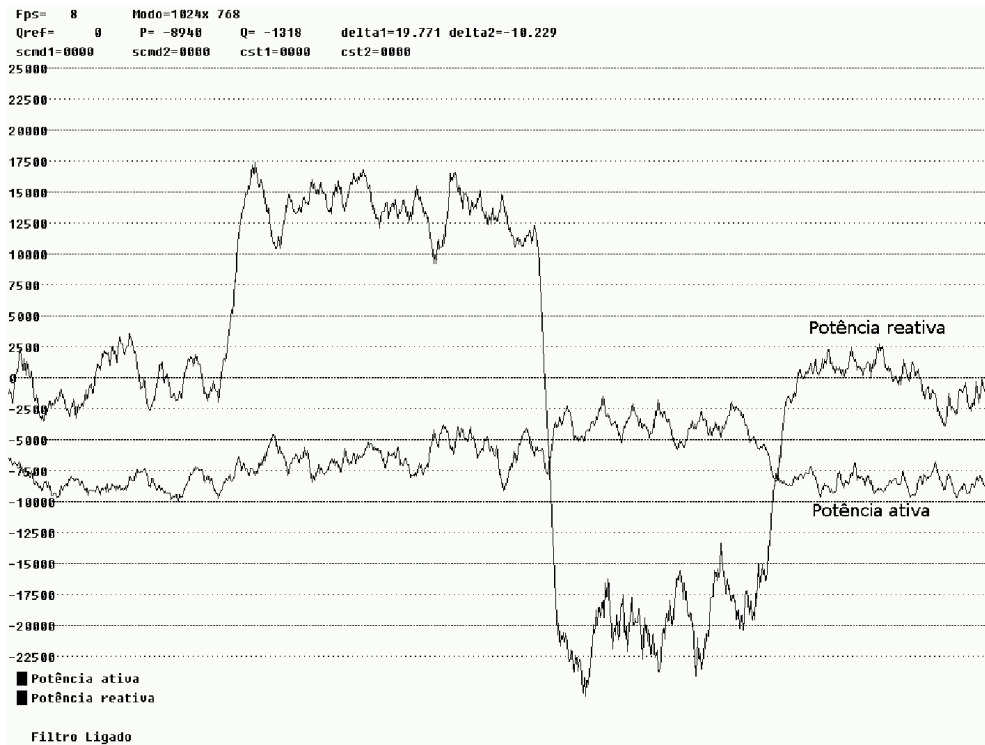


Figura 6.23: Tela de Potências do Supervisor Injetando Potência Ativa e Reativa.

Tabela 6.1: Valores do Esquema de Simulação da Planta no ATP

Dispositivo	Valor	Valor em PU
Potência instalada	30 kW	100%
Tensão de linha (pico)	180 V	100%
Resistência das chaves	$10\mu\Omega$	0,0003%
Resistência da Linha	$0,5\ \Omega$	15,43%
Reatância da Linha	$0,075\ \Omega$	2,33%
Reatância primária dos transformadores	$0,032\ \Omega$	1%
Reatância secundária dos transformadores	$0,032\ \Omega$	1%
Reatância do filtro capacitivo	$56,43\ \Omega$	1741,87%

Nas Figuras 6.24 e 6.25 são mostradas a tensão e a corrente de linha da fase “a” respectivamente, onde pode-se observar as distorções causadas pela redução das indutâncias dos transformadores e da linha. As formas de onda são bem próximas das obtidas experimentalmente e comprovam que as diferenças das simulações do Capítulo 2 são devidas realmente as diferenças dos valores das indutâncias na prática.

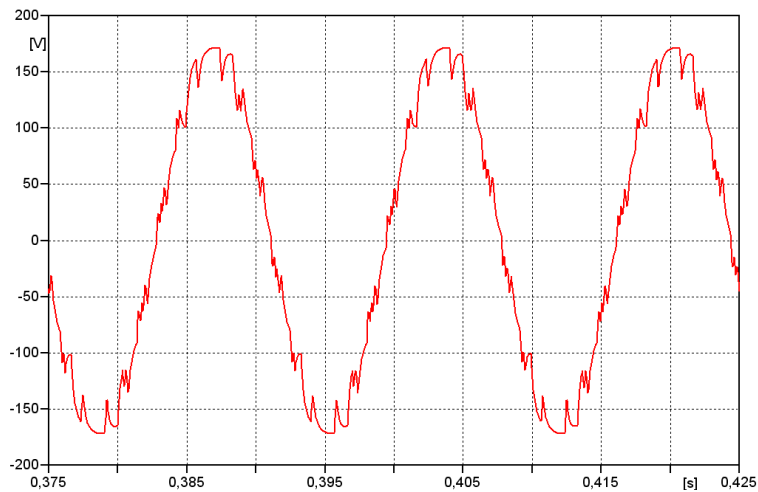


Figura 6.24: Tensão na Fase “a”.

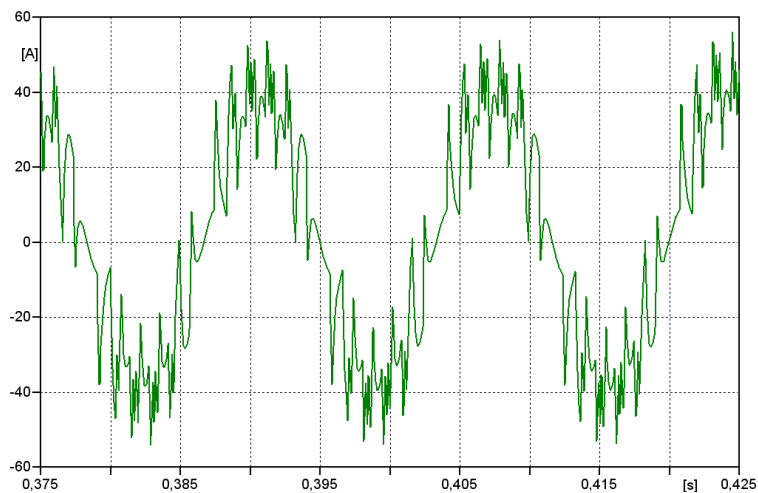


Figura 6.25: Corrente na Fase “a”.

6.4 Conversores CC-CA com Modulação PWM

A baixa qualidade das correntes obtidas com a técnica de chaveamento híbrido (multipulso + eliminação seletiva de harmônicos) motivou a modificação do controle de chaveamento dos inversores VSI do sistema de geração fotovoltaico. Como a estratégia de chaveamento foi implementada nos DSPs essa alteração foi executada de maneira rápida modificando apenas algumas linhas do algoritmo original de controle.

Seguindo a mesma metodologia adotada no Capítulo 2 foi escolhida a estratégia de chaveamento senoidal PWM (Mohan, Undeland e Robbins, 1995). Nessa estratégia

uma portadora triangular com frequência fixa é comparada com três sinais de referência triangular com amplitude, frequência e fase controladas. Essas tensões de referência são as tensões que os conversores devem sintetizar em seus terminais de saída para garantir que a potência convertida pelos painéis solares sejam injetadas na rede elétrica CA. Para evitar mudanças profundas na estratégia de chaveamento as tensões de referência trifásicas são sintetizadas com amplitude constante. Ou seja, as tensões de referência enviadas para o controlador PWM têm somente a frequência e a fase controladas.

A Figura 6.26 mostra a forma de onda de tensão de fase com 3 níveis distintos enquanto que a Figura 6.27 mostra a tensão de fase com 5 níveis obtidas pela conexão dos dois conversores CC-CA em série. Os resultados anteriores foram obtidos antes dos conversores serem conectados em paralelo com a rede CA. A frequência de chaveamento de cada um dos conversores é aproximadamente 6 kHz.

A Figura 6.28 mostra a forma de onda da tensão de fase e da corrente nos terminais dos conversores CC-CA quando o sistema está flutuando. Apesar do chaveamento em frequência elevada não observa-se uma melhora significativa na forma de onda da corrente. A Figura 6.29 mostra a tensão de fase e a corrente de linha quando o conversor CC-CA está injetando potência ativa no sistema elétrico quando o conversor CC-CC é controlado com um algoritmo de rastreamento de máxima potência (MPPT).

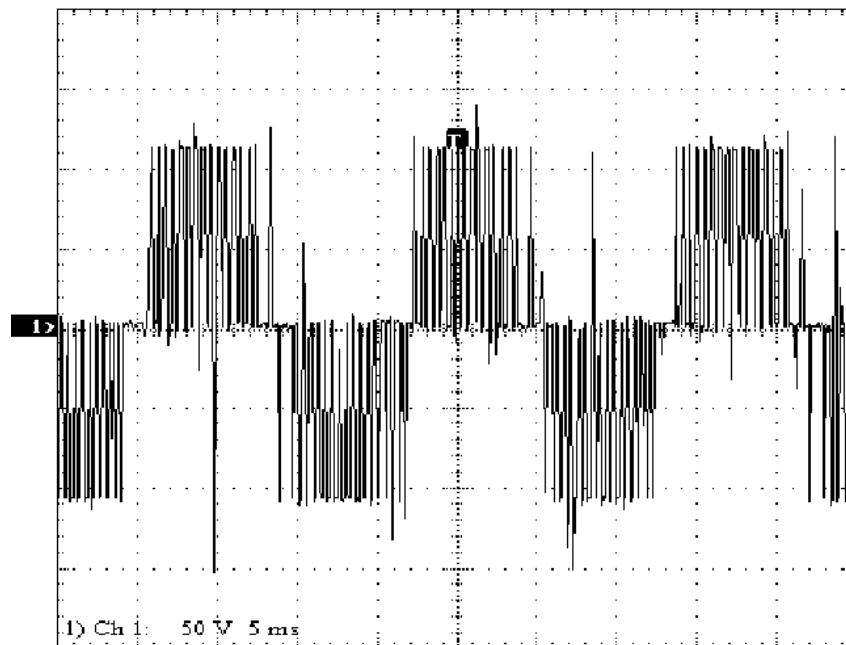


Figura 6.26: Tensão PWM de 3 Níveis na Saída de um Conversor em Aberto.

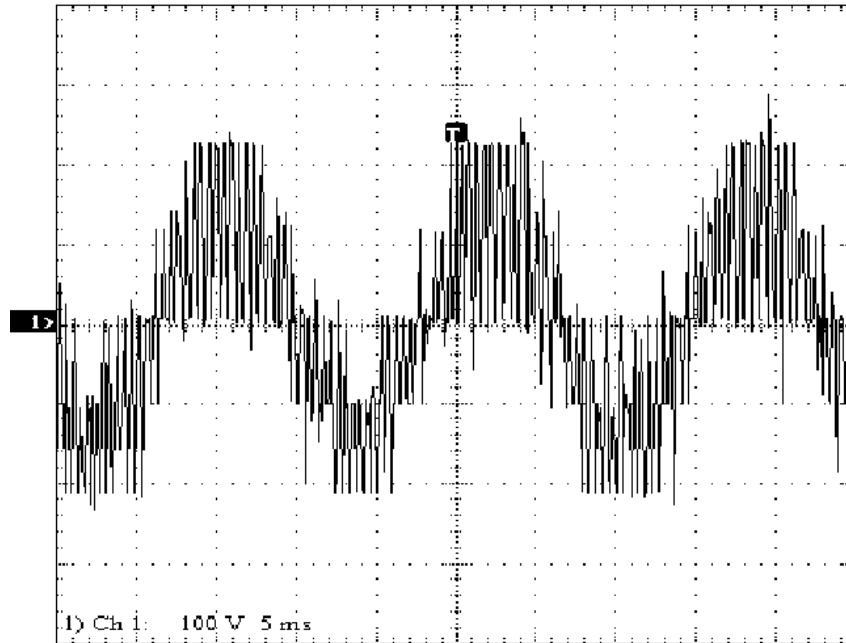


Figura 6.27: Tensão PWM de 5 Níveis na Saída dos Conversores em Aberto.

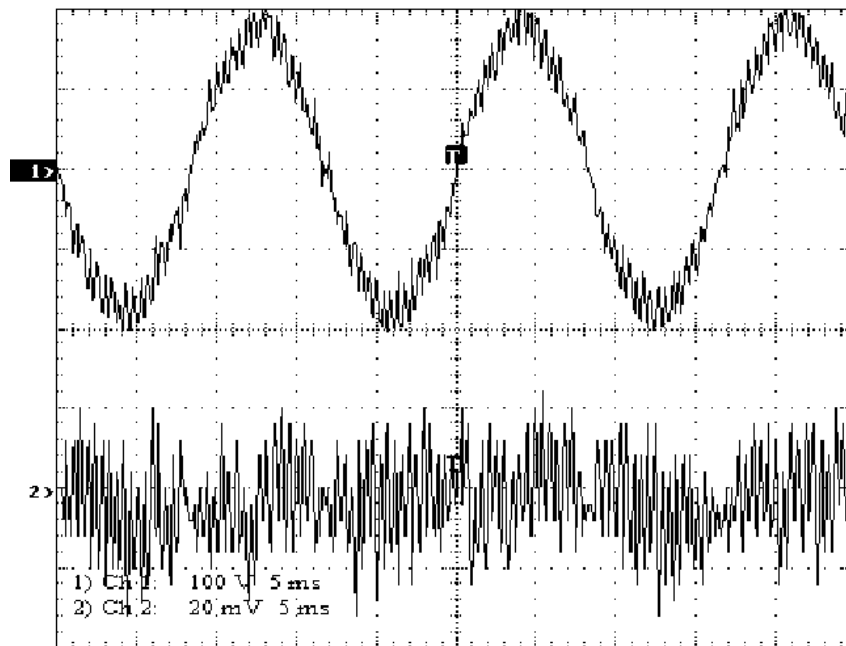


Figura 6.28: Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1mV/A] do Conversor Flutuando.

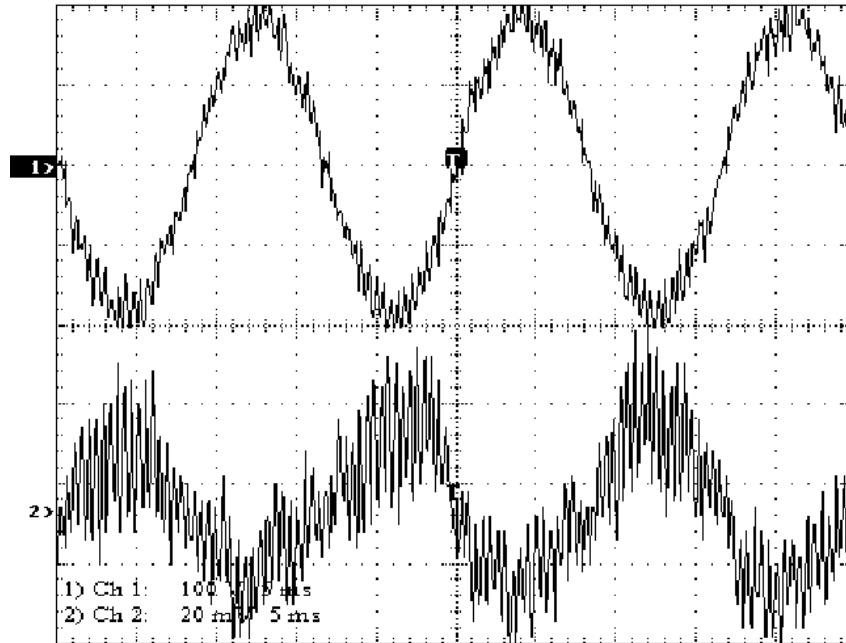


Figura 6.29: Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1mV/A] Injetando Potência Ativa com MPPT.

A Figura 6.30 mostra as formas de onda da tensão de fase e da corrente de linha quando os conversores CC-CA estão absorvendo 20 kVAR de potência reativa do sistema elétrico. Neste caso os conversores estáticos operam com uma característica indutiva, isto é, corrente atrasada de 90° ($\pi/2rad$) em relação à tensão de fase. A Figura 6.31 mostra as formas de onda da tensão de fase e da corrente de linha quando o conversor CC-CA está injetando 20 kVAR de potência reativa no sistema CA. Neste caso os conversores operam com uma característica capacitiva sintetizando uma corrente adiantada de 90° ($\pi/2rad$) em relação a tensão de fase. Pode se observar nas duas figuras que as correntes apresentam distorções devido ao baixo valor de indutância de dispersão dos transformadores de conexão utilizados.

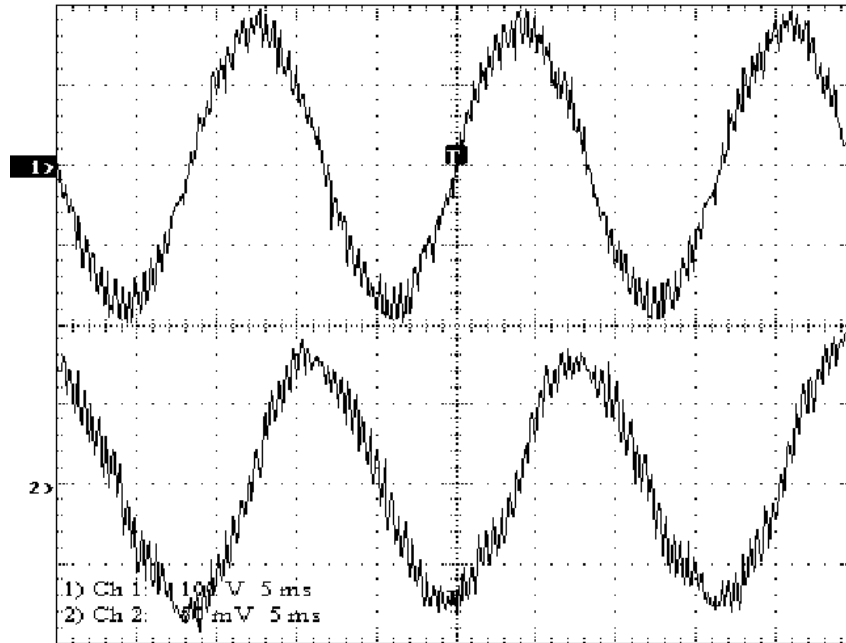


Figura 6.30: Tensão PWM (superior) e Corrente PWM (inferior) [escala: $1\text{mV}/\text{A}$] com o Sistema Sintetizando 20 kVAR Indutivo.

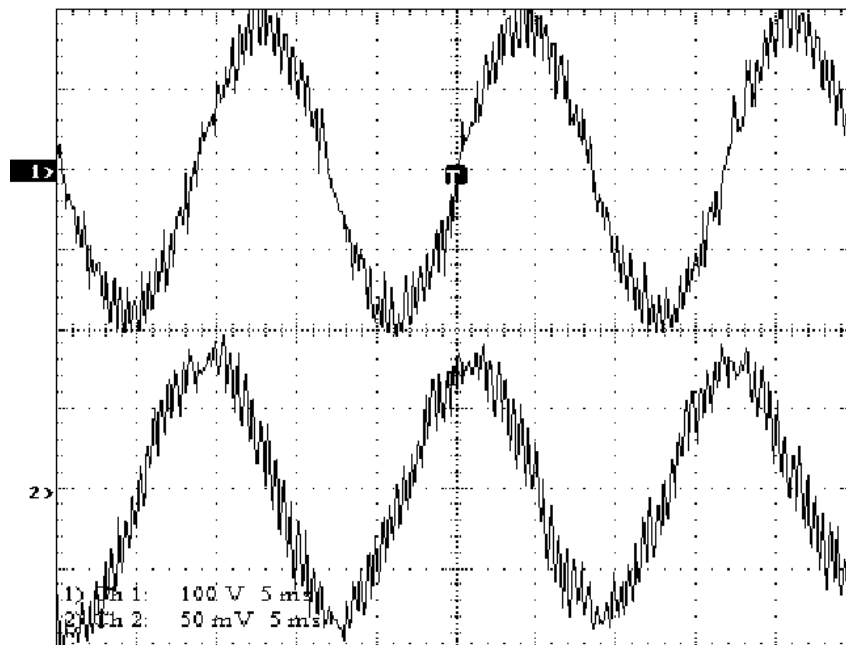


Figura 6.31: Tensão PWM (superior) e Corrente PWM (inferior) [escala: $1\text{mV}/\text{A}$] com o Sistema Sintetizando 20 kVAR Capacitivo.

A Figura 6.32 mostra as forma de onda da tensão e da corrente de linha quando os conversores VSI's estão absorvendo 35 kVAR de potência reativa do sistema elétrico e a Figura 6.33 mostra as formas de onda da tensão e da corrente de linha para o caso em que os conversores injetam 35 kVAR de potência reativa. Note que a forma de onda da

corrente de linha melhora quando uma potência reativa maior é absorvida/gerada pelos conversores.

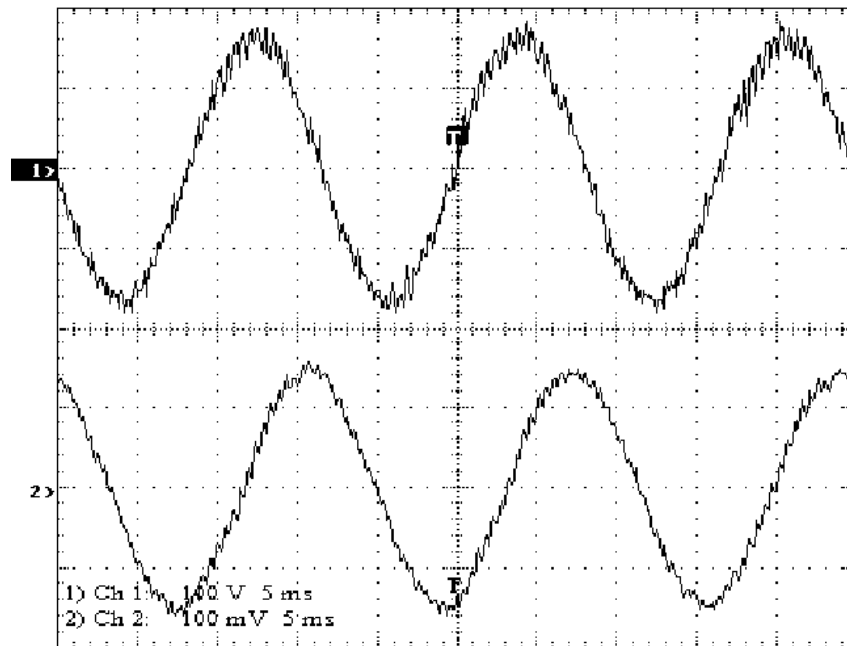


Figura 6.32: Tensão PWM (superior) e Corrente PWM (inferior) [escala: 1mV/A] com o Sistema Sintetizando 35 kVAR Indutivo.

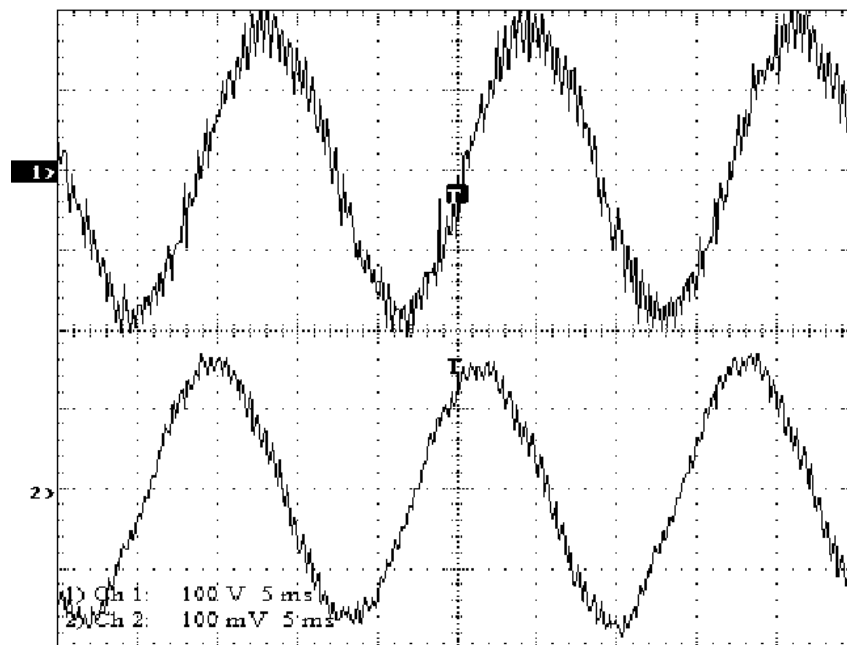


Figura 6.33: Tensão (superior) e Corrente (inferior) [escala: 1mV/A] [escala: 1mV/A] com o Sistema Sintetizando 35 kVAR Capacitivo.

6.5 Rotinas dos DSPs

Para a verificação do tempo de execução das rotinas implementadas nos DSPs, no início de cada interrupção um pino externo é ligado e no final da interrupção é desligado, o que permite medir externamente com um osciloscópio o tempo de execução do código da rotina de interrupção e verificar a compatibilidade com a frequência de amostragem adotada. Alguns algoritmos que não possuem tempo de execução crítico utilizam variáveis do tipo *ponto-flutuante* que são emuladas pelo processador de *ponto-fixa*, a aritmética de *ponto-flutuante* emulada é muito mais lenta que a aritmética de *ponto-fixa*, mas apresenta maior precisão nos resultados dos cálculos.

Na Figura 6.34 é mostrado o tempo de execução da interrupção do algoritmo “MPPT”, a frequência de amostragem utilizada é de 500Hz , o tempo de execução aproximado é de $375\mu\text{s}$, que é aproximadamente 5 vezes menor que o período da frequência de amostragem de $2000\mu\text{s}$. O algoritmo “MPPT” é implementado em linguagem c e assembly e utiliza variáveis de *ponto-flutuante* pois o tempo de execução não é crítico.

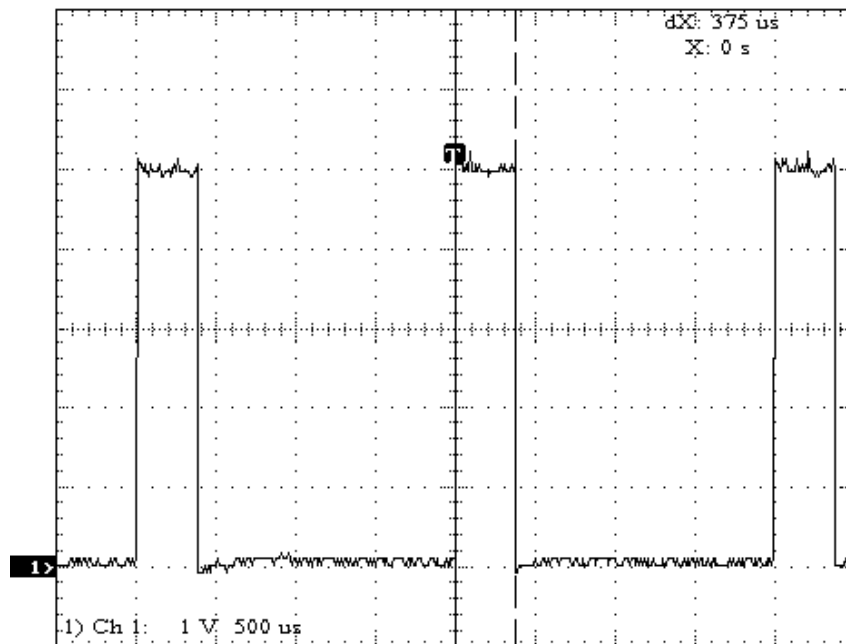


Figura 6.34: Tempo de Execução da Rotina “MPPT”.

Na Figura 6.35 é mostrado o tempo de execução da interrupção do algoritmo de “controle”, a frequência de amostragem utilizada é de 1kHz , o tempo de execução aproximado é de $730\mu\text{s}$, que é menor que o período da frequência de amostragem de $1000\mu\text{s}$. O algoritmo de “controle” é implementado em linguagem c e assembly e utiliza variáveis

de *ponto-flutuante* pois a precisão é crítica neste caso.

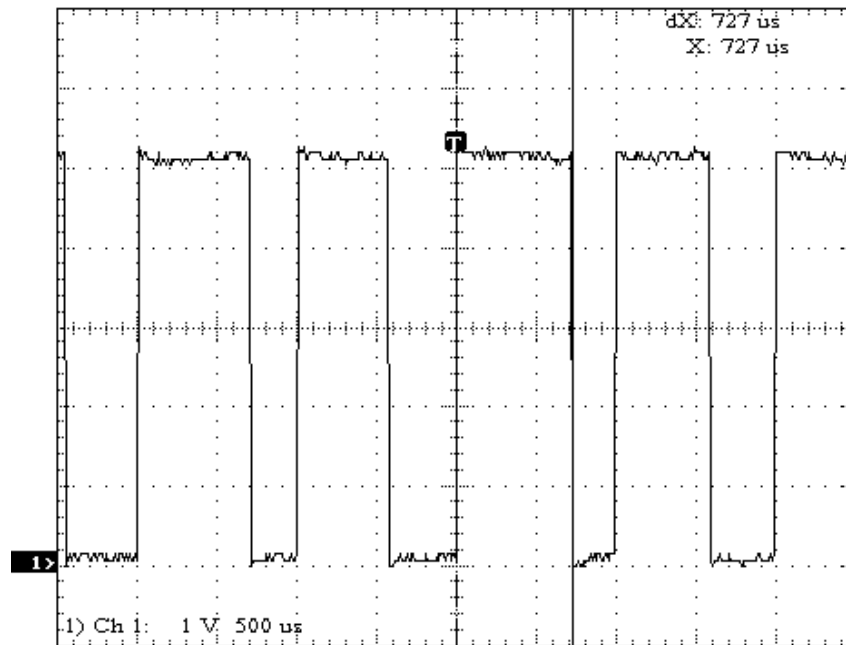


Figura 6.35: Tempo de Execução da Rotina “controle”.

Na Figura 6.36 é mostrado o tempo de execução da interrupção do algoritmo de “disparo”, a frequência de amostragem utilizada é de $35kHz$, o tempo de execução aproximado é de $25\mu s$, que é menor que o período da frequência de amostragem de $\cong 28,6\mu s$. O algoritmo de “disparo” é implementado totalmente em assembly e utiliza variáveis de *ponto-fixe* pois o tempo de execução é crítico neste caso.

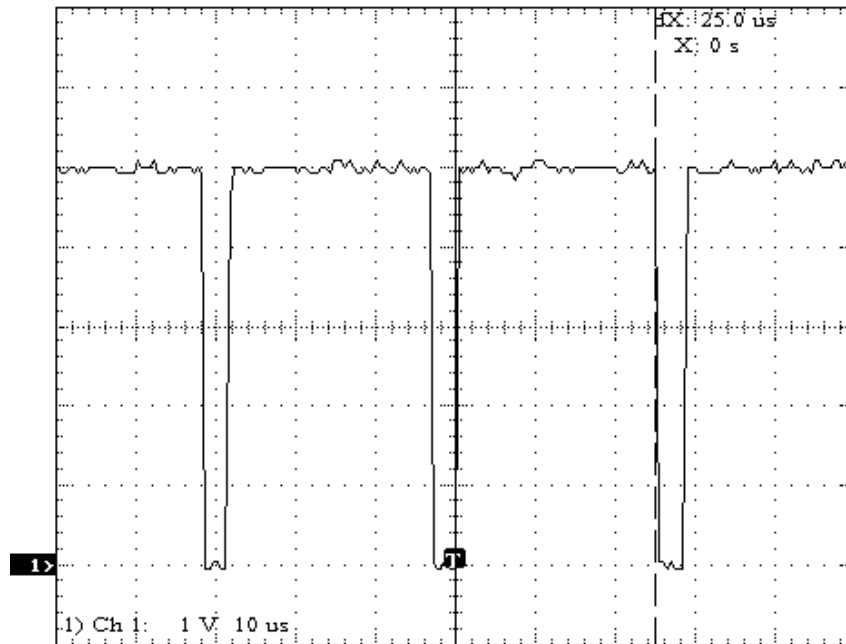


Figura 6.36: Tempo de Execução da Rotina “disparo”.

Na Figura 6.37 é mostrado o tempo de execução da interrupção do algoritmo de “disparo pwm”, a frequência de amostragem utilizada é de $30kHz$, o tempo de execução aproximado é de $29\mu s$, que é menor que o período da frequência de amostragem de $\cong 33,33\mu s$. O algoritmo de “disparo pwm” é implementado totalmente em assembly e utiliza variáveis de *ponto-fixe* pois o tempo de execução é crítico neste caso.

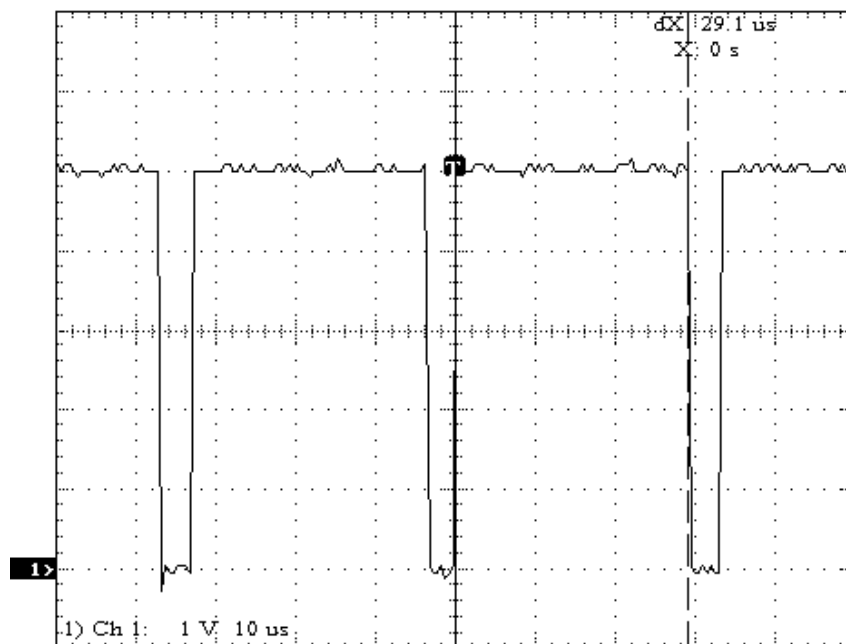


Figura 6.37: Tempo de Execução da Rotina “disparo pwm”.

O tempo de execução é crítico nos algoritmos de disparo pois a resolução dos disparos está ligada diretamente a frequência de amostragem e pode ser calculada para $60Hz$ por:

$$\Delta\theta = \frac{360^\circ \cdot 60}{F_s} \text{ ou } \Delta\theta = \frac{2\pi \cdot 60}{F_s} \quad (6.1)$$

No caso do algoritmo “disparo” a resolução é de $\Delta\theta = 0,617^\circ$, a resolução do algoritmo “disparo pwm” é de $\Delta\theta = 0,72^\circ$. Quanto menor a resolução, menor são os erros de disparo. Para diminuir o tempo de execução dos algoritmos de disparo foi utilizada a função *seno* baseada em séries infinitas, que são mais rápidas que as de interpolação por tabela.

6.6 Conclusões Parciais

Neste Capítulo foram demonstrados os resultados do ensaios dos conversores CC-CC e CC-CA operando como compensador estático e como sistema de geração interligado a rede elétrica CA. Os resultados experimentais apresentam algumas discrepâncias quando comparados com as formas de onda dos resultados das simulações. Essas discrepâncias podem ser explicadas pelas baixas indutâncias de dispersão dos transformadores de conexão e da linha. Contudo esses resultados não invalidam os modelos e a estratégia de controle utilizadas que tiveram desempenho equivalente aos previstos pelas simulações. Também foram demonstrados os testes de tempo de execução dos algoritmos utilizados nos DSP, e discutidos os pontos críticos dos mesmos.

Capítulo 7

Conclusões e Trabalhos Futuros

7.1 Conclusões Finais

Neste trabalho foram apresentadas as etapas da implementação de um sistema experimental fotovoltaico de geração de energia elétrica operando em paralelo com a rede elétrica. Foram discutidas e analisadas as partes constituintes do sistema de geração distribuída, os conversores estáticos utilizados e suas topologias. Modelos digitais dos dispositivos da planta de geração juntamente com os algoritmos de controle foram desenvolvidos para avaliar o desempenho do sistema. Foram discutidas as técnicas de rastreamento do ponto de máxima potência dos painéis fotovoltaicos e também a técnica de disparo multipulso com eliminação seletiva de harmônicos para aplicação no controle dos conversores CC-CC e CC-CA respectivamente.

Um algoritmo digital de sincronismo (PLL) foi desenvolvido para detectar a frequência e fase da tensão de um sistema de potência CA com precisão e rapidez. O circuito PLL foi implementado em um processador digital de sinal (DSP) e utilizado para sincronizar os disparos das chaves semicondutoras dos conversores estáticos CC-CA com objetivo de garantir que a potência convertida pelos painéis solares fotovoltaicos seja transferida para a rede CA com eficiência.

Todos os controladores do sistema de geração foram digitalizados e implementados em DSPs com o objetivo de controlar os conversores CC-CC e CC-CA da planta. Com o desenvolvimento de um *emulador* do DSP utilizado, todos algoritmos puderam ser testados antes de serem utilizados no sistema.

Em paralelo foram projetados e confeccionados circuitos de condicionamento para os DSPs acionarem os *drivers* dos conversores.

Foi desenvolvido um sistema de gerenciamento de falhas e um sistema supervisorio. Esses sistemas foram projetados para serem executados em um computador remoto e permitirem a observação do funcionamento da planta de geração fotovoltaica tanto como compensador estático de reativo como sistema de geração disperso, interligado ao sistema elétrico.

Os resultados experimentais, apesar de discrepâncias nas formas de onda das correntes, devido às reatâncias de dispersão dos transformadores de conexão, demonstraram que as estratégias de controle adotadas e implementadas funcionam perfeitamente atendendo às especificações de projeto.

Todas as simulações e resultados experimentais demonstraram que as topologias e as técnicas adotadas para o controle dos conversores CC-CC e CC-CA são perfeitamente empregadas no controle do sistema experimental de geração de energia elétrica implementado.

7.2 Trabalhos Futuros

Como temas para futuros desdobramentos desse trabalho de pesquisa são sugeridos:

1. Investigar a utilização aplicação de técnicas SVM-PWM (*Space Vector Modulation – Pulse Width Modulation*) para chavear os conversores CC-CA para permitir mais flexibilidade no controle.
2. Integrar as variáveis de controle dos conversores CC-CC no sistema supervisorio dos conversores CC-CA e implementar um protocolo para controlar e monitorar a planta de geração transmitindo dados pela internet.
3. Investigar e desenvolver técnicas de rastreamento de máxima potência mais refinadas para controle dos conversores CC-CC.
4. Investigar as conseqüências sobre a qualidade da energia elétrica do sistema elétrico da UFJF com a operação da planta de geração fotovoltaica.

Referências Bibliográficas

- Akagi, H., Kanazawa, Y. e Nabae, A. (1984), ‘Instantaneous reactive power compensator comprising switching devices without energy storage components’, *IEEE Transactions on Industry Applications* **IA-20**(3), 625–630.
- ANSI/IEEE-519 (1992), *IEEE Guide for Harmonic Control and Reactive Compensation of Static Power Converters*.
- ANSI/IEEE-929 (2000), *IEEE Recommended Practice for Utility Interface of Photovoltaic Systems*.
- ATP (2005), ‘Alternative transients program/electromagnetic transients program’, *ATP*.
- Barbosa, P. G. (2000), Compensador Série Síncrono Estático Baseado em Conversores VSI Multipulso, PhD thesis, COPPE / UFRJ, Rio de Janeiro, RJ, Brasil.
- Barbosa, P. G., Rolim, L. G. B., Watanabe, E. H. e Hanitsch, R. (1998), ‘Control startegy for grid-connected dc-ac converters with load power factor correction’, *IEE Proceedings online*.
- BOSCH (2005), ‘Bosch can protocol’, *BOSCH*.
- Carletti, R. L. (2005), Sistema de geração dispersa baseado em células fotovoltaicas: Estudo das estratégias de chaveamento e de controle dos conversores cc-ca, Master’s thesis, UFJF, Juiz de Fora, MG, Brasil.
- Carletti, R. L., Lopes, L. C. G. e Barbosa, P. G. (2003), ‘A dispersed generation system based on photovoltaic cells: converter configuration ans switching strategy’, *The 7th Brazilian Power Eletronics Conference* pp. 404 – 409.
- Castañer, L. e Silvestre, S. (2002), *Modelling Photovoltaic Systems*, John Wiley and Sons.
- Dugan, R. C., Key, T. S. e Ball, G. J. (2006), ‘Distributed resources standards’, *IEEE Industry Applications Magazine* **12**(1), 27–34.
- Hohm, D. P. e Ropp, M. E. (2003), ‘Comaparitive study of maximum power point tracking algorithms’, *Progress in Photovoltaics: Research and Applications* (11), 47–62.

- Instruments, T. (1992), *TMS320F2x User's Guide*, <http://www.ti.com>.
- Jiang, J.-A., Huang, T.-L., Hsio, Y.-T. e Chen, C.-H. (2005), 'Maximum power tracking for photovoltaic power systems', *Tamkang Journal of Science and Engineering* **8**(2), 147–153.
- John, V., Zhihong, Y. e Kolwalkar, A. (2004), 'Investigation of anti-islanding protection of power converter based distributed generators using frequency domain analysis', *IEEE Transactions on Power Electronics* **19**(5), 1177–1183.
- Koutroulis, E., Kalaitzakis, K. e Voulgaris, N. (2001), 'Development of a microcontroller-based photovoltaic maximum power point tracking control system', *IEEE Transactions on Power Electronics* **16**(1), 46–54.
- Mohan, N., Undeland, T. e Robbins, W. P. (1995), *Power Electronics: Converters, Applications and Design*, 2nd. edn, John Wiley e Sons Inc.
- Ogata, K. (1994), *Discrete-Time Control Systems*, 2nd edn, Prentice Hall.
- Ogata, K. (1997), *Modern Control Engineering*, 3rd edn, Prentice Hall.
- Ogrodzki, J. (1994), *Circuit Simulation Methods and Algorithms*, CRC-PRess.
- Raghuram, R. (1989), *Computer Simulation of Electronic Circuits*, John Wiley e Sons.
- Rodrigues, M. C. B. (2004), Inversor boost multinível em corrente e sua aplicação no processamento de energia em sistemas fotovoltaicos monofásicos conectados à rede elétrica, Master's thesis, Universidade Federal de Juiz de Fora.
- Santos, J. e Antunes, F. L. (2003), 'Maximum power point tracker for pv systems', *Rio 3 - World CLimate e Energy Event* pp. 75–80.
- Sasso, E. M., Sotelo, G. G., Ferreira, A. A., Watanabe, E. H., Aredes, M. e Barbosa, P. G. (2002), 'Investigação dos modelos de circuitos de sincronismo trifásicos baseados na teoria das potências real e imaginária instantâneas (p -pll e q -pll)', *Anais do Congresso Brasileiro de Automática*.
- Sayigh, A. (2003), 'World renewable energy', *Oral Plenary of The Third World Conference on Photovoltaic Conversion* pp. 2560–2561. Osaka, Japão.
- SCILAB (2005), 'Scientific laboratory rule book', *SCILAB*.
- SDL (2005), 'Simple directmedia layer', *SDL team*.
- Smith, S. (1999), *The Scientist and Engineer's Guide to Digital Signal Processing*, second edn, California Techinal Publishing, www.dspguide.com, chapter 3, pp. 48–58.

Watanabe, E. H., Stephan, R. M. e Aredes, M. (1993), 'New concepts of instantaneous active and reactive power for three phase system and generic loads', *IEEE Transactions on Power Delivery*.

Apêndice A

Princípio de Funcionamento do Sistema de Geração Fotovoltaico

Conforme mencionado no Capítulo 2, o controlador dos conversores CC-CA do sistema de geração fotovoltaico é semelhante ao controlador de um compensador estático síncrono (STATCOM). O princípio de funcionamento desse controlador pode ser explicado através do circuito da Figura A.1 (a) onde o sistema elétrico e o inversor são representados, em regime permanente, pelas fontes \dot{V}_S e \dot{V}_I , respectivamente. Desprezando os harmônicos gerados pelos conversores estáticos e observando o diagrama fasorial da Figura A.1 (b) pode-se escrever que:

$$P_I = 3 \frac{V_I V_S}{X_L} \sin \delta \quad (\text{A.1})$$

e,

$$Q_I = 3 \frac{V_S}{X_L} [V_S - V_I \cos \delta] \quad (\text{A.2})$$

onde δ é o ângulo de defasagem entre as tensões do sistema e do conversor em (*rad*), X_L representa a reatância equivalente de Thévenin composta pela reatância de dispersão dos transformadores de conexão dos conversores e da linha de transmissão em (Ω).

De (A.1) e (A.2) pode-se concluir, em relação ao fluxo de potência ativa, que:

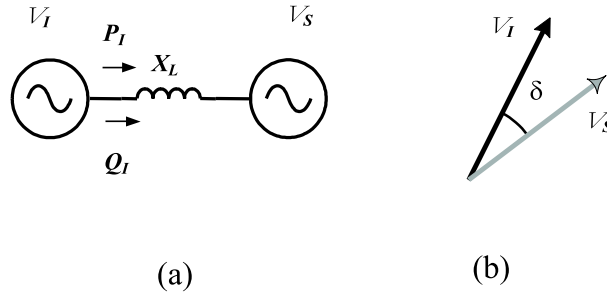


Figura A.1: (a) Diagrama Unifilar do Sistema Fotovoltaico; (b) Diagrama Fasorial.

- i.* Se a tensão V_I está adiantada em relação a tensão V_S , $-90^\circ < \delta < 0^\circ$, haverá um fluxo de potência ativa do conversor para a rede CA, ou seja, $P_I < 0$;
- ii.* Se a tensão V_I está atrasada em relação a tensão V_S , $0^\circ < \delta < +90^\circ$, haverá um fluxo de potência ativa da rede CA para o conversor, ou seja, $P_I > 0$;
- iii.* Se a tensão V_I está em fase com a tensão V_S ($\delta = 0^\circ$) e $|V_I| = |V_S|$, não haverá fluxo de potência ativa e reativa nos terminais do conversor.

Já, considerando o caso em que o conversor esteja entregando uma quantidade de potência ativa à rede CA, pode-se ter a situação em que:

- i.* Se $V_I \cos \delta > V_S$ haverá, além do fluxo de potência ativa, um fluxo de potência reativa $Q_I < 0$ nos terminais do conversor;
- ii.* Se $V_I \cos \delta < V_S$ haverá, além do fluxo de potência ativa, um fluxo de potência reativa $Q_I < 0$ nos terminais do conversor;
- iii.* Por fim, se a tensão V_I está em fase com a tensão V_S ($\delta = 0^\circ$) e $V_I = V_S$, não haverá fluxo de potência ativa e reativa nos terminais do conversor.

Baseado nessas características é que foi desenvolvido o controlador do sistema de geração fotovoltaico descrito na Seção 2.5. Quando o conversor *boost* injeta a energia CC convertida pelos painéis solares fotovoltaicos no elo CC da planta fotovoltaica a tensão CC e conseqüentemente as tensões trifásicas geradas pelos conversores CC-CA têm suas amplitudes aumentadas. Com o aumento da amplitude das tensões alternadas a potência reativa nos terminais dos conversores altera seu valor. O controlador PI altera o ângulo de fase δ fazendo com que os conversores passem a fornecer potência ativa para a rede CA.

Apêndice B

Listagem dos *Models* Usados no ATP/EMTP para Simulação

B.1 Modelos para Acionamento do Conversor CC-CC

B.1.1 diodo_pv.mod

```
MODEL diodo_pv

comment *****
* Luis Claudio Gambôa Lopes 06/11/2002 *
* *
* Function: fotovoltaic painel *
* Inputs : voltage and sun power *
* Output : current *
* *
***** endcomment

INPUT V

DATA Ns {dflt:3}
Np {dflt:1}
Vt {dflt:3.6872}
Is {dflt:4.256e-5}
Rs {dflt:0.001}

VAR Vd
Vi
Ix
x0
x1
fx1
dfox1
io
k

OUTPUT io

INIT

write('----- PAINEL Module By Gambôa -----')

ENDINIT

EXEC
comment*****
teste
*****endcomment

Vi:=V

IF Vi =0 THEN
Vi:=1e-6
ELSE
Vi:=Vi/Ns
ENDIF

IF Vi > 500 THEN
Vi:=500
```

```
ENDIF

x0:=100
x1:=0
k:=0

WHILE abs(x0-x1) > 1e-6 AND k < 100 DO

x1:=x0
fx1:=((x1-Vi)/Rs)+(Np*Is*(exp(x1/Vt)-1))
dfox1:= (1/Rs)+((Np*Is/Vt)*exp(x1/Vt))

IF dfox1 = 0 THEN
x0:=200
ELSE
x0:=x1-(fx1/dfox1)
ENDIF

k:=k+1
ENDWHILE

io:= (x0-Vi)/Rs

ENDEXEC

ENDMODEL

RECORD
diodo_pv.io as pnI
diodo_pv.V as pnV

comment ****
teste
**** endcomment
```

B.1.2 mppt.mod

```
MODEL mppt

comment *****
* Luis Claudio Gambôa Lopes 06/11/2002 *
* *
* Function: fotovoltaic painel *
* Inputs : voltage and sun power *
* Output : current *
* *
***** endcomment

INPUT vm
im
vo
io

DATA ialfa {dflt:1}

VAR
alfa
p0
p1
pd
vd
```



```

po
tp
ndp      dp
        vf
        if0
if1
vf0
        vf1
io0
io1
OUTPUT dp
ndp

INIT

        write('----- PAINEL Module By Gamb6a -----')
        tp:=0
        alfa:=ialfa
        p0:=0
        p1:=0
        pd:=0
        dp:=0
        vd:=50
        if0:=0
        vf0:=0
        io0:=0
ENDINIT

EXEC
comment*****
        teste
*****endcomment

vf1:=vf0
vf0:=(vm*timestep)+(0.005*vf1)/(0.005+timestep)

if1:=if0
if0:=(im*timestep)+(0.005*if1)/(0.005+timestep)

if tp > 499 then
        tp:=0

        p1:= p0
        p0:= vf0*if0

        pd:= p0-p1

        if pd < 0 then
                vd:= vd-alfa
                alfa:=-1*alfa
        else
                vd:= vd+alfa
        endif

        if vd < 0 then
                vd:=0
        endif

        if vd > (499*0.45) then
                vd:=(499*0.45)
        endif

endif

        tp:=tp+10

        if tp < vd then
                dp:=1
                ndp:=0
        else
                dp:=0
                ndp:=1
        endif

io1:=io0
io0:=(io*timestep)+(0.005*io1)/(0.005+timestep)

if io0 < 0 then
        io0:=1e-3
endif

        po:=io0*vo

```

```

ENDEXEC
ENDMODEL

RECORD
        mppt.vf0 as VI
        mppt.vo as VO
        mppt.if0 as II
        mppt.io0 as IO
        mppt.p0 as PI
        mppt.po as PO
        mppt.dp as DP
        mppt.tp as TP
        mppt.pd as PD
        mppt.alfa as ALFA
        mppt.vd as VD

comment ****
teste
**** endcomment

```

B.2 Modelos para Acionamento do Conversor CC-CA

B.2.1 pll.mod

```

MODEL pll

comment *****
* Luis Claudio Gamb6a Lopes 30/01/2006 *
* *
* Function: detect phase angle and generate trigger *
* Inputs : voltage phase a, b and c *
* Output : trigger for phase a, b and c *
***** endcomment

INPUT v1 -- tensão de entrada fase A [V]
        v2 -- tensão de entrada fase B [V]
        v3 -- tensão de entrada fase C [V]

DATA K {dflt:999} -- Gain
        TI {dflt:0.002} -- Int Time

VAR dp1 -- Erro de potência [W]
        dp2
        w -- Frequência angular [Rad/s]
        w1
        t01 -- angulo de saída VSC1 [Rad]
        t1 -- memory for output angle [Rad]
        t2 -- memory for output angle [Rad]
        t3
        va -- Clark transformation of v1 [V]
        vb -- Clark transformation of v2 [V]
        ia -- feedback current [A]
        ib -- feedback current [A]
        TS

OUTPUT t1

INIT t1:=1
        ia:=COS(0)
        ib:=SIN(0)
        write('----- PLL Module By Gamb6a -----')
        va:=0
        vb:=0
        dp1:=0
        t1:=0
        t2:=0
        w:=0
        TS:=timestep
        write('K =',K)
        write('TI =',TI)
        write('T =',TS)
ENDINIT

EXEC
        va:=sqrt(2/3)*(v1-((v2+v3)/2))
        vb:=sqrt(2/3)*((sqrt(3)/2)*(v2-v3))

        IF va = 0 AND vb = 0 THEN
                va:=1e-9
        ENDIF

```

```

dp2:=dp1
dp1:=(va*ia+vb*ib)/sqrt((va*va)+(vb*vb))

w1:=w;
w:=((((K*TS)*((TS/(2*TI))+1))*dp1
+((K*TS)*((TS/(2*TI))-1))*dp2))/0.0002)+w1

t3:=t2
t2:=t1
t1:=((K*TS)*((TS/(2*TI))+1))*dp1
+((K*TS)*((TS/(2*TI))-1))*dp2+(2*t2)-t3

WHILE t1 < 0 DO
t1:=t1+(2*pi)
t2:=t2+(2*pi)
t3:=t3+(2*pi)
ENDWHILE

WHILE t1 > (2*pi) DO
t1:=t1-(2*pi)
t2:=t2-(2*pi)
t3:=t3-(2*pi)
ENDWHILE

ia:=COS(t1)
ib:=SIN(t1)

ENDEXEC

ENDMODEL

RECORD
pll.w as w
pll.t1 as delta

comment ****
RECORD
pll.w as w
pll.t1 as tetha
**** endcomment

```

B.2.2 disparo.mod

```

MODEL disparo

comment *****
* Luis Claudio Gambôa Lopes 06/11/2002 *
* *
* Function: detect phase angle and generate trigger *
* Inputs : voltage phase a, b and c *
* Output : trigger for phase a, b and c *
* *
*****endcomment

INPUT v1 -- tensão de entrada fase A [V]
v2 -- tensão de entrada fase B [V]
v3 -- tensão de entrada fase C [V]
delta -- angulo de referência [rad]

DATA K {dflt:999} -- Gain
TI {dflt:0.002} -- Int Time
alfa1 {dflt:0.15259} -- Harmonic Elimination angle1
alfa2 {dflt:0.42582} -- Harmonic Elimination angle2
alfa3 {dflt:0.48454} -- Harmonic Elimination angle3

VAR dp1 -- Erro de potência [W]
dp2
w -- Frequência angular [Rad/s]
w1
t01 -- angulo de saída VSC1 [Rad]
t1 -- memory for output angle [Rad]
t2 -- memory for output angle [Rad]
t3
va -- Clark transformation of v1 [V]
vb -- Clark transformation of v2 [V]
ia -- feedback current [A]
ib -- feedback current [A]
TS
tx1
twopi
PLLa
PLLb
PLLc
Thta
Thtb
Thtc
Sfa
Sfb
Sfc
NSfa
NSfb

```

```

NSfc

OUTPUT Sfa -- trigger phase a [V]
Sfb -- trigger phase b [V]
Sfc -- trigger phase c [V]
NSfa -- trigger phase a (INVERTED) [V]
NSfb -- trigger phase b (INVERTED) [V]
NSfc -- trigger phase c (INVERTED) [V]

INIT t1:=1
ia:=COS(0)
ib:=SIN(0)
twopi:=2*pi
write('----- PLL Module By Gambôa -----')
va:=0
vb:=0
dp1:=0
t1:=0
t2:=0
w:=0
TS:=timestep
write('K =',K)
write('TI =',TI)
write('T =',TS)
ENDINIT

EXEC
va:=sqrt(2/3)*(v1-((v2+v3)/2))
vb:=sqrt(2/3)*((sqrt(3)/2)*(v2-v3))

dp2:=dp1
dp1:=(va*ia+vb*ib)/sqrt((va*va)+(vb*vb))

w1:=w;
w:=((((K*TS)*((TS/(2*TI))+1))*dp1
+((K*TS)*((TS/(2*TI))-1))*dp2))/0.0002)+w1

t3:=t2
t2:=t1
t1:=((K*TS)*((TS/(2*TI))+1))*dp1
+((K*TS)*((TS/(2*TI))-1))*dp2+(2*t2)-t3

WHILE t1 < 0 DO
t1:=t1+(2*pi)
t2:=t2+(2*pi)
t3:=t3+(2*pi)
ENDWHILE

WHILE t1 > (2*pi) DO
t1:=t1-(2*pi)
t2:=t2-(2*pi)
t3:=t3-(2*pi)
ENDWHILE

ia:=COS(t1)
ib:=SIN(t1)

tx1:=t1+delta
WHILE tx1 < 0 DO
tx1:=tx1+(2*pi)
ENDWHILE

WHILE tx1 > (2*pi) DO
tx1:=tx1-(2*pi)
ENDWHILE

comment*****
---- reference angles for the converter
*****endcomment
PLLa:= tx1
PLLb:= tx1 - twopi/3
PLLc:= tx1 + twopi/3

WHILE PLLa < 0 DO
PLLa:=PLLa+(2*pi)
ENDWHILE

WHILE PLLa > (2*pi) DO
PLLa:=PLLa-(2*pi)
ENDWHILE

WHILE PLLb < 0 DO
PLLb:=PLLb+(2*pi)
ENDWHILE

WHILE PLLb > (2*pi) DO
PLLb:=PLLb-(2*pi)
ENDWHILE

WHILE PLLc < 0 DO
PLLc:=PLLc+(2*pi)
ENDWHILE

```

```

        WHILE PLLc > (2*pi) DO
            PLLc:=PLLc-(2*pi)
        ENDWHILE

Thta:= PLLa
Thtb:= PLLb
Thtc:= PLLc

comment*****
-----FASE A-----
*****endcomment

        Sfa:=0

        IF Thta >= alfa1
            AND Thta <= alfa2
            THEN
                Sfa:=1
            ENDIF

        IF Thta >= alfa3
            AND Thta <= (pi-alfa3)
            THEN
                Sfa:=1
            ENDIF

        IF Thta >= (pi-alfa2)
            AND Thta <= (pi-alfa1)
            THEN
                Sfa:=1
            ENDIF

        IF Thta >= pi
            THEN
                Thta:= Thta-pi
                Sfa:=1
                IF Thta >= alfa1
                    AND Thta <= alfa2
                    THEN
                        Sfa:=0
                    ENDIF
            ENDIF

        IF Thta >= alfa3
            AND Thta <= (pi-alfa3)
            THEN
                Sfa:=0
            ENDIF

        IF Thta >= (pi-alfa2)
            AND Thta <= (pi-alfa1)
            THEN
                Sfa:=0
            ENDIF
        ENDIF

comment*****
-----FASE B-----
*****endcomment
        Sfb:=0

        IF Thtb >= alfa1
            AND (Thtb <= alfa2)
            THEN
                Sfb:=1
            ENDIF

        IF Thtb >= alfa3
            AND Thtb <= (pi-alfa3)
            THEN
                Sfb:=1
            ENDIF

        IF Thtb >= (pi-alfa2)
            AND Thtb <= (pi-alfa1)
            THEN
                Sfb:=1
            ENDIF

        IF Thtb >= pi THEN
            Thtb:= Thtb-pi
        Sfb:=1
            IF Thtb >= alfa1
                AND Thtb <= alfa2
                THEN
                    Sfb:=0
                ENDIF

        IF Thtb >= alfa3
            AND Thtb <= (pi-alfa3)
            THEN
                Sfb:=0
            ENDIF

        IF Thtb >= pi-alfa2
            AND Thtb <= (pi-alfa1)
            THEN
                Sfb:=0
            ENDIF

```

```

        ENDIF
        ENDIF

comment*****
-----FASE C-----
*****endcomment
        Sfc:=0

        IF Thtc >= alfa1
            AND (Thtc <= alfa2)
            THEN
                Sfc:=1
            ENDIF

        IF Thtc >= alfa3
            AND Thtc <= (pi-alfa3)
            THEN
                Sfc:=1
            ENDIF

        IF Thtc >= (pi-alfa2)
            AND Thtc <= (pi-alfa1)
            THEN
                Sfc:=1
            ENDIF

        IF Thtc >= pi THEN
            Thtc:= Thtc-pi
        Sfc:=1
            IF Thtc >= alfa1
                AND Thtc <= alfa2
                THEN
                    Sfc:=0
                ENDIF

            IF Thtc >= alfa3
                AND Thtc <= (pi-alfa3)
                THEN
                    Sfc:=0
                ENDIF

            IF Thtc >= (pi-alfa2)
                AND Thtc <= (pi-alfa1)
                THEN
                    Sfc:=0
                ENDIF

        NSfa:= NOT Sfa
        NSfb:= NOT Sfb
        NSfc:= NOT Sfc

        ENDEXEC

        ENDMODEL

        RECORD
            dispero.tx1 as dt
            dispero.Thta as dt2

        comment ****
        RECORD
            dispero.w as w
            dispero.t1 as tetha
        **** endcomment

        MODEL controle

comment *****
* Luis Claudio Gambøa Lopes 06/11/2002 *
* *
* Function: control reatiactive power *
* Inputs : voltage and currents phase a, b and c, and Qcref *
* Output : angle for converter 1 and 2 *
* *
***** endcomment

        INPUT BUSA
        BUSB
        BUSC
        IA
        IB
        IC

        Qcref

        DATA Kp {dflt:0.000025}
        Ki {dflt:0.001}
        tau {dflt:0.005}

        VAR VALFA
        VBETA
        IALFA

```

```

IBETA
Pcnv
Qcnv
QCerr
QCint
QCini
deltafini
deltaq
deltaf
deltat
delta1
delta2

Pcnv0
Qcnv0
Pcnv1
Qcnv1
fa0
fbl
fx
time

OUTPUT delta1 -- angle 1 [V]
delta2 -- angle 2 [V]

INIT
write('----- Controller Module By Gambôa -----')
deltat:=timestep
QCint:=0
deltafini:=0
QCini:=0
delta1:=0
delta2:=0
write('kp =',kp)
write('ki =',ki)
write('tau =',tau)

fx:=exp(-2*pi*timestep/0.02)
fa0:=1-fx
fbl:=fx
Pcnv:=0
Qcnv:=0
time:=0
ENDINIT

EXEC

comment*****
Controle de potencia Reativa
*****endcomment

VALFA:= sqrt(2/3)*(BUSA-(BUSB + BUSC)/2)
VBETA:= sqrt(2)*(BUSB-BUSC)/2

IALFA:= sqrt(2/3)*(IA - (IB + IC)/2)
IBETA:= sqrt(2)*(IB-IC)/2

comment*****
---- real and imaginary converter instantaneous powers
*****endcomment
Pcnv1:= Pcnv
Pcnv0:= (VALFA*IALFA)+(VBETA*IBETA)

Qcnv1:= Qcnv
Qcnv0:= (VBETA*IALFA)-(VALFA*IBETA)

comment*****
Filtrar
*****endcomment

Pcnv:=fa0*Pcnv0+fbl*Pcnv1

Qcnv:=fa0*Qcnv0+fbl*Qcnv1

comment*****
Pcnv:=Pcnv0
Qcnv:=Qcnv0
*****endcomment

comment*****
----- Controlador PI-----
*****endcomment

QCerr:= QCref-Qcnv

QCint:= QCini + QCerr * deltat
QCini:=QCint
deltaq:= Ki * QCint + Kp * QCerr

IF deltaq > pi/10 THEN
deltaq:=pi/10
ENDIF

IF deltaq < -pi/10 THEN
deltaq:=-pi/10
ENDIF

comment*****
filtro
deltaf:=(deltat/(tau+deltat))*deltaq+(1/(1+deltat/tau))*deltafini
deltafini:=deltaf
*****endcomment

deltaf:=deltaq

deltat:=deltaf-0.5236-0.2618
delta2:=deltaf-0.5236+0.2618

time:=time+timestep
ENDEXEC

ENDMODEL

RECORD
controle.Qcnv as QCNV1
controle.Pcnv as PCNV1
controle.BUSA as mVA
controle.IA as mIA
controle.deltaf as DELTAF

comment ****
teste
**** endcomment

```

Apêndice C

Listagem do Algoritmo de Controle para o TMS320F243

C.1 Arquivos de Cabeçalho

C.1.1 f243.h

```
-----
;Filename: f243.h
;
;Author: Luis CLaudio G. Lopes.
;
;Last Modified: 08/08/01
;
;Description: F243 DSP register definitions for assembly code
-----

;Core registers (DP_0)
IMR          .set 00004h    ;Interrupt mask reg
GREG         .set 00005h    ;Global memory allocation reg
IFR          .set 00006h    ;Interrupt flag reg

;System configuration an interrupt registers (DP_PF1)
SCSR         .set 07018h    ;System control & status reg 1
DINR        .set 0701Ch    ;Device identification reg
PIVR        .set 0701Eh    ;Peripheral interrupt vector reg
PIRQRO      .set 07010h    ;Peripheral interrupt request reg 0
PIRQR1      .set 07011h    ;Peripheral interrupt request reg 1

;External interrupt configuration registers (DP_PF1)
XINT1CR     .set 07070h    ;Ext interrupt 1 config reg
XINT2CR     .set 07071h    ;Ext interrupt 2 config reg

;Digital I/O registers (DP_PF2)
OCRA        .set 07090h    ;I/O mux control reg A
OCRB        .set 07092h    ;I/O mux control reg B
PADATDIR    .set 07098h    ;I/O port A data & dir reg
PBDATDIR    .set 0709Ah    ;I/O port B data & dir reg
PCDATDIR    .set 0709Ch    ;I/O port C data & dir reg
PDDATDIR    .set 0709Eh    ;I/O port D data & dir reg

;Watchdog timer (WD) registers (DP_PF1)
WDCNTR     .set 07023h    ;WD counter reg
WDKEY       .set 07025h    ;WD reset key reg
WDCR        .set 07029h    ;WD timer control reg

;ADC registers (DP_PF1)
;Analog-to-Digital Converter (ADC) registers
ADCTRL1     .set 07032h    ;ADC control reg 1
ADCTRL2     .set 07034h    ;ADC control reg 2
ADCFIFO1    .set 07036h    ;ADC FIFO
ADCFIFO2    .set 07038h    ;ADC FIFO

;Serial Peripheral Interface (SPI) registers (DP_PF1)
SPICCR      .set 07040h    ;SPI configuration control reg
SPICTL      .set 07041h    ;SPI operation control reg
SPISTS      .set 07042h    ;SPI status reg
SPIBRR      .set 07044h    ;SPI baud rate reg
SPIRXEMU    .set 07046h    ;SPI emulation buffer reg
SPIRXBUF    .set 07047h    ;SPI serial receive buffer reg
SPITXBUF    .set 07048h    ;SPI serial transmit buffer reg
SPIDAT      .set 07049h    ;SPI serial data reg
SPIPRI      .set 0704Fh    ;SPI priority control reg

;SCI registers (DP_PF1)
SCICCR      .set 07050h    ;SCI communication control reg
SCICTL1     .set 07051h    ;SCI control reg 1
SCIHBAUD    .set 07052h    ;SCI baud-select reg, high bits

SCI_LBAUD    .set 07053h    ;SCI baud-select reg, low bits
SCI_CTL2     .set 07054h    ;SCI control reg 2
SCI_RXST     .set 07055h    ;SCI receiver status reg
SCI_RXEMU    .set 07056h    ;SCI emulation data buffer reg
SCI_RXBUF    .set 07057h    ;SCI receiver data buffer reg
SCI_TXBUF    .set 07059h    ;SCI transmit data buffer reg
SCI_PRI      .set 0705Fh    ;SCI priority control reg

;Event Manager (EV) registers (DP_EV)
GPTCON      .set 07400h    ;GP timer control reg
T1CNT       .set 07401h    ;GP timer 1 counter reg
T1CMPR      .set 07402h    ;GP timer 1 compare reg
T1PR        .set 07403h    ;GP timer 1 period reg
T1CON       .set 07404h    ;GP timer 1 control reg
T2CNT       .set 07405h    ;GP timer 2 counter reg
T2CMPR      .set 07406h    ;GP timer 2 compare reg
T2PR        .set 07407h    ;GP timer 2 period reg
T2CON       .set 07408h    ;GP timer 2 control reg
COMCON      .set 07411h    ;Compare control reg
ACTR        .set 07413h    ;Compare action control reg
DBTCON      .set 07415h    ;Dead-band timer control reg
CMPR1       .set 07417h    ;compare reg 1
CMPR2       .set 07418h    ;compare reg 2
CMPR3       .set 07419h    ;compare reg 3
CAPCON      .set 07420h    ;Capture control reg
CAPFIFO     .set 07422h    ;Capture FIFO status reg
CAP1FIFO    .set 07423h    ;Capture Channel 1 FIFO top
CAP2FIFO    .set 07424h    ;Capture Channel 2 FIFO top
CAP3FIFO    .set 07425h    ;Capture Channel 3 FIFO top
CAP1FBOT    .set 07427h    ;Bottom reg of capture FIFO stack 1
CAP2FBOT    .set 07428h    ;Bottom reg of capture FIFO stack 2
CAP3FBOT    .set 07429h    ;Bottom reg of capture FIFO stack 3
EVIMRA      .set 0742Ch    ;EVA interrupt mask reg A
EVIMRB      .set 0742Dh    ;EVA interrupt mask reg B
EVIMRC      .set 0742Eh    ;EVA interrupt mask reg C
EVIFRA      .set 0742Fh    ;EVA interrupt flag reg A
EVIFRB      .set 07430h    ;EVA interrupt flag reg B
EVIFRC      .set 07431h    ;EVA interrupt flag reg C

;Controller Area Network (CAN) registers (DP_CAN1)
MDER        .set 07100h    ;CAN mailbox direction/enable reg
TCR         .set 07101h    ;CAN transmission control reg
RCR         .set 07102h    ;CAN receive control reg
MCR         .set 07103h    ;CAN master control reg
BCR2        .set 07104h    ;CAN bit config reg 2
BCR1        .set 07105h    ;CAN bit config reg 1
ESR         .set 07106h    ;CAN error status reg
GSR         .set 07107h    ;CAN global status reg
CEC         .set 07108h    ;CAN trans and rcv err counters
CAN_IFR     .set 07109h    ;CAN interrupt flag reg
CAN_IMR     .set 0710Ah    ;CAN interrupt mask reg
LAMO_H      .set 0710bh    ;CAN local acceptance mask MBX0/1
LAMO_L      .set 0710ch    ;CAN local acceptance mask MBX0/1
LAM1_H      .set 0710dh    ;CAN local acceptance mask MBX2/3
LAM1_L      .set 0710eh    ;CAN local acceptance mask MBX2/3

;Controller Area Network (CAN) mailboxes (DP_CAN2)
MSGIDOL     .set 07200h    ;CAN message ID for mailbox 0
; (lower 16 bits)
MSGIDOH     .set 07201h    ;CAN message ID for mailbox 0
; (upper 16 bits)
MSGCTRL0    .set 07202h    ;CAN RTR and DLC for mailbox 0
MBX0A       .set 07204h    ;CAN 2 of 8 bytes of mailbox 0
MBX0B       .set 07205h    ;CAN 2 of 8 bytes of mailbox 0
MBX0C       .set 07206h    ;CAN 2 of 8 bytes of mailbox 0
MBX0D       .set 07207h    ;CAN 2 of 8 bytes of mailbox 0
MSGID1L     .set 07208h    ;CAN message ID for mailbox 1
; (lower 16 bits)
MSGID1H     .set 07209h    ;CAN message ID for mailbox 1
; (upper 16 bits)
MSGCTRL1    .set 0720Ah    ;CAN RTR and DLC for mailbox 1
MBX1A       .set 0720Ch    ;CAN 2 of 8 bytes of mailbox 1
```

```

MBX1B      .set 0720Dh ;CAN 2 of 8 bytes of mailbox 1
MBX1C      .set 0720Eh ;CAN 2 of 8 bytes of mailbox 1
MBX1D      .set 0720Fh ;CAN 2 of 8 bytes of mailbox 1
MSGID2L    .set 07210h ;CAN message ID for mailbox 2
           ;(lower 16 bits)
MSGID2H    .set 07211h ;CAN message ID for mailbox 2
           ;(upper 16 bits)
MSGCTRL2   .set 07212h ;CAN RTR and DLC for mailbox 2
MBX2A      .set 07214h ;CAN 2 of 8 bytes of mailbox 2
MBX2B      .set 07215h ;CAN 2 of 8 bytes of mailbox 2
MBX2C      .set 07216h ;CAN 2 of 8 bytes of mailbox 2
MBX2D      .set 07217h ;CAN 2 of 8 bytes of mailbox 2
MSGID3L    .set 07218h ;CAN message ID for mailbox 3
           ;(lower 16 bits)
MSGID3H    .set 07219h ;CAN message ID for mailbox 3
           ;(upper 16 bits)
MSGCTRL3   .set 0721Ah ;CAN RTR and DLC for mailbox 3
MBX3A      .set 0721Ch ;CAN 2 of 8 bytes of mailbox 3
MBX3B      .set 0721Dh ;CAN 2 of 8 bytes of mailbox 3
MBX3C      .set 0721Eh ;CAN 2 of 8 bytes of mailbox 3
MBX3D      .set 0721Fh ;CAN 2 of 8 bytes of mailbox 3
MSGID4L    .set 07220h ;CAN message ID for mailbox 4
           ;(lower 16 bits)
MSGID4H    .set 07221h ;CAN message ID for mailbox 4
           ;(upper 16 bits)
MSGCTRL4   .set 07222h ;CAN RTR and DLC for mailbox 4
MBX4A      .set 07224h ;CAN 2 of 8 bytes of mailbox 4
MBX4B      .set 07225h ;CAN 2 of 8 bytes of mailbox 4
MBX4C      .set 07226h ;CAN 2 of 8 bytes of mailbox 4
MBX4D      .set 07227h ;CAN 2 of 8 bytes of mailbox 4
MSGID5L    .set 07228h ;CAN message ID for mailbox 5
           ;(lower 16 bits)
MSGID5H    .set 07229h ;CAN message ID for mailbox 5
           ;(upper 16 bits)
MSGCTRL5   .set 0722Ah ;CAN RTR and DLC for mailbox 5
MBX5A      .set 0722Ch ;CAN 2 of 8 bytes of mailbox 5
MBX5B      .set 0722Dh ;CAN 2 of 8 bytes of mailbox 5
MBX5C      .set 0722Eh ;CAN 2 of 8 bytes of mailbox 5
MBX5D      .set 0722Fh ;CAN 2 of 8 bytes of mailbox 5

;I/O space mapped registers
WSGR       .set OFFFh ;Wait-state generator control reg

;*****
;Other useful definitions below (not addresses) *
;*****

;Data page definitions for LDP instruction
DP_PF1     .set 224 ;sys regs, WD, SPI, SCI
           ;(0x7000 - 0x707F)
DP_PF2     .set 225 ;ADC, GPIO (0x7080 - 0x70FF)
DP_CAN1    .set 226 ;CAN control regs (0x7100 - 0x717F)
DP_CAN2    .set 228 ;CAN mailboxes 1-5 (0x7200 - 0x727F)
DP_EV      .set 232 ;Event manager (0x7400 - 0x747F)

;Bit codes for bit test instruction (BIT)
BIT15     .set 00000h ;Bit code for testing bit 0
BIT14     .set 00001h ;Bit code for testing bit 1
BIT13     .set 00002h ;Bit code for testing bit 2
BIT12     .set 00003h ;Bit code for testing bit 3
BIT11     .set 00004h ;Bit code for testing bit 4
BIT10     .set 00005h ;Bit code for testing bit 5
BIT9      .set 00006h ;Bit code for testing bit 6
BIT8      .set 00007h ;Bit code for testing bit 7
BIT7      .set 00008h ;Bit code for testing bit 8
BIT6      .set 00009h ;Bit code for testing bit 9
BIT5      .set 0000Ah ;Bit code for testing bit 10
BIT4      .set 0000Bh ;Bit code for testing bit 11
BIT3      .set 0000Ch ;Bit code for testing bit 12
BIT2      .set 0000Dh ;Bit code for testing bit 13
BIT1      .set 0000Eh ;Bit code for testing bit 14
BIT0      .set 0000Fh ;Bit code for testing bit 15

CANESR     .set 7106h ; CAN Error Status Register
CANGSR     .set 7107h ; CAN Global Status Register
CANCEC     .set 7108h ; CAN Transmit and Receive Err counters

CANIFR     .set 7109h ; CAN Interrupt Flag Registers
CANIMR     .set 710Ah ; CAN Interrupt Mask Registers

CANLAMOH   .set 710Bh ; CAN Local Acceptance Mask MBx0/1
CANLAMOL   .set 710Ch ; CAN Local Acceptance Mask MBx0/1

CANLAM1H   .set 710Dh ; CAN Local Acceptance Mask MBx2/3
CANLAM1L   .set 710Eh ; CAN Local Acceptance Mask MBx2/3

; CAN Mailboxes

CANMSGID0L .set 7200h ; CAN Message ID for mailbox 0
           ;(lower 16 bits)
CANMSGID0H .set 7201h ; CAN Message ID for mailbox 0
           ;(upper 16 bits)
CANMSGCTRL0 .set 7202h ; CAN RTR and DLC
CANMBX0A   .set 7204h ; CAN 2 of 8 bytes of Mailbox 0
CANMBX0B   .set 7205h ; CAN 2 of 8 bytes of Mailbox 0
CANMBX0C   .set 7206h ; CAN 2 of 8 bytes of Mailbox 0
CANMBX0D   .set 7207h ; CAN 2 of 8 bytes of Mailbox 0

CANMSGID1L .set 7208h ; CAN Message ID for mailbox 1
           ;(lower 16 bits)
CANMSGID1H .set 7209h ; CAN Message ID for mailbox 1
           ;(upper 16 bits)
CANMSGCTRL1 .set 720Ah ; CAN RTR and DLC
CANMBX1A   .set 720Ch ; CAN 2 of 8 bytes of Mailbox 1
CANMBX1B   .set 720Dh ; CAN 2 of 8 bytes of Mailbox 1
CANMBX1C   .set 720Eh ; CAN 2 of 8 bytes of Mailbox 1
CANMBX1D   .set 720Fh ; CAN 2 of 8 bytes of Mailbox 1

CANMSGID2L .set 7210h ; CAN Message ID for mailbox 2
           ;(lower 16 bits)
CANMSGID2H .set 7211h ; CAN Message ID for mailbox 2
           ;(upper 16 bits)
CANMSGCTRL2 .set 7212h ; CAN RTR and DLC
CANMBX2A   .set 7214h ; CAN 2 of 8 bytes of Mailbox 2
CANMBX2B   .set 7215h ; CAN 2 of 8 bytes of Mailbox 2
CANMBX2C   .set 7216h ; CAN 2 of 8 bytes of Mailbox 2
CANMBX2D   .set 7217h ; CAN 2 of 8 bytes of Mailbox 2

CANMSGID3L .set 7218h ; CAN Message ID for mailbox 3
           ;(lower 16 bits)
CANMSGID3H .set 7219h ; CAN Message ID for mailbox 3
           ;(upper 16 bits)
CANMSGCTRL3 .set 721Ah ; CAN RTR and DLC
CANMBX3A   .set 721Ch ; CAN 2 of 8 bytes of Mailbox 3
CANMBX3B   .set 721Dh ; CAN 2 of 8 bytes of Mailbox 3
CANMBX3C   .set 721Eh ; CAN 2 of 8 bytes of Mailbox 3
CANMBX3D   .set 721Fh ; CAN 2 of 8 bytes of Mailbox 3

CANMSGID4L .set 7220h ; CAN Message ID for mailbox 4
           ;(lower 16 bits)
CANMSGID4H .set 7221h ; CAN Message ID for mailbox 4
           ;(upper 16 bits)
CANMSGCTRL4 .set 7222h ; CAN RTR and DLC
CANMBX4A   .set 7224h ; CAN 2 of 8 bytes of Mailbox 4
CANMBX4B   .set 7225h ; CAN 2 of 8 bytes of Mailbox 4
CANMBX4C   .set 7226h ; CAN 2 of 8 bytes of Mailbox 4
CANMBX4D   .set 7227h ; CAN 2 of 8 bytes of Mailbox 4

CANMSGID5L .set 7228h ; CAN Message ID for mailbox 5
           ;(lower 16 bits)
CANMSGID5H .set 7229h ; CAN Message ID for mailbox 5
           ;(upper 16 bits)
CANMSGCTRL5 .set 722Ah ; CAN RTR and DLC
CANMBX5A   .set 722Ch ; CAN 2 of 8 bytes of Mailbox 5
CANMBX5B   .set 722Dh ; CAN 2 of 8 bytes of Mailbox 5
CANMBX5C   .set 722Eh ; CAN 2 of 8 bytes of Mailbox 5
CANMBX5D   .set 722Fh ; CAN 2 of 8 bytes of Mailbox 5

```

C.1.2 can.h

```

;=====
;* File name: can.h
;* Description: Header File
;* Author: Thiago Magaldi
;* References: SPRA500
;* Last Modified: 04/09/2003
;=====

```

; CAN Registers.

```

CANMDER    .set 7100h ; CAN Mailbox Direction/Enable register
CANTCR     .set 7101h ; CAN Transmission Control Register
CANRCR     .set 7102h ; CAN Receive Control Register
CANMCR     .set 7103h ; CAN Master Control Register

CANBCR2    .set 7104h ; CAN Bit Configuration Register 2
CANBCR1    .set 7105h ; CAN Bit Configuration Register 1

```

C.2 Mppt

C.2.1 _mppt.asm

```

;=====
;* File name: _mppt.asm
;* Description:
;* Author: Luis Claudio Gambôa Lopes
;* References:
;* Last Modified: 09/11/2004
;=====
TGG        .set 1 ;use toggle

```

```

;20e6/(4*(9999+1))=500
INT_rate .set 9999 ;500Hz sampling rate

;20e6/(1*(499+1))=40000
PWM_rate .set 499 ;40000Hz sampling rate

.global _init, _ReadPMod, _PWM, _RSTINT
.global _vm, _im, _vo, _duty, _intmain
.global _sdta, _s_send, _s_receive, _c_int0

ADCDELAY .macro
nop
nop
nop
nop
.endm

;*****
; inicializa labels para o f243 *
;*****
.include "f243.h"

.sect ".data"
;_Vbat .word 0
_vm .word 0
_im .word 0
_vo .word 0
_Vref .word 0
_duty .word 0
_sdta .word 0
temp .word 0 ;variavel de uso geral
dout .word 0

;*****
; setup dos vetores de interrupção *
;*****
.sect "vectors"
RST: B _c_int0 ;00h reset
int1: B int1 ;02h INT1
int2: B _intmain ;04h INT2
int3: B int3 ;06h INT3
int4: B int4 ;08h INT4
int5: B int5 ;0Ah INT5 (reservado pelo DSK)
int6: B int6 ;0Ch INT6
int7: B int7 ;0Eh reserved
int8: B int8 ;10h INT8 user-defined
int9: B int9 ;12h INT9 user-defined
int10: B int10 ;14h INT10 user-defined
int11: B int11 ;16h INT11 user-defined
int12: B int12 ;18h INT12 user-defined
int13: B int13 ;1Ah INT13 user-defined
int14: B int14 ;1Ch INT14 user-defined
int15: B int15 ;1Eh INT15 user-defined
int16: B int16 ;20h INT16 user-defined
int17: B int17 ;22h TRAP
int18: B int18 ;24h NMI
int19: B int19 ;26h reserved
int20: B int20 ;28h INT20 user-defined
int21: B int21 ;2Ah INT21 user-defined
int22: B int22 ;2Ch INT22 user-defined
int23: B int23 ;2Eh INT23 user-defined
int24: B int24 ;30h INT24 user-defined
int25: B int25 ;32h INT25 user-defined
int26: B int26 ;34h INT26 user-defined
int27: B int27 ;36h INT27 user-defined
int28: B int28 ;38h INT28 user-defined
int29: B int29 ;3Ah INT29 user-defined
int30: B int30 ;3Ch INT30 user-defined
int31: B int31 ;3Eh INT32 user-defined

.sect ".text"
_init:
;*****
; Desativa interrupcoes *
;*****
SETC INTM
;*****
; desliga todos registros do EV *
;*****
LACL #0
LDP #DP_EV
SACL GPTCON
SACL T1CNT
SACL T1CMPR
SACL T1PR
SACL T1CON
SACL T2CNT
SACL T2CMPR
SACL T2PR
SACL T2CON
SACL COMCON
SACL ACTR
SACL DBTCON
SACL CMPR1

SACL CMPR2
SACL CMPR3
SACL CAPCON
SACL CAPFIFO

;*****
; desabilita watchdog *
;*****
LDP #DP_PF1
SPLK #01101000b,WDCR
;
; |11111111
; 76543210
; bit 7 0: clear WD flag
; bit 6 1: disable de dog
; bit 5-3 101: must be written as 101
; bit 2-0 000: WDCLK divider =1

;*****
; setup da saida *
;*****
;*****
; setup dos pinos compartilhados de I/O *
;*****
LDP #DP_PF2
LACC OCRA
OR #0011000000000000b
;
; |1111111111111111
; FEDCBA9876543210
; bit 15 0: 0=IOPB7, 1=TCLKIN
; bit 14 0: 0=IOPB6, 1=TDIR
; bit 13 1: 0=IOPB5, 1=T2PWM/T2CMP
; bit 12 0: 0=IOPB4, 1=T1PWM/T1CMP
; bit 11 0: 0=IOPB3, 1=PWM6
; bit 10 0: 0=IOPB2, 1=PWM5
; bit 9 0: 0=IOPB1, 1=PWM4
; bit 8 0: 0=IOPB0, 1=PWM3
; bit 7 0: 0=IOPB7, 1=PWM2
; bit 6 0: 0=IOPB6, 1=PWM1
; bit 5 0: 0=IOPB5, 1=CAP3
; bit 4 0: 0=IOPB4, 1=CAP2/QEP1
; bit 3 0: 0=IOPB3, 1=CAP1/QEP0
; bit 2 0: 0=IOPB2, 1=XINT1
; bit 1 0: 0=IOPB1, 1=SCIRXD
; bit 0 0: 0=IOPB0, 1=SCITXD
SACL OCRA

SPLK 0000h,PDDATDIR

;*****
; setup do registro de controle do sistema *
;*****
LDP #DP_PF1
SPLK #0000000000000000b,SCSR
;
; |1111111111111111
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: CLKOUT=CPUCLK
; bit 13-12 00: IDLE selected for low-power mode
; bit 11-1 0's: reserved
; bit 0 1: clear the ILLADR bit

;*****
; setup do gerador de espera *
;*****
LDP #temp
SPLK #0000000000000000b,temp
;
; |1111111111111111
; FEDCBA9876543210
; bit 15-11 0s: reserved
; bit 10-9 00: bus visibility off
; bit 8-6 001: 1 wait-state for I/O space
; bit 5-3 000: 0 wait-state for data space
; bit 2-0 000: 0 wait-state for program space
;
; OUT temp, WSGR

;*****
; setup das interrupcoes *
;*****
LDP #0h
LACL #000010b
SACL IFR
OR IMR
SACL IMR

;*****
; setup do ADC *
;*****
;ADC:
LDP #DP_PF1
SPLK #0000000000000000b,ADCTRL2 ;init ADCTRL2 register
;
; |1111111111111111
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: 0 = ADCINTFLAG set when FIFO has at least 1 word
; bit 13 1: 1 = clear EVSOCP bit

```

```

; bit 12 1: 1 = clear EXTSOCP bit
; bit 11 0: 0 = high interrupt priority
; bit 10 0: 1 = enable conversion start by EV
; bit 9 0: 1 = enable conversion start by ADCSOC pin
; bit 8 0: 0 = reserved
; bit 7-6 00: 00 = ADCFIFO2 status - read-only
; bit 5 0: 0 = reserved
; bit 4-3 00: 00 = ADCFIFO1 status - read-only
; bit 2-0 000: 000 = clock prescaler, ADCCLK=CPUCLK/1

;*****
; setup dos registros de modo da CPU *
;*****
SETC SXM ; usa o modo de sinalizacao extendido
CLRC OVM ; normaliza overflow
SPM 0 ; seta ALU para multiplicacao fracionaria

;*****
; setup do contador *
;*****
contador:
LDP #DP_EV

; configuracao T1CON GP Timer 1

SPLK #0000001001000000b,T1CON ;GP Timer control
;
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 00: 00 = stop and hold mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

; configuracao T2CON GP Timer 2

SPLK #000000001000010b,T2CON ;GP Timer control
;
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 00: 00 = stop and hold mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 1: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

; configuracao GPTCON

SPLK #0000000001001010b,GPTCON ;GP Timer control
;
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: T2STAT - read-only
; bit 13 0: T1STAT - read-only
; bit 12-11 00: reserved
; bit 10-9 00: T2TADC, 00 = no ADCSOC
; bit 8-7 00: T1TADC, 10 = ADCSOC on period match
; bit 6 1: 1 = enable all timer compare outputs
; bit 5-4 00: reserved
; bit 3-2 10: 00 = T2PIN forced low
; bit 1-0 00: 00 = T1PIN forced low

; configuracoes T1PR,T1CMR e T1CNT
SPLK #INT_rate,T1PR ;periodo
SPLK #OFFFEh,T1CNT ;limpa contador
; configuracoes T2PR,T2CMR e T2CNT
SPLK #PWM_rate,T2PR ;periodo
SPLK #0010h,T2CMR
SPLK #OFFFEh,T2CNT ;limpa contador

;*****
; inicio do contador *
;*****
LDP #DP_EV
; inicializa GP Timer 1

SPLK #0001001001000000b,T1CON ;GP Timer control
;
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

; inicializa GP Timer 2
SPLK #00010000001000010b,T2CON ;GP Timer control
;
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

;*****
; setup das interrupcoes do EV *
;*****
LDP #DP_EV
SPLK #OFFFh, EVIFRA ;clear interrupt flags
SPLK #OFFFh, EVIFRB ;clear interrupt flags
SPLK #OFFFh, EVIFRC ;clear interrupt flags

; period timer1 match interrupt
SPLK #00080h, EVIMRA ;enable desired interrupts
SPLK #00000h, EVIMRB ;enable desired interrupts
SPLK #00000h, EVIMRC ;enable desired interrupts

;-----
; SCI Configuration
;-----

LDP #DP_PF1
SPLK #0007h, SCICCR
SPLK #0003h, SCICTL1
SPLK #0000h, SCICTL2
SPLK #0001h, SCIHBAUD
SPLK #0003h, SCILBAUD ; Baud Rate=9600 bps (9615,...)
SPLK #0023h, SCICTL1

; SPLK #0002h, SCIPRI

;-----
; Configure the shared pins (serial)
;-----

LDP #DP_PF2
LACC OCRA
OR #0003h
SACL OCRA ; pinos SCITX e SCIRX

;*****
; Ativa interrupcoes *
;*****
CLRC INTM

RET

;*****
; interrupcao *
;*****

_RSTINT
;toggle

.if TGG
LDP #dout
LACC dout
AND #OFF00h
SACL dout
;disparo externo
LDP #DP_PF2
SACL PDDATDIR
.endif

LDP #DP_EV
SPLK #OFFFh, EVIFRA ;clear interrupt flags
;
CLRC INTM
LDP #_vm
RET

;*****
; interrupcao *
;*****

_ReadPMod:
;toggle

.if TGG
LDP #dout
LACC dout
OR #00FFh

```



```

}

void
sendint (char* name,unsigned int n)
{
    unsigned int c1, c2, c3, c4, c5;
    int i;

    c1 = (n / 10000);
    c2 = ((n - c1 * 10000) / 1000);
    c3 = ((n - c1 * 10000 - c2 * 1000) / 100);
    c4 = ((n - c1 * 10000 - c2 * 1000 - c3 * 100) / 10);
    c5 = ((n - c1 * 10000 - c2 * 1000 - c3 * 100 - c4 * 10));

    for(i=0;i<strlen(name);i++)
    {
        sdat=name[i];
        s_send();
    }

    sdat=c1+48;
    s_send();
    sdat=c2+48;
    s_send();
    sdat=c3+48;
    s_send();
    sdat=c4+48;
    s_send();
    sdat=c5+48;
    s_send();
}

void
sendfloat (char* name,float f)
{
    unsigned int c1, c2, c3, c4, c5;
    int i;
    unsigned int n;

    n=(unsigned int)(f*100);

    c1 = (n / 100000);
    c2 = ((n - c1 * 100000) / 10000);
    c3 = ((n - c1 * 100000 - c2 * 10000) / 1000);
    c4 = ((n - c1 * 100000 - c2 * 10000 - c3 * 1000) / 100);
    c5 = ((n - c1 * 100000 - c2 * 10000 - c3 * 1000 - c4 * 100));

    for(i=0;i<strlen(name);i++)
    {
        sdat=name[i];
        s_send();
    }

    sdat=c1+48;
    s_send();
    sdat=c2+48;
    s_send();
    sdat=c3+48;
    s_send();
    sdat=c4+48;
    s_send();
    sdat=c5+48;
    s_send();
}

void
sendfloat2 (char* name,float f)
{
    unsigned long int c00, c0, c1, c2, c3, c4, c5;
    int i;
    unsigned long int n;

    n=(unsigned long int)(f*100);

    c00 = (n / 10000000);
    c0 = ((n - c00 * 10000000) / 1000000);
    c1 = ((n - c00 * 10000000 - c0 * 1000000) / 100000);
    c2 = ((n - c00 * 10000000 - c0 * 1000000 - c1 * 100000) / 10000);
    c3 = ((n - c00 * 10000000 - c0 * 1000000 - c1 * 100000 - c2 * 10000) / 1000);
    c4 = ((n - c00 * 10000000 - c0 * 1000000 - c1 * 100000 - c2 * 10000 - c3 * 1000) / 100);
    c5 = ((n - c00 * 10000000 - c0 * 1000000 - c1 * 100000 - c2 * 10000 - c3 * 1000 - c4 * 100));

    for(i=0;i<strlen(name);i++)
    {
        sdat=name[i];
        s_send();
    }

    sdat=c00+48;
    s_send();
    sdat=c0+48;
    s_send();
    sdat=c1+48;
    s_send();
    sdat=c2+48;
    s_send();
    sdat=c3+48;
    s_send();
    sdat=c4+48;
    s_send();
    sdat=c5+48;
    s_send();
}

}

sdat=name[i];
s_send();
}

sdat=c00+48;
s_send();
sdat=c0+48;
s_send();
sdat=c1+48;
s_send();
sdat=c2+48;
s_send();
sdat=c3+48;
s_send();
sdat=c4+48;
s_send();
sdat=c5+48;
s_send();
}

void
main()
{
    unsigned int l_Vpv;
    unsigned int l_Ipv;
    unsigned int l_vo;
    unsigned int l_duty;
    float v;

    char L1[]=" Vpv= ";
    char L2[]=" Ipv= ";
    char L3[]=" Vo= ";
    char L4[]=" Duty= ";
    char L5[]=" Pot= ";

    /*brute force loading*/
    L1[0]=' ';
    L1[1]='V';
    L1[2]='p';
    L1[3]='v';
    L1[4]='v';
    L1[5]=' ';
    L1[6]=0;
    L2[0]=' ';
    L2[1]=' ';
    L2[2]=' ';
    L2[3]=' ';
    L2[4]='I';
    L2[5]='p';
    L2[6]='v';
    L2[7]='v';
    L2[8]=' ';
    L2[9]=0;
    L3[0]=' ';
    L3[1]=' ';
    L3[2]=' ';
    L3[3]=' ';
    L3[4]='V';
    L3[5]='o';
    L3[6]='v';
    L3[7]=' ';
    L3[8]=0;
    L4[0]=' ';
    L4[1]=' ';
    L4[2]=' ';
    L4[3]=' ';
    L4[4]='D';
    L4[5]='u';
    L4[6]='t';
    L4[7]='y';
    L4[8]=' ';
    L4[9]=' ';
    L4[10]=0;
    L5[0]=' ';
    L5[1]=' ';
    L5[2]=' ';
    L5[3]=' ';
    L5[4]='P';
    L5[5]='o';
    L5[6]='t';
    L5[7]=' ';
    L5[8]=' ';
    L5[9]=0;

    duty=100;
    vd=100;
    p[0]=0;
    p[1]=0;
    alfa=1;

    fv[0]=0;
    fv[1]=0;
    fi[0]=0;
    fi[1]=0;
}

```

```

init();

while(1)
{
l_Vpv=vm;
l_Ipv=im;
l_vo=vo;
l_duty=duty;

/*
sendint(L1,l_Vpv);
sendint(L2,l_Ipv);
sendint(L3,l_duty);
*/

v=l_Vpv*0.293*1.5;
sendfloat(L1,v);
sdat='V';
s_send();

v=l_Ipv*0.039*1.5;
sendfloat(L2,v);
sdat='A';
s_send();

v=l_vo*0.3216;
sendfloat(L3,v);
sdat='V';
s_send();

v=l_duty/4.99;
sendfloat(L4,v);
sdat='%';
s_send();

v=(l_Vpv*0.293*1.5)*(l_Ipv*0.039*1.5);

sendfloat2(L5,v);
sdat='W';
s_send();

sdat=13;
s_send();
sdat=10;
s_send();
}
}

```

C.3 Comando

C.3.1 _comando.asm

```

;=====
;* File name: _comando.asm
;* Description:
;* Author: Luis Claudio Gambôa Lopes
;* References:
;* Last Modified: 09/11/2004
;=====

TGG .set 0 ;use toggle
VDC .set 1 ;use VDC

.title "transmit"

.include "f243.h"
.include "can.h"

.global _c_int0,_comando,_ADC,_TRANSMIT,_START
.global _va,_vb,_vc,_ia,_ib,_ic,_delta1,_delta2
.global _scmd1,_scmd2,_RECEIVE,_cst1,_cst2,_wf1,_wf2,_vdc
.global _s_send,_RSTINT,_s_receive,_sdat

ADCDELAY .macro
nop
nop
nop
nop
nop
.endm

;-----
; Seções da memória
;-----

; memory.cmd

;-----
; Constant definitions

```

```

;-----
;ADC_rate .set 1249 ;4000Hz sampling rate
ADC_rate .set 4999 ;1000Hz sampling rate
dp_d .set 0300h>>7

PHANTOM .set 8160h

;-----
; Variáveis na memória de dados
;-----

.data
temp .word 0 ; variável temporária de uso geral
_delta1 .word 0
_delta2 .word 0
_va .word 0
_vb .word 0
_vc .word 0
_ia .word 0
_ib .word 0
_ic .word 0
_vdc .word 0
_sdat .word 0
dout .word 0
_scmd1 .word 0 ;server comand conversor1
_scmd2 .word 0 ;server comand conversor2
_cst1 .word 0 ;client status conversor1
_cst2 .word 0 ;client status conversor2
_wf1 .word 0 ;client wf conversor1
_wf2 .word 0 ;client wf conversor2

;-----
; Programa Principal
;-----

.sect "vectors"

RSVECT B _c_int0 ; Reset Vector
INT1 B INT1 ; Interrupt Level 1
INT2 B _comando ; Interrupt Level 2
INT3 B INT3 ; Interrupt Level 3
INT4 B INT4 ; Interrupt Level 4
INT5 B INT5 ; Interrupt Level 5
INT6 B INT6 ; Interrupt Level 6
RESERVED B RESERVED ; Reserved
SW_INT8 B SW_INT8 ; Software Interrupt
SW_INT9 B SW_INT9 ; Software Interrupt
SW_INT10 B SW_INT10 ; Software Interrupt
SW_INT11 B SW_INT11 ; Software Interrupt
SW_INT12 B SW_INT12 ; Software Interrupt
SW_INT13 B SW_INT13 ; Software Interrupt
SW_INT14 B SW_INT14 ; Software Interrupt
SW_INT15 B SW_INT15 ; Software Interrupt
SW_INT16 B SW_INT16 ; Software Interrupt
TRAP B TRAP ; Trap vector
NMI B NMI ; Non-maskable Interrupt
EMU_TRAP B EMU_TRAP ; Emulator Trap
SW_INT20 B SW_INT20 ; Software Interrupt
SW_INT21 B SW_INT21 ; Software Interrupt
SW_INT22 B SW_INT22 ; Software Interrupt
SW_INT23 B SW_INT23 ; Software Interrupt
SW_INT24 B SW_INT24 ; Software Interrupt
SW_INT25 B SW_INT25 ; Software Interrupt
SW_INT26 B SW_INT26 ; Software Interrupt
SW_INT27 B SW_INT27 ; Software Interrupt
SW_INT28 B SW_INT28 ; Software Interrupt
SW_INT29 B SW_INT29 ; Software Interrupt
SW_INT30 B SW_INT30 ; Software Interrupt
SW_INT31 B SW_INT31 ; Software Interrupt

.text

_START:
;*****
; Desativa interrupcoes *
;*****
SETC INTM
; SETC CNF ; Config Block B0 to Program memory.
;*****
; desliga todos registros do EV *
;*****
LACL #0
LDP #DP_EV
SACL GPTCON
SACL T1CNT
SACL T1CMPR
SACL T1PR
SACL T1CON
SACL T2CNT
SACL T2CMPR
SACL T2PR
SACL T2CON
SACL COMCON
SACL ACTR
SACL DETCON
SACL CMPR1

```

```

SACL CMPR2
SACL CMPR3
SACL CAPCON
SACL CAPFIFO

;*****
; desabilita watchdog *
;*****
LDP #DP_Pf1
SPLK #01101000b,WDCR
;
; | | | | | | | |
; 76543210
; bit 7 0: clear WD flag
; bit 6 1: disable de dog
; bit 5-3 101: must be writen as 101
; bit 2-0 000: WDCLK divider =

;*****
; setup do registro de controle do sistema *
;*****
LDP #DP_Pf1
SPLK #000000000000000b,SCSR
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: CLKOUT=CPUCLK
; bit 13-12 00: IDLE selected for low-power mode
; bit 11-1 0's: reserved
; bit 0 1: clear the ILLADR bit

;*****
; setup do gerador de espera *
;*****
LDP #dp_d
SPLK #000000000000000b,temp
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15-11 0s: reserved
; bit 10-9 00: bus visibility off
; bit 8-6 001: 1 wait-state for I/O space
; bit 5-3 000: 0 wait-state for data space
; bit 2-0 000: 0 wait-state for program space
OUT temp, WSGR

;*****
; setup das interrupções *
;*****
LDP #0h
LACL #000010b
SACL IFR
OR IMR
SACL IMR

;*****
; setup do ADC *
;*****
;ADC:
LDP #DP_Pf1
SPLK #0000000000000010b,ADCTRL2 ;init ADCTRL2 register
SPLK #000000000000000b,ADCTRL2 ;init ADCTRL2 register
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: 0 = ADCINTFLAG set when FIFO has at least 1 word
; bit 13 1: 1 = clear EVSOCP bit
; bit 12 1: 1 = clear EXTSOCP bit
; bit 11 0: 0 = high interrupt priority
; bit 10 0: 1 = enable conversion start by EV
; bit 9 0: 1 = enable conversion start by ADCSOC pin
; bit 8 0: 0 = reserved
; bit 7-6 00: 00 = ADCFIF02 status - read-only
; bit 5 0: 0 = reserved
; bit 4-3 00: 00 = ADCFIF01 status - read-only
; bit 2-0 000: 000 = clock prescaler, ADCLK=CPUCLK/1

;*****
; setup dos registros de modo da CPU *
;*****
SETC SXM ; usa o modo de sinalizacao extendido
CLRC OVM ; normaliza overflow
SPM 0 ; seta ALU para multiplicacao fracionaria

;*****
; setup do contador *
;*****
contador:
LDP #DP_EV

;configuração T1CON GP Timer 1

SPLK #0000001001000000b,T1CON ;GP Timer control
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

;*****
; setup das interrupções do EV *
;*****
LDP #DP_EV
SPLK #OFFFh, EVIFRA ;clear interrupt flags
SPLK #OFFFh, EVIFRB ;clear interrupt flags
SPLK #OFFFh, EVIFRC ;clear interrupt flags

;period timer1 match interrupt
SPLK #00080h, EVIMRA ;enable desired interrupts
SPLK #00000h, EVIMRB ;enable desired interrupts
SPLK #00000h, EVIMRC ;enable desired interrupts

; bit 13 0: reserved
; bit 12-11 00: 00 = stop and hold mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

;configuração T2CON GP Timer 2
SPLK #0000001001000010b,T2CON ;GP Timer control
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 00: 00 = stop and hold mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 1: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

;configuração GPTCON
SPLK #000000001001010b,GPTCON ;GP Timer control
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: T2STAT - read-only
; bit 13 0: T1STAT - read-only
; bit 12-11 00: reserved
; bit 10-9 00: T2TOADC, 00 = no ADCSOC
; bit 8-7 00: T1TOADC, 10 = ADCSOC on period match
; bit 6 1: 1 = enable all timer compare outputs
; bit 5-4 00: reserved
; bit 3-2 10: 00 = T2PIN forced low
; bit 1-0 00: 00 = T1PIN forced low

;configurações T1PR,T1CMPR e T1CNT
SPLK #ADC_rate,T1PR ;periodo
SPLK #OFFFh,T1CNT ;limpa contador
;configurações T1PR,T1CMPR e T1CNT
SPLK #ADC_rate,T2PR ;periodo
SPLK #0010h,T2CMPR
SPLK #OFFFh,T2CNT ;limpa contador

;*****
; inicio do contador *
;*****
LDP #DP_EV
;inicializa GP Timer 1

SPLK #0001001001000000b,T1CON ;GP Timer control
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

;inicializa GP Timer 2
SPLK #0001001001000010b,T2CON ;GP Timer control
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13 0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8 010: 010 = x/4 prescaler
; bit 7 0: 0 = use own TENABLE bit
; bit 6 1: 1 = enable timer
; bit 5-4 00: 00 = CPUCLK is clock source
; bit 3-2 00: 00 = reload compare reg on underflow
; bit 1 0: 0 = disabled timer compare
; bit 0 0: 0 = use own period register

;*****
; setup das interrupções do EV *
;*****

```

```

;bit 0-3  DLC Data length code=1000 = 8 bytes (data bytes)
;bit 4    RTR=0 - Data frame
;bit 5-15 Reserved

;-----
; SCI Configuration
;-----
LDP #DP_PF1
SPLK #0007h, SCICCR
SPLK #0003h, SCICTL1
SPLK #0000h, SCICTL2
SPLK #0001h, SCILBAUD
SPLK #0003h, SCILBAUD ; Baud Rate=9600 bps (9615,...)
SPLK #0023h, SCICTL1
; SPLK #0002h, SCIPRI

;-----
; Configure the shared pins (CAN)
;-----
LDP #DP_PF2
LACC OCRA
OR #0003h
SACL OCRA ; pinos SCITX e SCIRX
LACC OCRB
OR #00C0h
SACL OCRB ; pinos CANTX e CANRX

LDP #dp_d
SPLK #0FF00h,dout ;todos saida

; LDP #DP_CAN1
; SPLK #3F7Fh, CANIMR ; Enable all CAN interrupts

;-----
; CAN Bit Timing Configuration - for 1Mbps, CLKOUT= 20MHz
;-----
LDP #DP_CAN1
SPLK #0001000000000000b, CANMCR
; bit 12 CCR Change configuration request
; CCR=1: write access to BCRn registers
W_CCE BIT CANGSR, BIT4 ; TC=CCE , Wait for Change configuration
BCND W_CCE, NTC

SPLK #0000000000000000b, CANBCR2
; bit 0-7 BRP=0 Baud Rate Prescaler
; bit 8-15 Reserved

SPLK #000000011111010b, CANBCR1
;
; FEDCBA9876543210
; bit 0-2 TSEG2=2
; bit 3-6 TSEG1=15
; bit 7 SAM=1 Sample point setting
; bit 8-9 SJW=1 Synchronization jump width
; bit A SBG=0 - synchronization on the falling edge only
; bit B-F Reserved

SPLK #0000000000000000b, CANMCR
; bit 12 CCR Change configuration request
; CCR=0: request normal operation

W_NCCE BIT CANGSR, BIT4 ; TC=CCE
BCND W_NCCE, TC

;-----
; Disable CAN Mailboxes before writing
;-----
SPLK #0000000000000000b, CANMDER
;
; FEDCBA9876543210
; bit 0-5 MEN=0 - disable all mailboxes
; bit 6-7 MDn Mailbox direction for mailbox n
; bit 8-15 Reserved

;-----
; Writing in CAN Mailbox 5 - Transmit
;-----
LDP #DP_CAN2
SPLK #101011000010101b, CANMSGID5H
;
; FEDCBA9876543210
;bit 0-12 IDH - Upper 13 bits of extended identifier
;bit 13 AAM=1 - if a matching remote request is received,
;the CAN Peripheral
; answers by sending the contents of the mailbox
;bit 14 AME=0
;bit 15 IDE=1 - the received message has an extended ID (29 bits)

SPLK #110111000010101b, CANMSGID5L
;bit 0-15 IDL - Lower part of extended identifier

SPLK #0000000000001000b, CANMSGCTRL5
;
; FEDCBA9876543210

;-----
; Writing in CAN Mailbox 2 - Receive
;-----
LDP #DP_CAN2
SPLK #101011000010101b, MSGID2H
;
; FEDCBA9876543210
;bit 0-12 IDH - Upper 13 bits of extended identifier
;bit 13 AAM=1 - if a matching remote request is received,
;the CAN Peripheral
; answers by sending the contents of the mailbox
;bit 14 AME=0
;bit 15 IDE=1 - the received message has an extended ID (29 bits)

SPLK #110111000010101b, MSGID2L
;bit 0-15 IDL - Lower part of extended identifier

SPLK #0000000000001000b, MSGCTRL2
;
; FEDCBA9876543210
;bit 0-3  DLC Data length code=1000 = 8 bytes (data bytes)
;bit 4    RTR=0 - Data frame
;bit 5-15 Reserved

;-----
; Writing in CAN Mailbox 3 - Receive
;-----
LDP #DP_CAN2
SPLK #101011000010101b, MSGID3H
;
; FEDCBA9876543210
;bit 0-12 IDH - Upper 13 bits of extended identifier
;bit 13 AAM=1 - if a matching remote request is received,
;the CAN Peripheral
; answers by sending the contents of the mailbox
;bit 14 AME=0
;bit 15 IDE=1 - the received message has an extended ID (29 bits)

SPLK #110111000010100b, MSGID3L
;bit 0-15 IDL - Lower part of extended identifier

SPLK #0000000000001000b, MSGCTRL3
;
; FEDCBA9876543210
;bit 0-3  DLC Data length code=1000 = 8 bytes (data bytes)
;bit 4    RTR=0 - Data frame
;bit 5-15 Reserved

;-----
; Enable Mailbox 2,3,5 after writing
;-----
LDP #DP_CAN1
SPLK #000000011101100b, CANMDER
;
; FEDCBA9876543210
; bit 0 ME0=0 - disable mailbox 0
; bit 1 ME1=0 - disable mailbox 1
; bit 2 ME2=1 - enable mailbox 2
; bit 3 ME3=1 - enable mailbox 3
; bit 4 ME4=0 - disable mailbox 4
; bit 5 ME5=1 - enable mailbox 5
; bit 6 MD2=1 - mailbox 2 receive
; bit 7 MD3=1 - mailbox 3 receive
; bit 8-15 Reserved

LDP #DP_EV
SPLK #0FFFh, EVIFRA ;clear interrupt flags
CLRC INTM ; Enable interrupts
; (para interrupção do Debugger)

SETC XF

LDP #dp_d

RET

;MAIN:
; CLRC XF
; RPT #080h ; repeat next instruction 129 times
; NOP
; RPT #080h ; repeat next instruction 129 times
; NOP
; RPT #080h ; repeat next instruction 129 times

```

```

; NOP
; CALL _comando
; B MAIN
; .else
; LDP #DP_EV
; SPLK #0FFFFh, EVIFRA ;clear interrupt flags
; CLRC INTM
;main:
; CLRC XF
; RPT #080h ; repeat next instruction 129 times
; NOP
; RPT #080h ; repeat next instruction 129 times
; NOP
; RPT #080h ; repeat next instruction 129 times
; NOP
; B main
; .endif

_ADC:
SETC XF

SETC INTM

CALL readADC

LDP #dp_d
LACL _va
SUB #512
SACL _va
LACL _vb
SUB #512
SACL _vb
LACL _vc
SUB #512
SACL _vc
LACL _ia
SUB #512
; NEG
SACL _ia
LACL _ib
SUB #512
; NEG
SACL _ib
LACL _ic
SUB #512
; NEG
SACL _ic
RET

_TRANSMIT:
;-----
; TRANSMIT
;-----
LDP #DP_EV

LDP #DP_CAN1
SPLK #0080h, CANTCR
;bit 7 TRS5=1 - write access to mailbox 5 is denied (negado),
;and the the message will be transmitted

W_TA: BIT CANTCR, BIT15 ; TC=TA5 - transmission acknowledge
BCND W_TA, NTC

LDP #_delta1
LACL _delta1
LDP #DP_CAN2
SACL CANMBX5A

LDP #_scmd1
LACL _scmd1
LDP #DP_CAN2
SACL CANMBX5B

LDP #_delta2
LACL _delta2
LDP #DP_CAN2
SACL CANMBX5C

LDP #_scmd2
LACL _scmd2
LDP #DP_CAN2
SACL CANMBX5D

LDP #DP_CAN1
SPLK #8000h, CANTCR ; Reset TA5 (and CANIFR)

_RSTINT:

;togle
.if TGG
LDP #dout
LACC dout
AND #0FF00h
SACL dout
;disparo externo

LDP #DP_PF2
SACL PDDATDIR
.endif

LDP #DP_EV
SPLK #0FFFFh, EVIFRA ;clear interrupt flags
CLRC INTM

RET

;-----
; RECEIVE
;-----
_RECEIVE:
LDP #DP_CAN1
BIT RCR, BIT6 ; TC=RMP2 - receive message pending
BCND W_RMP1, NTC

SPLK #0040h, RCR ; Reset RMP2 (and CANIFR)

LDP #DP_CAN2
LACL MBX2A
LDP #dp_d
SACL _cst1
LDP #DP_CAN2
LACL MBX2B
LDP #dp_d
SACL _wf1
W_RMP1:
LDP #DP_CAN1
BIT RCR, BIT7 ; TC=RMP2 - receive message pending
BCND W_RMP2, NTC

SPLK #0080h, RCR ; Reset RMP2 (and CANIFR)

LDP #DP_CAN2
LACL MBX3A
LDP #dp_d
SACL _cst2
LDP #DP_CAN2
LACL MBX3B
LDP #dp_d
SACL _wf2
W_RMP2:
RET

;*****
;* Routine Name: readADC *
;* Measure of phase magnitude *
;* Output : va, vb and vc *
;*****

readADC:
;togle
.if TGG
LDP #dout
LACC dout
OR #00FFh
LDP #DP_PF2
SACL PDDATDIR
.endif

CLRC SXM
LDP #DP_PF1
SPLK #1010100100000001b,ADCTRL1 ;init ADCTRL1 register
;
; | | | | | | | | | | | | | | | |
; FEDCBA9876543210
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled
; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done

ADCDELAY
BIT ADCTRL1,8
BCND $-1, TC
ADCDELAY
LDP #DP_PF1
LACC ADCFIF0,10
LDP #dp_d
SACL _va

LDP #DP_PF1
SPLK #1010100100000001b,ADCTRL1 ;init ADCTRL1 register
;
; | | | | | | | | | | | | | | | |
; FEDCBA9876543210
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled

```

```

; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done
    ADCDELAY
    BIT ADCTRL1,8
    BCND $-1, TC
    ADCDELAY
    LDP #DP_PPF1
    LACC ADCFIFO1,10
    LDP #dp_d
    SACH _vb

    LDP #DP_PPF1
SPLK #1010100100000101b,ADCTRL1 ;init ADCTRL1 register
;
;
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled
; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done
    ADCDELAY
    BIT ADCTRL1,8
    BCND $-1, TC
    ADCDELAY
    LDP #DP_PPF1
    LACC ADCFIFO1,10
    LDP #dp_d
    SACH _vc

    LDP #DP_PPF1
SPLK #1010100100000111b,ADCTRL1 ;init ADCTRL1 register
;
;
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled
; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done
    ADCDELAY
    BIT ADCTRL1,8
    BCND $-1, TC
    ADCDELAY
    LDP #DP_PPF1
    LACC ADCFIFO1,10
    LDP #dp_d
    SACH _ia

    LDP #DP_PPF1
SPLK #1010100100001001b,ADCTRL1 ;init ADCTRL1 register
;
;
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled
; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done
    ADCDELAY
    BIT ADCTRL1,8
    BCND $-1, TC
    ADCDELAY
    LDP #DP_PPF1
    LACC ADCFIFO1,10
    LDP #dp_d
    SACH _ib

    LDP #DP_PPF1
SPLK #1010100100001011b,ADCTRL1 ;init ADCTRL1 register
;
;
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled
; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done
    ADCDELAY
    BIT ADCTRL1,8
    BCND $-1, TC
    ADCDELAY
    LDP #DP_PPF1
    LACC ADCFIFO1,10
    LDP #dp_d
    SACH _ic

    .if VDC
    LDP #DP_PPF1
SPLK #10101001000001101b,ADCTRL1 ;init ADCTRL1 register
;
;
; bit 15-14 10: 10 = stop after current conversion on emu suspend
; bit 13 1: 1 = immediate start of conversion
; bit 12 0: 1 = ADC2 enabled
; bit 11 1: 1 = ADC1 enabled
; bit 10 0: 1 = continuous mode enabled
; bit 9 0: 1 = ADC interrupt enabled
; bit 8 0: ADCINTFLAG, write 1 to clear
; bit 7 0: ADCEOC, read-only
; bit 6-4 000: ADC2CHSEL (001=ch 1)
; bit 3-1 001: ADC1CHSEL (000=ch 0)
; bit 0 0: 1 = start conversion when current conversion done
    ADCDELAY
    BIT ADCTRL1,8
    BCND $-1, TC
    ADCDELAY
    LDP #DP_PPF1
    LACC ADCFIFO1,10
    LDP #dp_d
    SACH _vdc
    .endif

    SETC SXM
    RET

_s_send:
    LDP #_sdat
    LACL _sdat
    LDP #DP_PPF1
;t2: BIT SCICTL2, BIT7
; BCND t2, NTC
    SACL SCITXBUF
    LDP #_sdat
    RET

_s_receive:
    LDP #DP_PPF1
t1: BIT SCIRXST, BIT6
BCND t1, NTC
    LACC SCIRXBUF
    LDP #_sdat
    SACL _sdat
    RET

.end

C.3.2 comando.c

#include<math.h>

/*
#define DEBUG
*/

extern int va, vb, vc, ia, ib, ic, vdc,sdat;
extern int delta1,delta2;
extern int scmd1,scmd2,cst1,cst2,wf1,wf2;
extern void START(void);
extern void ADC(void);
extern void RECEIVE(void);
extern void TRANSMIT(void);
extern void RSTINT(void);
extern void s_send(void);
extern void s_receive(void);

float VALFA, VBETA,IALFA,IBETA,deltaq;
/*float deltaf_ini,deltaf,tau,fa,fb;*/
unsigned long vl,il;
unsigned int cst;
float P_cnv,Q_cnv,QCerr,QCref,QCint,QCini;
float falfai,fbetai;
float falfai,fbetai;
int b[4];

```

```

long sref,sp,sq;
int sdelta1;
int sdelta2;
int trip1,trip2,tp1,tp2;
float fa0,fb1;
float P_cnv0,Q_cnv0,P_cnv1,Q_cnv1;

#define deltat 5e-4

float ki,kp;

float ddf;

#ifdef DEBUG
float t=0,d=0.02;
#define dt 0.09425;
interrupt void comando(void);
#endif

void
main()
{
  QCref=0;
  P_cnv=0;
  Q_cnv=0;
  QCini=0;
  sref=0;
  /*deltaf_ini=0;*/
  falfav=sqrt(2.0/3.0)/1.7067;
  fbetav=(sqrt(2.0)/2.0)/1.7067;
  falfai=sqrt(2.0/3.0)/2.56;
  fbetai=(sqrt(2.0)/2.0)/2.56;
  /*
  ki=0.001;
  kp=0.000025;
  tau=0.005;
  */
  ki=0.001;
  kp=0.000025;
  /*tau=0.005;*/

  /*
  fa=deltat/(tau+deltat);
  fb=1.0/(1.0+deltat/tau);
  */
  /* fx=exp(-2*M_PI*timestep/0.08);*/

  fa0=0.03850884;
  fb1=0.96149116;

  cst1=0;
  cst2=0;
  scmd1=0;
  scmd2=0;
  trip1=0;
  trip2=0;

  START();
  while(1)
  {
    QCref=(sref);
    sp=(long)(P_cnv);
    sq=(long)(Q_cnv);
    sdelta1=delta1+5461;
    sdelta2=delta2+5461;

    cst=0;
    /*vl e il estão elevadas ao quadrado*/
    if(il > 62500) cst= cst | 0x81;
    if(vl < 2500) cst= cst | 0x82;
    if(vl > 62500) cst= cst | 0x84;

    /*vdc está multiplicado por (1024/300) = 3,4133*/
    // if(vdc < 48) cst= cst | 0x88;
    if(vdc > 990) cst= cst | 0x90;

    if(((trip1+trip2)==0)&&((tp1+tp2)!= (trip1+trip2)))
    {
      scmd1=0x00;
      scmd2=0x00;
      QCref=0;
      P_cnv=0;
      P_cnv0=0;
      P_cnv1=0;
      Q_cnv=0;
      Q_cnv0=0;
      Q_cnv1=0;
      QCini=0;
      sref=0;
      /* deltaf_ini=0;*/
    };
    tp1=trip1;
    tp2=trip2;

    /*if((cst != 0)||((cst1 != 0)&&(cst1 != 0x88))
    ||((cst2 != 0)&&(cst1 != 0x88)))*/
    if(((trip1+trip2)!=0)||((cst != 0))
    {
      scmd1=0xFF;
      scmd2=0xFF;
    }

    sdat=0;
    while(sdat != 0xAA)s_receive();
    s_send();
    while(sdat != 0x55)s_receive();
    s_send();

    s_receive();
    b[0]=sdat;
    s_send();
    s_receive();
    b[1]=sdat;
    s_send();
    s_receive();
    b[2]=sdat;
    s_send();
    s_receive();
    b[3]=sdat;
    s_send();
    sref=(11*b[3]<<24)+(11*b[2]<<16)+(11*b[1]<<8)+b[0];

    s_receive();
    b[0]=sdat;
    s_send();
    s_receive();
    b[1]=sdat;
    s_send();
    trip1=(11*b[1]<<8)+b[0];

    s_receive();
    b[0]=sdat;
    s_send();
    s_receive();
    b[1]=sdat;
    s_send();
    trip2=(11*b[1]<<8)+b[0];

    b[0]=(0x000000FF1&sdelta1);
    b[1]=(0x0000FF001&sdelta1)>>8;
    s_receive();
    sdat=b[0];
    s_send();
    s_receive();
    sdat=b[1];
    s_send();

    b[0]=(0x000000FF1&sdelta2);
    b[1]=(0x0000FF001&sdelta2)>>8;
    s_receive();
    sdat=b[0];
    s_send();
    s_receive();
    sdat=b[1];
    s_send();

    b[0]=(0x000000FF1&sp);
    b[1]=(0x0000FF001&sp)>>8;
    b[2]=(0x00FF00001&sp)>>16;
    b[3]=(0xFF0000001&sp)>>24;
    s_receive();
    sdat=b[0];
    s_send();
    s_receive();
    sdat=b[1];
    s_send();
    s_receive();
    sdat=b[2];
    s_send();
    s_receive();
    sdat=b[3];
    s_send();

    b[0]=(0x000000FF1&sq);
    b[1]=(0x0000FF001&sq)>>8;
    b[2]=(0x00FF00001&sq)>>16;
    b[3]=(0xFF0000001&sq)>>24;
    s_receive();
    sdat=b[0];
    s_send();
    s_receive();
    sdat=b[1];
    s_send();
    s_receive();
    sdat=b[2];
    s_send();
    s_receive();
  }
}

```



```

sdatt=b[3];
s_send();

b[0]=(0x000000FF1&cst1);
b[1]=(0x000000FF1&cst);
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();

b[0]=(0x000000FF1&cst2);
b[1]=(0x000000FF1&cst);
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();

b[0]=(0x000000FF1&wf1);
b[1]=(0x0000FF001&wf1)>>8;
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();

b[0]=(0x000000FF1&wf2);
b[1]=(0x0000FF001&wf2)>>8;
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();

b[0]=(0x000000FF1&v1);
b[1]=(0x0000FF001&v1)>>8;
b[2]=(0x00FF00001&v1)>>16;
b[3]=(0xFF0000001&v1)>>24;
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();
s_receive();
sdatt=b[2];
s_send();
s_receive();
sdatt=b[3];
s_send();

b[0]=(0x000000FF1&i1);
b[1]=(0x0000FF001&i1)>>8;
b[2]=(0x00FF00001&i1)>>16;
b[3]=(0xFF0000001&i1)>>24;
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();
s_receive();
sdatt=b[2];
s_send();
s_receive();
sdatt=b[3];
s_send();

b[0]=(0x000000FF1&vdc);
b[1]=(0x0000FF001&vdc)>>8;
s_receive();
sdatt=b[0];
s_send();
s_receive();
sdatt=b[1];
s_send();

#ifdef DEBUG
comando();
#endif
};
}

interrupt void
comando()
{
#ifdef DEBUG
ADC();
#else

```

```

va=(int)(512.0*sin(t));
vb=(int)(512.0*sin(t-2.0944+0.001));
vc=(int)(512.0*sin(t+2.0944+0.002));
ia=(int)(512.0*sin(t+d+0.003));
ib=(int)(512.0*sin(t-2.0944+d+0.004));
ic=(int)(512.0*sin(t+2.0944+d+0.005));

t+=dt;
#endif
/*
ia=va;
ib=vb;
ic=vc;
*/

//=====
// Controle de potencia Reativa
/*
VALFA = (sqrt(2.0/3.0)*(BUS_A-(BUS_B + BUS_C)/2.0))*1.7067;
VBETA = (sqrt(2.0)*(BUS_B-BUS_C)/2.0)*1.7067;

IALFA = (sqrt(2.0/3.0)*(IC_A - (IC_B + IC_C)/2.0))*2.56;
IBETA = (sqrt(2.0)*(IC_B-IC_C)/2.0)*2.56;

P_cnv = (d_VALFA*d_IALFA + d_VBETA*d_IBETA)/4.269152;

VALFA = (0.8164966*(va-(vb + vc)/2.0));
VBETA = (0.70710678*(vb-vc));

IALFA = (0.8164966*(ia - (ib + ic)/2.0));
IBETA = (0.70710678*(ib-ic));
*/
/*
VALFA= va+vb+vc;
VBETA= ia+ib+ic;
*/
VALFA = (falfav*(va-(vb + vc)/2.0));
VBETA = (fbetav*(vb-vc));

IALFA = (falfai*(ia - (ib + ic)/2.0));
IBETA = (fbetai*(ib-ic));

v1 = (long)(VALFA*VALFA+VBETA*VBETA);
i1 = (long)(IALFA*IALFA+IBETA*IBETA);

// ---- real and imaginary converter instantaneous powers

P_cnv1= P_cnv;
P_cnv0 = ((VALFA*IALFA + VBETA*IBETA));

Q_cnv1= Q_cnv;
Q_cnv0 = ((VBETA * IALFA - VALFA * IBETA));
//S_cnv = sqrt(P_cnv*P_cnv + Q_cnv*Q_cnv);

P_cnv=(fa0*P_cnv0)+(fb1*P_cnv1);
Q_cnv=(fa0*Q_cnv0)+(fb1*Q_cnv1);

//----- Controlador PI-----
QCerr = (QCref-Q_cnv); // sinal de erro

//QCint = QCini + QCerr * deltat2; // integrador

QCint = QCini + QCerr * 5.0e-4; // integrador
QCini=QCint;
/*
if(QCint > 1.570796327)QCini=1.570796327;
if(QCint < -1.570796327)QCini=-1.570796327;

deltaq =ki * QCint + kp * QCerr; // saida=int+proporcional

/*
deltaq =5.0e-4 * QCint + 1.25e-5* QCerr; //saida=int+prop
*/

/*
if(deltaq > 1.570796327)
{
deltaq=1.570796327;
}
if(deltaq < -1.570796327)
{
deltaq=-1.570796327;
};
*/

if(deltaq > 0.314)
{
deltaq=0.314;
}
if(deltaq < -0.314)
{

```

```

        deltaq=-0.314;
    };
/*
    if(deltaq > 1.03)
    {
        deltaq=1.03;
    }
    if(deltaq < -1.03)
    {
        deltaq=-1.03;
    };
*/

/*
// filtro
//deltaf=(deltat2/(tau+deltat2))*deltaq
//+(1.0/(1.0+deltat2/tau))*deltaf_ini;
//deltaf=0.047619047*deltaq+0.952380952*deltaf_ini;
*/

//deltaf=fa*deltaq+fb*deltaf_ini;

/*deltaf=0.000999*deltaq+0.999*deltaf_ini;*/
//deltaf_ini=deltaf;

/*delta1+=6827; 37.5 graus*/
/*delta2+=4096; 22.5 graus*/
/*
delta1-=4096; //-22.5 graus
delta2-=6827; //-37.5 graus
*/

//ddf=(deltaf*10430.22);
ddf=(deltaq*10430.22);
/*trocar na planta real*/
/*
delta1=(int)(ddf-5461.0);
delta2=(int)(ddf-5461.0);
*/

delta1=(int)(ddf-2730.5);
delta2=(int)(ddf-8191.5);

/*
delta1=(int)(ddf+8191.5);
delta2=(int)(ddf+2730.5);
*/
/*
delta1=(int)(-2730.5);
delta2=(int)(-8191.5);
*/
#ifndef DEBUG
RECEIVE();
/*
#80h erro
#01h BIT B0
#02h BIT B1
#04h BIT B2
#08h scmd
#10h f menor
#20h f maior
*/

TRANSMIT();
#endif
}

```

```

DAC .set 0 ;use output to DAC

FSIZE .set 16
FSHIFT .set 4

.if ASM
.title "pll"
.length 1000
.setsect ".data", 8100h,1
.setsect ".text", 08200h,0
.setsect ".vectors", 08000h,0
.setsect ".buffer", 08100h,0
.else
.global _c_int0, _dsp_pll
.endif

dp_d .set 0300h>>7
;20e6/(4*125)=40000
;20e6/(4*200)=25000

;ADC_rate .set 142 ;35000Hz sampling rate
ADC_rate .set 142

PI_2 .set 3FFFh
PI .set 7FFFh

;alfa1 .set 0754 ;04.1443 * M_PI/180 //==> eliminar 11,13 e 23
;alfa2 .set 2175 ;11.9490 * M_PI/180
;alfa3 .set 2474 ;13.5912 * M_PI/180
;alfa1 .set 1092 ; 6 * M_PI/180 //==> eliminar 11,13 e 23
;alfa2 .set 2731 ;15 * M_PI/180
;alfa3 .set 3277 ;18 * M_PI/180
alfa1 .set 806 ; 4.4262 * M_PI/180 //==> eliminar 11,13 e 23
alfa2 .set 3284 ;18.0406 * M_PI/180
alfa3 .set 4204 ;23.0919 * M_PI/180

DISP1 .set 00001h
DISP2 .set 00002h
DISP3 .set 00004h
DISP4 .set 00008h
DISP5 .set 00010h
DISP6 .set 00020h

;FMIN .set 60
;FMAX .set 170

FMIN .set 0
FMAX .set 17000

ADCDELAY .macro
nop
nop
nop
nop
.endm

;*****
; inicializa labels para o f243
;*****
.include "f243.h"

.sect ".data"
va .word 0
vb .word 0
vc .word 0
w0 .word 0
w1 .word 0
valfa0 .word 0
valfa1 .word 0
vbeta0 .word 0
ialfa1 .word 0
ibeta1 .word 0
q .word 0
dq1 .word 0
dq2 .word 0
theta0 .word 0
theta1 .word 0
theta .word 0
delta .word 0
dout .word 0
temp .word 0 ;variavel de uso geral
t .word 0 ;tempo de simulacao
count .word 0
scount .word 0
scmd .word 0 ;server comand
erro .word 0FFFFh ;status de erro
fw .word 0
wf .word 0
.space FSIZE*16
sstack .word 0,0,0,0,0
;sin stack

m_scmd .word 0
m_fw .word 0
m_erro .word 0
m_temp .word 0

```

C.4 Disparo

C.4.1 _disparo.asm

```

-----
;Filename: pllfinal.asm
;
;Author: Luis Claudio G. Lopes.
;
;Last Modified: 05/1/2005
;
;Description: pll
-----

NUM .set 0 ;Conversor 1(1) ou 2(0)
ASM .set 0 ;assembler ou code composer
EMU .set 0 ;use va, vb and vc emulated
BUF .set 0 ;use output buffer
TGG .set 0 ;use output pin

```

```

isr1 .word 0
isr0 .word 0
iacch .word 0
iaccl .word 0

.sect ".buffer"
buff1 .word 0
.space 199*16
ebuff1 .word 0

buff2 .word 0
.space 199*16
ebuff2 .word 0

;*****
; setup dos vetores de interrupção *
;*****
.sect ".vectors"
.if ASM
B init ;00h reset
int1: B int1 ;02h INT1
int2: B dsp_pll ;04h INT2
.else
B _c_int0 ;00h reset
int1: B int1 ;02h INT1
int2: B _dsp_pll ;04h INT2
.endif
int3: B int3 ;06h INT3
int4: B int4 ;08h INT4
int5: B int5 ;0Ah INT5 (reservado pelo DSK)
int6: B int6 ;0Ch INT6
int7: B int7 ;0Eh reserved
int8: B int8 ;10h INT8 user-defined
int9: B int9 ;12h INT9 user-defined
int10: B int10 ;14h INT10 user-defined
int11: B int11 ;16h INT11 user-defined
int12: B int12 ;18h INT12 user-defined
int13: B int13 ;1Ah INT13 user-defined
int14: B int14 ;1Ch INT14 user-defined
int15: B int15 ;1Eh INT15 user-defined
int16: B int16 ;20h INT16 user-defined
int17: B int17 ;22h TRAP
int18: B int18 ;24h NMI
int19: B int19 ;26h reserved
int20: B int20 ;28h INT20 user-defined
int21: B int21 ;2Ah INT21 user-defined
int22: B int22 ;2Ch INT22 user-defined
int23: B int23 ;2Eh INT23 user-defined
int24: B int24 ;30h INT24 user-defined
int25: B int25 ;32h INT25 user-defined
int26: B int26 ;34h INT26 user-defined
int27: B int27 ;36h INT27 user-defined
int28: B int28 ;38h INT28 user-defined
int29: B int29 ;3Ah INT29 user-defined
int30: B int30 ;3Ch INT30 user-defined
int31: B int31 ;3Eh INT32 user-defined

.sect ".text"
;*****
;* Routine Name: init (setup and main loop) *
;* Initialize DSP *
;* Pointed register AR3 and AR4 *
;*****
_c_int0:
init:
;*****
; Desativa interrupcoes *
;*****
SETC INTM
;*****
; desliga todos registros do EV *
;*****
LACL #0
LDP #DP_EV
SACL GPTCON
SACL T1CNT
SACL T1CMPR
SACL T1PR
SACL T1CON
SACL T2CNT
SACL T2CMPR
SACL T2PR
SACL T2CON
SACL COMCON
SACL ACTR
SACL DBTCON
SACL CMPR1
SACL CMPR2
SACL CMPR3
SACL CAPCON
SACL CAPFIFO

;*****
; desabilita watchdog *
;*****
LDP #DP_PF1

SPLK #01101000b,WDCR
;
; | | | | | | | |
; 76543210
; bit 7 0: clear WD flag
; bit 6 1: disable de dog
; bit 5-3 101: must be written as 101
; bit 2-0 000: WDCLK divider =1

;*****
; setup da saida *
;*****
; setup dos pinos compartilhados de I/O *
;*****
LDP #DP_PF2
LACC OCRA
OR #0011000000000000b
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15 0: 0=IOPB7, 1=TCLKIN
; bit 14 0: 0=IOPB6, 1=TDIR
; bit 13 1: 0=IOPB5, 1=T2PWM/T2CMP
; bit 12 0: 0=IOPB4, 1=T1PWM/T1CMP
; bit 11 0: 0=IOPB3, 1=PWM6
; bit 10 0: 0=IOPB2, 1=PWM5
; bit 9 0: 0=IOPB1, 1=PWM4
; bit 8 0: 0=IOPB0, 1=PWM3
; bit 7 0: 0=IOPB7, 1=PWM2
; bit 6 0: 0=IOPB6, 1=PWM1
; bit 5 0: 0=IOPB5, 1=CAP3
; bit 4 0: 0=IOPB4, 1=CAP2/QEP1
; bit 3 0: 0=IOPB3, 1=CAP1/QEP0
; bit 2 0: 0=IOPB2, 1=XINT1
; bit 1 0: 0=IOPB1, 1=SCIRXD
; bit 0 0: 0=IOPB0, 1=SCITXD
;
SACL OCRA

LACC OCRA
AND #0000000011111111b ;IOPB
FEDCBA9876543210
SACL OCRA

LACC OCRB
OR #0000000111000011b ;IOPD0 + IOPC
AND #111110111000011b ;IOPD1 + IOPC
FEDCBA9876543210
SACL OCRB

SPLK #0F800h,PBDATDIR ;
SPLK #0FF00h,PCDATDIR ;todos saida
SPLK #0FF00h,PDDATDIR ;todos saida

LDP #dp_d
SPLK #0FF00h,dout ;todos saida

;*****
; setup do registro de controle do sistema *
;*****
LDP #DP_PF1
SPLK #0000000000000000b,SCSR
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15 0: reserved
; bit 14 0: CLKOUT=CPUCLK
; bit 13-12 00: IDLE selected for low-power mode
; bit 11-1 0's: reserved
; bit 0 1: clear the ILLADR bit

;*****
; setup do gerador de espera *
;*****
LDP #dp_d
SPLK #0000000000000000b,temp
;
; | | | | | | | |
; FEDCBA9876543210
; bit 15-11 0s: reserved
; bit 10-9 00: bus visibility off
; bit 8-6 001: 1 wait-state for I/O space
; bit 5-3 000: 0 wait-state for data space
; bit 2-0 000: 0 wait-state for program space
;
OUT temp, WSGR

;*****
; setup das interrupcoes *
;*****
LDP #0h
LACL #000010b
SACL IFR
OR IMR
SACL IMR

;*****
; setup do ADC *
;*****
ADC:
LDP #DP_PF1
SPLK #0000000000000000b,ADCTRL2 ;init ADCTRL2 register

```

```

;          |||||||
;          FEDCBA9876543210
; bit 15   0: reserved
; bit 14   0: 0 = ADCINTFLAG set when FIFO has at least 1 word
; bit 13   1: 1 = clear EVSOCP bit
; bit 12   1: 1 = clear EXTSOCP bit
; bit 11   0: 0 = high interrupt priority
; bit 10   0: 1 = enable conversion start by EV
; bit 9    0: 1 = enable conversion start by ADCSOC pin
; bit 8    0: 0 = reserved
; bit 7-6  00: 00 = ADCFIFO2 status - read-only
; bit 5    0: 0 = reserved
; bit 4-3  00: 00 = ADCFIFO1 status - read-only
; bit 2-0  000: 000 = clock prescaler, ADCCLK=CPUCLK/1

;*****
; setup dos registros de modo da CPU *
;*****
SETC SXM      ; usa o modo de sinalizacao estendido
CLRC OVM     ; normaliza overflow
SPM 0        ; seta ALU para multiplicacao fracionaria

;*****
; setup do contador *
;*****
contador:
    LDP #DP_EV

; configuracao T1CON GP Timer 1

    SPLK #0000001001000000b,T1CON ;GP Timer control
;          |||||||
;          FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13    0: reserved
; bit 12-11 00: 00 = stop and hold mode
; bit 10-8  010: 010 = x/4 prescaler
; bit 7     0: 0 = use own TENABLE bit
; bit 6     1: 1 = enable timer
; bit 5-4   00: 00 = CPUCLK is clock source
; bit 3-2   00: 00 = reload compare reg on underflow
; bit 1     0: 0 = disabled timer compare
; bit 0     0: 0 = use own period register

; configuracao T2CON GP Timer 2

    SPLK #00000010010000010b,T2CON ;GP Timer control
;          |||||||
;          FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13    0: reserved
; bit 12-11 00: 00 = stop and hold mode
; bit 10-8  010: 010 = x/4 prescaler
; bit 7     0: 0 = use own TENABLE bit
; bit 6     1: 1 = enable timer
; bit 5-4   00: 00 = CPUCLK is clock source
; bit 3-2   00: 00 = reload compare reg on underflow
; bit 1     0: 0 = disabled timer compare
; bit 0     0: 0 = use own period register

; configuracao GPTCON

    SPLK #0000000001001010b,GPTCON ;GP Timer control
;          |||||||
;          FEDCBA9876543210
; bit 15    0: reserved
; bit 14    0: T2STAT - read-only
; bit 13    0: T1STAT - read-only
; bit 12-11 00: reserved
; bit 10-9  00: T2TOADC, 00 = no ADCSOC
; bit 8-7   00: T1TOADC, 10 = ADCSOC on period match
; bit 6     1: 1 = enable all timer compare outputs
; bit 5-4   00: reserved
; bit 3-2   10: 00 = T2PIN forced low
; bit 1-0   00: 00 = T1PIN forced low

; configuracoes T1PR,T1CMPR e T1CNT
SPLK #ADC_rate,T1PR ;periodo
    SPLK #OFFFEh,T1CNT ;limpa contador
; configuracoes T1PR,T1CMPR e T1CNT
; SPLK #ADC_rate,T2PR ;periodo
; SPLK #0010h,T2CMPR
; SPLK #OFFFEh,T2CNT ;limpa contador

;*****
; inicio do contador *
;*****
    LDP #DP_EV
; inicializa GP Timer 1

    SPLK #0001001001000000b,T1CON ;GP Timer control
;          |||||||
;          FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13    0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8  010: 010 = x/4 prescaler
; bit 7     0: 0 = use own TENABLE bit
; bit 6     1: 1 = enable timer
; bit 5-4   00: 00 = CPUCLK is clock source
; bit 3-2   00: 00 = reload compare reg on underflow
; bit 1     0: 0 = disabled timer compare
; bit 0     0: 0 = use own period register

; bit 6     1: 1 = enable timer
; bit 5-4   00: 00 = CPUCLK is clock source
; bit 3-2   00: 00 = reload compare reg on underflow
; bit 1     0: 0 = disabled timer compare
; bit 0     0: 0 = use own period register

; inicializa GP Timer 2

    SPLK #00010010010000010b,T2CON ;GP Timer control
;          |||||||
;          FEDCBA9876543210
; bit 15-14 00: 10 = stop immediately on emulator suspend
; bit 13    0: reserved
; bit 12-11 10: 10 = continuous-up count mode
; bit 10-8  010: 010 = x/4 prescaler
; bit 7     0: 0 = use own TENABLE bit
; bit 6     1: 1 = enable timer
; bit 5-4   00: 00 = CPUCLK is clock source
; bit 3-2   00: 00 = reload compare reg on underflow
; bit 1     0: 0 = disabled timer compare
; bit 0     0: 0 = use own period register

; setup das interrupcoes do EV *
;*****

    LDP #DP_EV
    SPLK #OFFFh, EVIFRA ;clear interrupt flags
    SPLK #OFFFh, EVIFRB ;clear interrupt flags
    SPLK #OFFFh, EVIFRC ;clear interrupt flags

; period timer1 match interrupt
    SPLK #00080h, EVIMRA ;enable desired interrupts
    SPLK #00000h, EVIMRB ;enable desired interrupts
    SPLK #00000h, EVIMRC ;enable desired interrupts

    LDP #dp_d

    LAR AR3,#buff1 ;buffer preparation
    LAR AR4,#buff2 ;buffer preparation

    LACL theta
    ADD #03FFFh
    CALL dsp_sin
    SACL ialfa1
    LACL theta
    CALL dsp_sin
    LACL ibeta1

;-----
; CAN Bit Timing Configuration - for 1Mbps, CLKOUT= 20MHz
;-----

    LDP #DP_CAN1
    SPLK #0001000000000000b, MCR
; bit 12 CCR Change configuration request
; CCR=1: write access to BCRn registers

.if EMU
.else
W_CCE BIT GSR, BIT4 ; TC=CCE, Wait for Change configuration
BCND W_CCE, NTC
.endif

    SPLK #000000000000000b, BCR2
; bit 0-7 BRP=0 Baud Rate Prescaler
; bit 8-15 Reserved

    SPLK #0000000111111010b, BCR1
;          |||||||
;          FEDCBA9876543210
; bit 0-2 TSEG2=2
; bit 3-6 TSEG1=15
; bit 7 SAM=1 Sample point setting
; bit 8-9 SJW=1 Synchronization jump width
; bit A SBG=0 - synchronization on the falling edge only
; bit B-F Reserved

    SPLK #000000000000000b, MCR
; bit 12 CCR Change configuration request
; CCR=0: request normal operation

W_NCCE BIT GSR, BIT4 ; TC=CCE
BCND W_NCCE, TC

;-----
; Disable CAN Mailboxes before writing
;-----

    SPLK #000000000000000b, MDER
;          |||||||
;          FEDCBA9876543210
; bit 0-5 MEn=0 - disable all mailboxes
; bit 6-7 MDn Mailbox direction for mailbox n
; bit 8-15 Reserved

;-----
; Writing in CAN Mailbox 2 - Receive
;-----

```

```

;-----
LDP #DP_CAN2
SPLK #1010111000010101b, MSGID2H
;
; FEDCBA9876543210
;bit 0-12 IDH - Upper 13 bits of extended identifier
;bit 13 AAM=1 - if a matching remote request is received,
;the CAN Peripheral
; answers by sending the contents of the mailbox
;bit 14 AME=0
;bit 15 IDE=1 - the received message has an extended ID (29 bits)

SPLK #1101110000110101b, MSGID2L
;bit 0-15 IDL - Lower part of extended identifier

SPLK #0000000000001000b, MSGCTRL2
;
; FEDCBA9876543210
;bit 0-3 DLC Data length code=1000 = 8 bytes (data bytes)
;bit 4 RTR=0 - Data frame
;bit 5-15 Reserved

;-----
; Writing in CAN Mailbox 5 - Transmit
;-----

LDP #DP_CAN2
SPLK #1010111000010101b, MSGID5H
;
; FEDCBA9876543210
;bit 0-12 IDH - Upper 13 bits of extended identifier
;bit 13 AAM=1 - if a matching remote request is received,
;the CAN Peripheral
; answers by sending the contents of the mailbox
;bit 14 AME=0
;bit 15 IDE=1 - the received message has an extended ID (29 bits)
.if NUM
SPLK #1101110000110110b, MSGID5L
.else
SPLK #1101110000110100b, MSGID5L
.endif
;bit 0-15 IDL - Lower part of extended identifier

SPLK #0000000000001000b, MSGCTRL5
;
; FEDCBA9876543210
;bit 0-3 DLC Data length code=1000 = 8 bytes (data bytes)
;bit 4 RTR=0 - Data frame
;bit 5-15 Reserved

SPLK #0000h, MBX5A ; Message to transmit (2 bytes)
SPLK #0000h, MBX5B
SPLK #0000h, MBX5C
SPLK #0000h, MBX5D ; total= 8 bytes

;-----
; Enable Mailbox 2,5 after writing
;-----

LDP #DP_CAN1
SPLK #0000000001100100b, MDER
;
; FEDCBA9876543210
; bit 0 MEO=0 - disable mailbox 0
; bit 1 ME1=0 - disable mailbox 1
; bit 2 ME2=1 - enable mailbox 2
; bit 3 ME3=0 - disable mailbox 3
; bit 4 ME4=0 - disable mailbox 4
; bit 5 ME5=0 - disable mailbox 5
; bit 6 MD2=1 - mailbox 2 receive
; bit 7 MD3=0 - mailbox 3 transmit
; bit 8-15 Reserved

;*****
; Ativa interrupcoes *
;*****

LDP #dp_d
.if EMU
SPLK #0,t
main:
CALL dsp_pll
LDP #dp_d
LACL t
ADD #786 ;10430*377/FS
ADD #98
SACL t
B main
.else
LDP #DP_EV
SPLK #0FFFh, EVIFRA ;clear interrupt flags
CLRC INTM

```

```

LDP #dp_d
;
; LACC #0
;
; SACL erro
main:
;
; nop
;
; nop
;
; B main

;le variaveis
BLDD #fw,m_fw
BLDD #scmd,m_scmd
BLDD #erro,m_erro

LDP #DP_PF2
LACC PBDATDIR
LDP #m_temp
SACL m_temp

;verifica status
LACC m_fw
SUB #FMIN
BCND ERR,LEQ

LACC m_fw
SUB #FMAX
BCND ERR,GEQ

;LER B
BIT m_temp,BIT0
BCND ERR,TC

BIT m_temp,BIT1
BCND ERR,TC

BIT m_temp,BIT2
BCND ERR,TC

;testa comando
BIT m_scmd,BIT0
BCND ERR,TC

SPLK #0h,m_erro

CLRC XF
B CANTRA

ERR:

;define erro
SPLK #80h,m_erro
LDP #DP_PF2
LACC PBDATDIR
LDP #m_temp
SACL m_temp

BIT m_temp,BIT0
BCND fe1,NTC
LACC m_erro
OR #01h
SACL m_erro

fe1:
BIT m_temp,BIT1
BCND fe2,NTC
LACC m_erro
OR #02h
SACL m_erro

fe2:
BIT m_temp,BIT2
BCND fe3,NTC
LACC m_erro
OR #04h
SACL m_erro

fe3:
BIT m_scmd,BIT0
BCND fe4,NTC
LACC m_erro
OR #08h
SACL m_erro

fe4:
LACC m_fw
SUB #FMIN
BCND fe5,GT
LACC m_erro
OR #10h
SACL m_erro

fe5:
LACC m_fw
SUB #FMAX
BCND fe6,LT
LACC m_erro
OR #20h
SACL m_erro

fe6:

```

```

SETC XF

CANTRA:
;transmitir erro
LDP #DP_CAN1
SPLK #0080h, TCR
;bit 7 TRS5=1 - write access to mailbox 5 is denied (negado),
;and the message will be transmitted

W_TA: BIT TCR, BIT15 ; TC=TA5 - transmission acknowledge
BCND W_TA, NTC

LDP #m_erro
LACL m_erro
LDP #DP_CAN2
SACL MBX5A

LDP #m_fw
LACL m_fw
LDP #DP_CAN2
SACL MBX5B

LDP #DP_CAN1
SPLK #8000h, TCR ; Reset TA5 (and CANIFR)

LDP #m_erro
LACC m_erro
SACL erro
;debug

B main

.endif

;*****
; interrupcao *
;*****
;* Routine Name: dsp_pll (main interrupt) *
;* Detect phase angle *
;* Output : theta and gpio pulses *
;* Pointed register AR3 and AR4 *
;*****

_dsp_pll:
dsp_pll:
I$$SAVE:
LAR AR1,#isr1
MAR *,AR1
SST #1,**
SST #0,**
SACH **,0
SACL *,0

;togle
.if TGG
LDP #dp_d
LACC dout
OR #00C0h
SACL dout
;disparo externo
LACC dout
LDP #DP_PF2
SACL PDDATDIR
LDP #dp_d
.endif

.if EMU = 0
SETC INTM

CALL readADC

LDP #dp_d
LACL va
SUB #512
SACL va
LACL vb
SUB #512
SACL vb
LACL vc
SUB #512
SACL vc
SETC INTM
.else
LDP #dp_d
LACL t
CALL dsp_sin
RPT #5
SFR
SACL va
LACL t
SUB #21845
CALL dsp_sin
RPT #5
SFR
SACL vb
LACC #0
SUB va
SUB vb
SACL vc
.endif
LDP #dp_d
; BLDD #valfa0, valfa1
; BLDD #vbeta0, vbeta1
; valfa[0]=(short)(1024.0*(vaf-0.5*(vbf+vcf)));
LACC va,1
SUB vb
SUB vc
SFR
SACL valfa0
vbeta[0]=(short)(1024.0*((sqrt(3.0)/2.0)*vbf
-(sqrt(3.0)/2.0)*vcf));
LACC vb
SUB vc
SACL vbeta0
LT vbeta0
MPY #3547
PAC
SACH vbeta0,4
q=muladd(valfa0,ialfa1,vbeta0,ibeta1);
LT valfa0
MPY ialfa1
PAC
LT vbeta0
MPY ibeta1
APAC
ADD 0400h
SACH q,5
dq2 = dq1;
BLDD #dq1,dq2
dq1 = q - qr;
BLDD #q,dq1
w1=w0;
BLDD #w0,w1
w0=muladd(4437,dq1,-4292,dq2)+w1;
; LT dq1
; MPY #008AAh
; PAC
; LT dq2
; MPY #0F79Eh
; APAC
; SACH w0,2
; LT dq1
; MPY #0091eh
; MPY #3000;ganho PLL
; PAC
; LT dq2
; MPY #0F8E9h
; MPY #-1999
; APAC
; ADD 0800h
; SACH w0,4

LACC w0
ADD w1
SACL w0
theta1=theta0;
BLDD #theta0,theta1
theta0=w0+theta1;
CLRC SXM
ADD theta1
SACL theta0
LACC theta0
ADD #03FFFh
CALL dsp_sin
SACL ialfa1
LACC theta0
CALL dsp_sin
SACL ibeta1
SETC SXM
;diminui delta de theta
LACL theta0
ADD delta
SACL theta

;verifica se ha erro
LACC erro
BCND sdisp,EQ

;desliga disparo
LACC dout
OR #DISP1
OR #DISP2
OR #DISP3
OR #DISP4

```

```

OR #DISP5
OR #DISP6
SACL dout
    B enddisp
sdisp:
;
    disparo 1
    LACL theta
    CALL dsp_disparo
    BCND d1lt, LT
d1gt:
    LACC dout
    OR #DISP2
    AND #~ DISP1
    B d1end
d1lt:
    LACC dout
    OR #DISP1
    AND #~ DISP2
d1end:
    SACL dout
;
    disparo 2
    LACL theta
    SUB #21845
    CALL dsp_disparo
    BCND d2lt, LT
d2gt:
    LACC dout
    OR #DISP4
    AND #~ DISP3
    B d2end
d2lt:
    LACC dout
    OR #DISP3
    AND #~ DISP4
d2end:
    SACL dout
;
    disparo 3
    LACL theta
    ADD #21845
    CALL dsp_disparo
    BCND d3lt, LT
d3gt:
    LACC dout
    OR #DISP6
    AND #~ DISP5
    B d3end
d3lt:
    LACC dout
    OR #DISP5
    AND #~ DISP6
d3end:
enddisp:
    SACL dout

.if DAC
;saida dac
    LDP #dp_d
    LACC theta
    RPT #3
    SFR
    OR #OFF00h
    LDP #DP_P2
    SACL PBDATDIR
    LDP #dp_d
    LACC theta
    RPT #11
    SFR
    OR #OFF00h
    LDP #DP_P2
    SACL PCDATDIR
;
    LDP #dp_d
;
    LACC dq1
;
    OR #OFF00h
;
    LDP #DP_P2
;
    SACL PCDATDIR
.endif

;buffer
.if BUF
buffer:
    LDP #dp_d
    LACC scount
    SUB #1
    SACL scount
    BCND endb,GT
    SPLK #8,scount
;LACL theta0
;LACL dq1
;LACL w0
LACL va
;buffer1
    MAR *,AR3
    SACL **;
;coloca resultado no buffer

    LACL dout
                                AND #DISP1
                                ;LACL va
                                ;LACL valfa0
;buffer2
    MAR *,AR4
    SACL **;
;coloca resultado no buffer

    LACL #200
;buffersize
    SUB count
    BCND end_buff ,GT
    SPLK #0,count
    LAR AR3,#buff1 ;buffer preparation
    LAR AR4,#buff2 ;buffer preparation
end_buff:
    LACL count;
    ADD #1;
    SACL count;
endb:
.endif

;-----
; RECEIVE
;-----
CAN:
;
    LDP #dp_d
;
    LACL #0
;
    SACL delta
;
    B W_RMP ;teste

    LDP #DP_CAN1

    BIT RCR, BIT6
    BCND W_RMP, NTC
;W_FLAG2 BIT CAN_IFR, BIT10
;
    BCND W_FLAG2, NTC

    SPLK #0040h, RCR
; Reset RMP2 (and CANIFR)

.if NUM
    LDP #DP_CAN2
    LACL MBX2A
    LDP #dp_d
    SACL delta
    LDP #DP_CAN2
    LACL MBX2B
    LDP #dp_d
    SACL scmd
.else
    LDP #DP_CAN2
    LACL MBX2C
    LDP #dp_d
    SACL delta
    LDP #DP_CAN2
    LACL MBX2D
    LDP #dp_d
    SACL scmd
.endif
.endif
W_RMP:

;monitora erros de freq
efreq:
    LDP #dp_d
    LACC w0
    ABS
    SACL temp

    LAR AR5,#wf+1
    MAR *,AR5
LDP #wf
RPT #FSIZE-1
    BLDD **,#wf
    LAR AR5,#wf+FSIZE-1
    BLDD #temp,*
    LAR AR5,#wf
    LACC #0;
    RPT #FSIZE-1
    ADD **
    RPT #FSHIFT-1
    SFR
    SACL fw

;toggle
    LDP #dp_d
    LACC dout
    .if TGG
    AND #OFF3Fh
    SACL dout
    .endif
;disparo externo
    LACC dout
    LDP #DP_P2

```



```

;PREG : angle * 0.0372
LACC #0c426h,13 ; in high ACCU
APAC ;ACCU : -0.2338 + angle * 0.0372
SACH * ,3 ;temporary value
;angle in TREG
MPY * ;multiply by temporary value
;temporary value
;stack : angle/rad/TEMP
LACC #6f69h,10 ;0.0544
APAC ;ACCU: temp + 0.0544
SACH *,2 ;temp
;stack : angle/RAD/temp
;angle in TREG
;stack : angle/rad/TEMP
MPY * ;temp
LACC #7dc5h,13 ;0.9826
APAC ;ACCU=0.9826+temp
SACH *,3 ;treg = rad
;temp
MPY * ;rad * temp
;temp * angle
LACC #05532,4 ;0.0013
APAC
SACH * ,3 ;result
LACC *- ;stack : angle/SIN
RET
sin_angle_neg
LT ** ;variable in size_sin
;stack : angle/X
MPY #0c90h ;abscisse scale modif degre -> rad
PAC
SACH *,4 ;angle in rad
;stack : angle/RAD
LT **
;stack : angle/rad/X
MPY #985h ;0.0372
;
;PREG : angle * 0.0372
LACC #03bdah,13 ; in high ACCU
APAC ;ACCU : -0.2338 + angle * 0.0372
SACH * ,3 ;temporary value
;angle in TREG
MPY * ;multiply by temporary value
;temporary value
;stack : angle/rad/TEMP
LACC #6f69h,10 ;0.0544
APAC ;ACCU: temp + 0.0544
SACH *,2 ;temp
;stack : angle/RAD/temp
;angle in TREG
;stack : angle/rad/TEMP
MPY * ;temp
LACC #7dc5h,13 ;0.9826
SPAC ;ACCU=0.9826+temp
SACH *,3 ;treg = rad
MPY * ;rad * temp
;temp * angle
LACC #05532,4 ;0.0013
APAC
SACH *,3 ;result
LACC *- ;stack : angle/SIN
RET
.end

```