

# **Universidade Federal de Itajubá**

Programa de Pós-graduação em Engenharia Elétrica

## **Implementação de um Compensador Ativo Série para Sistemas de Distribuição**

Dissertação apresentada à Universidade Federal de Itajubá para a obtenção do grau de Mestre em Ciências em Engenharia Elétrica.

**Wilson Cesar Sant'Ana**

Itajubá, 28 de Maio de 2004



**Wilson Cesar Sant'Ana**

# **Implementação de um Compensador Ativo Série para Sistemas de Distribuição**

Dissertação apresentada à Universidade Federal de Itajubá para a obtenção do grau de Mestre em Ciências em Engenharia Elétrica.

**Área de Concentração:**  
Automação e Sistemas Elétricos Industriais

**Orientador:**  
Luiz Eduardo Borges da Silva

**Co-Orientador:**  
Valberto Ferreira da Silva

Itajubá, 28 de Maio de 2004



# ***Abstract***

This work presents a practical implementation of a Series Active Power Filter for Voltage Sags and Voltage Harmonics compensation. Aspects concerning equipment development and testing are also discussed. It is shown how a 13.8kV bus can be compensated, even using 220V low cost IGBTs. Problems of the proposed topology are shown as well their solutions.

A contribution to the area, related to algorithms development for Active Power Filters, is also given. It is described a noise-tolerant algorithm, based on the traditional id-iq method. It is explained why traditional techniques are sensitive to noise and shows where the vulnerability is located. Simulation results are presented to compare the proposed method against the traditional. Experimental results are also shown.



# ***Resumo***

Este trabalho apresenta uma implementação de um Compensador Ativo Série para correção de Voltage Sags e Distorções Harmônicas de Tensão. Aspectos relacionados ao desenvolvimento do equipamento e testes são discutidos. Mostra-se como um barramento de 13.8kV pode ser compensado, mesmo com a utilização de IGBTs de baixo custo a 220V. Problemas da topologia proposta são mostrados, bem como suas soluções.

Uma contribuição para a área, relacionada ao desenvolvimento de algoritmos para Compensadores Ativos também é dada. Descreve-se um algoritmo insensível a ruídos, baseado no método id-iq. Explica-se porque as técnicas convencionais são sensíveis ao ruído e mostra-se onde a vulnerabilidade está localizada. Resultados de simulações são apresentados para comparar a técnica proposta contra a convencional. Resultados experimentais também são mostrados.





# ***Agradecimentos***

Agradeço à CAPES e ao povo brasileiro pela ajuda financeira nestes 24 meses de trabalho.

Agradeço à Companhia Energética do Maranhão (CEMAR) pelo apoio financeiro ao projeto que deu origem a este trabalho.

Agradeço aos professores Luiz Eduardo, Valberto e Germano pela oportunidade e pelos desafios que me ofereceram para enriquecer minha experiência profissional.

Agradeço a todos que, direta ou indiretamente, contribuíram para a realização deste trabalho; e em especial ao professor Ismael Noronha, pelas aulas de oficina e caça a ruídos.

Agradeço e dedico este trabalho aos meus pais e familiares por compreenderem e perdoarem a minha ausência.

Agradeço a Deus por ter me mostrado que Ele, realmente, “escreve certo por linhas tortas”; e que, tudo o que nos acontece tem um propósito, embora não entendamos no momento. E este propósito é SEMPRE em nosso benefício.



# Sumário

<b>Abstract</b> .....	i
<b>Resumo</b> .....	iii
<b>Agradecimentos</b> .....	v
<b>Sumário</b> .....	vii
<b>Capítulo 1 - Introdução</b>	
1.1 – Considerações Iniciais .....	1 - 3
1.2 - Referências .....	1 - 5
<b>Capítulo 2 – Caracterização de Voltage Sags</b>	
2.1 - Introdução .....	2 - 3
2.2 – Sag Tipo A .....	2 - 4
2.3 – Sag Tipo B .....	2 - 5
2.4 – Sag Tipo C .....	2 - 6
2.5 – Sag Tipo D .....	2 - 7
2.6 – Sag Tipo E .....	2 - 8
2.7 – Sag Tipo F .....	2 - 9
2.8 – Sag Tipo G .....	2 - 10
2.9 - Referências .....	2 - 11
<b>Capítulo 3 – Compensadores Ativos</b>	
3.1 - Introdução .....	3 - 3
3.2 – Algoritmos de Controle para Compensadores Ativos .....	3 - 6
3.2.1 – Transformação de Coordenadas .....	3 - 10
3.2.2 – Método id-iq .....	3 - 19
3.2.3 – Método id-iq Modificado para Imunidade Contra Ruídos .....	3 - 23
3.2.4 – Comparação entre os Métodos Convencional e Modificado .....	3 - 27
3.2.5 – Compensação em Malha Fechada .....	3 - 29
3.2.5.1 – Ajuste dos Ganhos do Regulador PI .....	3 - 30
3.2.5.2 – Compensação de Harmônicos em Malha Fechada .....	3 - 33
3.2.5.3 – Compensação de Voltage Sags em Malha Fechada .....	3 - 36
3.3 - Referências .....	3 - 50

## **Capítulo 4 – Equipamento Desenvolvido**

<b>4.1 - Introdução</b> .....	4 - 3
<b>4.2 – Condicionamento de Sinais</b> .....	4 - 5
<b>4.3 – VSI e Modulação PWM</b> .....	4 - 8
<b>4.3.1 – Modulação PWM</b> .....	4 - 10
4.3.1.1 – Modulação Bipolar.....	4 - 11
4.3.1.2 – Modulação Unipolar .....	4 - 13
4.3.1.3 – Modulador Desenvolvido .....	4 - 17
4.3.1.3.1 – Gerador de Portadora Triangular.....	4 - 18
4.3.1.3.2 – Comparador Digital.....	4 - 20
4.3.1.3.3 – Interface PWM X VSI.....	4 - 21
4.3.1.3.4 – Barramento .....	4 - 23
<b>4.4 – Filtro Passivo</b> .....	4 - 25
<b>4.5 – Fotografias do Equipamento Desenvolvido</b> .....	4 - 30

## **Capítulo 5 – Resultados Experimentais**

<b>5.1 – Ambiente de Teste</b> .....	5 - 3
<b>5.2 – Resultados Obtidos</b> .....	5 - 7
<b>5.2.1 – Teste a vazio</b> .....	5 - 7
<b>5.2.2 – Teste com Carga Não Linear</b> .....	5 - 11

<b>Conclusão</b> .....	6 - 1
------------------------	-------

## **Apêndice A – Implementação do Algoritmo de Médias Móveis em uma S-Function**

<b>A.1 - Introdução</b> .....	A - 3
<b>A.2 – O Algoritmo de Médias Móveis</b> .....	A - 3
<b>A.3 – Conversão do Algoritmo para S-Function</b> .....	A - 10
<b>A.4 – Exemplo de utilização da S-Function <i>media_movel</i></b> .....	A - 13
<b>A.5 - Referências</b> .....	A - 16

## **Apêndice B – PLL – Phase Locked Loop**

<b>B.1 - Introdução</b> .....	B - 3
<b>B.2 - PLL Trifásica</b> .....	B - 4
<b>B.2.1 - Cálculo dos ganhos do Controlador PI</b> .....	B - 6
<b>B.3 - Simulação Computacional</b> .....	B - 7
<b>B.4 - Referências</b> .....	B - 10

## **Apêndice C – Software de Controle**..... C - 1

## **Apêndice D – Placas Confeccionadas**

**Apêndice D\_1 – Placa Gerador de Portadora Triangular**

**Apêndice D\_2 – Placa Oscilador a Cristal**

**Apêndice D\_3 – Placa Comparador Digital**

**Apêndice D\_4 – Placa Interface PWM X VSI**

**Apêndice D\_5 – Placa Interface CPU X Comando**

**Apêndice D\_6 – Placa Barramento**



# ***Capítulo 1***

## ***Introdução***

<b>1.1 – Considerações Iniciais.....</b>	<b>1 - 3</b>
<b>1.2 - Referências.....</b>	<b>1 - 5</b>





## 1.1 – Considerações Iniciais

A qualidade da energia elétrica tem se tornado uma grande preocupação nas últimas décadas. Isto é devido à popularização e à utilização cada vez maior de equipamentos eletrônicos, tanto pelos pequenos consumidores residenciais, quanto pelos grandes consumidores industriais.

O problema se torna ainda maior quando se considera que ao mesmo tempo em que estes equipamentos eletrônicos são a principal causa de poluição do sistema, eles também são os elementos mais sensíveis a esta poluição (que eles próprios criam).

Os semicondutores utilizados nos equipamentos eletrônicos, na maior parte dos casos, trabalham ou em sua região de corte (ausência de corrente) ou em sua região de saturação (presença de corrente); portanto a característica Tensão x Corrente destes dispositivos é não-linear. Esta não-linearidade causa distorções harmônicas nas correntes drenadas por estes equipamentos.

Mesmo que a concessionária esteja fornecendo uma tensão perfeitamente senoidal, este tipo de carga exige uma corrente não senoidal para que possa trabalhar corretamente. Esta corrente distorcida, ao circular pelas impedâncias do sistema, acaba por distorcer a tensão de fornecimento (dependendo da robustez do alimentador).

Como esta carga não-linear, normalmente, está ligada em um barramento que alimenta outras cargas (lineares ou não), estas outras cargas irão receber uma tensão distorcida. Dessa forma, mesmo as cargas lineares (que não poluem) irão receber uma tensão poluída.

Deve-se ressaltar que os equipamentos eletrônicos, embora sejam a principal causa de distorção harmônica, não são a única.

A distorção na tensão já vem desde a Geração. Pois as máquinas síncronas, que convertem a energia mecânica das turbinas em energia elétrica, possuem imperfeições construtivas; sendo que a tensão gerada é um sinal senoidal aproximado.

A tensão também sofre distorções na Transmissão. Pois existe um grande número de elementos não-lineares neste percurso, tais como os transformadores, cujo núcleo ferromagnético sofre saturação.

Entretanto a maior causa dos problemas de distorção harmônica são os dispositivos eletrônicos.

Outro problema da (má) Qualidade de Energia são os Afundamentos de Tensão (ou Voltage Sags).

Os Voltage Sags são variações de curta duração (entre 0,5 ciclos de onda na frequência fundamental e 1 minuto) no valor eficaz da tensão que alimenta uma carga. Embora estes eventos sejam extremamente rápidos, o tempo gasto para restabelecer o funcionamento de certos processos industriais pode se estender de várias horas a até mesmo alguns dias.

Isto tem causado grandes prejuízos às indústrias no mundo todo.

Segundo um estudo patrocinado pelo EPRI (Electric Power Research Institute), somente nos Estados Unidos da América, os prejuízos causados por Voltage Sags são computados em bilhões de dólares, [1.1] e [1.2].

De acordo com o mesmo estudo, realizado em um período de 27 meses, entre 1993 e 1995, 92% dos eventos de Voltage Sag causam uma tensão remanescente de 40 a 50% da tensão nominal nos pontos estudados, sendo que a maioria destes eventos dura menos do que 2 segundos.

Os Voltage Sags possuem, na maioria dos casos, duas causas: entrada de grandes cargas no sistema e faltas nos sistemas de transmissão ou de distribuição.

A primeira causa não apresenta tanta importância quanto a segunda. Pois, normalmente a entrada em operação destas grandes cargas ocorre com aviso prévio. Porém estas altas correntes de partida, ao circularem pelas impedâncias do sistema, podem causar uma grande queda de tensão em redes com baixo nível de curto-circuito.

Já a segunda causa apresenta ocorrência imprevisível, e está relacionada a curto circuito em linhas aéreas e em equipamentos.

---

---

O **Capítulo 2** apresenta uma classificação dos Voltage Sags quanto à forma como estes são vistos pelas cargas.

O **Capítulo 3** apresenta os métodos para correção de Distorções Harmônicas e correção de Voltage Sags. Uma comparação é realizada entre as técnicas convencionais de compensação com uma nova técnica desenvolvida especialmente para o cálculo correto das compensações em ambientes ruidosos.

O **Capítulo 4** apresenta o equipamento desenvolvido para a compensação de tensões em sistemas de distribuição, em 13.8kV. Descreve-se o funcionamento das partes constituintes do equipamento, tais como o sistema de condicionamento e aquisição de dados, modulação PWM, inversor de frequência e os cuidados necessários para o cálculo do filtro passivo do inversor.

O **Capítulo 5** apresenta o ambiente de testes desenvolvido para o equipamento e também apresenta os resultados obtidos da compensação em 13.8kV.

O **Apêndice A** apresenta uma técnica de médias móveis para aumentar a eficiência computacional dos algoritmos de compensação. Também mostra como implementar esta técnica em uma S-Function do Simulink / Matlab.

O **Apêndice B** apresenta uma técnica de PLL (Phase Locked Loop), utilizada nos algoritmos de compensação para sincronismo.

## 1.2 - Referências

[1.1] Electrotek Concepts, Inc., “An Assessment of Distribution System Power Quality”, Electric Power Research Institute, Palo Alto, CA, Statistical Summary Rep., Final Rep. EPRI TR-106294-V2, vol.2, May 1996.

[1.2] W.E.Brumsickle, R.S.Schneider, G.A.Luckjiff, D.M.Divan, M.F.McGranagham, “Dynamic Sag Correctors: Cost-Effective Industrial Power Line Conditioning”, in *IEEE Transactions on Industry Applications*, vol.37, no.1, January/February,2001, pp.212-216.



# ***Capítulo 2***

## ***Caracterização de Voltage Sags***

<b>2.1 - Introdução.....</b>	<b>2 - 3</b>
<b>2.2 – Sag Tipo A.....</b>	<b>2 - 4</b>
<b>2.3 – Sag Tipo B.....</b>	<b>2 - 5</b>
<b>2.4 – Sag Tipo C.....</b>	<b>2 - 6</b>
<b>2.5 – Sag Tipo D.....</b>	<b>2 - 7</b>
<b>2.6 – Sag Tipo E.....</b>	<b>2 - 8</b>
<b>2.7 – Sag Tipo F.....</b>	<b>2 - 9</b>
<b>2.8 – Sag Tipo G.....</b>	<b>2 - 10</b>
<b>2.9 - Referências.....</b>	<b>2 - 11</b>



## 2.1 - Introdução

A norma **IEEE Std.1159-1995** classifica os eventos de variação de tensão de acordo com o quadro da **Figura 2.1**. Esta figura é cópia, apenas traduzida para o Português, da *Figura 1.18* da referência [2.1].

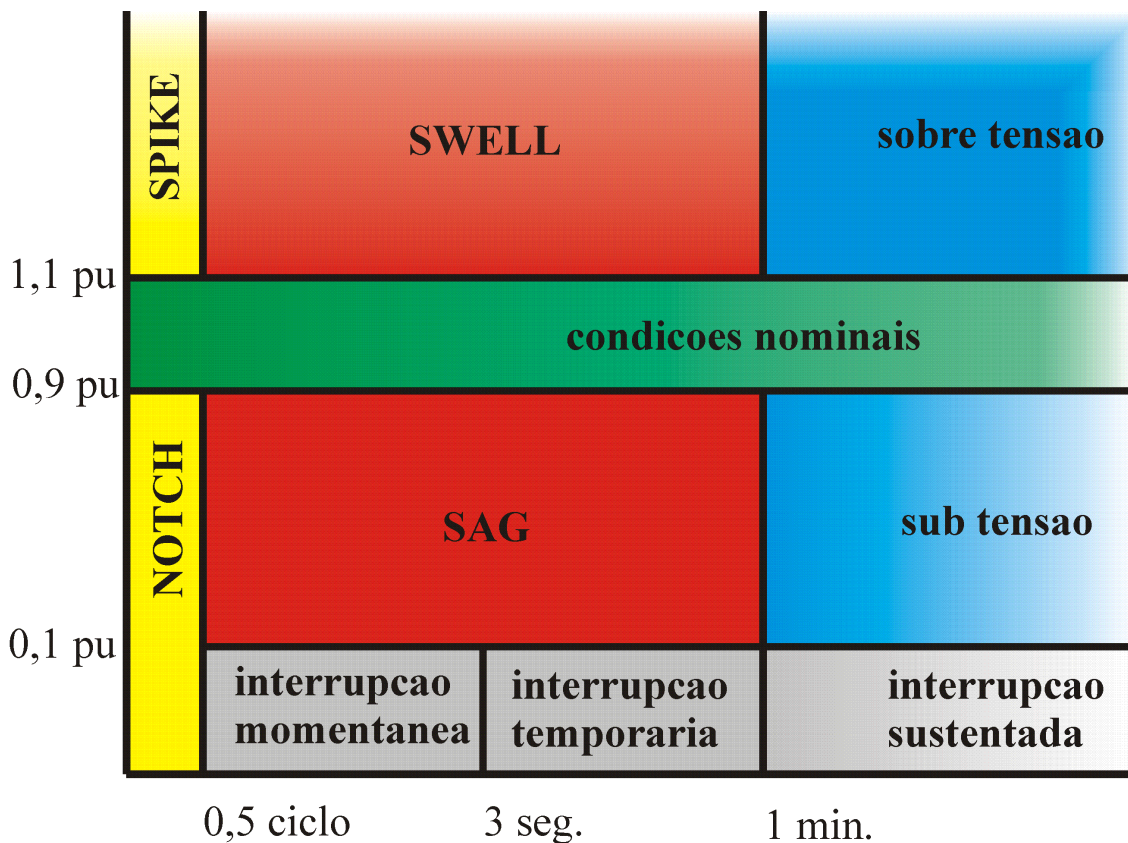


Figura 2.1 – Definições de eventos de variação de tensão segundo a norma **IEEE Std.1159-1995**.

Os valores indicados na **Figura 2.1** se referem à tensão residual no ponto de análise.

Este trabalho é focado, principalmente, na correção de Voltage Sags. Embora o equipamento desenvolvido também seja capaz de corrigir Voltage Swells, Sobre-Tensões e Sub-Tensões.

Os Voltage Sags são causados, principalmente, por faltas nos sistemas de transmissão e de distribuição. Porém a forma como a carga irá ver os distúrbios depende de diversos fatores, tais como o tipo, a localização e a impedância da falta; o tipo de conexão de transformadores existentes entre a carga e o local da falta; atuação do sistema de proteção para isolar a falta; etc.

A referência [2.1] mostra uma análise em que qualquer tipo de Voltage Sag pode ser classificado em um entre sete tipos. As próximas seções deste Capítulo resumem esta análise, identificando as equações básicas para cada um deles, bem como os diagramas fasoriais e formas de onda em relação ao tempo.

## 2.2 – Sag Tipo A

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{2}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{2}V\sqrt{3}\right)$$

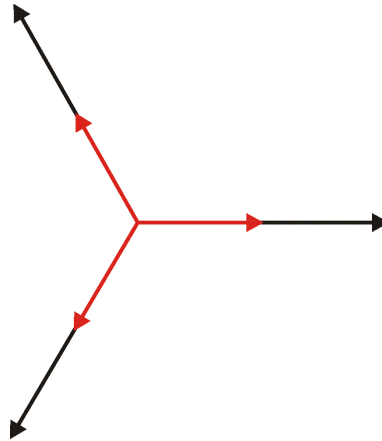


Figura 2.2 – Diagrama Fasorial – Sag Tipo A

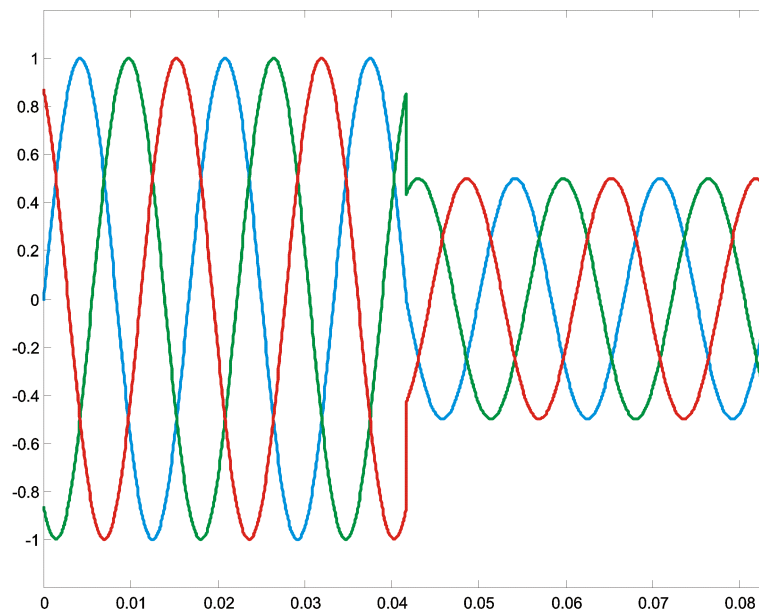


Figura 2.3 – Formas de Onda – Sag Tipo A



### 2.3 – Sag Tipo B

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}\right) - j\left(\frac{1}{2}\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}\right) + j\left(\frac{1}{2}\sqrt{3}\right)$$

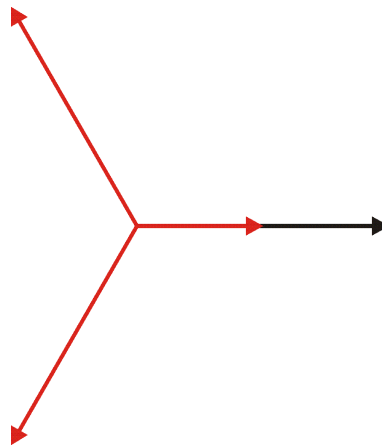


Figura 2.4 – Diagrama Fasorial – Sag Tipo B

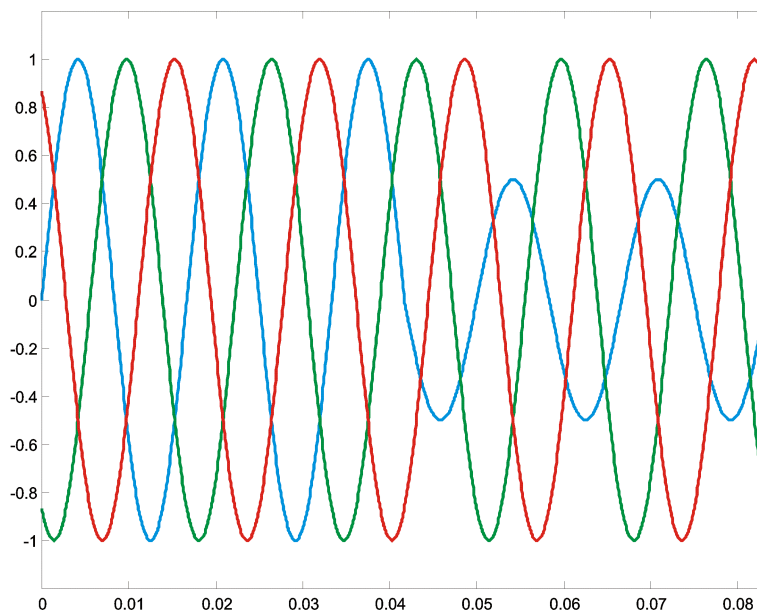


Figura 2.5 – Formas de Onda – Sag Tipo B

## 2.4 – Sag Tipo C

$$v_a = (1) + j(0)$$

$$v_b = \left(-\frac{1}{2}\right) - j\left(\frac{1}{2}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}\right) + j\left(\frac{1}{2}V\sqrt{3}\right)$$

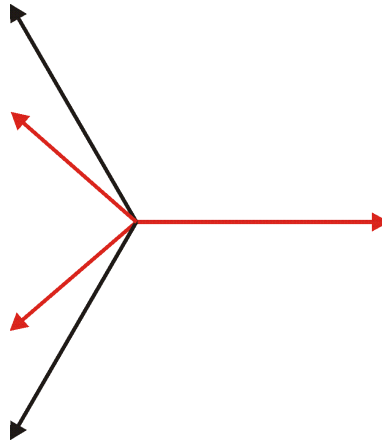


Figura 2.6 – Diagrama Fasorial – Sag Tipo C

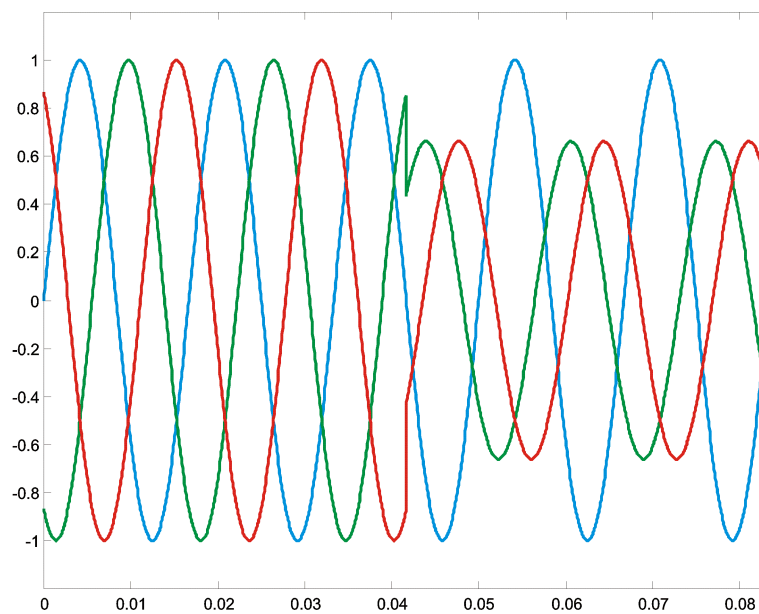


Figura 2.7 – Formas de Onda – Sag Tipo C

## 2.5 – Sag Tipo D

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{2}\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{2}\sqrt{3}\right)$$

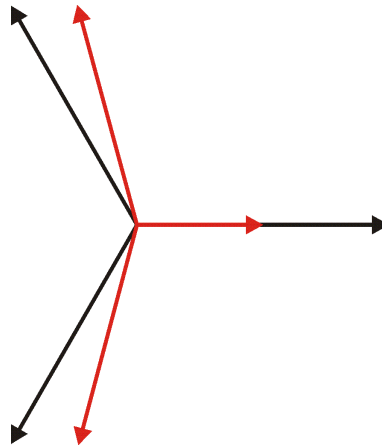


Figura 2.8 – Diagrama Fasorial – Sag Tipo D

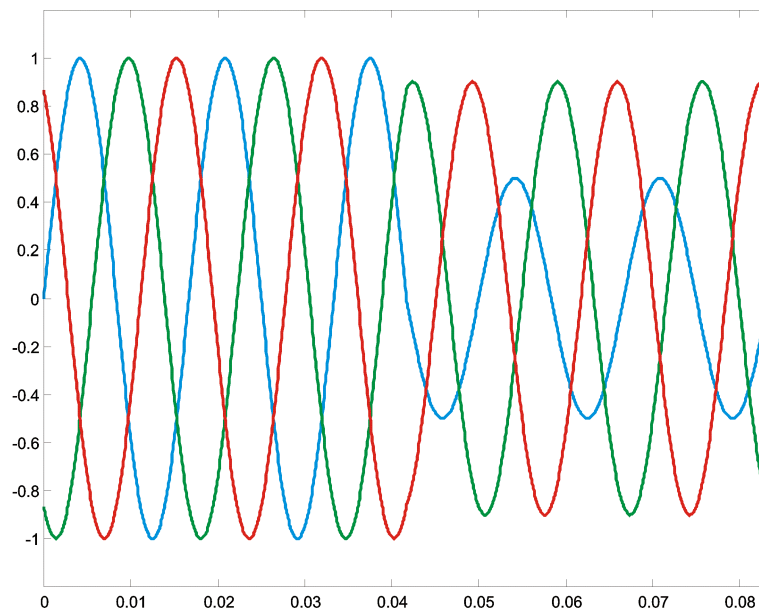


Figura 2.9 – Formas de Onda – Sag Tipo D

## 2.6 – Sag Tipo E

$$v_a = (1) + j(0)$$

$$v_b = \left(-\frac{1}{2} V\right) - j\left(\frac{1}{2} V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2} V\right) + j\left(\frac{1}{2} V\sqrt{3}\right)$$

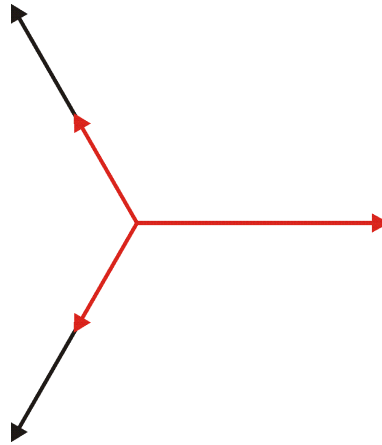


Figura 2.10 – Diagrama Fasorial – Sag Tipo E

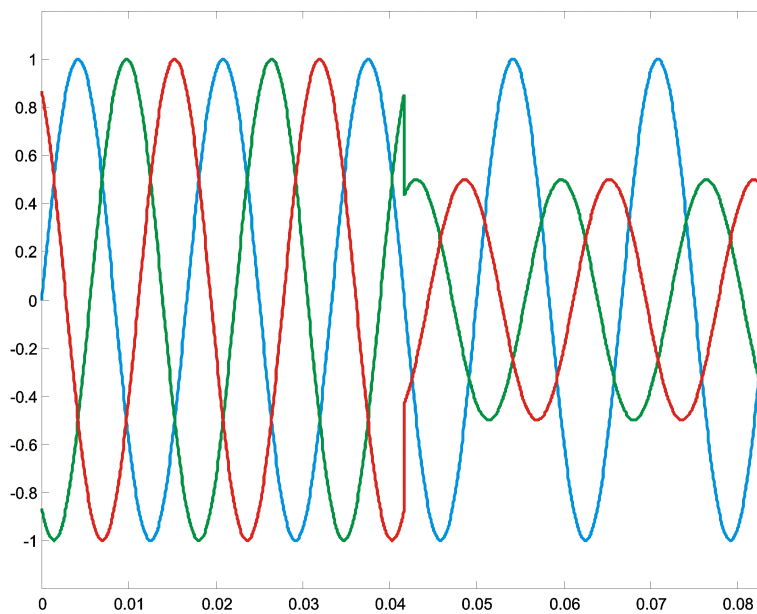


Figura 2.11 – Formas de Onda – Sag Tipo E

## 2.7 – Sag Tipo F

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{3}\sqrt{3} + \frac{1}{6}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{3}\sqrt{3} + \frac{1}{6}V\sqrt{3}\right)$$

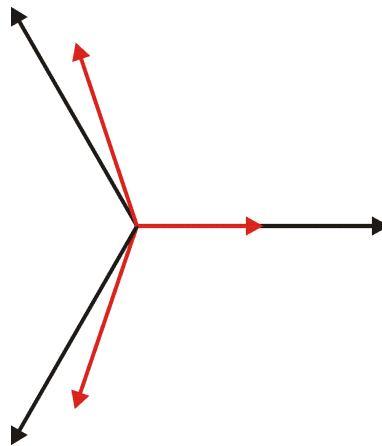


Figura 2.12 – Diagrama Fasorial – Sag Tipo F

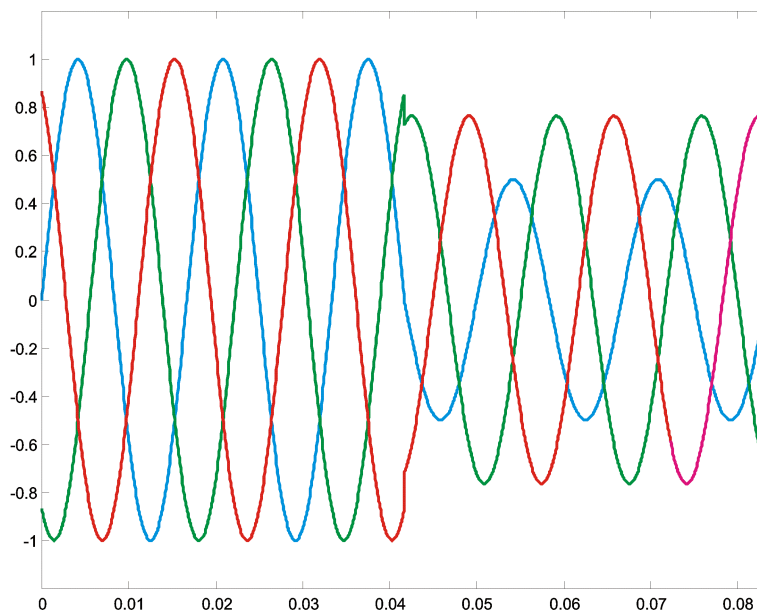


Figura 2.13 – Formas de Onda – Sag Tipo F

## 2.8 – Sag Tipo G

$$v_a = \left( \frac{2}{3} + \frac{1}{3} V \right) + j(0)$$

$$v_b = \left( -\frac{1}{3} - \frac{1}{6} V \right) - j \left( \frac{1}{2} V \sqrt{3} \right)$$

$$v_c = \left( -\frac{1}{3} - \frac{1}{6} V \right) + j \left( \frac{1}{2} V \sqrt{3} \right)$$

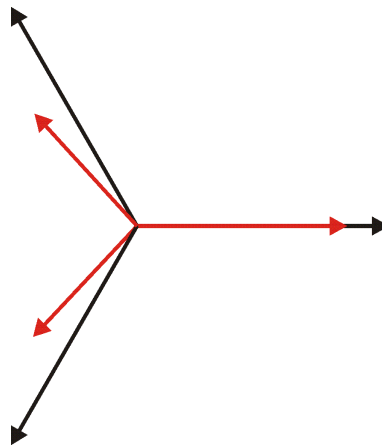


Figura 2.14 – Diagrama Fasorial – Sag Tipo G

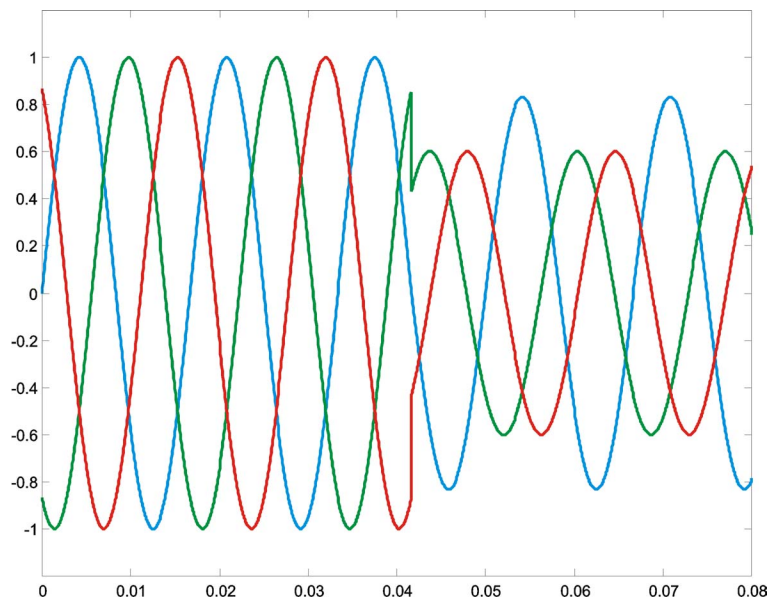


Figura 2.15 – Formas de Onda – Sag Tipo G

## **2.9 - Referências**

[2.1] M.H.J.Bollen, “Understanding Power Quality Problems: Voltage Sags and Interruptions”, IEEE Press, New York, 2000.





# Capítulo 3

## Compensadores Ativos

<b>3.1 - Introdução</b> .....	3 - 3
<b>3.2 – Algoritmos de Controle para Compensadores Ativos</b> .....	3 - 6
<b>3.2.1 – Transformação de Coordenadas</b> .....	3 - 10
<b>3.2.2 – Método id-iq</b> .....	3 - 19
<b>3.2.3 – Método id-iq Modificado para Imunidade Contra Ruídos</b> .....	3 - 23
<b>3.2.4 – Comparação entre os Métodos Convencional e Modificado</b> .....	3 - 27
<b>3.2.5 – Compensação em Malha Fechada</b> .....	3 - 29
3.2.5.1 – Ajuste dos Ganhos do Regulador PI.....	3 - 30
3.2.5.2 – Compensação de Harmônicos em Malha Fechada .....	3 - 33
3.2.5.3 – Compensação de Voltage Sags em Malha Fechada.....	3 - 36
<b>3.3 - Referências</b> .....	3 - 50



### 3.1 - Introdução

Para um melhor entendimento do que são os Compensadores Ativos pode-se fazer uma analogia entre o sistema elétrico e o sistema de abastecimento de água.

Hoje em dia a água dos rios e represas não se encontra em condições próprias para o consumo. Caso alguma pessoa beba diretamente dessa água, muito provavelmente, ficará doente.

Para que essa água possa ser consumida, é necessário que ela passe por um tratamento. Ou seja, adicionam-se a ela certos produtos que irão realizar a limpeza da água, tornando-a própria para o consumo.

O mesmo ocorre com o sistema elétrico, devido à grande quantidade de cargas não lineares o sistema encontra-se “poluído”. E caso essa energia poluída seja entregue a equipamentos sensíveis, esses equipamentos também ficarão “doentes”. A “doença”, neste caso, será perda de produção, perda de vida útil, e, em alguns casos, a perda do equipamento.

Para que estes equipamentos sensíveis possam operar satisfatoriamente em um sistema poluído, é necessário que a energia que os alimenta também passe por um tratamento. Este tratamento pode ser entendido como uma espécie de filtragem.

Até pouco tempo atrás essa filtragem era realizada utilizando filtros passivos. Dependendo da topologia utilizada esses filtros podem apresentar baixa impedância para as componentes de alta frequência, os filtros Passa-Altas. Estes filtros devem ser ligados em paralelo com a carga, de forma que as altas frequências sejam atraídas para o ponto Comum. Os filtros também podem apresentar alta impedância para as componentes de alta frequência, estes tipos de filtros são conhecidos como Passa-Baixas, e devem ser ligados em série com a carga, de forma que as componentes de alta frequência fiquem “armazenadas” no filtro sob forma de queda de tensão. Essa duas configurações podem ser vistas na **Figura 3.1**.

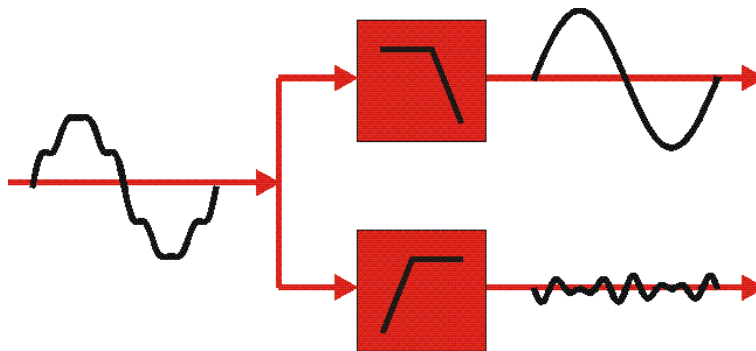


Figura 3.1 – Filtros Passivos

Porém, os filtros passivos, por serem circuitos sintonizados, precisam ser dimensionados apenas para uma determinada aplicação. Caso o “ambiente” em que ele está inserido se modifique, o filtro já não consegue um bom desempenho; e em alguns casos pode até tornar a situação pior, entrando em ressonância com o sistema e causando a queima de alguns equipamentos.

Para contornar estes problemas, surgiram os Compensadores (Filtros) Ativos. Os Compensadores Ativos são equipamentos “inteligentes”, que possuem um computador digital como elemento central. A todo instante este computador está calculando o tipo de compensação necessária para eliminar as componentes indesejadas presentes no sistema naquele momento. Dessa forma, ele se adapta perfeitamente à qualquer variação nos parâmetros do sistema em que ele está inserido. A **Figura 3.2** ilustra este processo.

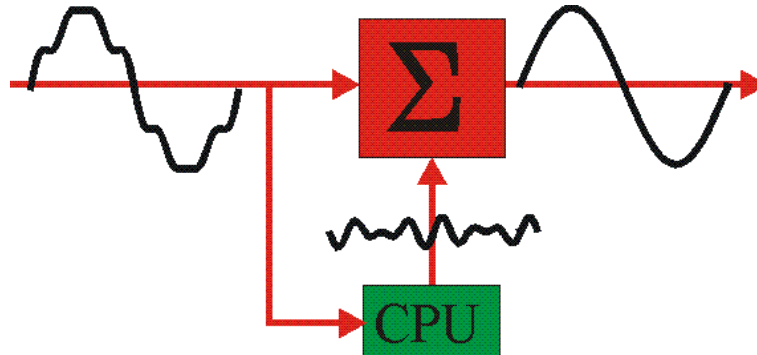


Figura 3.2 – Compensadores Ativos

**Na literatura existente sobre o assunto, alguns autores preferem o termo Filtros Ativos, enquanto outros preferem Compensadores Ativos. Neste texto será usada a denominação Compensador Ativo, pois o termo é mais apropriado devido à forma como a “poluição” é eliminada.**

**Pelo conceito de Filtro subentende-se um equipamento que absorva as componentes indesejadas. Enquanto que pelo conceito de Compensador subentende-se um equipamento que gere uma compensação para as componentes indesejadas, eliminando-as.**

Nota-se, na **Figura 3.2**, que o princípio de funcionamento dos Compensadores Ativos é uma soma do sinal de entrada (que contém a “parte boa” da energia e a parte indesejada) com o sinal de compensação (que contém a parte indesejada, porém defasada de 180°), como resultado desta soma tem-se somente a “parte boa” da energia, ou seja, a parte que efetivamente realiza trabalho.

Foram discutidos no **Capítulo 2** os problemas de qualidade de energia. Tanto os problemas de corrente quanto os problemas de tensão podem ser corrigidos com a utilização dos Compensadores Ativos, basta a utilização da topologia correta.

Quando se deseja corrigir problemas relacionados à corrente, deve-se realizar uma soma de correntes. Deve-se, portanto instalar o compensador em paralelo com a carga, pois esta é única forma de se somar duas correntes. → **Lei dos Nós (de Kirchoff)**

A **Figura 3.3** ilustra esta situação. Tem-se como carga um circuito não linear. Este circuito drena da concessionária de energia uma corrente com formato não senoidal. Esta corrente distorcida prejudica as outras cargas (lineares ou não) ligadas ao mesmo barramento, e por consequência a própria concessionária, que tem sua imagem ligada à qualidade do produto que oferece. Deve-se, portanto instalar um Compensador Ativo Paralelo (ou Shunt) para gerar uma corrente que contenha apenas a parcela da

distorção, que a carga necessita para funcionar. Dessa forma, a concessionária precisará fornecer apenas a corrente Senoidal.

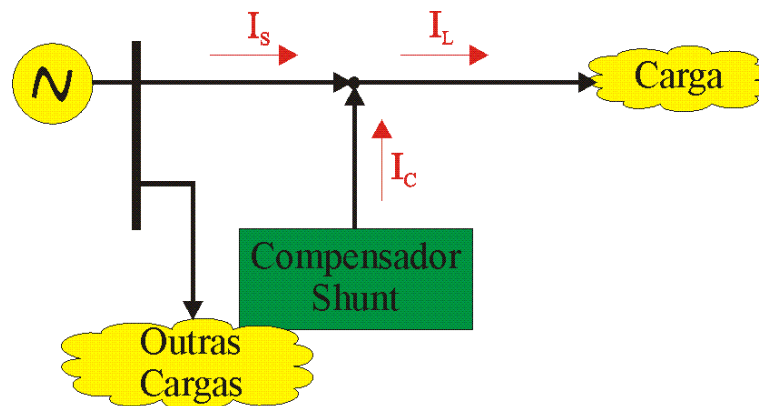


Figura 3.3 – Compensador Ativo Paralelo

Quando se deseja corrigir problemas relacionados à tensão, deve-se realizar uma soma de tensões. Deve-se, portanto instalar o compensador em série com a carga, pois esta é única forma de se somar duas tensões. → *Lei das Malhas (de Kirchoff)*

A **Figura 3.4** ilustra esta situação. A concessionária de energia está com uma tensão poluída com harmônicos e também com alto índice de afundamentos de tensão (*Voltage Sags*). A carga é extremamente sensível a estes problemas. Sendo assim, deve-se instalar um Compensador Ativo Série para gerar os Anti-Harmônicos, que tornam a tensão na carga senoidal e também para gerar a compensação para os *Sags*.

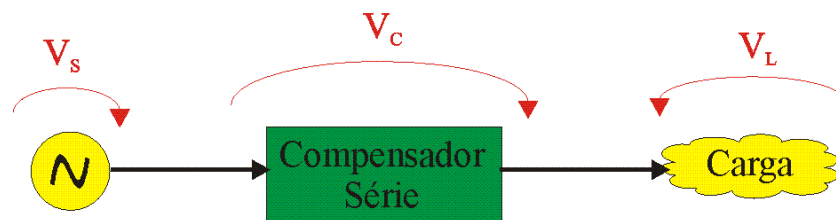


Figura 3.4 – Compensador Ativo Série

**Nota-se que um Compensador Série é o dual de um Compensador Paralelo.**

**O Compensador Série protege a carga contra os problemas de tensão, que são gerados pela concessionária. Portanto é instalado em Série.**

**O Compensador Paralelo protege a concessionária contra os problemas de corrente, que são gerados pela carga. Portanto é instalado em Paralelo.**

### 3.2 – Algoritmos de Controle para Compensadores Ativos

Partindo da idéia de que um Compensador Ativo é um equipamento que deve gerar a compensação para um determinado problema, segue-se o raciocínio seguinte (que pode ser aplicado tanto para filtros série quanto para filtros shunt):

Tem-se na entrada de aquisição de dados um sinal composto pela fundamental ( $X_{Fund}$ ) e pelos harmônicos ( $X_{Harm}$ ). Deseja-se que, após a compensação, a grandeza  $X$  (que pode ser Corrente ou Tensão) contenha apenas a componente fundamental. Portanto, deve-se somar uma compensação ( $X_{Comp}$ ) ao sinal de entrada para compensá-la. Conforme equação (Eq. 3.1).

$$(X_{Fund} + X_{Harm}) + X_{Comp} = X_{Fund} \quad (\text{Eq. 3.1})$$

Sendo assim, percebe-se que o compensador precisará gerar  $X_{Comp} = -X_{Harm}$ .

Deve-se, portanto encontrar alguma forma para gerar ( $-X_{Harm}$ ).

Uma primeira tentativa é realizar uma filtragem do tipo passa-baixas (de forma digital) no sinal de entrada. Dessa forma, ter-se-ia o sinal  $X_{Fund}$ . E o sinal de compensação seria dado pela equação (Eq. 3.2).

$$X_{Comp} = X_{Fund} - X_{Entrada} \quad (\text{Eq. 3.2})$$

Esta técnica está ilustrada na Figura 3.5.

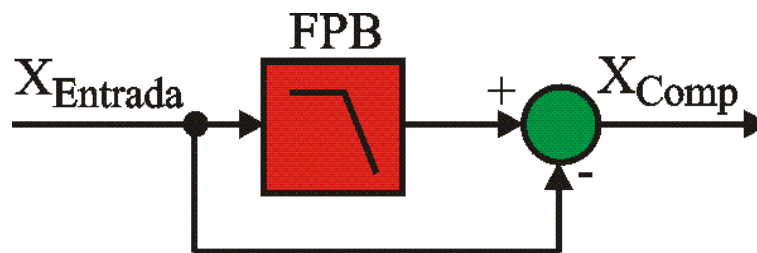


Figura 3.5 – Técnica Simples para geração do Sinal de Compensação

Supondo que o filtro passa-baixas utilizado fosse um filtro ideal, ter-se-ia o sinal de compensação ilustrado na Figura 3.6 (parte central da figura), assim como o sinal já compensado (parte inferior) e o sinal de entrada (parte superior). Utilizou-se como entrada um sinal com 20% de 3º harmônico somado à fundamental.

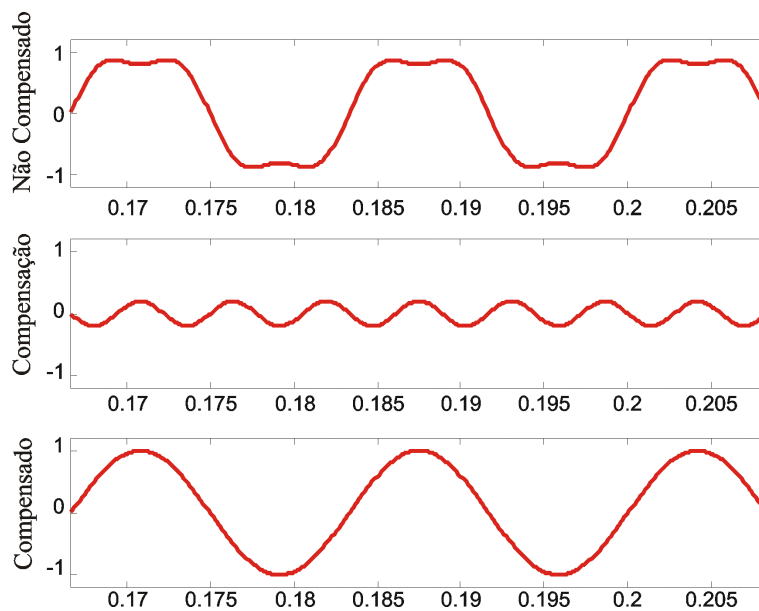


Figura 3.6 – Simulação para Filtro Passa-Baixas ideal

Nota-se que o Compensador cumpriu corretamente sua função. Porém existe um problema nesta análise. Para o algoritmo da Figura 3.5 funcionar corretamente é necessário que o filtro passa-baixas seja um filtro ideal. Porém isto não é possível. Os filtros, necessariamente, seguem uma lei que rege seu comportamento dinâmico. Este comportamento dinâmico faz com que o sinal filtrado apresente uma defasagem com relação ao sinal não filtrado. Mais detalhes sobre os filtros passa-baixas serão discutidos no **Capítulo 4, seção 4.4**.

Caso seja utilizado um filtro do tipo Butterworth de 2ª ordem, com frequência de corte em 60Hz, o resultado seria semelhante ao que pode ser visto na **Figura 3.7**.

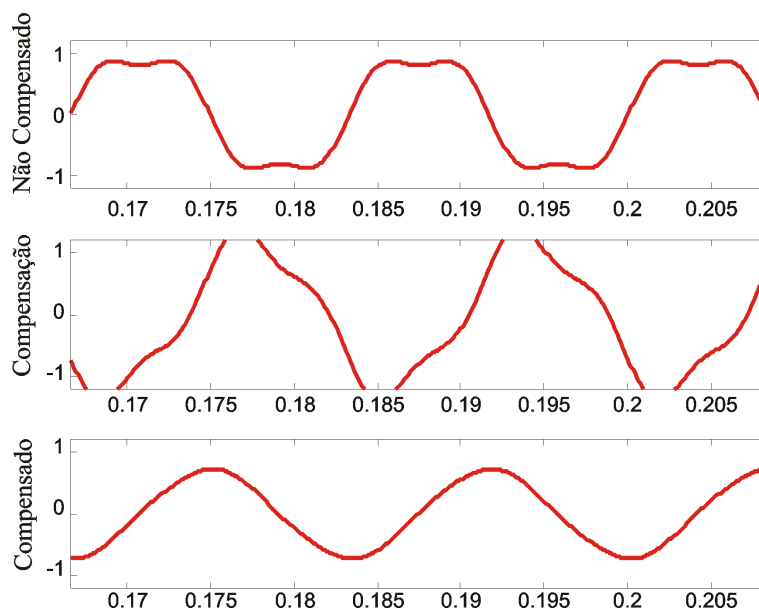


Figura 3.7 – Simulação para Filtro Passa Baixas Real, frequência de corte em 60Hz

Da **Figura 3.7**, nota-se que o Compensador conseguiu eliminar o 3º harmônico da entrada, entretanto o sinal de compensação não foi calculado corretamente, conforme pode-se comparar com a **Figura 3.6**. Devido à defasagem, causada pelo filtro, o Compensador também corrigiu a componente na frequência Fundamental e deixou o sinal Compensado igual ao sinal que ele vê na saída do filtro (defasado, atenuado e ligeiramente distorcido).

Um filtro Butterworth de 2ª ordem defasa a componente com mesma frequência da frequência de corte de 90°. A **Figura 3.8** apresenta uma comparação entre a componente de 60Hz do sinal de entrada do Compensador e o sinal de saída do filtro passa-baixas. Nota-se que o sinal Compensado da **Figura 3.7** é exatamente o mesmo sinal da saída do filtro da **Figura 3.8**.

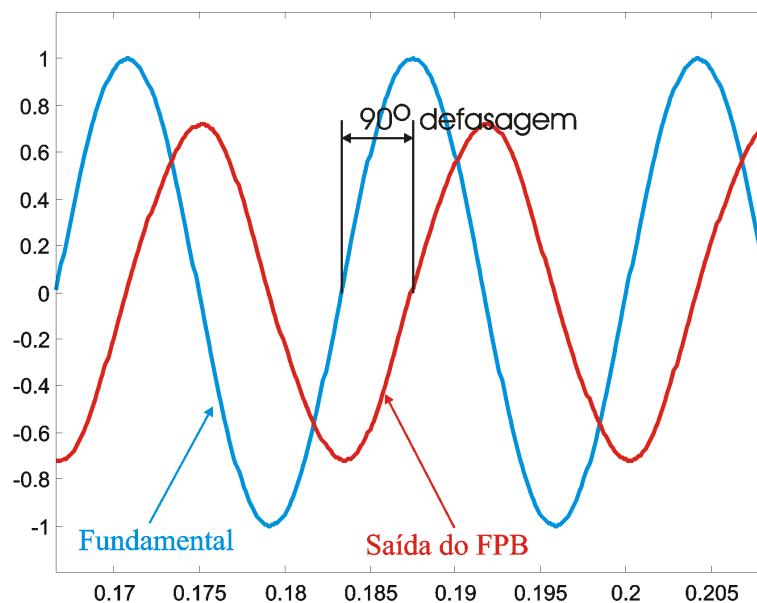


Figura 3.8 – Efeito do Filtro Passa Baixas, frequência de corte em 60Hz

Se a frequência de corte do filtro for deslocada para um valor superior, por exemplo 120Hz, a defasagem da componente de 60Hz será menor. Vale lembrar que se o sinal de entrada possuir uma componente em 120Hz, esta componente seria defasada de 90°.

Mantendo-se o mesmo sinal de entrada utilizado para gerar as **Figuras 3.6, 3.7 e 3.8**, realizou-se uma nova simulação, desta vez substituindo-se o Filtro Passa Baixas por um com frequência de corte em 120Hz. O resultado pode ser observado nas **Figuras 3.9 e 3.10**.



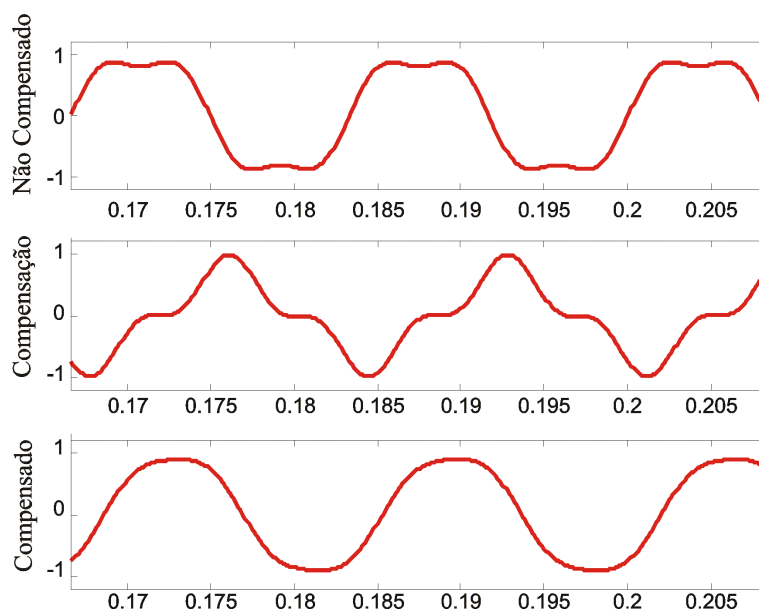


Figura 3.9 - Simulação para Filtro Passa Baixas Real, frequência de corte em 120Hz

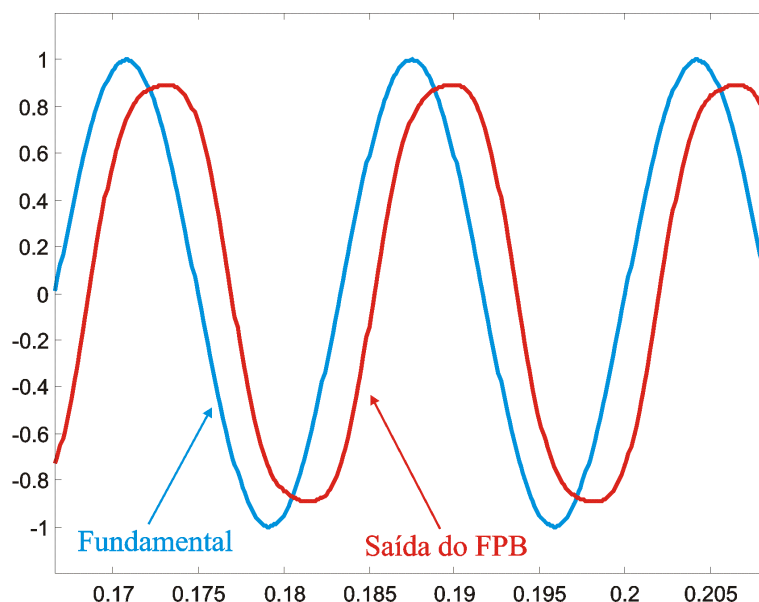


Figura 3.10 - Efeito do Filtro Passa Baixas, frequência de corte em 120Hz

Nota-se, da **Figura 3.10**, que a defasagem diminuiu, porém ela ainda existe, e com valor considerável. Pode-se perceber, na **Figura 3.9** (central), que como a defasagem diminuiu, o Compensador precisa inserir menos compensação de 60Hz do que na **Figura 3.7**. Outro detalhe que vale a pena ser observado é o fato de que tanto o sinal compensado, quanto a saída do filtro ainda apresentam componentes de 3º harmônico. Isto ocorre pois a frequência de corte do filtro (120Hz) ficou muito próxima da frequência do sinal que ele deveria eliminar (180Hz), permitindo a passagem de parte do sinal.

Observa-se, então, que estas tentativas utilizando Filtros Passa Baixas para geração do sinal de compensação não irão resultar em algo muito produtivo. Sendo assim deve-se partir para o estudo de técnicas mais poderosas.

### 3.2.1 – Transformação de Coordenadas

Esta técnica consiste, basicamente, em uma mudança para um outro universo matemático. Um universo onde a componente da frequência fundamental se torna um sinal DC, sendo, portanto facilmente filtrado, sem problemas de defasagem.

Primeiramente o sistema trifásico é convertido para um sistema de 2 vetores ortogonais estacionários. Para tanto é aplicada a transformação de Clarke, cuja matriz de transformação é dada na equação (Eq. 3.3).

$$\begin{bmatrix} v_0 \\ v_\alpha \\ v_\beta \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 1/\sqrt{2} \\ 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \cdot \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad (\text{Eq. 3.3})$$

Sendo  $v_0$  a componente de seqüência zero do sinal, que só existirá em sistemas a 4 fios e desequilibrados (para o caso de correntes) ou sistemas desbalanceados (para o caso de tensões).

Nota-se que a transformação de Clarke isola a componente de seqüência zero, sendo assim esta componente deve ser compensada de forma separada, conforme será mostrado na seção 3.2.3.

Considerando a componente de seqüência positiva de um sistema senoidal puro (apenas a frequência fundamental):

$$v_{a+} = U_+ \cdot \sin(\omega t + \theta_1) \quad (\text{Eq. 3.4})$$

$$v_{b+} = U_+ \cdot \sin(\omega t + \theta_1 - 120^\circ) \quad (\text{Eq. 3.5})$$

$$v_{c+} = U_+ \cdot \sin(\omega t + \theta_1 + 120^\circ) \quad (\text{Eq. 3.6})$$

Substituindo (Eq. 3.4), (Eq. 3.5) e (Eq. 3.6) em (Eq. 3.3), tem-se:

$$v_{\alpha+} = U_+ \cdot \sqrt{\frac{3}{2}} \cdot \sin(\omega t + \theta_1) \quad (\text{Eq. 3.7})$$

$$v_{\beta+} = -U_+ \cdot \sqrt{\frac{3}{2}} \cdot \cos(\omega t + \theta_1) \quad (\text{Eq. 3.8})$$

Para a seqüência negativa tem-se:

$$v_{a-} = U_- \cdot \sin(\omega t + \theta_1) \quad (\text{Eq. 3.9})$$

$$v_{b-} = U_- \cdot \sin(\omega t + \theta_1 + 120^\circ) \quad (\text{Eq. 3.10})$$

$$v_{c-} = U_- \cdot \sin(\omega t + \theta_1 - 120^\circ) \quad (\text{Eq. 3.11})$$

Substituindo (Eq. 3.9), (Eq. 3.10) e (Eq. 3.11) em (Eq. 3.3), tem-se:

$$v\alpha_- = U_- \cdot \sqrt{\frac{3}{2}} \cdot \sin(\omega t + \theta_1) \quad (\text{Eq. 3.12})$$

$$v\beta_- = U_- \cdot \sqrt{\frac{3}{2}} \cdot \cos(\omega t + \theta_1) \quad (\text{Eq. 3.13})$$

A **Figura 3.11** apresenta o diagrama fasorial da transformação, nota-se que os sinais **alfa** estão em fase com o sinal da **fase A**, e que os sinais **beta** estão defasados entre si de 180° e defasados de 90° com relação aos sinais **alfa**.

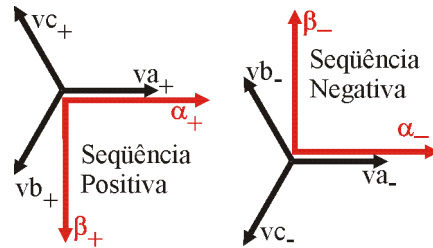


Figura 3.11 – Transformação de Clarke – Representação Fasorial

A **Figura 3.12** representa a mesma informação do diagrama fasorial da **Figura 3.11**, porém no domínio do tempo. Nota-se que os sinais **alfa** e **beta** ainda são sinais senoidais.

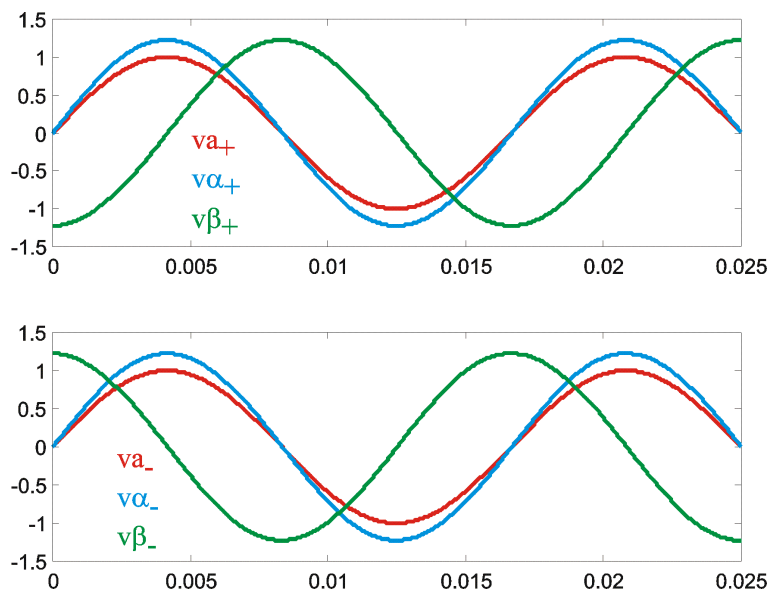


Figura 3.12 - Transformação de Clarke – Representação no Domínio do Tempo

Após a primeira transformação de coordenadas, uma segunda é realizada, desta vez, criam-se 2 vetores ortogonais que giram em sincronismo com a frequência da rede, esta transformação também é conhecida como transformação de Park, cuja matriz de transformação é dada em **(Eq. 3.14)**.

$$\begin{bmatrix} v_d \\ v_q \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} v_\alpha \\ v_\beta \end{bmatrix} \quad (\text{Eq. 3.14})$$

Considerando o sinal  $\cos \theta$ , da equação (Eq. 3.14), como uma onda do tipo  $\cos(\omega t + \theta_2)$  e  $\sin \theta$  como  $\sin(\omega t + \theta_2)$  e substituindo (Eq. 3.7) e (Eq. 3.8) em (Eq. 3.14), tem-se:

$$v_{d+} = U_+ \cdot \sqrt{\frac{3}{2}} \cdot \sin(\theta_1 - \theta_2) \quad (\text{Eq. 3.15})$$

$$v_{q+} = -U_+ \cdot \sqrt{\frac{3}{2}} \cdot \cos(\theta_1 - \theta_2) \quad (\text{Eq. 3.16})$$

Nota-se que tanto (Eq. 3.15) quanto (Eq. 3.16) são sinais DC, cujos valores dependem da amplitude da onda de seqüência positiva e da defasagem entre os ângulos  $\theta_1$  e  $\theta_2$ . Normalmente utiliza-se uma PLL (maiores detalhes sobre a implementação de PLLs podem ser vistos no Apêndice B) para gerar os sinais  $\cos \theta$  e  $\sin \theta$ , dessa forma  $\theta_1 = \theta_2$ . Nesta condição  $v_{d+} = 0$  e  $v_{q+} = -U_+ \cdot \sqrt{\frac{3}{2}}$ . Esta situação se encontra ilustrada na Figura 3.13.

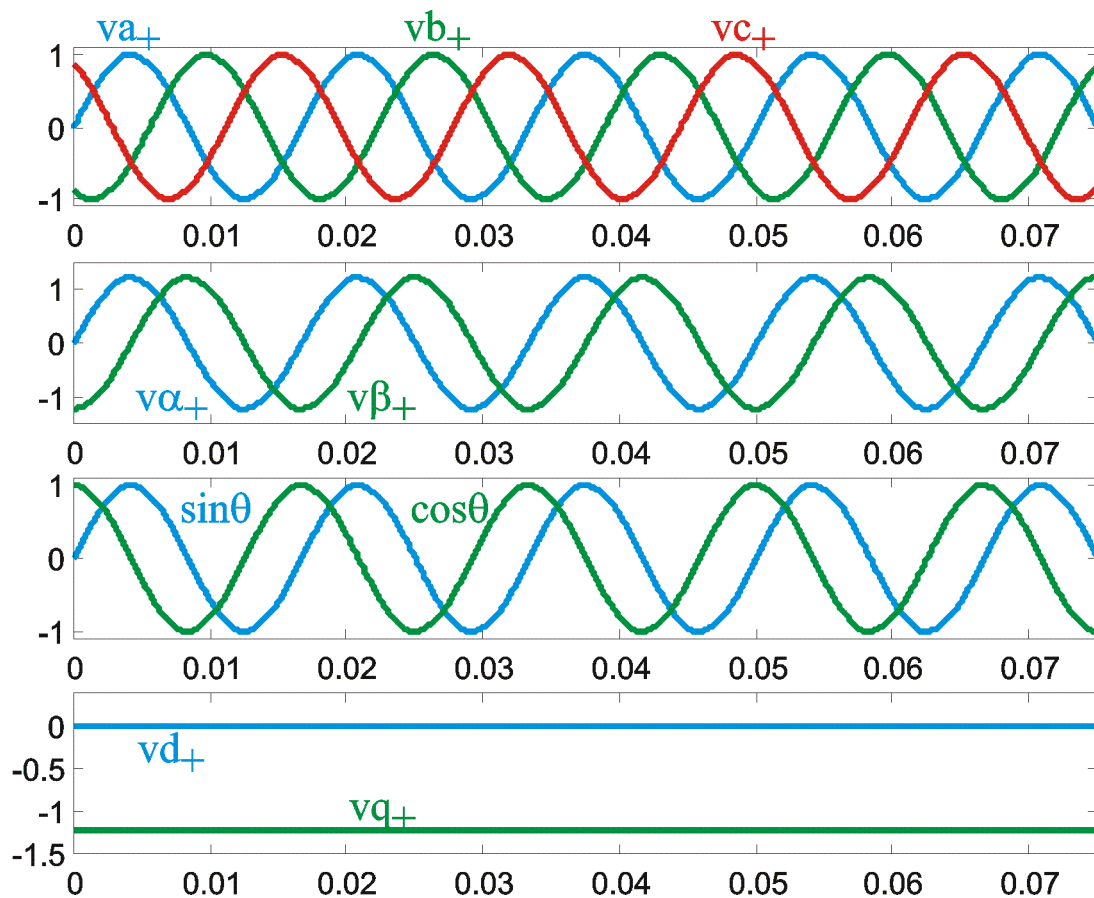


Figura 3.13 – Transformação de Park – Freq. de Entrada igual a Fundamental

Caso a frequência de  $\cos\theta$  e  $\sin\theta$  não seja a mesma dos sinais  $v_a$ ,  $v_b$  e  $v_c$ , a diferença entre  $\theta_1$  e  $\theta_2$  não será uma constante e sim uma função no tempo. Então os sinais  $v_{d+}$  e  $v_{q+}$  serão oscilatórios. Isto está ilustrado na **Figura 3.14**. Tem-se como entrada um sistema trifásico senoidal com frequência angular igual a  $2\pi \cdot 60 \cdot 4$ , e frequência angular de  $\cos\theta$  e  $\sin\theta$  igual a  $2\pi \cdot 60$ .

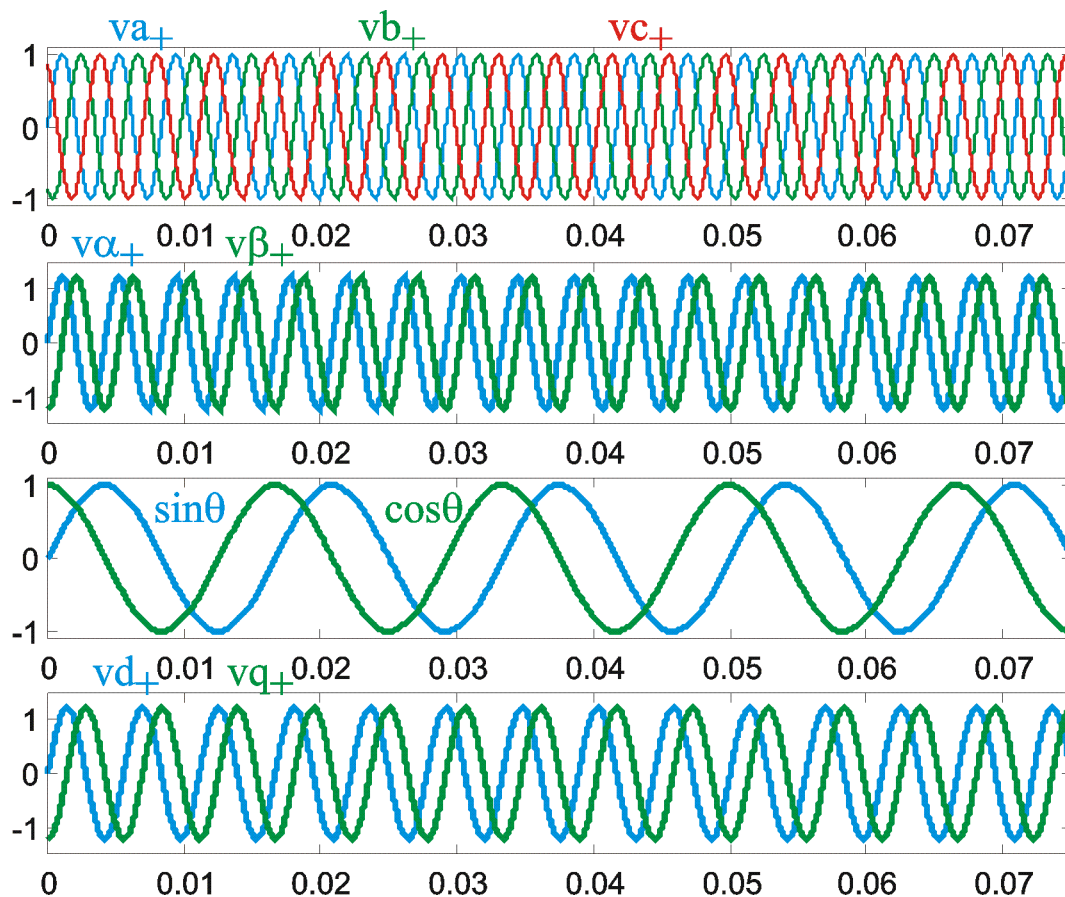


Figura 3.14 – Transformação de Park - Freq. de Entrada múltipla da Fundamental

Nota-se, na **Figura 3.14**, que os sinais  $v_{d+}$  e  $v_{q+}$  são sinais oscilatórios. A frequência desta oscilação, por agora, não interessa. O que interessa é que o valor médio desta oscilação é zero.

Na **Figura 3.13** analisou-se o comportamento dos sinais, quando a frequência de entrada era a mesma da dos sinais  $\cos\theta$  e  $\sin\theta$ . Na **Figura 3.14** analisou-se o comportamento dos sinais, quando a frequência de entrada era diferente da dos sinais  $\cos\theta$  e  $\sin\theta$ , porém sem nenhuma componente na frequência destes sinais.

A **Figura 3.15** apresenta uma situação em que a frequência dos sinais de entrada é uma composição de duas frequências,  $2\pi \cdot 60 \cdot 4$  e  $2\pi \cdot 60$ ; enquanto que a frequência dos sinais  $\cos\theta$  e  $\sin\theta$  continua igual a  $2\pi \cdot 60$ .

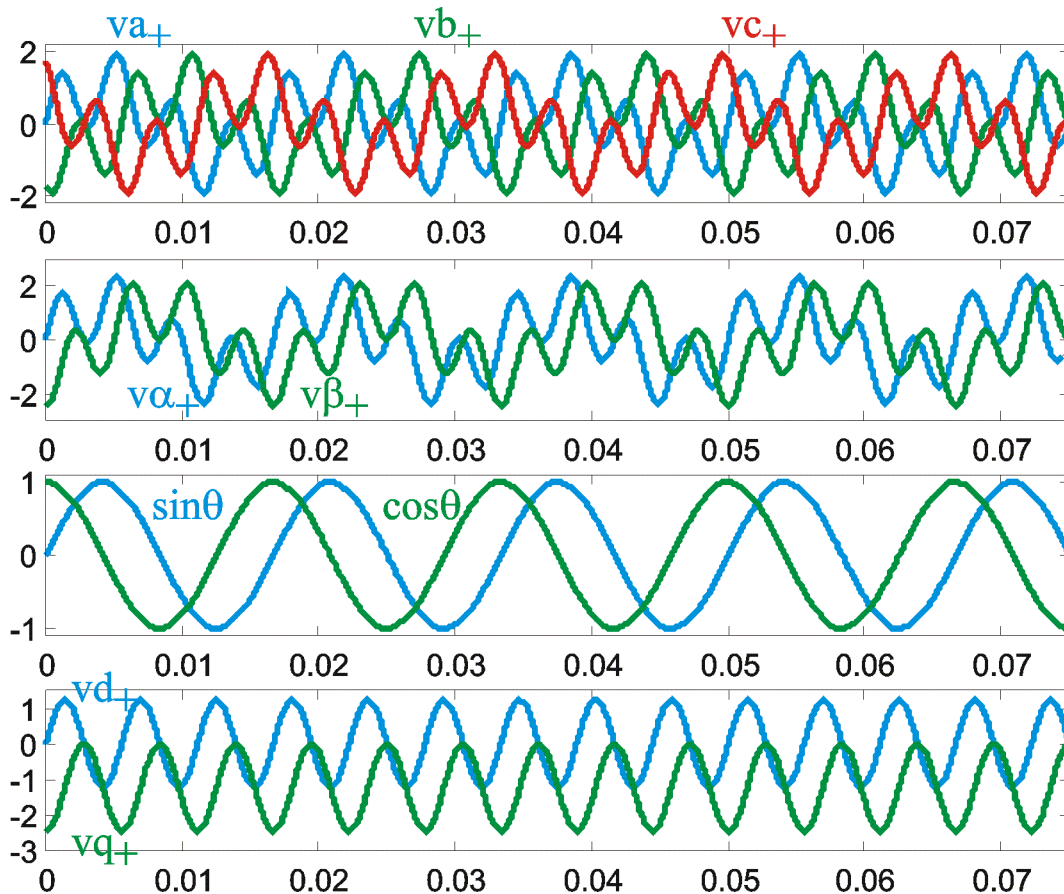


Figura 3.15 – Transformação de Park - Freq. de Entrada Composta

Nota-se, na **Figura 3.15**, que os sinais  $vd_+$  e  $vq_+$  são sinais oscilatórios (contribuição da componente de frequência  $2 \cdot \pi \cdot 60 \cdot 4$ ) sobrepostos a um nível DC (contribuição da componente de frequência  $2 \cdot \pi \cdot 60$ ). Caso uma filtragem fosse aplicada aos sinais  $vd_+$  e  $vq_+$  ter-se-ia o mesmo resultado da **Figura 3.13**. Ou seja uma filtragem nos sinais  $vd_+$  e  $vq_+$  remove todas as componentes harmônicas da entrada do sistema, deixando apenas a frequência fundamental.

Uma filtragem para se retirar o nível DC de um sinal oscilatório é muito mais simples de se realizar do que aquela implementada na **Figura 3.5**. Como o sinal resultante do processo de filtragem é um sinal DC, a defasagem imposta pelo filtro não importa, uma vez que sinais DC não possuem ângulo de fase.

Para as componentes de seqüência negativa, segue-se o mesmo raciocínio; substituindo-se (**Eq. 3.12**) e (**Eq. 3.13**) em (**Eq. 3.14**), tem-se:

$$vd_- = U_- \cdot \sqrt{\frac{3}{2}} \cdot \sin(2\omega t + \theta_1 + \theta_2) \quad (\text{Eq. 3.19})$$

$$vq_- = U_- \cdot \sqrt{\frac{3}{2}} \cdot \cos(2\omega t + \theta_1 + \theta_2) \quad (\text{Eq. 3.20})$$

Nota-se que a componente de seqüência negativa aparece como um sinal oscilatório cuja frequência é o dobro da frequência fundamental. A **Figura 3.16**

apresenta o resultado de uma simulação para um sinal de entrada contendo apenas a frequência fundamental, porém com seqüência de fase negativa.

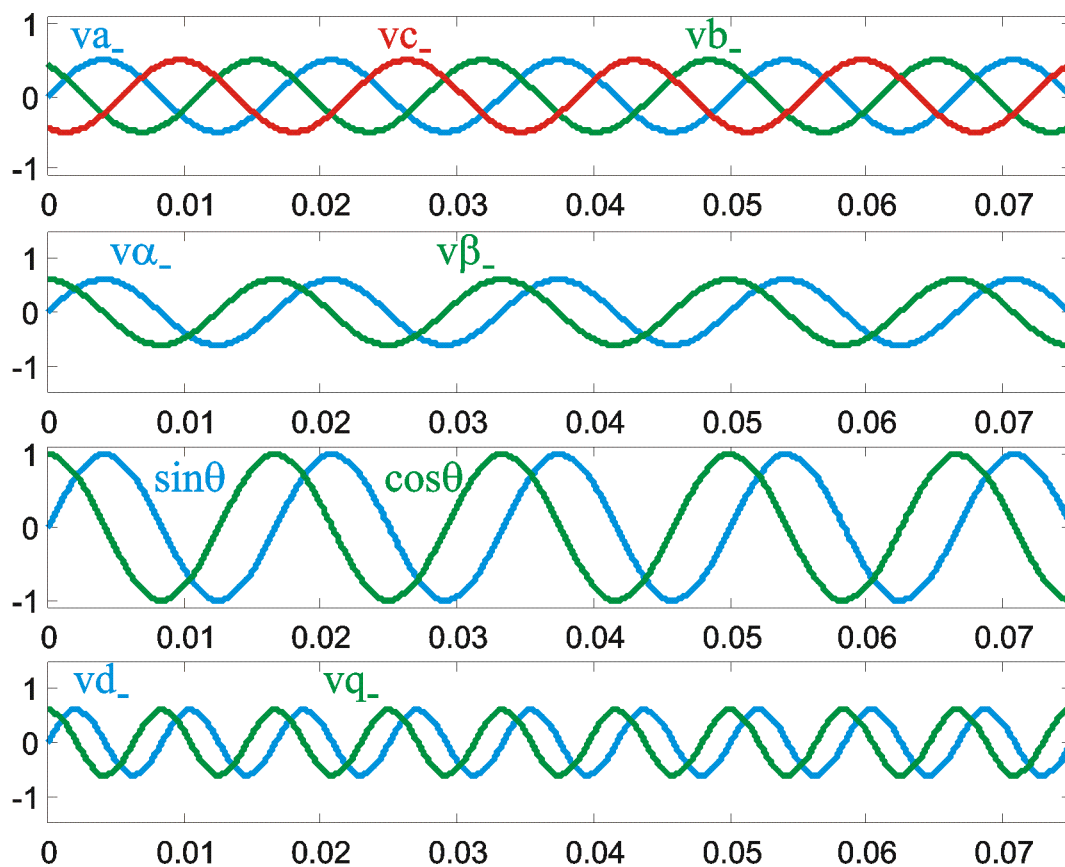


Figura 3.16 – Transformação de Park - Entrada com seq. de fases negativa

A simulação da **Figura 3.16** mostrou o comportamento dos sinais **d** e **q** para uma entrada contendo apenas componentes de seqüência negativa, na frequência fundamental. Enquanto que a simulação da **Figura 3.13** mostrou o comportamento dos sinais **d** e **q** para uma entrada contendo apenas componentes de seqüência positiva, na frequência fundamental. Para a geração da **Figura 3.16** utilizou-se  $U_-$  igual **0.5**. Na **Figura 3.13** utilizou-se  $U_+$  igual **1.0**.

A **Figura 3.17** apresenta uma simulação para um sinal de entrada contendo componentes de seqüência positiva e componentes de seqüência negativa. Utilizou-se  $U_+$  igual **1.0** e  $U_-$  igual **0.5**.

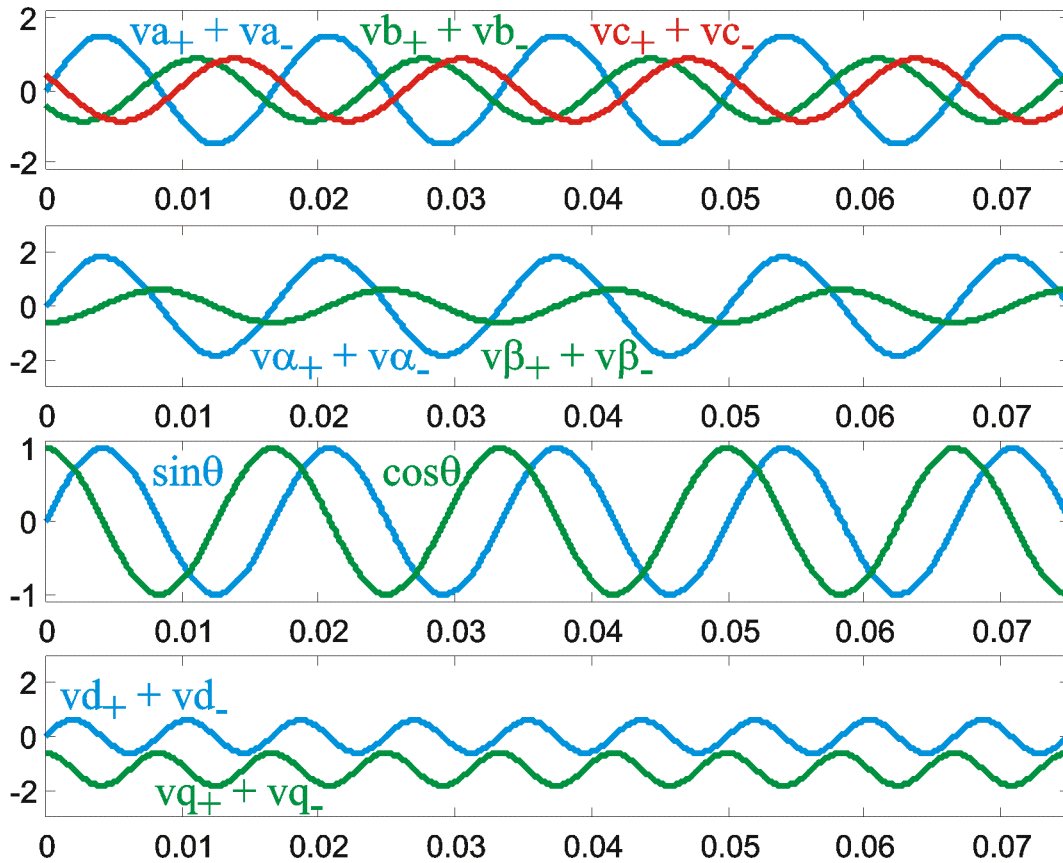


Figura 3.17 – Transformação de Park - Entrada com seq. positiva e negativa

Nota-se, na **Figura 3.17**, que os sinais  $\mathbf{vd}$  e  $\mathbf{vq}$  são sinais oscilatórios (contribuição da componente de seqüência negativa) sobrepostos a um nível DC (contribuição da componente de seqüência positiva). Caso uma filtragem fosse aplicada aos sinais  $\mathbf{vd}$  e  $\mathbf{vq}$  ter-se-ia o mesmo resultado da **Figura 3.13**. Ou seja uma filtragem nos sinais  $\mathbf{vd}$  e  $\mathbf{vq}$  remove todas as componentes de seqüência negativa da entrada do sistema, deixando apenas a componente de seqüência positiva.

Portanto, no domínio da Transformada de Park apenas a componente de seqüência positiva na mesma freqüência de  $\mathbf{cos}\theta$  e  $\mathbf{sen}\theta$  será vista como um nível DC. A componente de seqüência negativa na mesma freqüência de  $\mathbf{cos}\theta$  e  $\mathbf{sen}\theta$  será vista como um sinal oscilatório. E as componentes de outras freqüências (de seqüência positiva ou de seqüência negativa) também serão vistas como sinais oscilatórios.

O processo da transformação de Park não é nada mais do que uma mudança de referencial. É uma outra forma de se olhar para o sistema trifásico. Um observador estático vê o sistema trifásico como três vetores, que rotacionam em torno de um ponto, com uma freqüência angular  $\omega$ . No domínio da transformada de Park, o observador também rotaciona com freqüência angular  $\omega$ . E o observador vê como um sinal constante os sinais que estão na mesma velocidade em que ele está; e vê como oscilações os sinais que estão com velocidades diferentes.

Para facilitar a visualização destes conceitos, pode-se imaginar uma família (um pai, uma mãe e dois filhos) em um parque de diversões.

Os dois filhos estão em uma montanha russa e os pais estão aguardando por eles no “chão”.



O pai olha para a mãe e ela está parada. A mãe olha para o pai e ele está parado. Pois os dois estão no mesmo referencial. Porém eles olham para os filhos, na montanha russa, e vêem estes em movimento.

Um irmão olha para o outro irmão e o vê parado, embora a montanha russa esteja em movimento sob o ponto de vista dos pais. Porém se os irmãos olharem para o “chão”, eles verão os pais em movimento.

O efeito da Transformação de Park seria a construção de uma outra montanha russa exatamente igual à primeira, colocada ao lado dela. Têm-se os dois filhos na primeira montanha russa e os pais na segunda. Se a velocidade das duas montanhas russas for a mesma, os pais enxergarão os filhos parados; mesmo estando em montanhas russas diferentes.

Portanto, a Transformação de Park é um artifício matemático que cria um novo Universo onde se pode facilmente separar sinais, mesmo com a utilização de filtros tradicionais. Pois o sinal que se deseja separar aparece, no domínio de Park, como um nível DC.

---

---

Após os cálculos dos sinais no domínio de Park, separa-se a componente DC das componentes oscilatórias através de um processo de filtragem. É recomendável calcular o filtro para uma frequência de corte uma década abaixo (ou ainda menos) da componente de menor frequência que existir no sinal que se deseja eliminar. Isto é necessário para garantir uma boa filtragem.

Contanto que a frequência de corte do filtro esteja baixa o suficiente para filtrar bem o sinal, o Compensador Ativo conseguirá um resultado muito bom em regime permanente. Porém quanto mais baixa for a frequência de corte do filtro, mais tempo durará o período transitório da compensação.

Como o que interessa é apenas eliminar a parte oscilatória do sinal, em [3.1] foi proposta uma técnica de filtragem utilizando médias móveis. O uso deste tipo de técnica permite grandes reduções na duração do período transitório.

É válido ressaltar que um transitório rápido é muito importante para um Compensador Ativo de boa qualidade, seja ele do tipo série ou do tipo shunt. Porém, para compensação de **Voltage Sags**, um transitório rápido é de fundamental importância, pois estes são eventos que ocorrem por um período de tempo muito curto e a ação do compensador deve ser quase que imediata.

O **Apêndice A** apresenta uma explicação mais detalhada sobre as médias móveis.

Para a obtenção de um transitório o mais rápido possível é necessário dimensionar bem os parâmetros do filtro (o mesmo vale para os filtros de médias móveis), e este dimensionamento depende da menor frequência que se deseja eliminar dos sinais **d** e **q**.

As componentes harmônicas de seqüência positiva no sinal trifásico de entrada, como  $4*\omega$ ,  $7*\omega$ ,  $10*\omega$ , etc aparecerão nos sinais **d** e **q** como uma oscilação de  $3*\omega$ ,  $6*\omega$ ,  $9*\omega$ , etc respectivamente.

As componentes harmônicas de seqüência negativa no sinal trifásico de entrada, como  $2*\omega$ ,  $5*\omega$ ,  $8*\omega$ , etc também aparecerão nos sinais **d** e **q** como oscilações de  $3*\omega$ ,  $6*\omega$ ,  $9*\omega$ , etc respectivamente.

Ou seja, harmônicos de seqüência positiva de ordem **n** na entrada aparecerão como oscilações de ordem **n-1** nos sinais **d** e **q**. E harmônicos de seqüência negativa de ordem **n** na entrada aparecerão como oscilações de ordem **n+1** nos sinais **d** e **q**.

Porém, em situações de ocorrência de harmônicos desequilibrados (quando as tensões ou as correntes nas três fases não possuem a mesma amplitude, ou não estão defasados de  $120^\circ$  entre si) surgem componentes de seqüência negativa para harmônicos de seqüência positiva, e componentes de seqüência positiva para os harmônicos de seqüência negativa.

Por exemplo, o sétimo harmônico ( $7*\omega$ ) é um harmônico de seqüência positiva. Sendo assim, no domínio **dq**, surgirá uma oscilação com freqüência **n-1**, ou seja,  $6*\omega$ . Porém, para o caso deste harmônico estar desequilibrado, além da oscilação de freqüência **n-1**, surgirá uma outra oscilação sobreposta com freqüência **n+1**, ou seja  $8*\omega$ ; devido a parcela de seqüência negativa que irá aparecer no sinal trifásico de entrada.

Da mesma forma um harmônico do tipo  $2*\omega$ , normalmente contribuirá com uma oscilação de  $3*\omega$ . Porém, em caso de desequilíbrio deste harmônico, surgirá uma componente com freqüência  $1*\omega$  nos sinais **d** e **q**. E está será a menor freqüência possível de ocorrência no domínio **dq**.

Um filtro de médias móveis (conforme análise no **Apêndice A**) para eliminar uma oscilação de  $\omega$  rad/s terá um transitório de 1 ciclo de  $\omega$  rad/s. Ou seja se  $\omega$  for igual a  $2*\pi*60$ , o transitório demorará 1 ciclo de **60Hz**, portanto **16.66ms**. Infelizmente não há como reduzir este tempo.

Felizmente, a ocorrência de harmônicos de segunda ordem não é muito comum. Sendo assim, o filtro não precisa ter freqüência tão baixa. Porém, é aconselhável um estudo do tipo de harmônicos que se deseja eliminar antes do cálculo dos parâmetros do filtro.

Um **Voltage Sag** normalmente, devido ao desequilíbrio nas tensões, apresenta uma componente de seqüência negativa na freqüência fundamental (**60Hz**). Esta componente contribuirá com uma oscilação de **120Hz** no domínio **dq**. Portanto esta será a menor freqüência que deverá ser eliminada.

Um filtro de médias móveis para eliminar uma oscilação de  $2*\omega$  rad/s terá um transitório de 1 ciclo de  $2*\omega$  rad/s, ou seja  $\frac{1}{2}$  ciclo de **60Hz**. Um transitório de  $\frac{1}{2}$  ciclo já é aceitável, pois pela definição de **Voltage Sags**, estes têm duração entre  $\frac{1}{2}$  ciclo e 1 minuto.

---

As próximas seções apresentam algoritmos que utilizam a transformação de coordenadas como ferramenta principal.

### 3.2.2 – Método $id$ - $iq$

Este método consiste em se aplicar a Transformação de Clarke, **Eq. 3.3**, ao sinal trifásico amostrado. Após esta primeira transformação, aplica-se a Transformação de Park, (**Eq. 3.14**). Os sinais  $\sin\theta$  e  $\cos\theta$  são calculados através de uma PLL (maiores detalhes sobre a implementação de PLLs podem ser vistos no **Apêndice B**). Esta PLL é configurada para seguir a fase da componente de frequência fundamental.

Sendo assim, no domínio  $dq$ , apenas a componente de seqüência positiva na frequência fundamental será um nível DC. As demais componentes serão vistas como oscilações.

O método  **$id$ - $iq$**  utiliza apenas a parte oscilatória dos sinais  $d$  e  $q$ . Pois o objetivo deste método é gerar um sinal de compensação para todas as componentes que perturbam o sistema; sendo assim, a componente DC deve ser eliminada, e o sinal oscilatório deve ser utilizado para o cálculo das referências de compensação.

Para a obtenção da parcela oscilatória, isto pode ser feito utilizando-se um filtro do tipo passa-altas. A implementação de um filtro passa-altas terá o mesmo problema com relação à duração do período transitório relatado na **seção 3.2.1**. Sendo assim, pode-se converter um filtro passa-baixas de médias móveis em um filtro passa-altas, conforme estrutura da **Figura 3.18**.

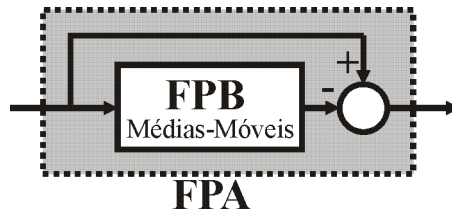


Figura 3.18 – Conversão de Passa-Baixas para Passa-Altas

Tendo-se a componente oscilatória no domínio  $dq$ , parte-se para o caminho reverso, ou seja, a volta para o referencial trifásico.

À componente oscilatória no referencial  $dq$  é aplicada a transformação de Park inversa, (**Eq. 3.21**), tendo como resultado a volta ao sistema de coordenadas bifásico estacionário.

$$\begin{bmatrix} v\alpha \\ v\beta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v_d \\ v_q \end{bmatrix} \quad (\text{Eq. 3.21})$$

Aos sinais no sistema bifásico estacionário é aplicada a transformação de Clarke inversa, (**Eq. 3.22**), tendo como resultado os sinais de compensação no referencial trifásico **a-b-c**.

$$\begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} = \sqrt{\frac{2}{3}} \cdot \begin{bmatrix} 1/\sqrt{2} & 1 & 0 \\ 1/\sqrt{2} & -1/2 & \sqrt{3}/2 \\ 1/\sqrt{2} & -1/2 & -\sqrt{3}/2 \end{bmatrix} \cdot \begin{bmatrix} v_0 \\ v\alpha \\ v\beta \end{bmatrix} \quad (\text{Eq. 3.22})$$

Os sinais de compensação  $va^*$ ,  $vb^*$  e  $vc^*$  podem, então, ser enviados a um modulador **PWM**, e posteriormente a um inversor para o sistema de potência seja compensado. O diagrama em blocos completo do método **id-iq** é dado na **Figura 3.19**.

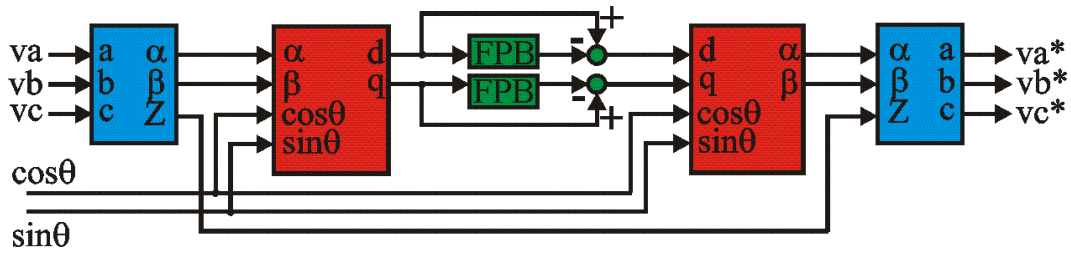


Figura 3.19 – Método id-iq

Nota-se, na **Figura 3.19** que a componente de seqüência zero não sofre qualquer tipo de processamento. Sendo assim, ela é separada durante a transformação de Clarke, (**Eq. 3.3**), e reagrupada na transformação de Clarke inversa, (**Eq. 3.22**).

As **Figuras 3.20** e **3.21** apresentam uma simulação para este algoritmo. Utilizou-se, na entrada, harmônicos desequilibrados, na composição descrita na **Tabela 3.1**.

Tabela 3.1  
Distorção Harmônica de Entrada

	Fase A	Fase B	Fase C
5 <sup>o</sup> harmônico	0.10%	0.15%	0.05%
7 <sup>o</sup> harmônico	0.05%	0.10%	0.15%
11 <sup>o</sup> harmônico	0.08%	0.03%	0.05%
13 <sup>o</sup> harmônico	0.05%	0.03%	0.08%

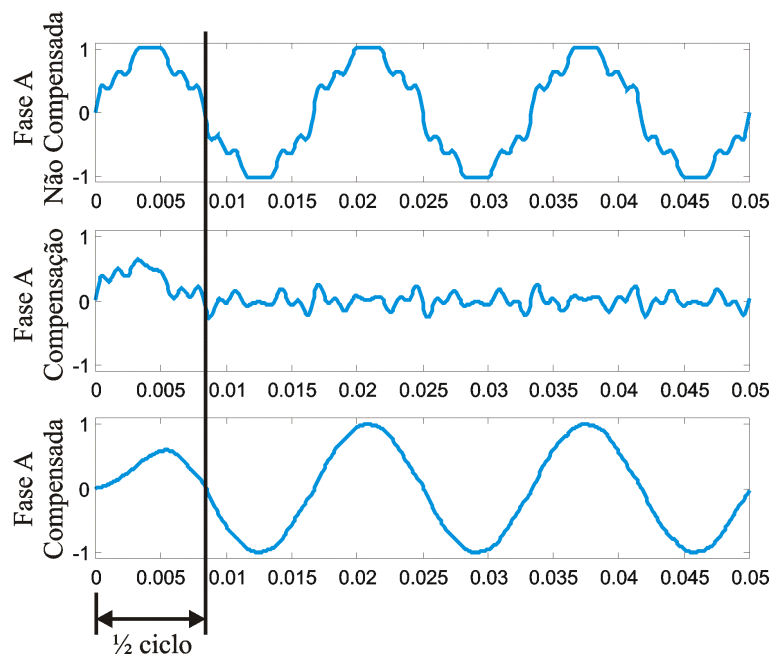


Figura 3.20 – Simulação do Algoritmo **id-iq**

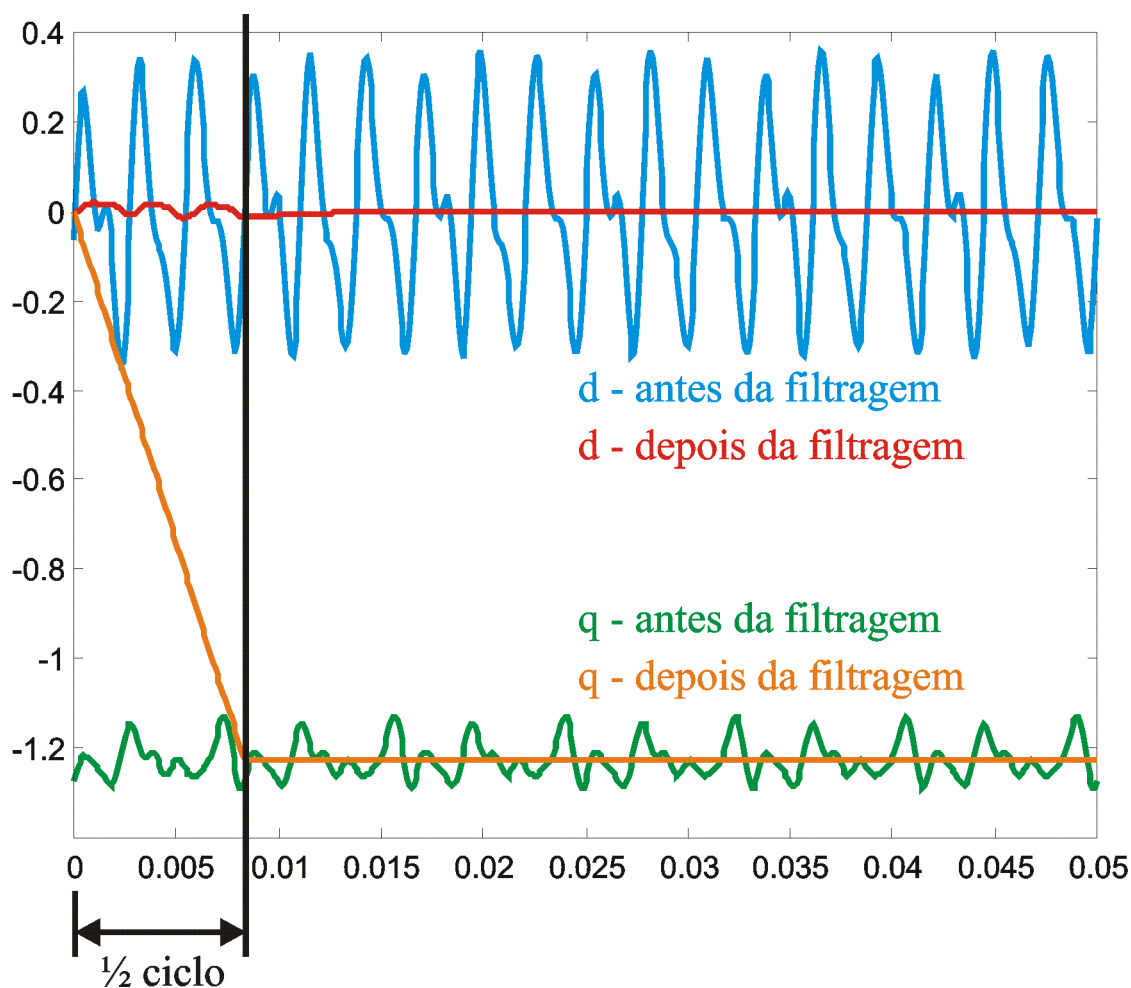


Figura 3.21 – Sinais no Domínio  $dq$

Nota-se, nas **Figuras 3.20 e 3.21**, um transitório com duração de  $\frac{1}{2}$  ciclo de **60Hz**, ou seja, **8.33ms**. Isto já era previsto, uma vez que o filtro de médias móveis está configurado para eliminar frequências a partir de **120Hz**.

Este método é muito utilizado em implementações de Compensadores Ativos. Alguns (dentre os diversos) trabalhos que utilizaram esta metodologia são as referências [3.1] e [3.2].

Porém, caso haja ruído no sistema de aquisição de dados, este ruído estará presente na parcela oscilatória dos sinais no domínio  $dq$ . E é justamente a parcela oscilatória que este método utiliza para gerar as compensações.

A **Figura 3.22** ilustra uma situação em que são mantidas todas as condições da simulação anterior (de acordo com **Tabela 3.1**) e é adicionado ruído branco aos sinais aquisitados.

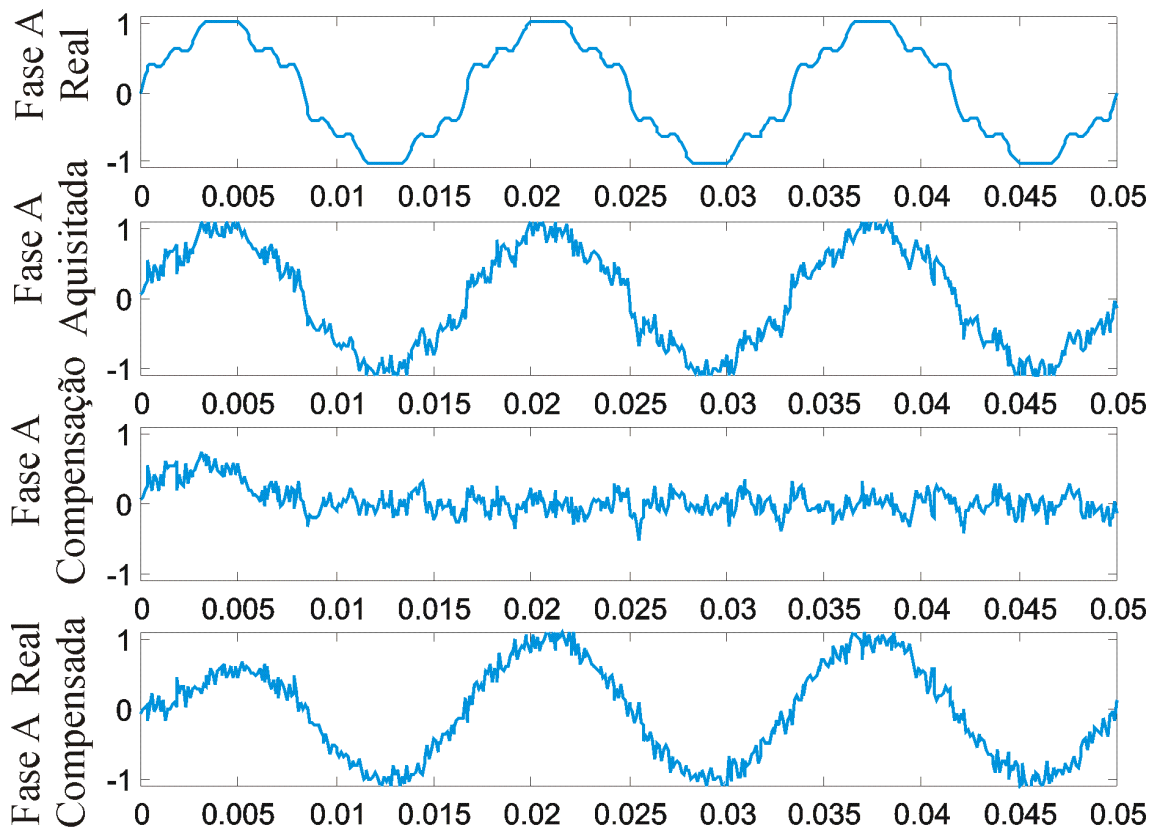


Figura 3.22 – Método id-iq com ruído na aquisição de dados

Nota-se, na **Figura 3.22**, que o ruído está presente apenas no sistema de aquisição de dados, e não no sistema de potência. O problema é que o algoritmo não tem como saber disso e gera uma compensação para o ruído da mesma forma como faz para os harmônicos. Sendo assim, o Compensador acaba por transferir todo o ruído da aquisição para a saída. Ou seja, o Compensador está “limpando” os harmônicos do sistema, porém está “sujando” o sistema com ruídos.

Deve-se, portanto, encontrar alguma forma de se calcular as compensações sem levar o ruído em consideração.

---

A próxima seção descreve a solução encontrada para o problema do ruído na aquisição de dados.

### 3.2.3 – Método **id-iq** Modificado para Imunidade Contra Ruídos

No método **id-iq** convencional, calculam-se os sinais no referencial  $dq$ , compostos por uma componente DC e uma componente oscilatória. Em seguida a componente DC é eliminada e a parte oscilatória é utilizada para o cálculo das referências de compensação. Caso haja contaminação com ruídos no sinal de aquisição, os ruídos estarão presentes na parte oscilatória, sendo assim o sinal de compensação também estará contaminado com os ruídos.

O método proposto neste trabalho consiste em uma adaptação do método **id-iq**, de forma que os ruídos não influenciem os cálculos das referências de compensação. Assim como o método **id-iq** original, o método modificado permite a eliminação de harmônicos de correntes (para compensadores shunt) ou de tensão (para compensadores série).

A técnica proposta consiste em se aplicar uma Transformada de Park para cada componente harmônica dos sinais de entrada. Ou seja, os sinais  $\sin\theta$  e  $\cos\theta$  deverão possuir a frequência do harmônico que se deseja eliminar.

Da explicação da **seção 3.2.1**, percebe-se que se os sinais  $\sin\theta$  e  $\cos\theta$  da transformada de Park tiverem uma frequência de  $n\omega t$ , as componentes harmônicas dos sinais de entrada nesta mesma frequência aparecerão como níveis DC no referencial  $dq$ , e todas as outras frequências aparecerão como oscilações.

A diferença entre esta técnica e a técnica convencional é que na técnica convencional eliminava-se a componente DC. Com a técnica proposta, elimina-se a componente oscilatória. Dessa forma, elimina-se junto com ela todos os ruídos.

Outra diferença importante é que na técnica convencional a componente de seqüência negativa aparecia na parte oscilatória, sendo, portanto compensada. Devido ao fato da técnica proposta compensar apenas a parte DC, deve-se acrescentar uma Transformação de Park de seqüência negativa, (**Eq. 3.23**), para cada harmônico.

$$\begin{bmatrix} vd_- \\ vq_- \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} v\alpha \\ v\beta \end{bmatrix} \quad (\text{Eq. 3.23})$$

Portanto na transformação de seqüência positiva as componentes de seqüência positiva daquela frequência aparecerão como níveis DC, e todas as outras frequências e a componente de seqüência negativa daquela frequência aparecerão como oscilações.

Da mesma forma, na transformação de seqüência negativa as componentes de seqüência negativa daquela frequência aparecerão como níveis DC, e todas as outras frequências e a componente de seqüência positiva daquela frequência aparecerão como oscilações.

As **Figuras 3.23** e **3.24** apresentam a técnica proposta. Realiza-se a transformação de Clarke, (**Eq. 3.3**), normalmente. Em seguida aplica-se uma transformação de Park de seqüência positiva, (**Eq. 3.14**), e uma de seqüência negativa, (**Eq. 3.23**), para cada harmônico do sistema. Extrai-se a componente DC utilizando-se um filtro passa-baixas, que pode ser do tipo Butterworth (porém torna o transitório lento) ou de médias móveis. Aplica-se uma transformação inversa de Park de seqüência positiva, (**Eq. 3.21**), e uma de seqüência negativa, (**Eq. 3.24**), para cada frequência. Somam-se todos os sinais do eixo **alfa**, e todos os sinais do eixo **beta**. Aplica-se a transformação inversa de Clarke, (**Eq. 3.22**), e como resultado têm-se os sinais de compensação,  $va^*$ ,  $vb^*$  e  $vc^*$ .

$$\begin{bmatrix} v\alpha \\ v\beta \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} vd_- \\ vq_- \end{bmatrix} \quad (\text{Eq. 3.24})$$

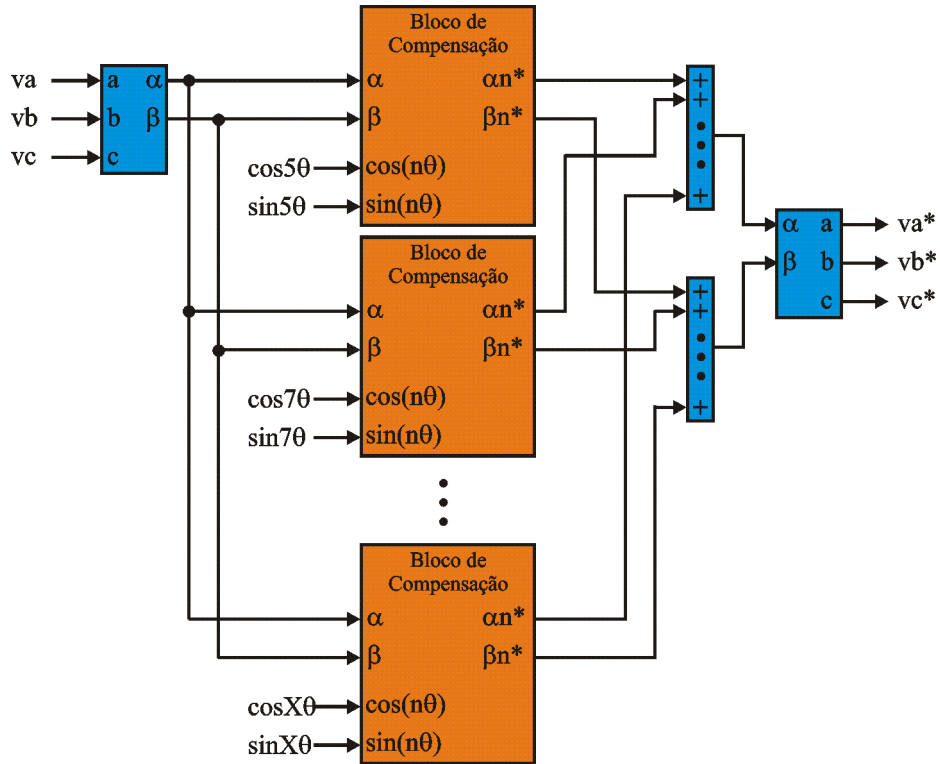


Figura 3.23 – Técnica Proposta

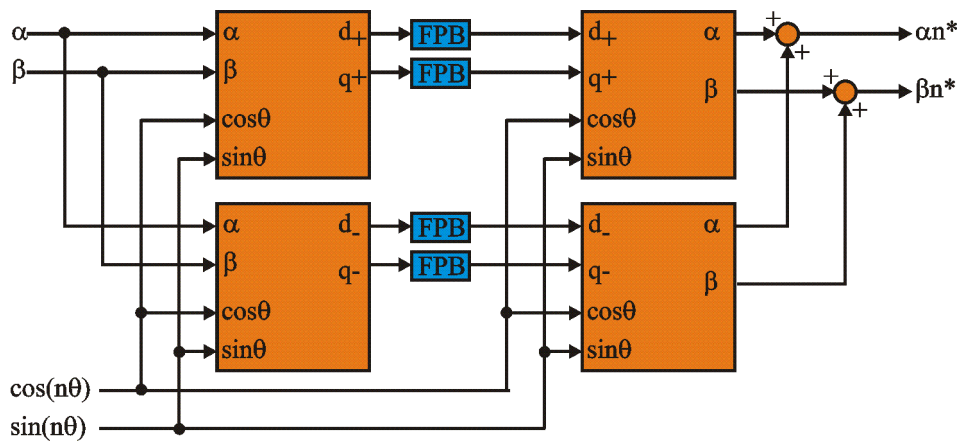


Figura 3.24 – Bloco de Compensação

Nota-se a repetição dos blocos da **Figura 3.24** na **Figura 3.23**. A única diferença é o argumento dos senos e co-senos. O ângulo  $\theta$ , que será multiplicado pela ordem do harmônico desejado vem de uma PLL.



Ainda falta acrescentar a componente de seqüência zero. Na técnica **id-iq**, convencional a componente de seqüência zero é simplesmente reagrupada ao sinal de compensação na transformada de Clarke inversa, (Eq. 3.22). Porém a componente de seqüência zero traz consigo a parcela dos ruídos da aquisição que não se anulam numa soma instantânea. Sendo assim, para compensação de corrente em sistemas a 4 fios, ou para compensação de tensões com desequilíbrios, deve-se compensar a seqüência zero de uma forma insensível aos ruídos da aquisição.

A forma encontrada de fazer isto é a aplicação da transformada de Park para cada harmônico de seqüência zero. O problema é que a seqüência zero consiste de três vetores de mesmo sentido e mesma orientação, e é por esta razão que ela é eliminada na transformação de Clarke.

Deve-se, portanto, utilizar um artifício que consiste em se considerar a seqüência zero dois vetores ortogonais, representando matematicamente o que seriam os sinais “**alfa e beta de seqüência zero**”, uma vez que o eixo **alfa-beta** não admite seqüência zero. Dessa forma, de acordo com as equações (Eq. 3.7) e (Eq. 3.8), tem-se que a parcela do eixo **alfa** da seqüência zero será a própria seqüência zero, e a parcela do eixo **beta** será a seqüência zero atrasada de **90°**, conforme **Figura 3.25**.

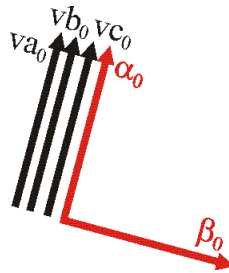


Figura 3.25 – Artifício para Conversão da seqüência Zero

A **Figura 3.26** apresenta o bloco de compensação quando se deseja compensar a seqüência zero. Nota-se apenas a transformação de Park de seqüência positiva para a seqüência zero, pois  $\beta_0$  sempre estará atrasado de **90°** de  $\alpha_0$ , uma vez que estes sinais foram artificialmente postos nesta condição.

O algoritmo completo com compensação de seqüência zero encontra-se na **Figura 3.27**.

Vale ressaltar que em Compensadores do tipo Shunt em sistemas a três fios, o algoritmo das **Figuras 3.23** e **3.24** já basta. Em sistemas com neutro é necessária a utilização do algoritmo das **Figuras 3.26** e **3.27**.

Em Compensadores do tipo Série também é necessária a utilização do algoritmo das **Figuras 3.26** e **3.27**, pois caso as tensões estejam desequilibradas, haverá componente de seqüência zero.

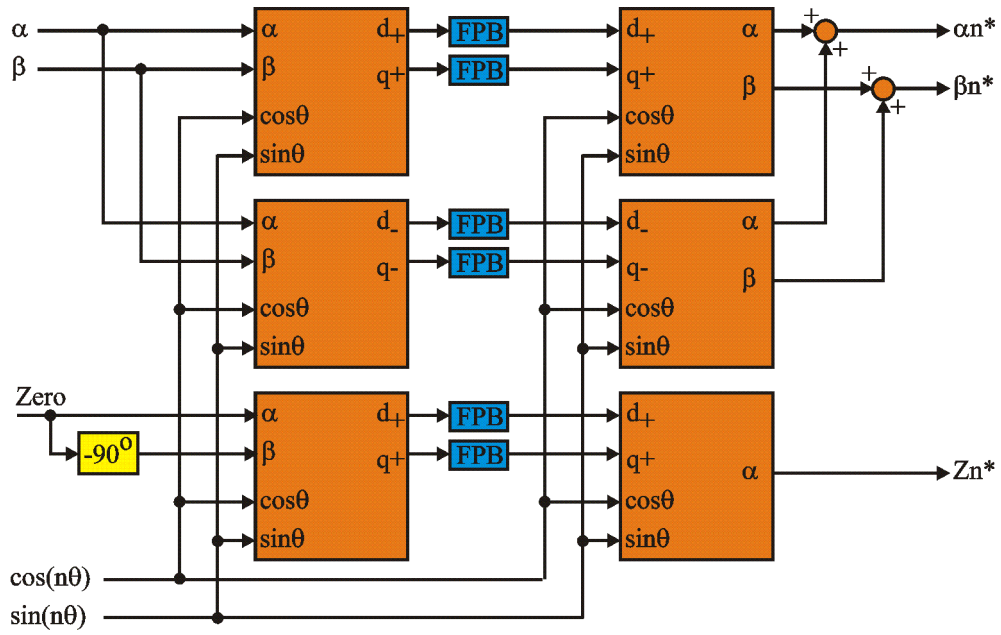


Figura 3.26 – Bloco de Compensação – com compensação de seqüência zero

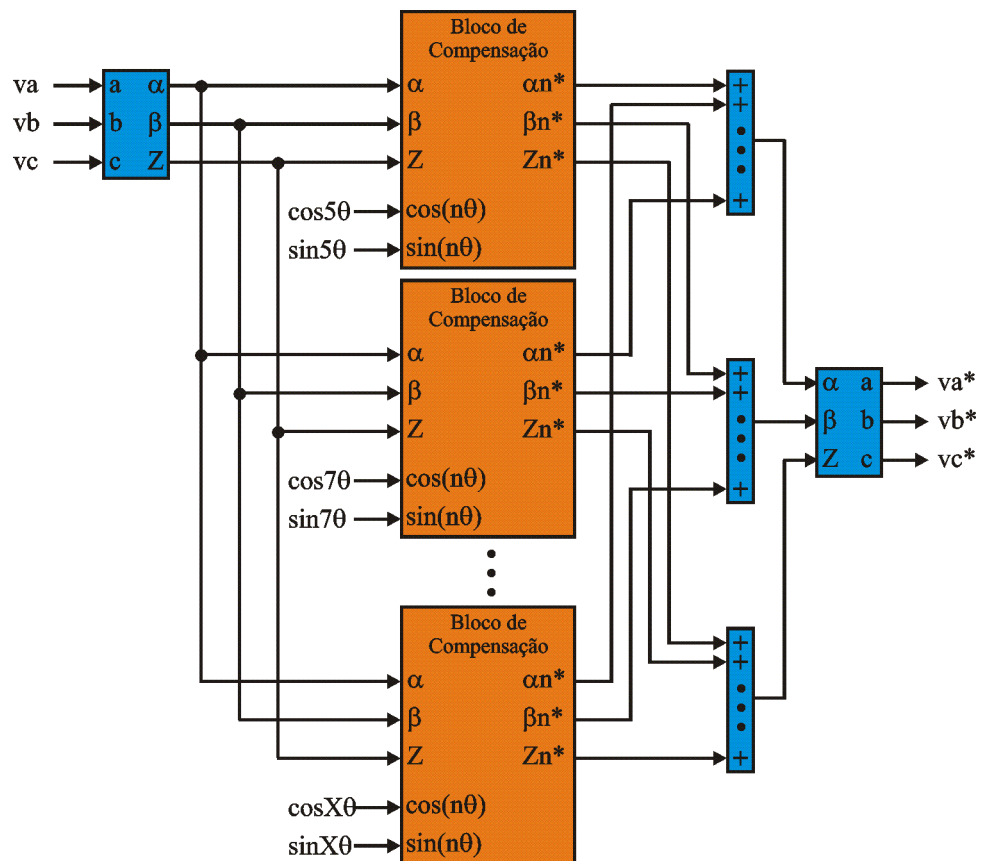


Figura 3.27 – Técnica Proposta – com compensação de seqüência zero

### 3.2.4 – Comparação entre os Métodos Convencional e Modificado

Os métodos  $id-iq$  convencional (**Figura 3.19**) e modificado (**Figuras 3.26 e 3.27**) foram recriados em uma mesma simulação, para que a performance dos dois pudesse ser comparada entre si.

Como sinais de entrada, tem-se um sistema trifásico com 1 pu de amplitude para a frequência fundamental. Somado ao sinal da fundamental foram acrescentados harmônicos desequilibrados, na composição dada na **Tabela 3.2**.

Tabela 3.2  
Distorção Harmônica de Entrada

	Fase A	Fase B	Fase C
5 <sup>o</sup> harmônico	0.10pu	0.15 pu	0.05 pu
7 <sup>o</sup> harmônico	0.05 pu	0.10 pu	0.15 pu
11 <sup>o</sup> harmônico	0.08 pu	0.03 pu	0.05 pu
13 <sup>o</sup> harmônico	0.05 pu	0.03 pu	0.08 pu

Nota-se um desequilíbrio exagerado entre as fases. Isto foi feito propositalmente para verificar a compensação de seqüência zero.

Com o sistema nesta configuração, sem a adição de ruídos, gerou-se a **Figura 3.28**. Tem-se na parte superior o sinal de entrada (não compensado), na parte central o sinal compensado pelo método  $id-iq$  convencional, e na parte inferior, o sinal compensado pelo método  $id-iq$  modificado.

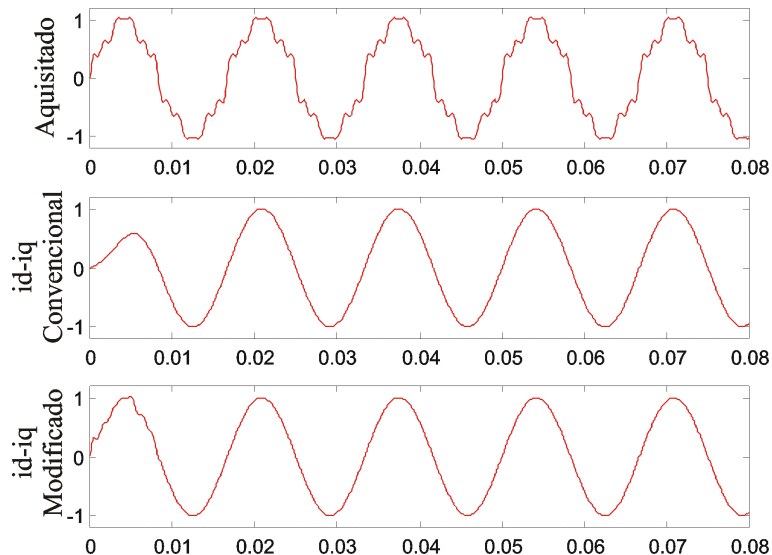


Figura 3.28 – Comparação sem Ruídos

Mantendo-se o sistema nas mesmas condições, acrescentou-se ruído branco na entrada dos compensadores, simulando ruído no sistema de aquisição de dados. Gerou-se a **Figura 3.29**. Nota-se que, no método  $id-iq$  convencional, o compensador gerou compensação para o ruído, contaminando o sinal compensado.

Observa-se, ainda na **Figura 3.29**, que no método modificado, o compensador gerou compensação para uma parcela do ruído. Isto se deve ao fato de o ruído branco possuir componentes em todas as frequências, sendo assim as parcelas de 5°, 7°, 11° e

13º harmônicos contaminaram a compensação. Deve-se notar, porém, a diferença entre o ruído gerado pelos dois métodos.

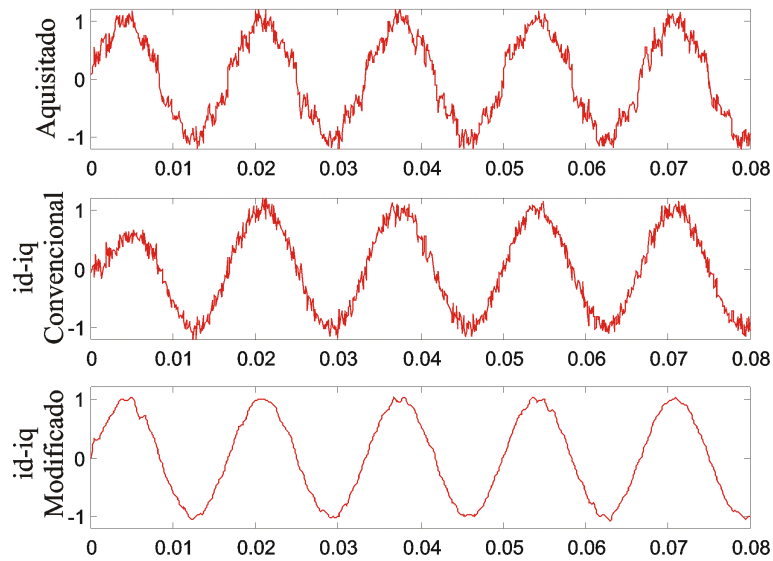


Figura 3.29 – Comparação com Ruído Branco

Outra simulação foi realizada, mantendo a mesma configuração de harmônicos e trocando o ruído branco por ruído de alta frequência (que não apresenta parcelas de 5º, 7º, 11º e 13º harmônicos). Utilizou-se uma soma de duas ondas com 2kHz e 5kHz. Gerou-se a **Figura 3.30**.

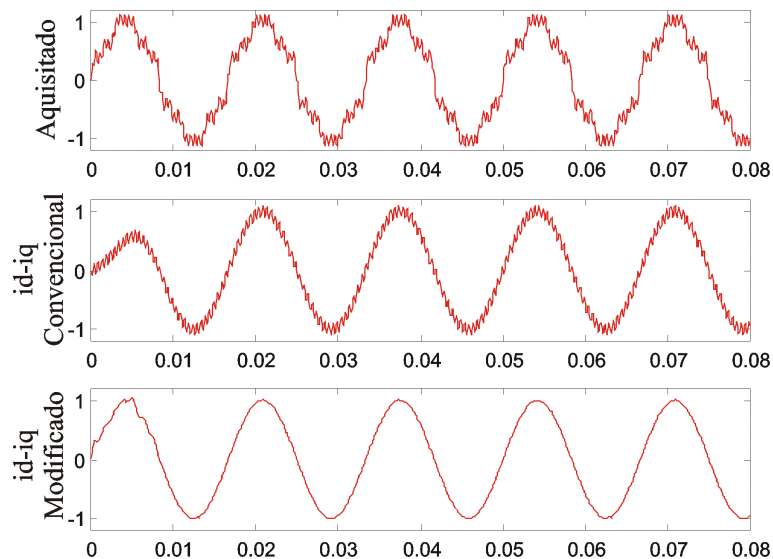


Figura 3.30 – Comparação com Ruído de alta frequência

No caso da simulação da **Figura 3.30**, nota-se que o método modificado não transfere ruído à saída.

### 3.2.5 – Compensação em Malha Fechada

A implementação do algoritmo de controle do compensador em malha aberta apresenta a vantagem da simplicidade; entretanto, nada garante que a compensação gerada nos inversores chegue corretamente (da mesma forma como foi gerada) até o ponto onde a soma de tensões ou correntes acontece.

Algoritmos em malha aberta geram um sinal de compensação, que será amplificado por um inversor de frequência. O ponto de acoplamento entre o Compensador Ativo e o sistema de potência, normalmente, é um transformador. Portanto, até que a compensação seja de fato realizada, existe um longo caminho que o sinal de compensação precisa percorrer; estando sujeito a perdas e distorções durante o trajeto.

Com a implementação do algoritmo em malha fechada, mesmo com as perdas e distorções ao longo do caminho, o sinal no ponto de compensação chega da forma como deveria chegar.

Uma pequena mudança no local da aquisição de dados precisa ser feita. Em Compensadores do tipo Série deve-se adquirir a tensão na carga; ao invés de se adquirir a tensão da concessionária, como antes era feito em malha aberta. Como se deseja manter a tensão na carga livre de distorções, desequilíbrios ou Voltage Sags, o algoritmo em malha fechada irá, a todo momento, verificar como está a tensão na carga, modificando o sinal de compensação até que os objetivos sejam alcançados.

O mesmo vale para os Compensadores do tipo Shunt. Deve-se *adquirir* a corrente do lado da concessionária, ao invés de se *adquirir* a corrente da carga, como antes era feito em malha aberta.

A correção do sinal de compensação, em função das perdas e distorções, é conseguida graças à mudança do ponto de aquisição (que equivale a uma realimentação do sinal de saída) e a utilização de um **regulador PI**.

A desvantagem da implementação em malha fechada é a dificuldade no cálculo dos ganhos do **regulador PI**. Existem diversas técnicas que podem ser utilizadas para calcular estes ganhos, as mais conhecidas são o **Método do Lugar das Raízes** e o **Método da Resposta em Frequência**. Estes métodos podem ser encontrados em livros de controle como [3.3] e [3.4]. O problema com estas técnicas é que é preciso conhecer a função de transferência do sistema para poder calcular os ganhos.

É necessário o modelamento matemático de todos os parâmetros envolvidos no processo. Ou seja, é preciso modelar o sistema de potência, o algoritmo de compensação (as transformações de coordenadas), as perdas e distorções, e tudo o que estiver envolvido.

Dada a complexidade da tarefa, optou-se pelo “método da tentativa e erro”.

3.2.5.1 – Ajuste dos Ganhos do Regulador PI

Desprezando-se qualquer dinâmica envolvida no sistema, exceto a do filtro de médias móveis, pode-se simplificar a malha de controle para o ajuste dos ganhos do Regulador PI. A malha de controle simplificada pode ser vista na **Figura 3.31**.

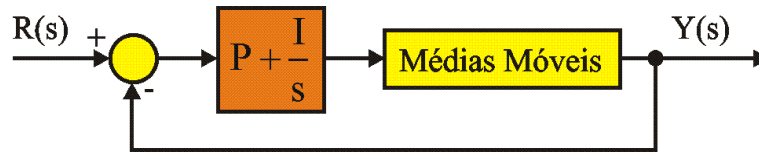


Figura 3.31 – Malha de Controle Simplificada para Ajuste dos Ganhos

A simulação da **Figura 3.31** tem como entrada um degrau de 0 até  $\sqrt{3/2}$  (que é o maior valor que poderá aparecer no referencial  $dq$ , se a amplitude dos sinais trifásicos,  $U$ , for igual a 1). Inicialmente mantém-se o ganho Integral ( $I$ ) igual a zero e, por tentativa e erro, ajusta-se o ganho Proporcional ( $P$ ) para que o transitório seja o menor possível. Este procedimento pode ser visto na **Figura 3.32**.

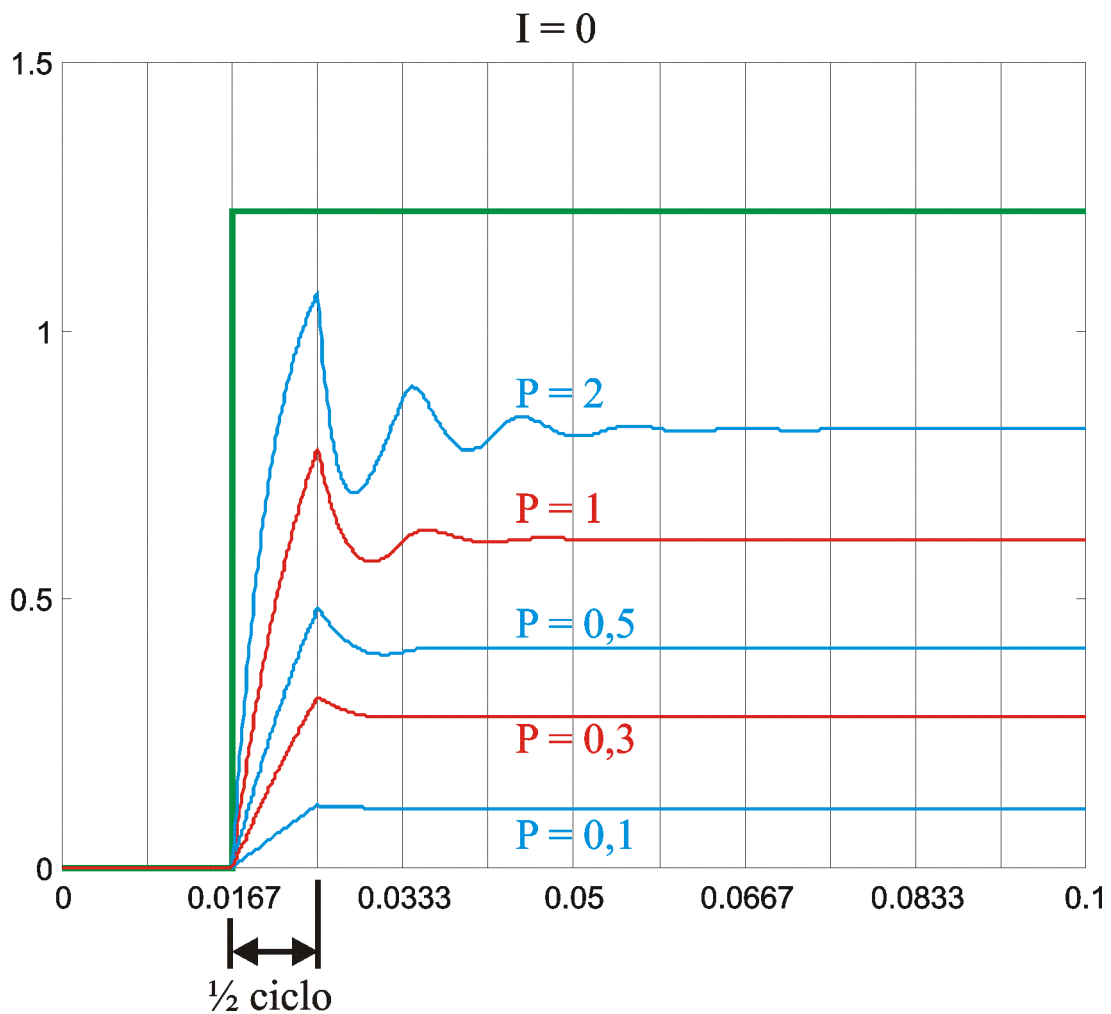


Figura 3.32 – Ajuste do Ganho Proporcional, com  $I=0$

Inicialmente, quando não se tem a menor idéia de que valor usar, utilizou-se  $P=1$ . O transitório foi de cerca de 2 ciclos. Dobrou-se o valor de  $P$ . Para  $P=2$ , o transitório aumentou para mais do que 3 ciclos, e o *overshoot* também aumentou. Nota-se, porém que o erro em regime permanente diminuiu.

O erro em regime permanente será corrigido com a parcela integral, sendo assim, deve-se apenas tentar diminuir o transitório. Como o aumento de  $P$  prejudicou o transitório, deve-se diminuí-lo.

Para  $P=0.5$  o transitório caiu para pouco mais do que 1 ciclo.

Para  $P=0.3$ , nota-se que não existe mais *overshoot*. Portanto a dinâmica em malha fechada do sistema deve estar com amortecimento crítico. Provavelmente este deve ser o melhor valor para  $P$ . Mas mesmo assim tentou-se diminuir  $P$  ainda mais.

Para  $P=0.1$  obteve-se o menor transitório. Porém o erro em regime permanente ficou muito grande. Provavelmente para corrigir este erro, mantendo o mesmo transitório o sistema irá apresentar oscilações. Mesmo assim fez-se o ajuste do ganho Integral com  $P=0.1$ , conforme **Figura 3.33**.

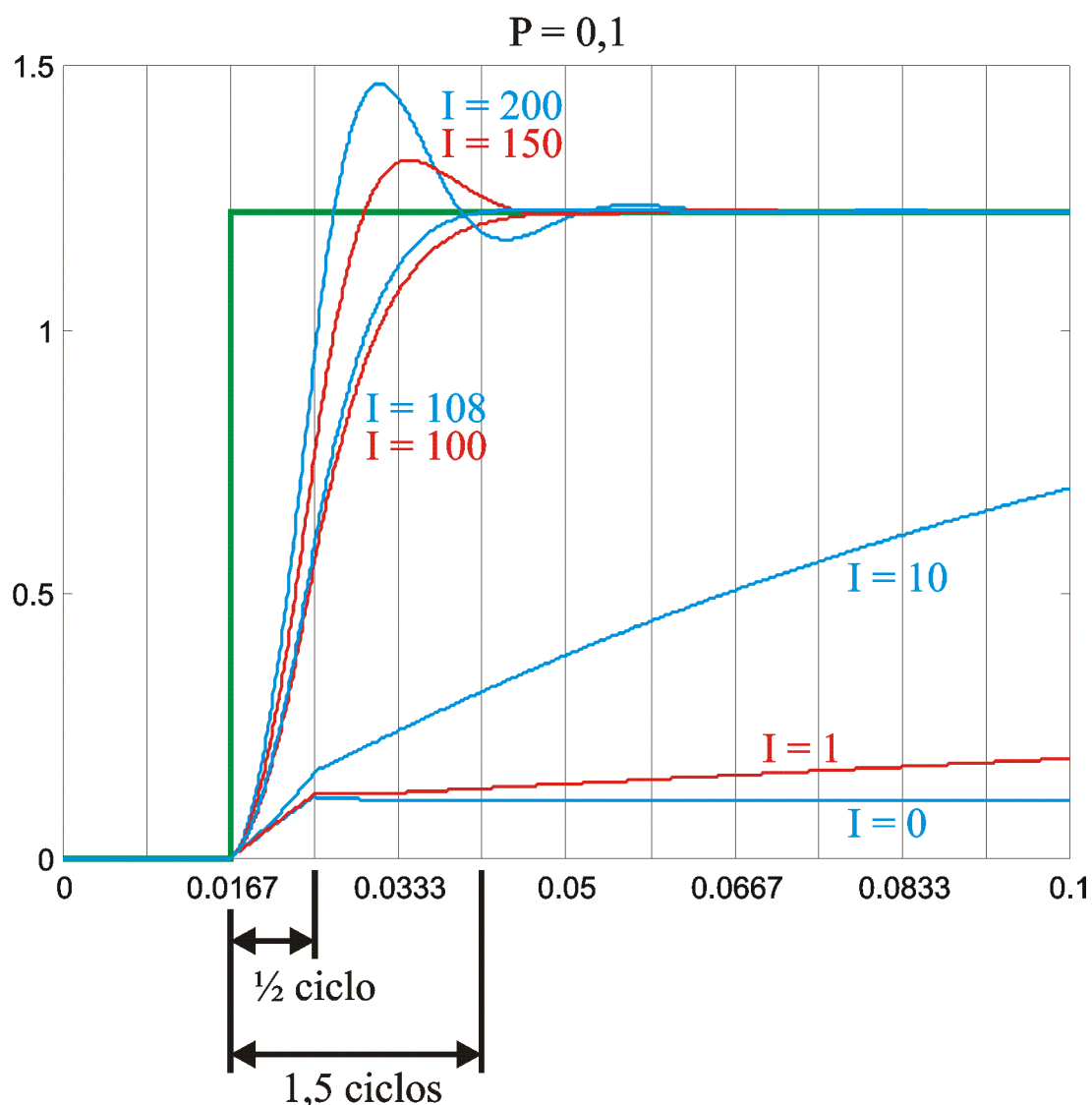


Figura 3.33 – Ajuste do Ganho Integral, com  $P=0.1$

Começando com  $I=1$ , nota-se que o sistema demorará muito para anular o erro em regime permanente. Sendo assim, vai-se aumentando  $I$ .

Para  $I=100$  o sistema demorou cerca de 1.75 ciclos para se estabilizar no regime permanente.

Para  $I=200$  o sistema passa a ter *overshoot*, sendo assim deve-se diminuir  $I$ .

Para  $I=150$  o sistema ainda está subamortecido. Normalmente é melhor trabalhar no amortecimento crítico, pois é a situação em que o sistema está mais rápido sem apresentar *overshoot*. Esta situação foi conseguida para  $I=108$ .

Nota-se que o sistema só entra, definitivamente, em regime permanente em 1.75 ciclos. Entretanto, um pouco antes de 1.5 ciclos já se tem uma boa aproximação.

Realizou-se o mesmo procedimento para  $P=0.3$ , conforme **Figura 3.34**.

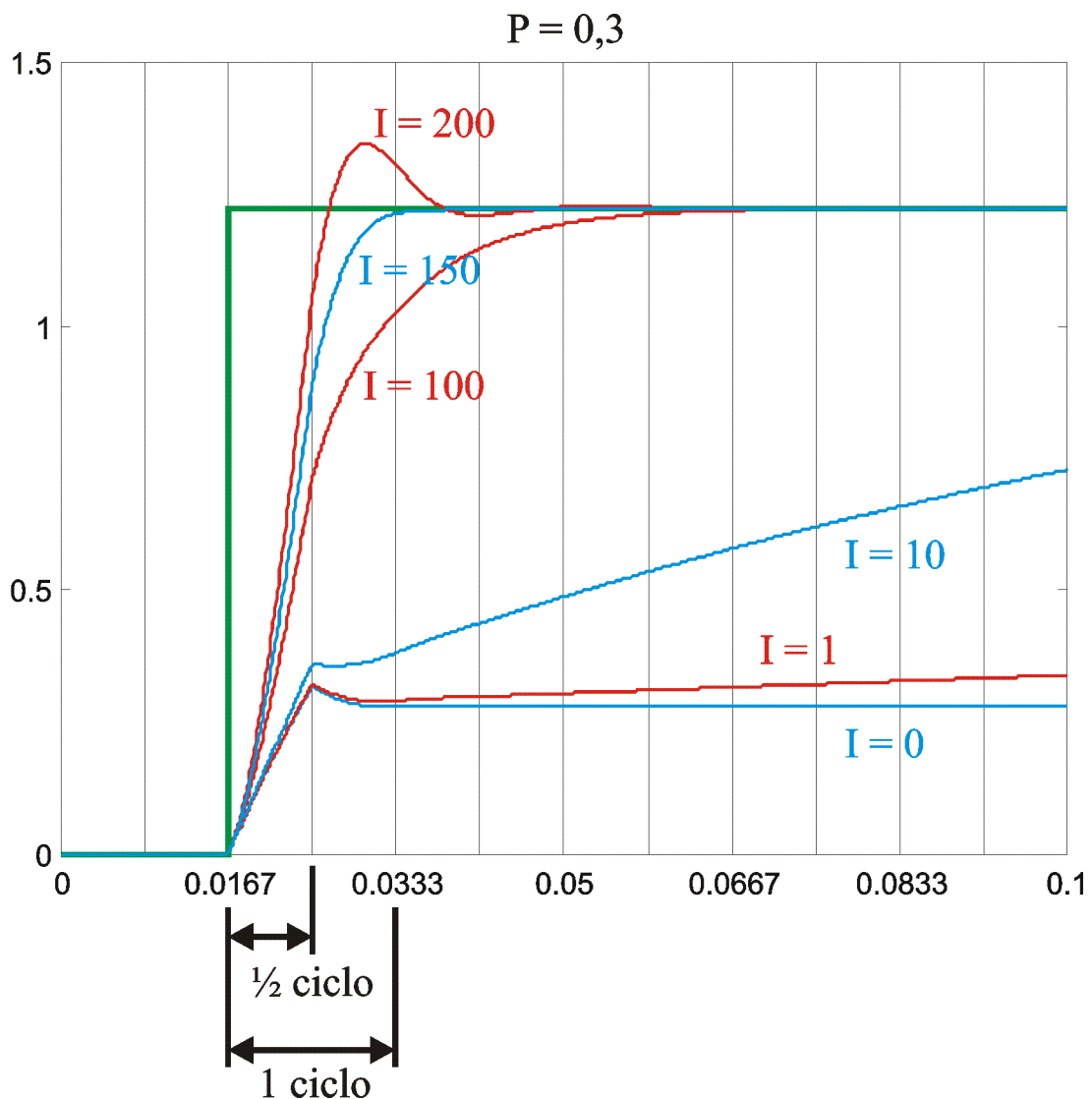


Figura 3.34 – Ajuste do Ganho Integral, com  $P=0.3$



Para  $P=0.3$  e  $I=150$  conseguiu-se atingir o regime permanente em pouco mais do que 1 ciclo e sem overshoot.

Deve-se ressaltar que um transitório menor do que  $\frac{1}{2}$  ciclo é impossível, devido à filtragem necessária para eliminar o sinal de 120Hz no eixo  $dq$ . Nota-se, nas Figuras 3.32, 3.33 e 3.34, que o regulador só começa a controlar o sinal a partir de  $\frac{1}{2}$  ciclo.

### 3.2.5.2 – Compensação de Harmônicos em Malha Fechada

A partir do algoritmo em malha aberta, Figuras 3.26 e 3.27, poucas mudanças precisam ser realizadas.

Tem-se na Figura 3.35 o diagrama em blocos de uma malha de controle com realimentação.

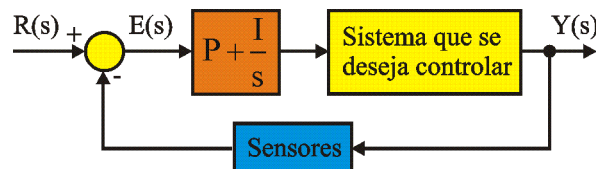


Figura 3.35 – Controle em Malha Fechada

Deseja-se que a saída  $Y(s)$  acompanhe o valor de referência ajustado em  $R(s)$ . A saída é realimentada e comparada com o valor de referência. O regulador irá trabalhar para fazer com que o erro  $E(s)$  fique em torno de zero. Nesta condição a saída estará igual a entrada.

Para o algoritmo de compensação de harmônicos, deseja-se que tanto a componente de seqüência positiva, quanto a de seqüência negativa, quanto a de seqüência zero sejam eliminadas. Para que isto aconteça os sinais no eixo  $dq$  devem ser cancelados. Portanto a entrada de referência  $R(s)$  será configurada com o valor zero. Como sensores tem-se todo o conjunto que gera os sinais  $d$  e  $q$ , ou seja, as Transformações de Clarke e de Park e o filtro de Médias Móveis. Isto está ilustrado na Figura 3.36.

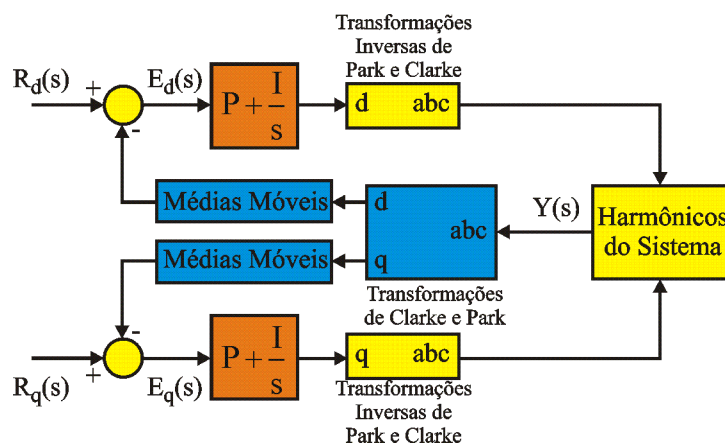


Figura 3.36 – Controle em malha fechada no eixo  $dq$

A **Figura 3.37** apresenta a modificação feita no Bloco de Compensação (da **Figura 3.26** em malha aberta) para controle em malha fechada. Nota-se a inserção de reguladores PI após cada filtro de médias.

Nota-se também uma multiplicação por (-1) antes de cada regulador PI, isto é necessário, pois conforme as **Figuras 3.35** e **3.36** o sinal de erro será o sinal de referência menos o sinal de realimentação. Como a entrada de referência deve ser zero (quer-se eliminar os harmônicos) o sinal de erro será o sinal de realimentação (saída das médias móveis) multiplicado por (-1).

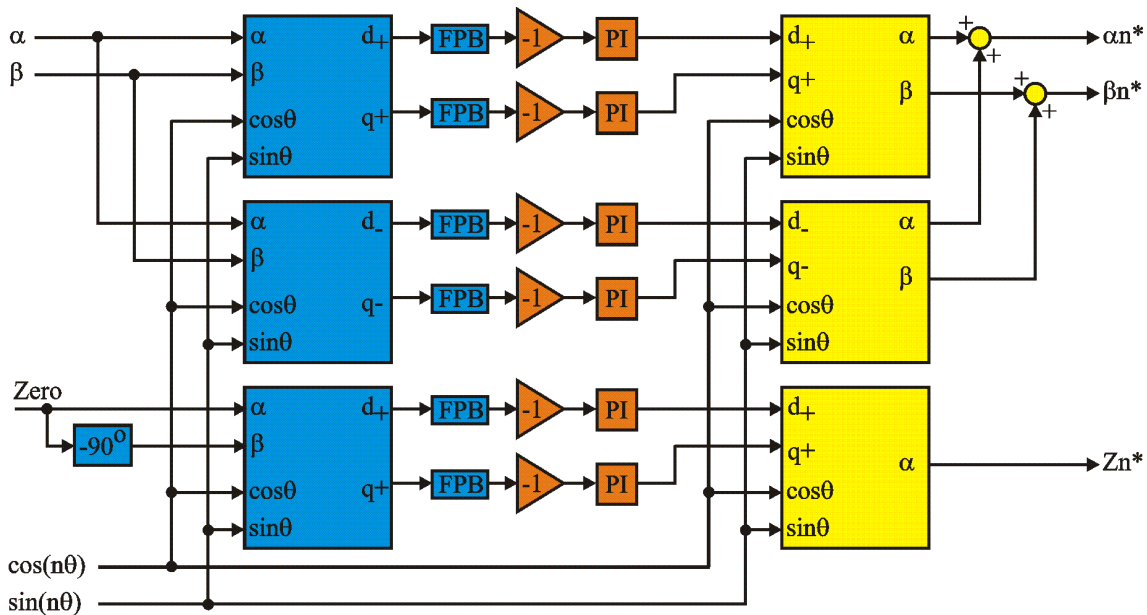


Figura 3.37 – Bloco de Compensação de Harmônicos, em malha fechada

O algoritmo da **Figura 3.37** foi simulado mantendo-se a mesma configuração de harmônicos desequilibrados da **Tabela 3.2**. Porém, inicialmente apenas com a fundamental. Repentinamente surgem os harmônicos. Com isto pode-se simular a performance dinâmica do algoritmo em malha fechada.

A **Figura 3.38** apresenta o resultado desta simulação, com adição de ruído branco, durante toda a simulação. Na parte superior, tem-se o sinal de tensão da concessionária. Caso o Compensador não estivesse sendo utilizado, este seria o sinal que estaria sendo entregue à carga. Na parte central, tem-se a tensão aquisitada que está entrando para o bloco de compensação. Esta tensão é igual a tensão da carga somada com os ruídos do sistema de aquisição de dados. Na parte inferior, tem-se a tensão compensada na carga.

Pelo mesmo motivo já explicado para o algoritmo de malha aberta, parte do ruído contamina o sinal de compensação. Porém nota-se uma nítida diferença entre a quantidade de ruídos na entrada do compensador e o ruído transferido para a saída.

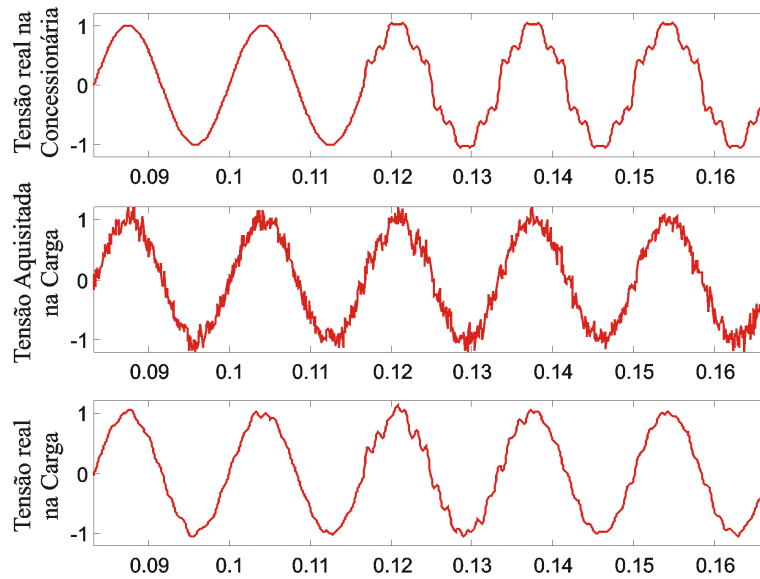


Figura 3.38 – Simulação do Algoritmo de Compensação de Harmônicos, em Malha Fechada, com adição de Ruído Branco

Mantendo-se as mesmas condições, apenas substituindo o ruído branco por ruído de alta frequência, gerou-se a **Figura 3.39**.

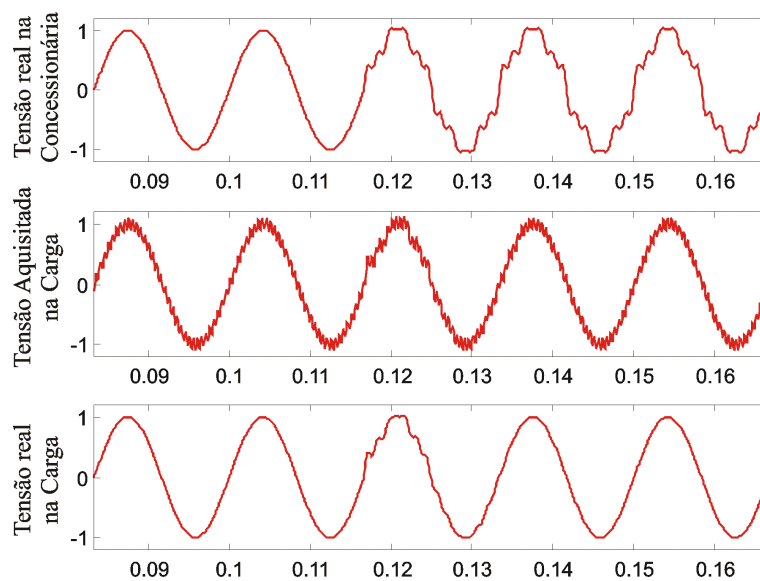


Figura 3.39 – Simulação do Algoritmo de Compensação de Harmônicos, em Malha Fechada, com adição de Ruído de Alta Frequência

Nota-se, tanto na **Figura 3.38**, como (e principalmente) na **Figura 3.39**, que a duração do transitório foi de cerca de 1 ciclo de 60Hz. Portanto a aproximação feita no modelo da Figura 3.31 está boa.

3.2.5.3 – Compensação de Voltage Sags em Malha Fechada

A compensação de Voltage Sags segue o mesmo raciocínio da compensação de harmônicos. No caso da compensação de harmônicos, desejava-se eliminar todas as componentes harmônicas, não importando se eram componentes de seqüência positiva, negativa ou zero. No caso da compensação de Voltage Sags, deseja-se eliminar as componentes de seqüência negativa na freqüência fundamental e de seqüência zero na freqüência fundamental. E também é necessário acrescentar uma parte à componente de seqüência positiva na freqüência fundamental.

Para eliminar as componentes de seqüência negativa e zero, utilizam-se as mesmas estruturas já utilizadas para a compensação de harmônicos. A única diferença é o argumento dos senos e co-senos das Transformações direta e inversa de Park, que para a compensação de Sags, deverão variar na freqüência angular da fundamental.

Para a compensação da componente de seqüência positiva é necessário que se saiba como deve estar a onda de tensão (com relação à fase e amplitude) naquele instante. Isto é conseguido através de uma PLL (ver **Apêndice B**).

De acordo com as equações (Eq. 3.15) e (Eq. 3.16), a informação de fase é dada pelo sinal no eixo **d** de seqüência positiva e a informação de amplitude é dada pelo sinal no eixo **q** de seqüência positiva. Deve-se, portanto utilizar mais um bloco de transformada direta de Park de seqüência positiva, para gerar os sinais **d** e **q** de referência. Conforme **Figura 3.40**.

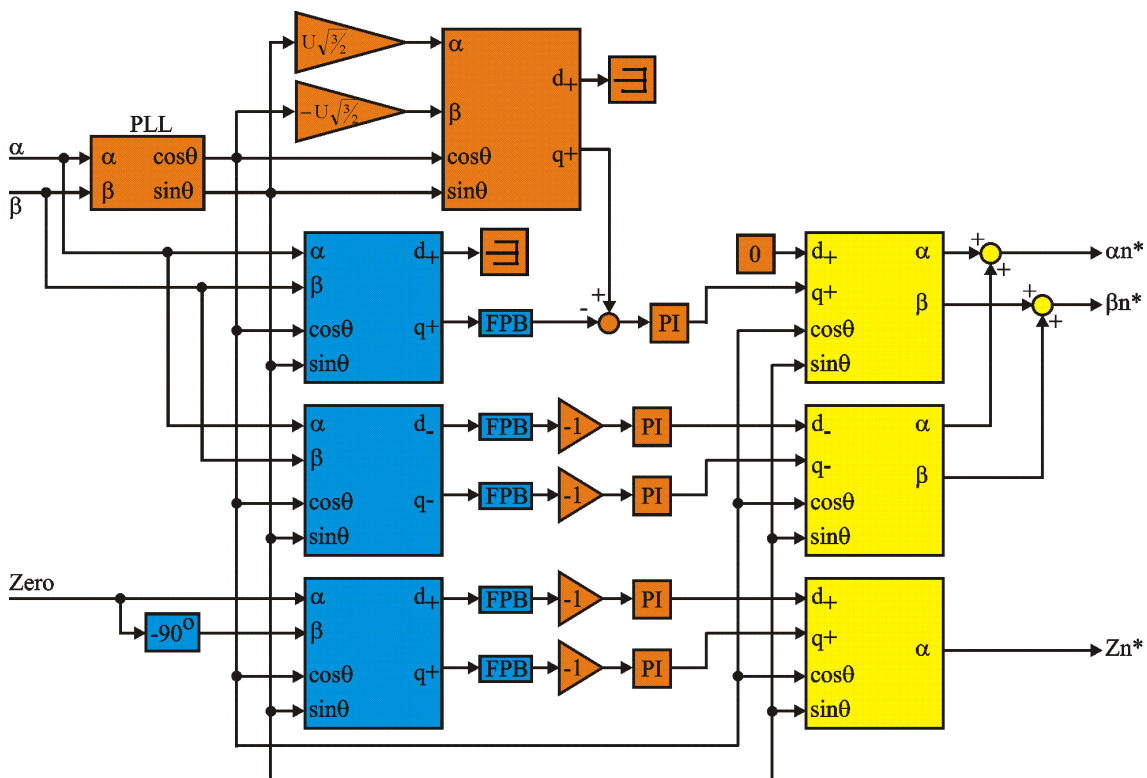


Figura 3.40 – Bloco de Compensação de Voltage Sags, em malha fechada

Nota-se, na **Figura 3.40**, que a entrada de referência para os sinais **d** e **q** de seqüência negativa e de seqüência zero é zero (ganho de -1 multiplicando a saída do filtro de médias). Para a compensação da seqüência positiva, a entradas de referência vêm da transformada de Park do sinal da PLL.

Realiza-se uma comparação com os sinais no eixo **dq** do sinal aqusitado. O regulador PI irá forçar o erro para zero, fazendo com que os valores de referência sejam atingidos. Nota-se, porém, que apenas o sinal do eixo **q** está sendo compensado; e que o sinal do eixo **d** está, aparentemente, em aberto.

Na verdade, este sinal não está em aberto. Ele está sendo corrigido pela malha interna da PLL. Caso a malha para este sinal fosse novamente fechada, a operação da PLL ficaria prejudicada.

A transformação de Park do sinal de referência precisa refletir o comportamento de um sistema trifásico, com as 3 amplitudes fixadas em um mesmo valor (**U**) e igualmente defasadas de 120°. O que poderia ser feito é, a partir da informação de ângulo da PLL, gerar três senoides nestas condições, realizar uma transformada de Clarke e uma de Park.

Isto, porém, geraria muitos cálculos desnecessários. Uma solução mais eficiente seria, a partir da informação de ângulo da PLL, gerar um seno e um co-seno, multiplicá-los por um ganho e realizar a transformada de Park diretamente com estes sinais. As equações (**Eq. 3.7**) e (**Eq. 3.8**) validam esta afirmação.

Pela (**Eq. 3.7**), e considerando que a PLL esteja funcionando corretamente, tem-se que o sinal **alfa** de referência será um ganho de  $U\sqrt{3/2}$  multiplicando o sinal de seno gerado pela PLL. Da mesma forma, pela (**Eq. 3.8**), tem-se que o sinal **beta** de referência será um ganho de  $-U\sqrt{3/2}$  multiplicando o sinal de co-seno gerado pela PLL. Esta simplificação fica melhor visualizada com a **Figura 3.41**.

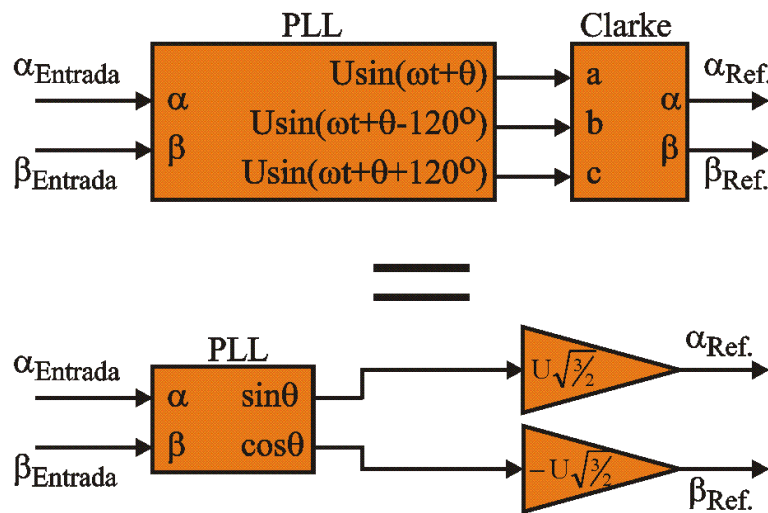


Figura 3.41 – Simplificação para se determinar os sinais **alfa** e **beta** de referência

Simulou-se o algoritmo da **Figura 3.40** para os sete tipos de Voltage Sags definidos no **Capítulo 2**. Em todas as simulações a seguir utilizou-se como entrada um sinal senoidal de 60Hz com amplitude 1pu. Em um determinado momento ocorre o Sag, com isto pode-se observar a duração do transitório.

### SAG do tipo A

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{2}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{2}V\sqrt{3}\right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 0.5 + j0$$

$$v_b = -0.25 - j0.433$$

$$v_c = -0.25 + j0.433$$

em coordenadas polares,

$$v_a = 0.5 \angle 0^\circ$$

$$v_b = 0.5 \angle -120^\circ$$

$$v_c = 0.5 \angle +120^\circ$$

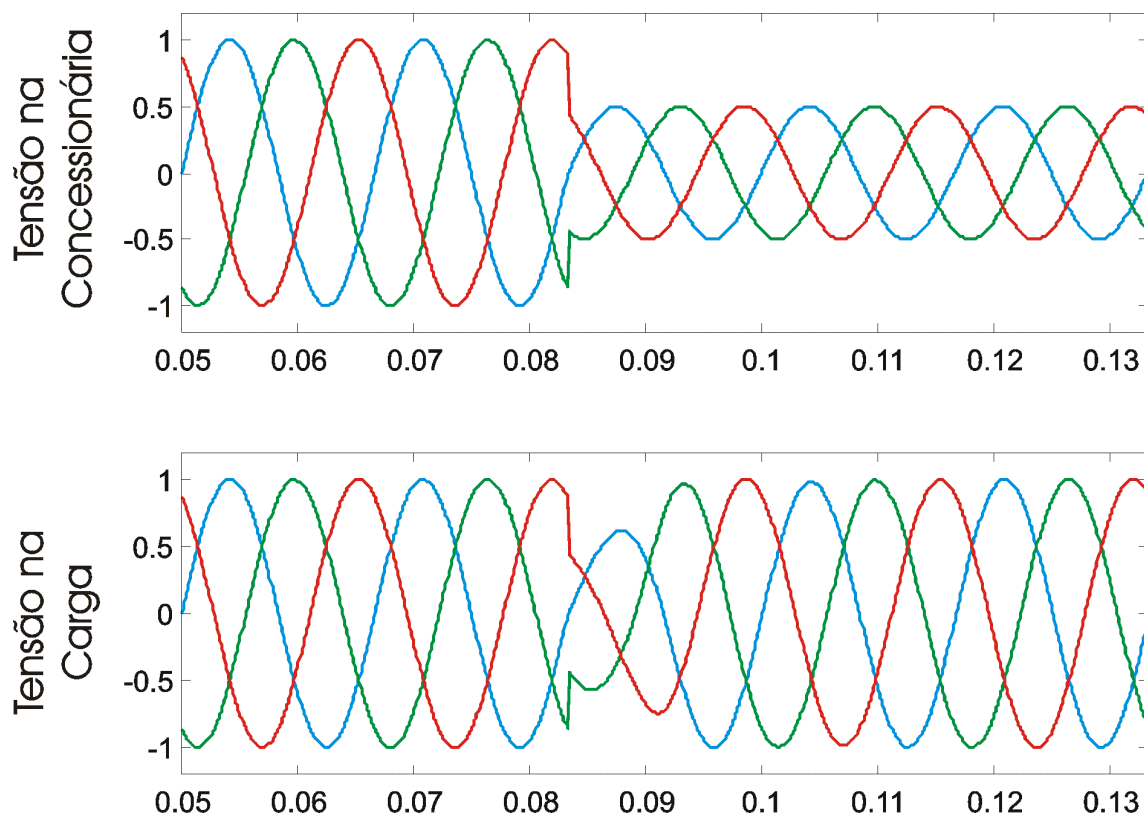


Figura 3.42 – Correção de SAG do Tipo A

**SAG do tipo B**

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}\right) - j\left(\frac{1}{2}\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}\right) + j\left(\frac{1}{2}\sqrt{3}\right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 0.5 + j0$$

$$v_b = -0.5 - j0.866$$

$$v_c = -0.5 + j0.866$$

em coordenadas polares,

$$v_a = 0.5 \angle 0^\circ$$

$$v_b = 1 \angle -120^\circ$$

$$v_c = 1 \angle +120^\circ$$

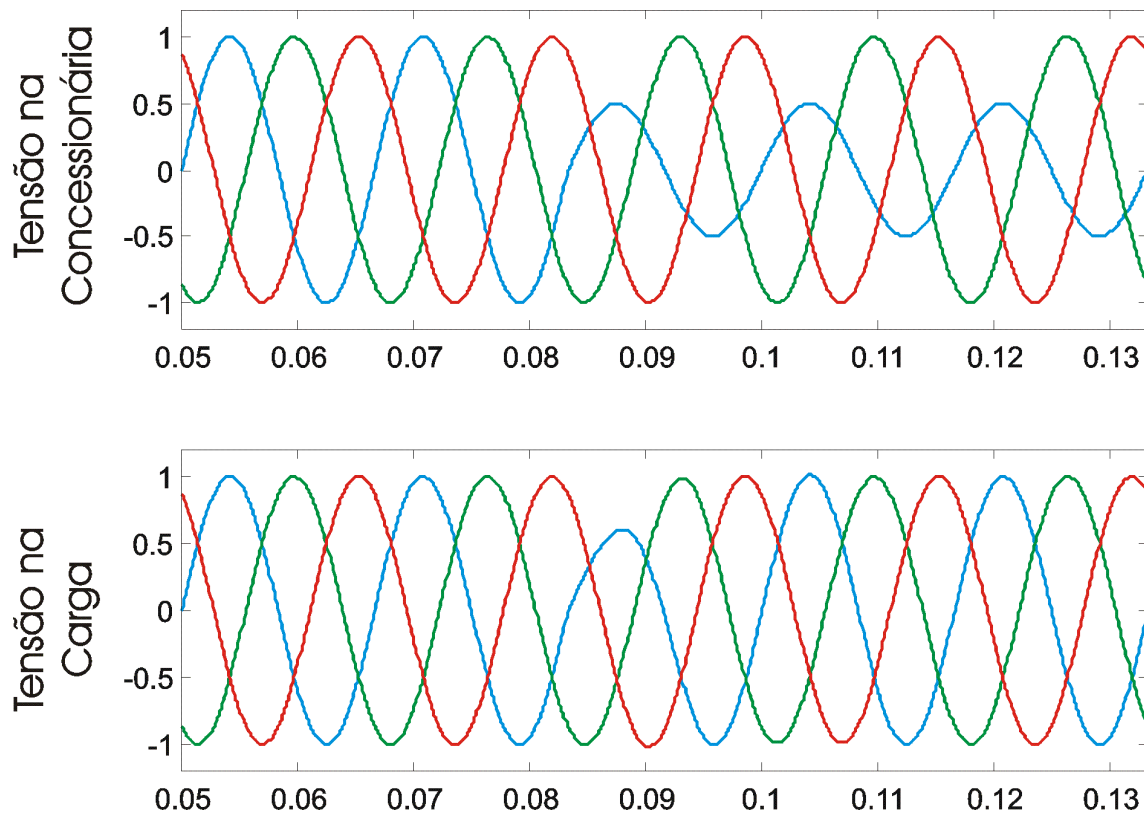


Figura 3.43 – Correção de SAG do Tipo B

**SAG do tipo C**

$$v_a = (1) + j(0)$$

$$v_b = \left(-\frac{1}{2}\right) - j\left(\frac{1}{2}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}\right) + j\left(\frac{1}{2}V\sqrt{3}\right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 1 + j0$$

$$v_b = -0.5 - j0.433$$

$$v_c = -0.5 + j0.433$$

em coordenadas polares,

$$v_a = 1 \angle 0^\circ$$

$$v_b = 0.661 \angle -139^\circ$$

$$v_c = 0.661 \angle +139^\circ$$

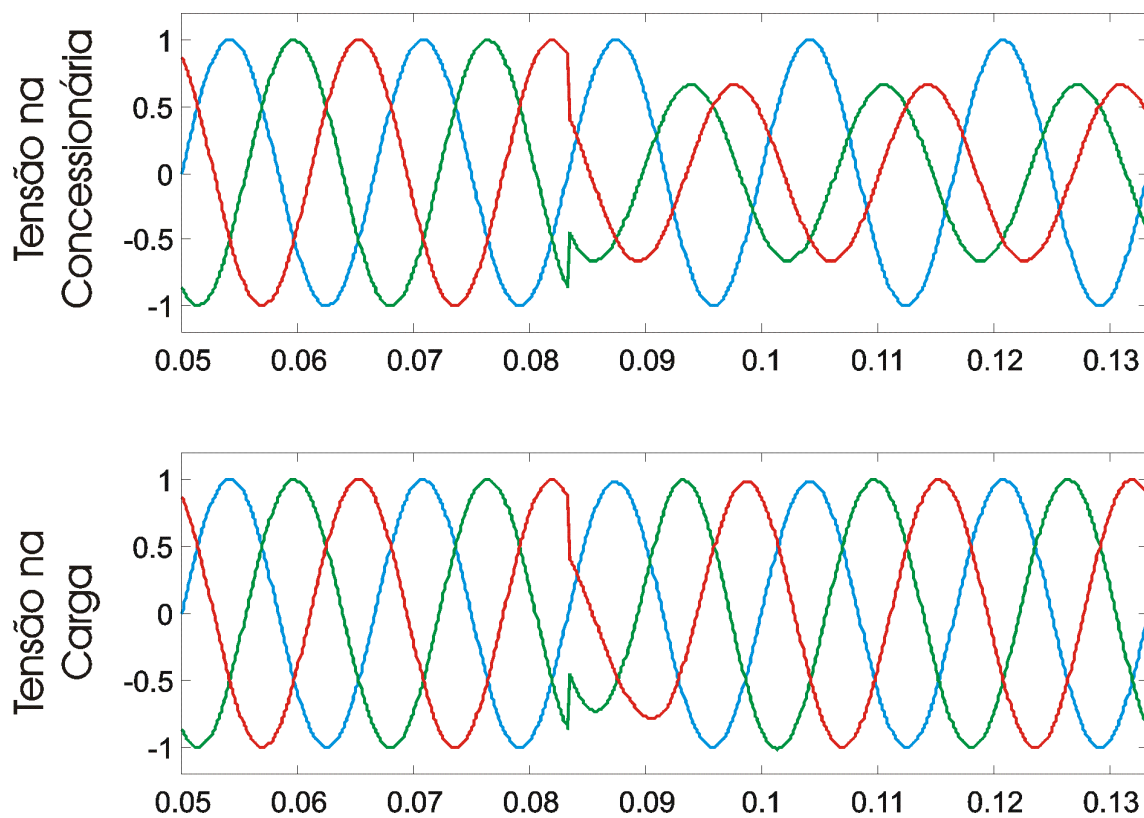


Figura 3.44 – Correção de SAG do Tipo C



**SAG do tipo D**

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{2}\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{2}\sqrt{3}\right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 0.5 + j0$$

$$v_b = -0.25 - j0.866$$

$$v_c = -0.25 + j0.866$$

em coordenadas polares,

$$v_a = 0.5 \angle 0^\circ$$

$$v_b = 0.901 \angle -106^\circ$$

$$v_c = 0.901 \angle +106^\circ$$

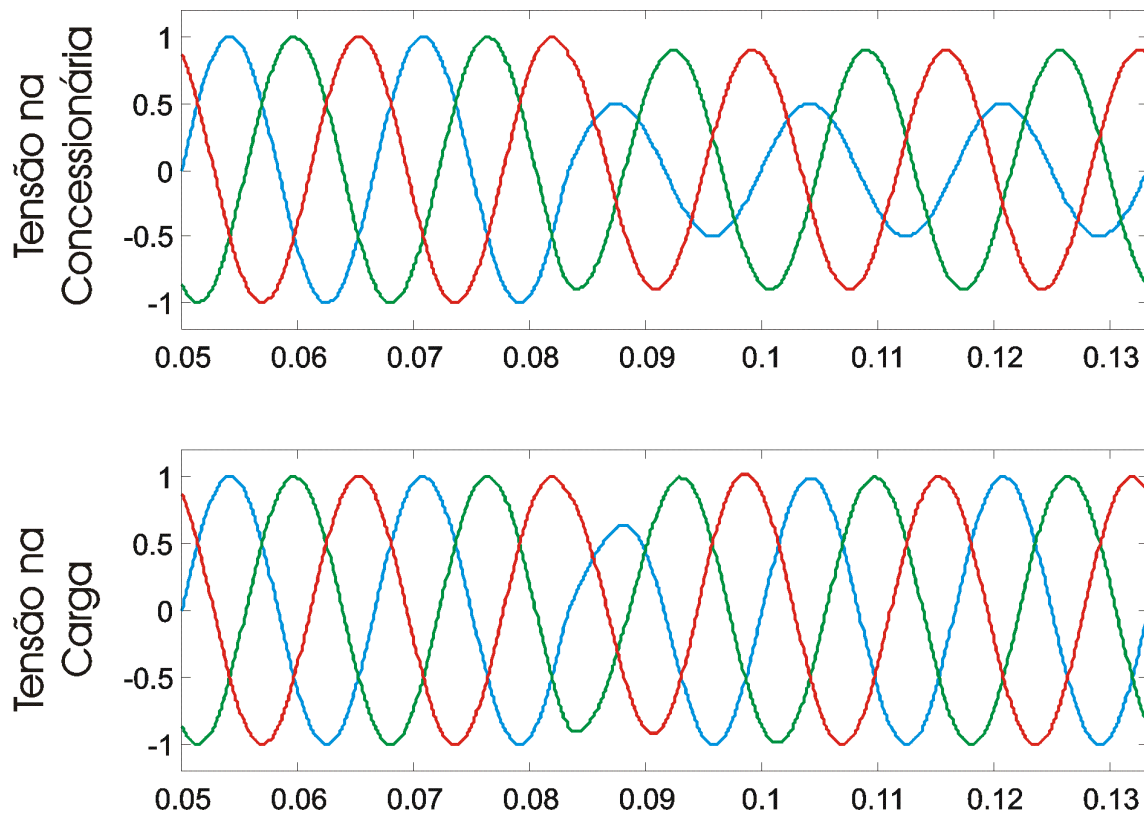


Figura 3.45 – Correção de SAG do Tipo D

**SAG do tipo E**

$$v_a = (1) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{2}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{2}V\sqrt{3}\right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 1 + j0$$

$$v_b = -0.25 - j0.433$$

$$v_c = -0.25 + j0.433$$

em coordenadas polares,

$$v_a = 1 \angle 0^\circ$$

$$v_b = 0.5 \angle -120^\circ$$

$$v_c = 0.5 \angle +120^\circ$$

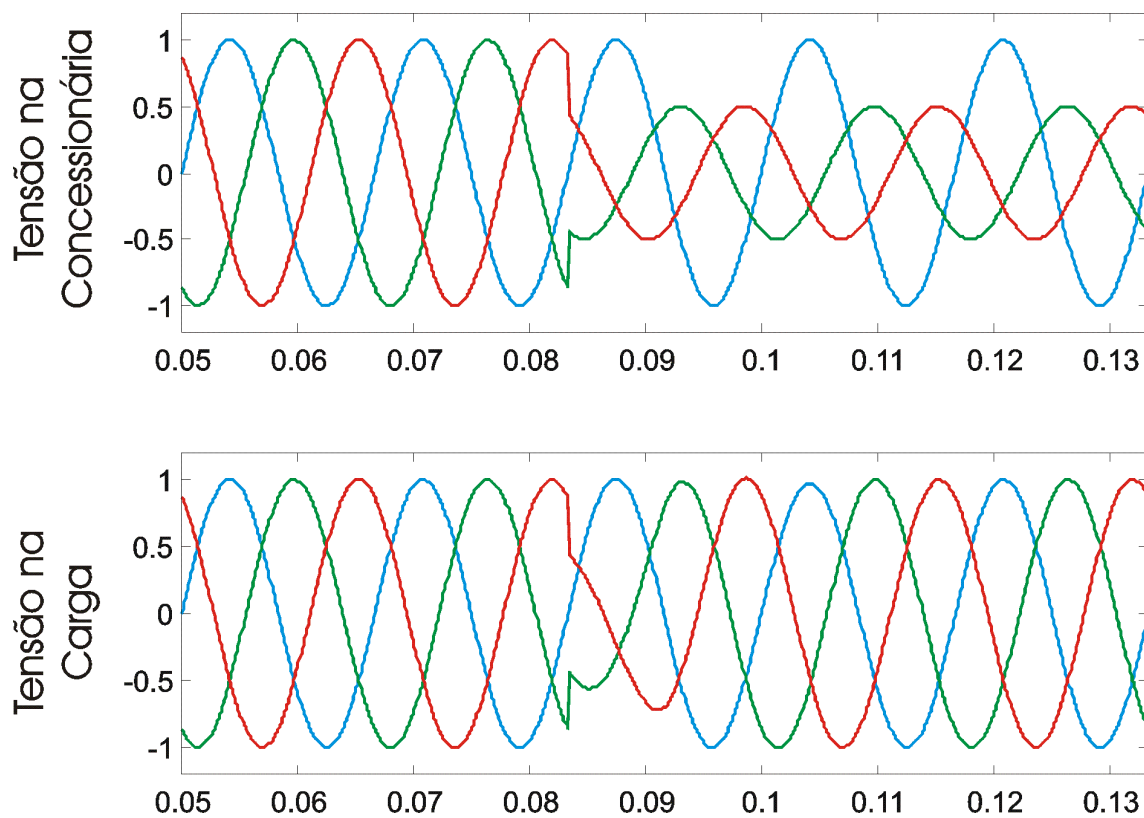


Figura 3.46 – Correção de SAG do Tipo E

**SAG do tipo F**

$$v_a = (V) + j(0)$$

$$v_b = \left(-\frac{1}{2}V\right) - j\left(\frac{1}{3}\sqrt{3} + \frac{1}{6}V\sqrt{3}\right)$$

$$v_c = \left(-\frac{1}{2}V\right) + j\left(\frac{1}{3}\sqrt{3} + \frac{1}{6}V\sqrt{3}\right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 0.5 + j0$$

$$v_b = -0.25 - j0.722$$

$$v_c = -0.25 + j0.722$$

em coordenadas polares,

$$v_a = 0.5 \angle 0^\circ$$

$$v_b = 0.764 \angle -109^\circ$$

$$v_c = 0.764 \angle +109^\circ$$

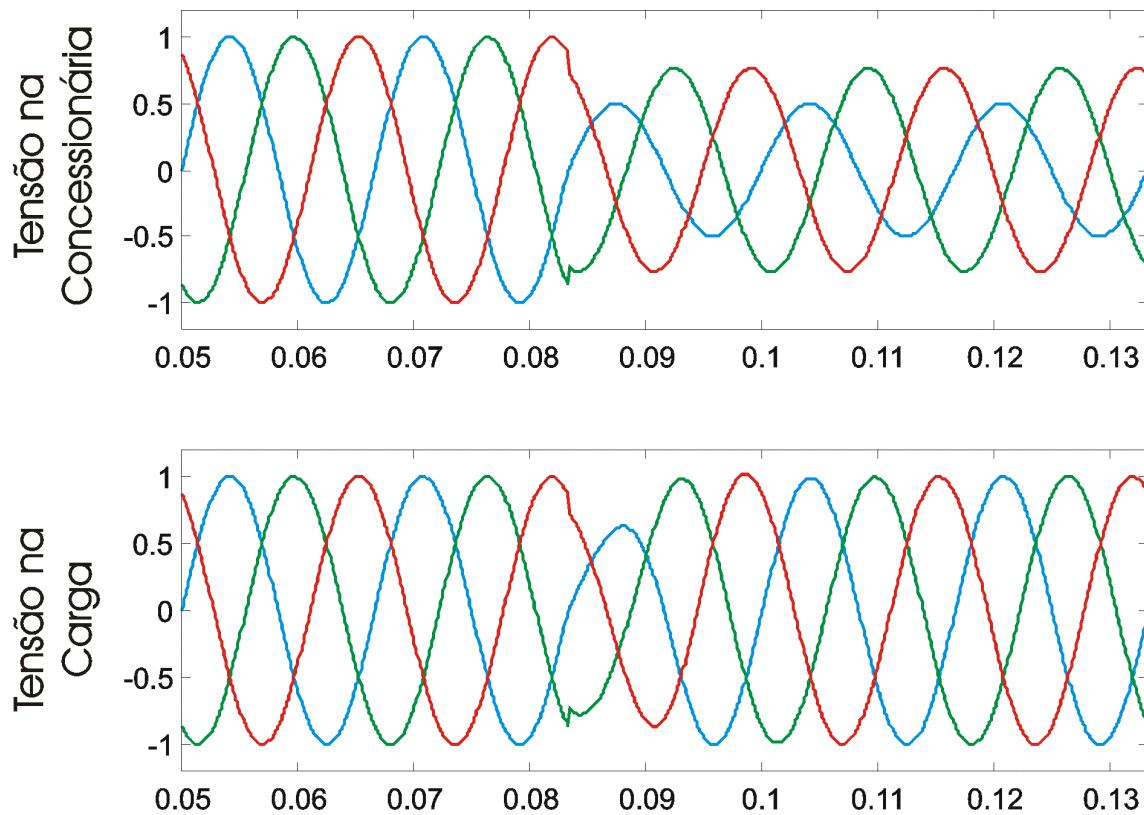


Figura 3.47 – Correção de SAG do Tipo F

**SAG do tipo G**

$$v_a = \left( \frac{2}{3} + \frac{1}{3}V \right) + j(0)$$

$$v_b = \left( -\frac{1}{3} - \frac{1}{6}V \right) - j\left( \frac{1}{2}V\sqrt{3} \right)$$

$$v_c = \left( -\frac{1}{3} - \frac{1}{6}V \right) + j\left( \frac{1}{2}V\sqrt{3} \right)$$

para  $V=0.5\text{pu}$ ,

$$v_a = 0.833 + j0$$

$$v_b = -0.417 - j0.433$$

$$v_c = -0.417 + j0.433$$

em coordenadas polares,

$$v_a = 0.833 \angle 0^\circ$$

$$v_b = 0.601 \angle -134^\circ$$

$$v_c = 0.601 \angle +134^\circ$$

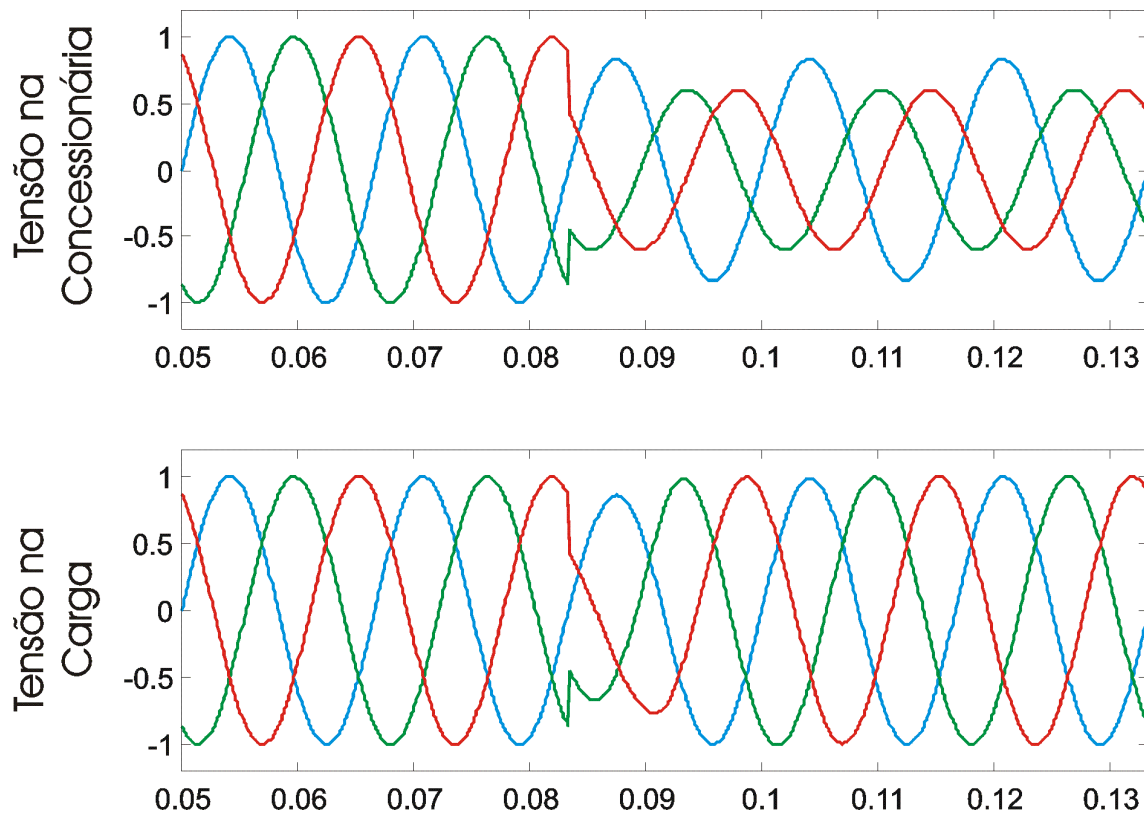


Figura 3.48 – Correção de SAG do Tipo G

Nota-se nas sete simulações anteriores que a duração do transitório foi em torno de 1 ciclo de 60Hz.

Existem equipamentos reguladores de tensão, como chaveadores de TAPs de transformadores, que apenas detectam que a tensão de uma determinada fase diminuiu e aplicam um ganho nesta tensão, fazendo-a voltar ao seu valor nominal. Nota-se nas **Figuras 3.44, 3.45, 3.47 e 3.48**, que para estes casos (SAGs do tipo **C, D, F e G**), esses reguladores não são eficazes, pois eles apenas corrigem o módulo da tensão e não corrigem a fase. Para estes casos apenas os Compensadores Ativos podem resolver o problema.

A **Figura 3.49** apresenta uma simulação para uma entrada com harmônicos e SAG do tipo B. Os harmônicos são desequilibrados, conforme Tabela 3.2. Adicionou-se ruído branco no sistema de aquisição de dados

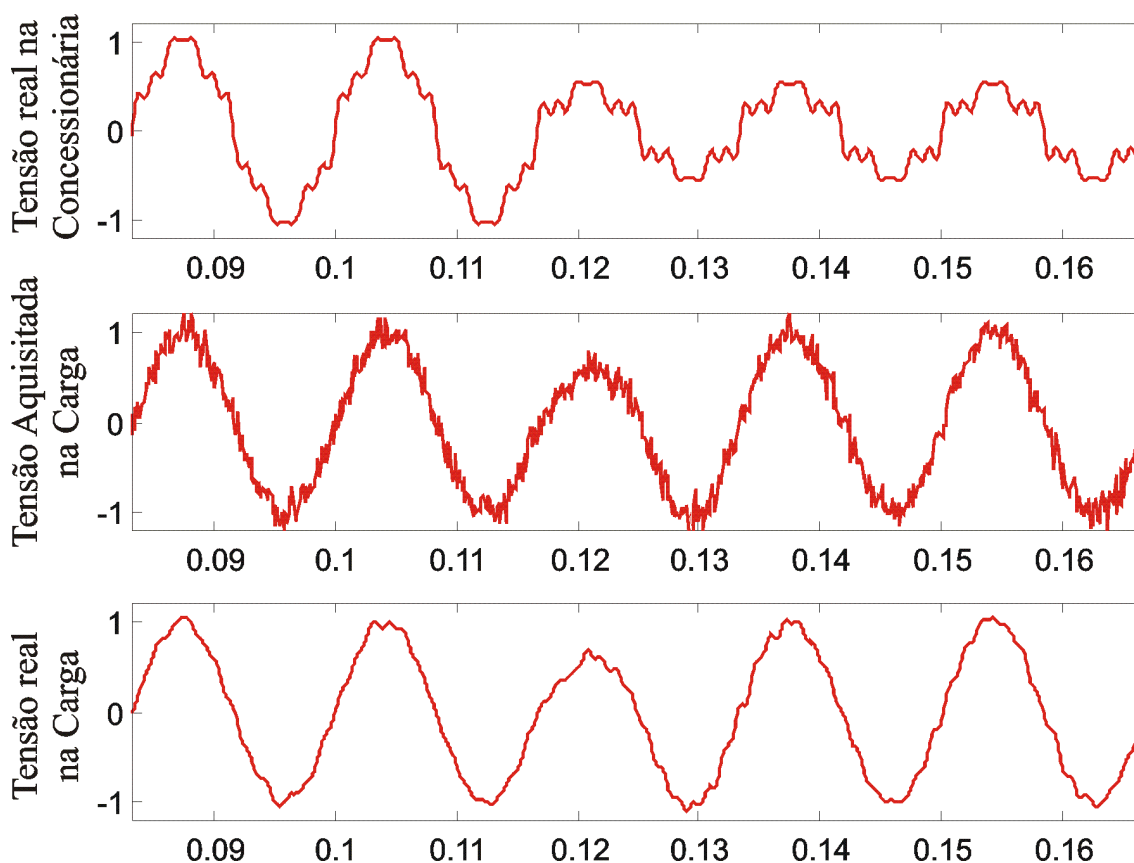


Figura 3.49 – Compensação de harmônicos e SAG, aquisição com ruído branco.

Nota-se, na **Figura 3.49**, um transitório em torno de 1 ciclo de 60Hz

A **Figura 3.50** apresenta a mesma simulação, porém substituindo o ruído branco por ruído de alta frequência.

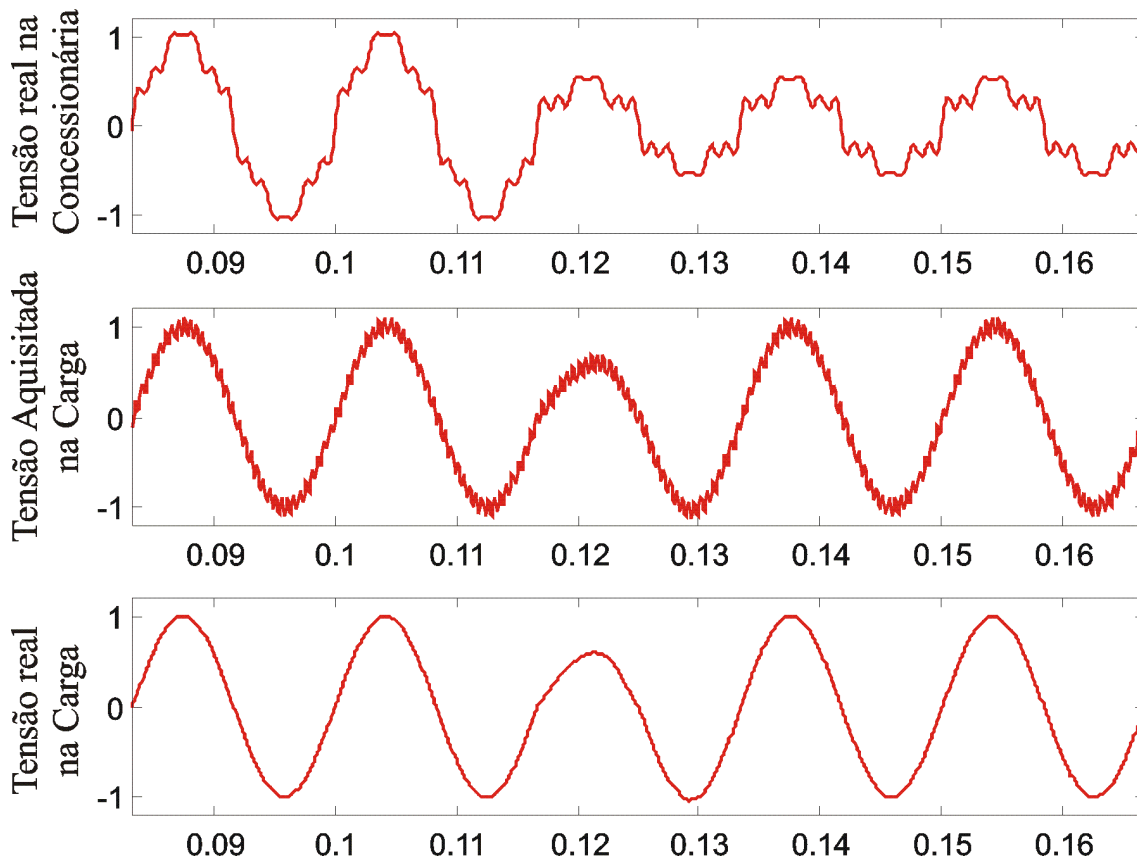


Figura 3.50 – Compensação de harmônicos e SAG, aquisição com ruído de alta frequência

Nota-se, **Figura 3.50**, um transitório em torno de 1 ciclo de 60Hz.

É interessante observar que em todas as simulações em malha fechada o transitório durou em torno de 1 ciclo, enquanto que em malha aberta durava apenas  $\frac{1}{2}$  ciclo. Porém, conforme foi explicado no início deste capítulo, certas perturbações podem ocorrer com o sinal de compensação após sua geração pelo algoritmo de controle. Por exemplo ripple no link DC dos inversores, não linearidades dos transformadores de acoplamento e perdas em geral.

Sendo assim, os algoritmos em malha aberta geram as compensações corretamente, mas como o problema está depois deles, e eles não tem qualquer tipo de realimentação, a tensão na carga não é compensada de forma satisfatória.

Como os algoritmos em malha fechada possuem realimentação do sinal de saída, caso surjam perturbações, elas são automaticamente corrigidas (dentro de um certo tempo).

Para ilustrar uma situação de perdas, inseriu-se um ganho de 0.5 entre a saída do compensador e o ponto de soma do sinal de compensação com o sinal não compensado. Este ganho foi colocado apenas na compensação da fase A, para criar uma situação de desequilíbrio. Simulou-se esta situação para uma entrada com os harmônicos desequilibrados da **Tabela 3.2** e com um SAG do tipo B. Para que as figuras ficassem mais legíveis, não foi inserido ruído na simulação, embora o algoritmo funcionasse da mesma forma se isto fosse feito.

A **Figura 3.51** apresenta o resultado desta simulação.

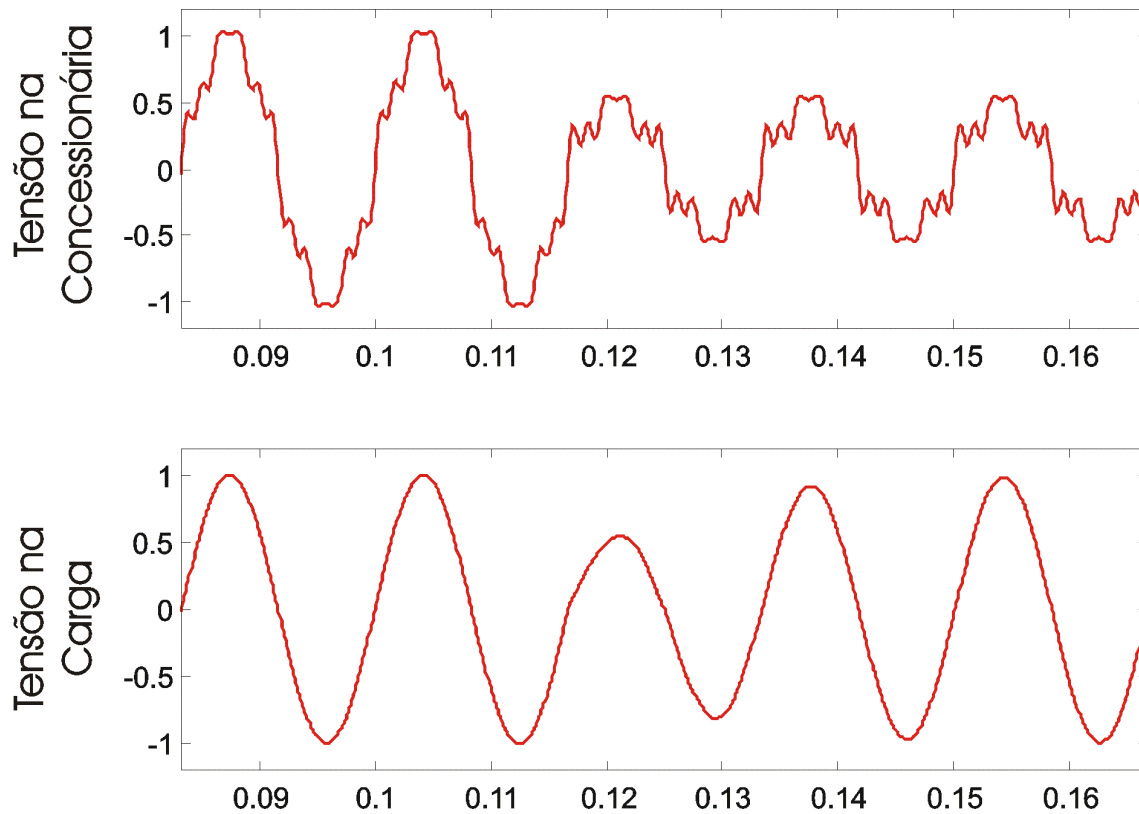


Figura 3.51 – Compensação de harmônicos, SAGs e perturbações

Nota-se, na **Figura 3.51**, que a tensão na carga foi compensada da mesma forma que anteriormente, mesmo com a perda inserida propositalmente após o compensador.

A **Figuras 3.52 e 3.53** mostram o sinal de compensação que o Compensador precisa gerar para vencer a perda (além dos harmônicos e do SAG) e como este sinal chega até o ponto de compensação. Deve-se ressaltar que o sinal chega ao ponto de compensação da maneira correta como deveria chegar, o sinal gerado pelo compensador que é alterado de forma a corrigir as perturbações que por ventura venham a ocorrer.

Para facilitar a visualização, a **Figura 3.52** mostra uma ampliação da região que necessita compensação de SAG. Conforme pode-se observar existe a componente de 60Hz e as componentes harmônicas neste sinal

E a **Figura 3.53** mostra uma ampliação da região em que o SAG ainda não ocorreu. Conforme se pode observar, apenas as componentes harmônicas precisam ser compensadas (e a perda também!), não existe componente de 60Hz neste sinal.

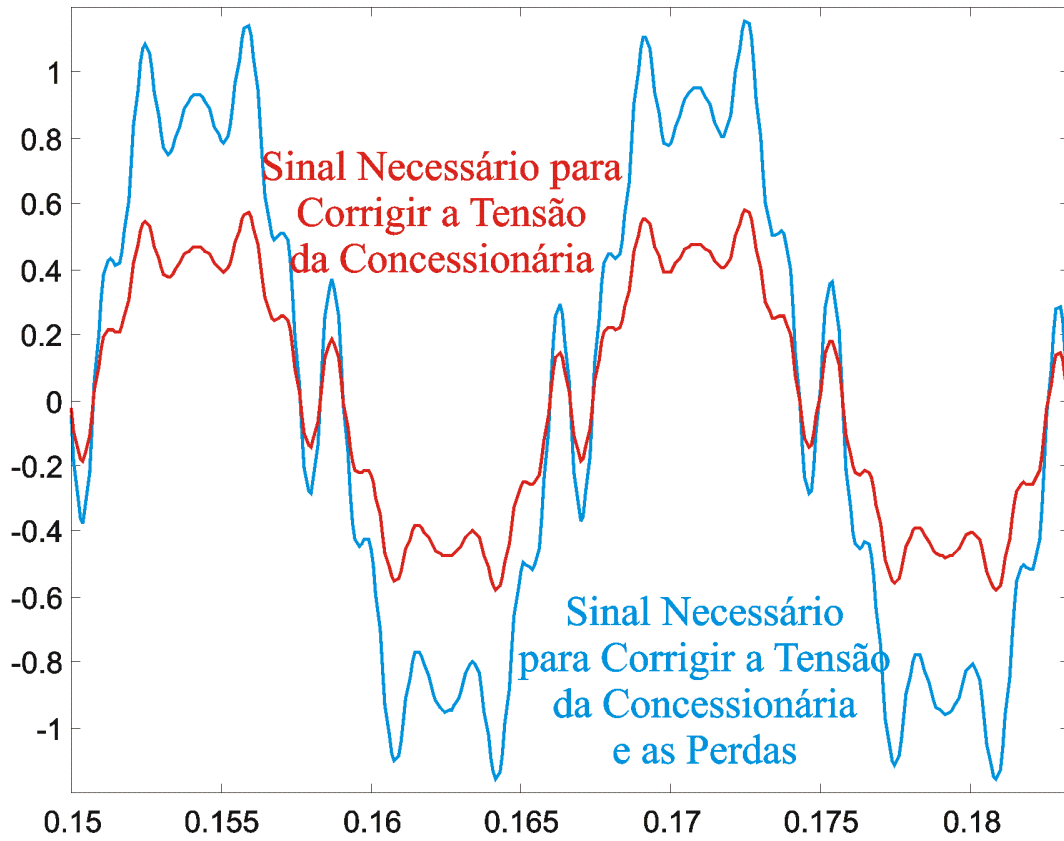


Figura 3.52 – Detalhe da região em que ocorre o SAG

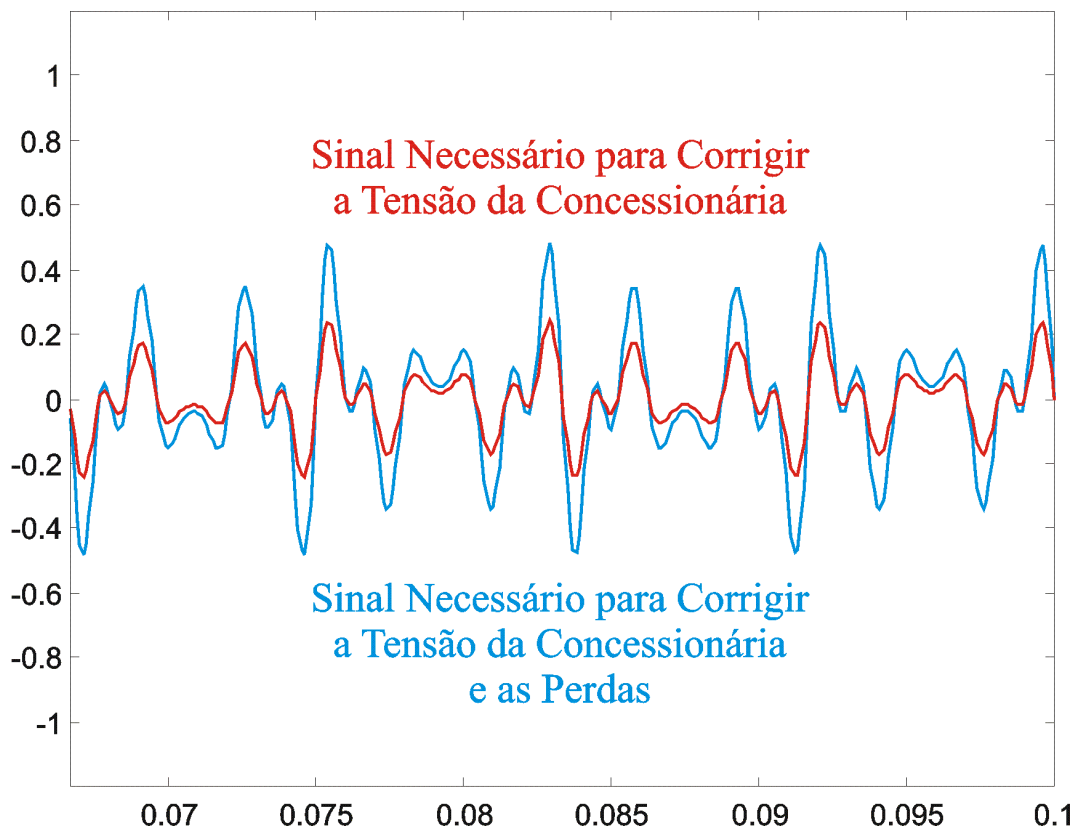


Figura 3.53 – Detalhe da região em que o SAG ainda não ocorreu



Portanto, mesmo apresentando a desvantagem de um transitório mais lento, o algoritmo em malha fechada é mais confiável, pois o regulador está a todo o momento monitorando a tensão da carga e deixando-a igual à referência.

Este tempo de transitório de 1 ciclo pode ser reduzido, com a utilização de reguladores melhor elaborados. Deve-se ressaltar, porém, que menos do que  $\frac{1}{2}$  ciclo de transitório, para compensadores de SAG é um tempo impossível de ser conseguido, mesmo em malha aberta. Pois durante um SAG desequilibrado surge uma componente de seqüência negativa na frequência fundamental. E, esta componente irá aparecer no referencial **dq** como um sinal de 120Hz.

Para compensação apenas de harmônicos este tempo pode ser reduzido.

### 3.3 - Referências

[3.1] L.E.L.Oliveira, L.E.B.Silva, V.F.Silva, G.L.Torres, J.O.P.Pinto, “Improving the dynamic response of active power filters based on the synchronous reference frame method”, in *Proceedings of the APEC*, vol. 2, March 2002, pp. 742-748.

[3.2] V.Souares, P.Verdelho, G.Marques, “Active Power Filter Control Circuit based on the Instantaneous Active and Reactive  $i_d$ - $i_q$  Method”, in *Proceedings of the PESC*, 1997, pp.1096-1101.

[3.3] C.L.Phillips, R.D.Harbor, “Sistemas de Controle e Realimentação” , MAKRON Books do Brasil Editora Ltda, São Paulo, 1997.

[3.4] K.Ogata, “Engenharia de Controle Moderno” , terceira edição, LTC – Livros Técnicos e Científicos Editora S.A., Rio de Janeiro, 2000.

# Capítulo 4

## Equipamento Desenvolvido

<b>4.1 - Introdução</b> .....	4 - 3
<b>4.2 – Condicionamento de Sinais</b> .....	4 - 5
<b>4.3 – VSI e Modulação PWM</b> .....	4 - 8
<b>4.3.1 – Modulação PWM</b> .....	4 - 10
4.3.1.1 – Modulação Bipolar.....	4 - 11
4.3.1.2 – Modulação Unipolar .....	4 - 13
4.3.1.3 – Modulador Desenvolvido .....	4 - 17
4.3.1.3.1 – Gerador de Portadora Triangular.....	4 - 18
4.3.1.3.2 – Comparador Digital.....	4 - 20
4.3.1.3.3 – Interface PWM X VSI.....	4 - 21
4.3.1.3.4 – Barramento .....	4 - 23
<b>4.4 – Filtro Passivo</b> .....	4 - 25
<b>4.5 – Fotografias do Equipamento Desenvolvido</b> .....	4 - 30



## 4.1 - Introdução

O algoritmo de controle em malha fechada do Compensador Ativo Série, descrito em detalhes no **Capítulo 3**, foi implementado para correção de Voltage SAGs e harmônicos de tensão em sistemas de distribuição, em **13800V**.

Este projeto foi realizado com recursos do Programa de Pesquisa e Desenvolvimento da Companhia Energética do Maranhão (**CEMAR**). No projeto, construiu-se um **UPQC** (Universal Power Quality Conditioner), ou seja, um Compensador Série e um Compensador Shunt, para compensação de tensões e correntes. Esta dissertação de mestrado descreve apenas o funcionamento do Compensador Série.

A **Figura 4.1** ilustra a representação unifilar do Compensador Ativo Série, e sua instalação no sistema.

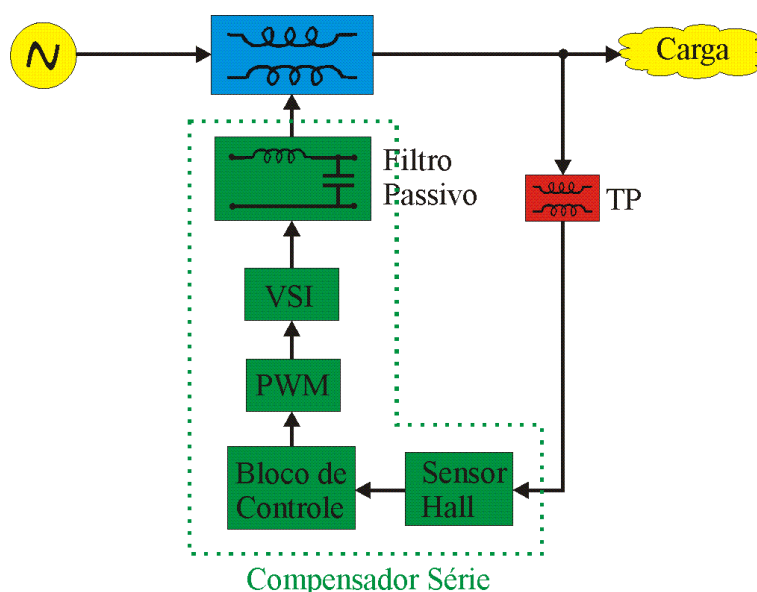


Figura 4.1 – Diagrama Unifilar do Compensador Série

Conforme explicado no **Capítulo 3**, o Compensador Série é um equipamento “inteligente” e, portanto, depende de um Computador Digital para funcionar. Este computador está representado na **Figura 4.1** pelo **Bloco de Controle**. Entretanto, um computador é um equipamento que realiza cálculos baseando-se em um sistema de numeração binário. Portanto, para que ele possa “entender” o que se passa com o sinal de tensão na carga, que é um sinal analógico, é necessário que ele esteja equipado com uma placa de aquisição de dados.

O computador utilizado foi um Embedded PC PCM-4823, da Advantech. Este computador é equipado com um processador 586 133MHz (performance equivalente à de um Intel 486 DX4 100MHz); e possui, praticamente, todos os periféricos (Ethernet, IDE, SVGA, portas serial e paralela, teclado e mouse) de um computador desktop do tipo IBM PC compatível, com a diferença de possuir suas dimensões reduzidas (145mm X 102mm). Foram projetados em dimensões reduzidas para o uso em sistemas embarcados, como por exemplo, computadores de bordo em automóveis.

A placa de aquisição de dados utilizada é uma PCM-3718H, também da Advantech. Esta placa permite amostrar até 16 canais em até 100kHz com resolução de 12 bits. Esta placa também é equipada com 2 portas de I/O digital de 8 bits.

Tanto a CPU quanto a placa de aquisição de dados estão representadas na **Figura 4.2**.

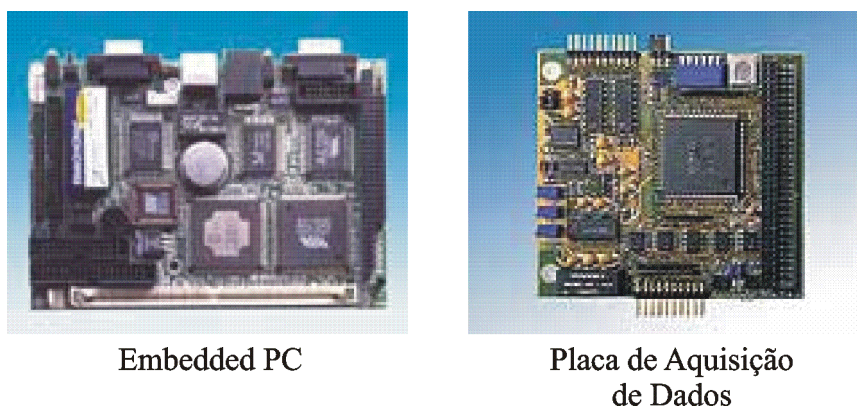


Figura 4.2 – Hardware do Bloco de Controle

De acordo com a **Figura 4.1**, a tensão na carga é monitorada através de 3 TPs com relação de  $13800$  para  $115/\sqrt{3}$ , ligados em estrela com neutro aterrado (na alta) e em delta (na baixa).

Para compatibilizar as tensões de saída dos TPs com as tensões de entrada da placa de aquisição de dados, foram utilizados sensores por efeito Hall. Além da compatibilização de tensões, estes sensores isolam galvanicamente o circuito de potência do circuito eletrônico.

Pela saída digital da PCM-3718H, o sinal de compensação é enviado, em 8 bits, para um modulador PWM. O modulador PWM gera os pulsos de chaveamento, que excitam o VSI.

O VSI amplifica esses pulsos em termos de potência.

O filtro passivo na saída do VSI elimina as componentes de altas frequências, tendo como resultado da filtragem o sinal de compensação.

Para acoplamento do compensador entre a concessionária e a carga, utiliza-se um transformador.

Para compensação de baixas tensões, este transformador pode ter relação de transformação **1:1**.

O objetivo deste trabalho é a compensação em 13.8kV. O problema é que a tensão máxima que IGBTs suportam é 1200V. Uma solução com inversores multiníveis até poderia ser implementada, porém isto complicaria demais o projeto, e, financeiramente, seria inviável.

A solução encontrada foi utilizar um transformador com relação de transformação **220:13800**. Desta forma, IGBTs de baixa tensão (e baixo custo) podem ser utilizados no VSI.

As próximas seções descrevem os blocos constituintes do diagrama da **Figura 4.1**.

## 4.2 – Condicionamento de Sinais

A placa de aquisição de dados trabalha com uma tensão de entrada de -5V até +5V. Sendo assim, é necessária a utilização de um circuito entre a CPU e os TPs.

Utilizou-se, para este fim, um circuito com sensores por efeito Hall. Estes sensores, além de converterem a tensão de  $115/\sqrt{3}$  RMS (portanto de  $-115 \cdot \sqrt{2}/\sqrt{3}$  até  $+115 \cdot \sqrt{2}/\sqrt{3}$ ) da saída dos TPs para um sinal de -15V até +15V, também isolam o circuito de potência do circuito eletrônico. Sendo assim, distúrbios no sistema de potência, que poderiam danificar a CPU, não se propagam.

Os sensores utilizados são do tipo **LV25-P**, fabricados pela LEM. Na verdade, estes dispositivos são sensores de correntes, porém, com a utilização de uma resistência (de valor apropriado) em serie com a entrada, pode-se conhecer o valor da tensão da entrada. Este processo pode ser visualizado na **Figura 4.3**.

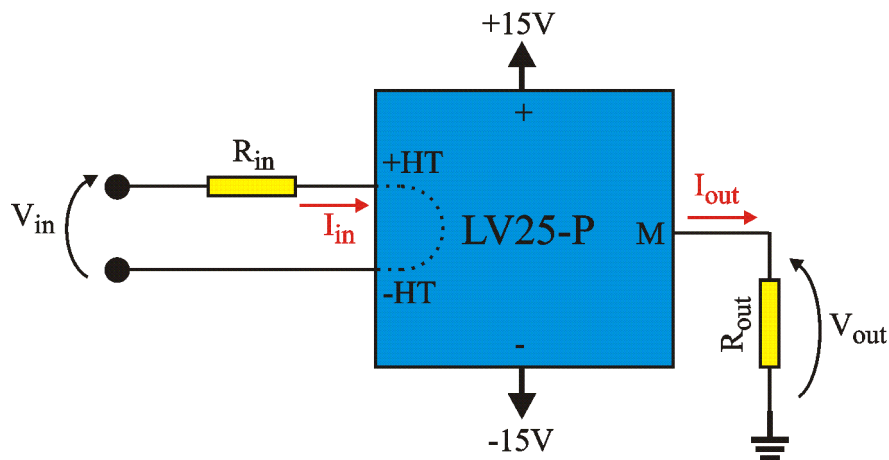


Figura 4.3 – Sensor Hall

A resistência de entrada deve ser calculada de forma que  $I_{in}$  fique em torno de 10mA RMS para a tensão de entrada nominal, conforme a equação (Eq. 4.1).

$$R_{in} = \frac{V_{in}}{10 \cdot 10^{-3}} = 100 \cdot V_{in} \quad (\text{Eq. 4.1})$$

Nota-se que essa resistência precisa suportar uma grande dissipação de potência. Sendo assim, ela precisa ser corretamente dimensionada, conforme a equação (Eq. 4.2).

$$P_{R_{in}} = \frac{V_{in}^2}{R_{in}} \quad (\text{Eq. 4.2})$$

A saída deste sensor também é em corrente, sendo é necessário uma conversão para tensão. Esta conversão é realizada por um resistor shunt ( $R_{out}$ ) na saída do circuito. A relação de transformação do sensor é de **10mA : 25mA**. Sendo assim, pode-se calcular  $R_{out}$  conforme a equação (Eq. 4.3).

$$R_{out} = \frac{V_{out}}{25 \cdot 10^{-3}} = 40 \cdot V_{out} \quad (\text{Eq. 4.3})$$

Deve-se, porém, manter  $R_{out}$  dentro do intervalo entre 100 e 350 ohms.

A tensão nominal de saída do TP é  $115/\sqrt{3}$ , sendo assim, para o cálculo de  $R_{in}$ :

$$R_{in} = 100 \cdot 115 / \sqrt{3} = 6639.53$$

O valor comercial mais próximo será  $6.8k\Omega$ . E a potência dissipada será:

$$P_{R_{in}} = \frac{(115/\sqrt{3})^2}{6.8 \cdot 10^3} = 0.648$$

Portanto deve-se utilizar um resistor de 1W ou superior.

Para um resistor  $R_{out}$  de valor  $330\Omega$ , a tensão de saída será

$$V_{out} = \frac{115/\sqrt{3}V}{6.8 \cdot 10^3 \Omega} \cdot \frac{25mA}{10mA} \cdot 330\Omega = 8.055V \text{ RMS, portanto de } -11,4V \text{ até } +11,4V \text{ de pico.}$$

Como a faixa dinâmica da entrada da placa de aquisição de dados é de -5V até +5V e a da saída do circuito dos sensores Hall é de -11,4V até +11,4V, deve-se utilizar um circuito com amplificadores para compatibilizar estas tensões. Deve-se ressaltar que pode ocorrer um SWELL (ver **Capítulo 2**) na tensão da concessionária, sendo assim, a tensão do Hall poderá variar de -15V até +15V, que é sua tensão de alimentação. Além dessa faixa o Hall entra em saturação nestes valores.

**Da equação (Eq. 4.3), nota-se que para um  $V_{out}$  RMS de  $5/\sqrt{2}$ , bastaria que  $R_{out}$  fosse igual à  $141,42 \Omega$ . Porém, em um ambiente ruidoso é interessante trabalhar com tensões mais altas, sendo assim, utilizou-se o valor máximo de resistência permitido.**

A utilização de um circuito com amplificadores apresenta a vantagem de realizar um melhor condicionamento no sinal; pois se pode controlar a impedância de entrada dos amplificadores, fazendo com que elas se tornem baixas o suficiente para não degradar o sinal de saída do Hall e, ao mesmo tempo, forçar uma circulação de corrente, evitando a contaminação por ruídos. A **Figura 4.4** ilustra um circuito amplificador diferencial.



Optou-se pela configuração diferencial pelo motivo desta ser mais imune a ruídos. Se o sinal for contaminado com ruído antes de chegar ao amplificador operacional (AmpOp) e se este ruído for igual nas duas entradas ele será eliminado, pois a tensão que vai ser amplificada é a subtração entre o potencial das entradas inversora e não inversora.

A relação entre as resistências do circuito da **Figura 4.4** define o ganho.  $R_1$  deve ser igual a  $R_2$  e  $R_3$  deve ser igual a  $R_4$ . E o ganho é dado por:  $R_3/R_1$ , então para um ganho de  $5/15$ , pode-se utilizar  $R_1=R_2=1500\Omega$  e  $R_3=R_4=510\Omega$ .

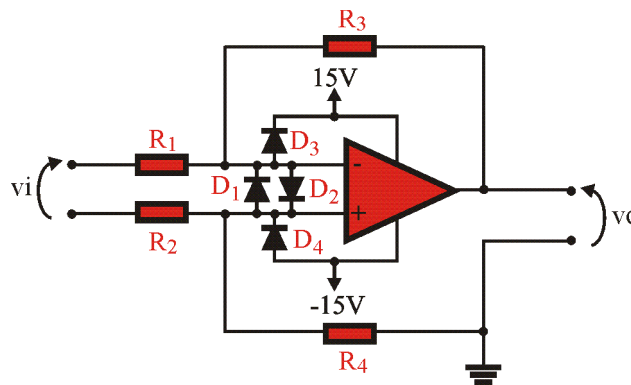


Figura 4.4 – Amplificador Diferencial

Pode-se notar, na **Figura 4.4**, a presença de alguns diodos. Estes diodos têm a função de proteger as entradas do AmpOp.  $D_1$  e  $D_2$  protegem contra sobre-tensão de modo diferencial, ou seja estes diodos não permitem que a diferença de tensão entre as entradas seja maior do que  $0,7V$ .  $D_3$  e  $D_4$  protegem contra sobre-tensão de modo comum, ou seja  $D_3$  não permite que a tensão nas entradas seja maior do que  $15V$  com relação ao terra e  $D_4$  não permite que a tensão nas entradas seja menor do que  $-15V$  com relação ao terra.

### 4.3 – VSI e Modulação PWM

Baseado na informação da tensão na carga, a CPU deve gerar uma tensão de compensação (de 0 a 220V) para mantê-la senoidal e com valor eficaz constante. Porém a CPU não é nada mais do que um circuito digital, sendo assim, ela trabalha com sinais binários, sob a forma de presença ou ausência de tensão.

Portanto, primeiramente, esta seqüência de 0s e 1s precisaria ser convertida para um sinal analógico e depois amplificada. O problema desta técnica é que não existem amplificadores capazes de fornecer a alta potência necessária para o equipamento.

Sendo assim, uma outra técnica deve ser utilizada. Será utilizado um método de modulação **PWM** (Pulse Width Modulation), cujo resultado é um trem de pulsos modulados em largura de pulso. Este trem de pulsos contém a informação do sinal modulante (referência de compensação), porém ainda é um trem de pulsos; ou seja, em determinados instantes há presença, e em outros há ausência de tensão.

Desta forma, pode-se utilizar um **VSI** (Voltage Source Inverter) para amplificar este trem de pulsos, pois se utiliza o trem de pulsos original (da CPU, com baixíssima capacidade de potência) apenas como comando de chaves controladas (**IGBTs** – Insulated Gate Bipolar Transistors). Essas chaves abrem e fecham um circuito DC de alta capacidade de potência, recriando o trem de pulsos original.

O trem de pulsos no circuito de potência representa a senóide modulante, porém ainda não é a própria senóide amplificada. Este trem de pulsos precisa ser demodulado, de forma que a informação modulante seja retirada. Um processo simples de demodulação para um sinal modulado em largura de pulsos é um **Filtro Passa-Baixas Passivo**. Este processo será ilustrado na **seção 4.4**.

Os **IGBTs** necessitam, portanto, chavear um circuito DC, que a partir daqui será chamado **link DC**. Este **link DC** poderia vir de um banco de baterias. O problema disto é que seria necessário um outro circuito para recarregar as baterias.

Uma forma mais simples de criar este **link DC** é o uso de um retificador. A grande desvantagem do circuito retificador em relação ao banco de baterias é que o **link DC** gerado poderá conter ripple. O ripple no **link DC** causa uma espécie de submodulação (com a frequência do ripple) no sinal de saída do **VSI** e, por conseguinte na tensão após o filtro passivo. Deve-se, portanto, construir um retificador com o menor ripple possível.

A forma utilizada neste projeto foi o uso de uma ponte retificadora trifásica com um grande capacitor para filtrar o ripple. Quanto maior for o valor do capacitor, menor será o ripple, entretanto a corrente de surto (no momento da entrada em operação do equipamento) nos diodos da ponte retificadora será maior. Sendo assim, deve-se escolher um valor de capacitância razoável para uma operação sem ripple e sem precisar de diodos muito caros.

Tem-se na **Figura 4.5** a representação do módulo VSI.

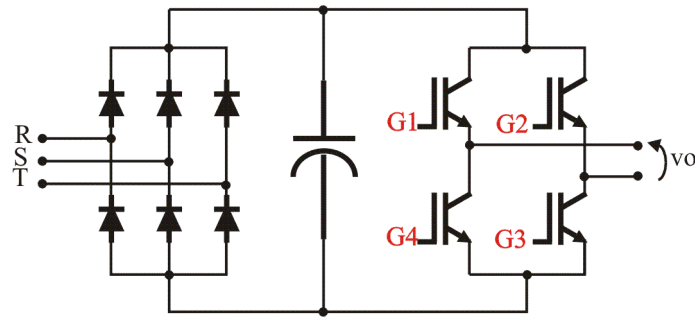


Figura 4.5 – Módulo VSI

Os sinais **G1**, **G2**, **G3** e **G4** são os pulsos de gatilho para os **IGBTs** da ponte. Um IGBT pode ser analisado como se fosse uma chave controlada. Um sinal em nível alto no gatilho do IGBT faz com que ele atue como uma chave fechada. Um sinal em nível baixo no gatilho do IGBT faz com que ele atue como uma chave aberta.

Da **Figura 4.5**, nota-se que **G1** e **G4** sempre devem ser sinais complementares. O mesmo vale para **G2** e **G3**. Isto deve ser feito para evitar um curto circuito no **link DC**.

O módulo Inversor (**VSI**) não precisou ser construído peça por peça. A empresa **Semikron** comercializa os módulos completos, com a ponte de IGBTs, retificador trifásico, banco de capacitores e drivers SKHI22A para os IGBTs. Utilizou-se o módulo B2U+B2CI, com IGBTs do tipo **SKM 75 GB 12 4 D**, para 75A/1200V. Tem-se na **Figura 4.6** uma fotografia deste módulo.

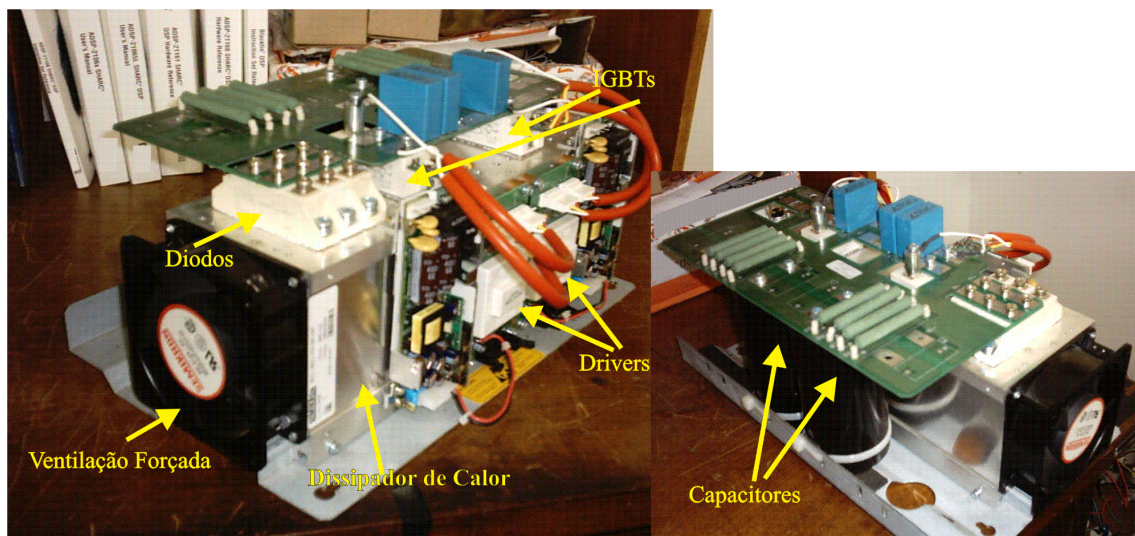


Figura 4.6 – Módulo VSI da Semikron

A próxima sub-seção explica os fundamentos da modulação PWM

### 4.3.1 – Modulação PWM

Modulação é um processo utilizado quando se deseja transmitir uma determinada informação através de um meio físico que impossibilitaria a transmissão direta desta informação. Em outras palavras, determinados meios físicos impossibilitam a transmissão direta de uma determinada informação; sendo assim, é necessário que esta informação seja modulada sobre um outro sinal (ou sinal de portadora). Cada sinal de portadora deve ser apropriado para um determinado meio físico.

Como exemplo, pode-se citar a transmissão de voz por rádio-frequência. O espectro de frequências da voz humana vai de 0Hz até 4kHz. Portanto o comprimento de onda da informação está na faixa dos 75km. Caso a transmissão utilizasse uma antena do tipo dipolo de meia-onda, o comprimento da antena deveria ser de 37.5km, o que torna a transmissão inviável. Sendo assim, utiliza-se um processo de modulação. O sinal de voz é utilizado para modular (imprimir suas características) sobre um sinal de portadora de alta frequência. Como a portadora possui alta frequência, a antena pode ter suas dimensões reduzidas; tornando, assim, viável o processo de transmissão da informação. Na recepção deve-se realizar o processo inverso, ou seja, uma demodulação para extrair o sinal de informação.

Com a modulação **PWM**, a idéia é a mesma. Tem-se como sinal modulante (informação) a referência de compensação, que deve ser amplificada para altos níveis de tensão e de corrente. O problema é que não existem amplificadores capazes deste tipo de amplificação. Para contornar este problema, o sinal modulante deve ser convertido para uma seqüência de pulsos. Estes pulsos podem ser amplificados, pois atuam como comandos para chaves de grande capacidade de potência. A onda resultante pode ser filtrada e a informação da senóide é, então, recuperada.

---

O processo de modulação **PWM** pode ser realizado de duas formas:

- Modulação Bipolar;
- Modulação Unipolar.

Basicamente os dois processos ocorrem da mesma forma, o sinal modulante é comparado com uma portadora triangular e o resultado desta comparação é o sinal modulado, ou seja, os pulsos de gatilho. A diferença entre os dois processos é a forma como a comparação é realizada e, por conseguinte o sinal modulado pelos dois processos. Conforme será explicado nas próximas **seções (4.3.1.1 e 4.3.1.2)** a modulação unipolar é um processo mais eficiente, porém sua implementação é um pouco mais complexa do que a bipolar.

4.3.1.1 – Modulação Bipolar

Tem-se na **Figura 4.7** e na **Figura 4.8** o princípio de funcionamento da geração de pulsos PWM com modulação Bipolar.

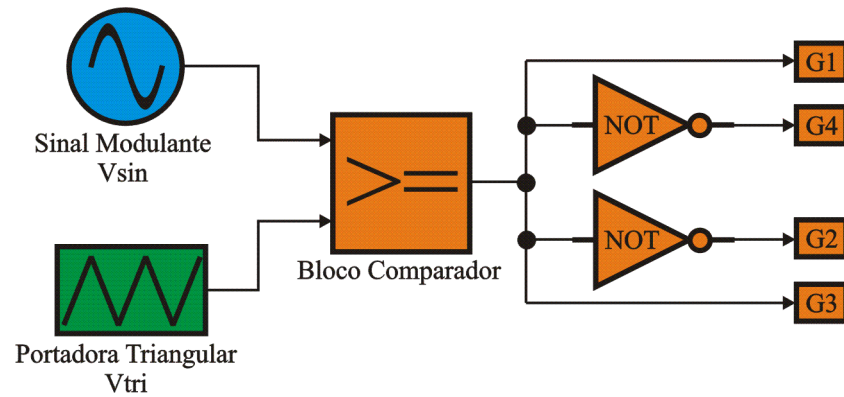


Figura 4.7 – Modulação PWM Bipolar – Diagrama em Blocos

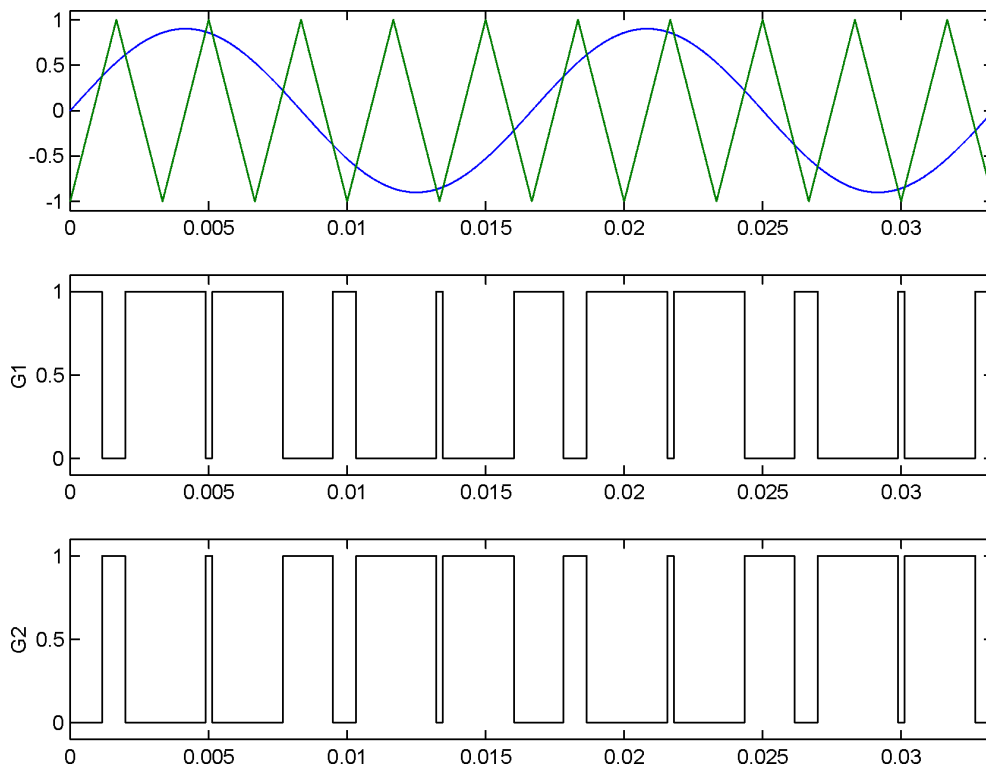


Figura 4.8 – Modulação PWM Bipolar

Os sinais modulante e portadora são comparados. Enquanto o sinal modulante for maior (ou igual) do que a portadora, **G1** estará em nível alto. E enquanto o sinal modulante for menor do que a portadora, **G1** estará em nível baixo.

Conforme mencionado anteriormente **G1** deve ser complementar a **G4** e o mesmo vale com relação a **G2** e **G3**. Para uma forma de onda do tipo bipolar na saída do VSI, é necessário que os IGBTs 1 e 3 estejam no mesmo estado, fechados ou abertos e, por conseguinte o mesmo acontecerá com os IGBTs 2 e 4. A **Figura 4.9** apresenta uma análise das possíveis situações de chaveamento.

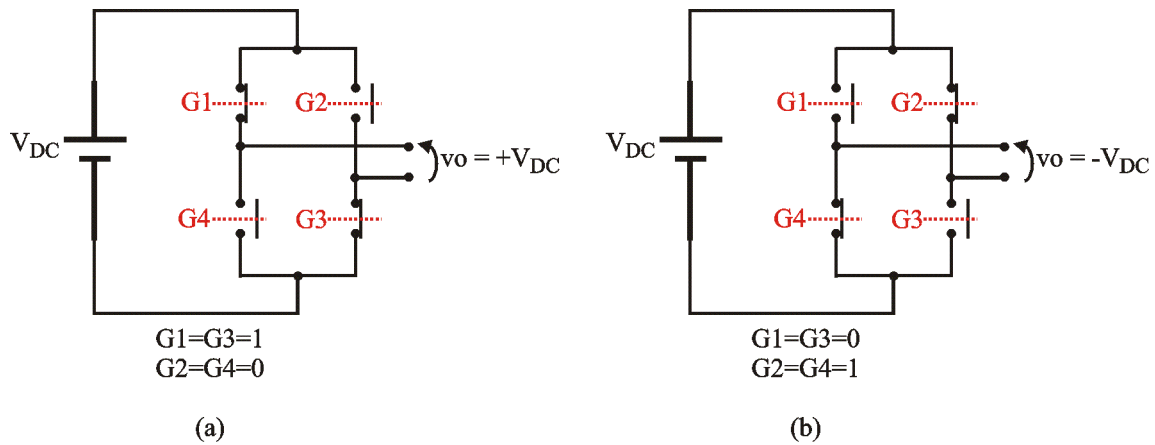


Figura 4.9 – Modulação PWM Bipolar – Situações possíveis de chaveamento

Na **Figura 4.9 (a)**, tem-se a situação em que  $G1=G3=1$  e  $G2=G4=0$ . Nesta situação tem-se  $+V_{DC}$  na saída do VSI. Na **Figura 4.9 (b)**, tem-se a situação em que  $G1=G3=0$  e  $G2=G4=1$ . Nesta situação tem-se  $-V_{DC}$  na saída do VSI.

A **Figura 4.10** mostra a tensão de saída do VSI, a partir dos pulsos de gatilho.

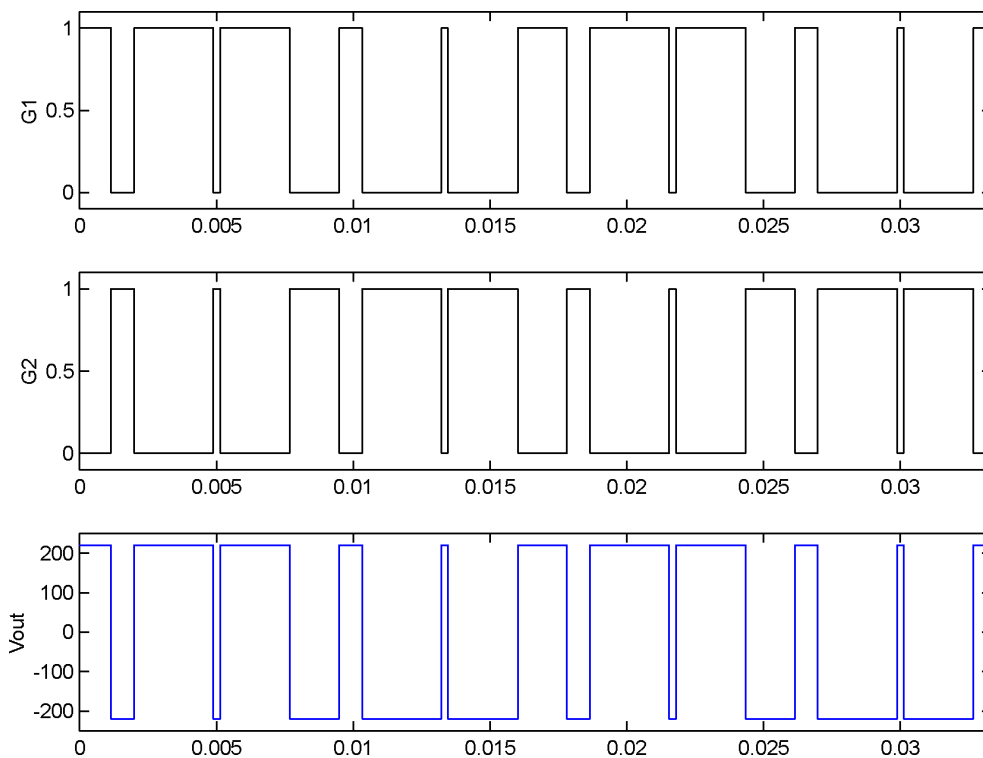


Figura 4.10 – Modulação PWM Bipolar – Saída do VSI

Nota-se que, a cada troca de estados, a tensão de saída da ponte apresenta chaveamentos de  $+V_{DC}$  para  $-V_{DC}$ . Por isto este tipo de modulação é conhecido como **Modulação PWM Bipolar**, pois os chaveamentos ocorrem entre as duas polaridades do **link DC** a cada troca de estados. Nota-se, também, que a frequência da tensão de saída é a mesma frequência de chaveamento dos IGBTs da ponte.

4.3.1.2 – Modulação Unipolar

Visando aumentar a frequência da tensão de saída, sem aumentar a frequência de chaveamento dos IGBTs, criou-se a Modulação Unipolar.

Originalmente este tipo de modulação era realizado conforme se vê na **Figura 4.11** e na **Figura 4.12**.

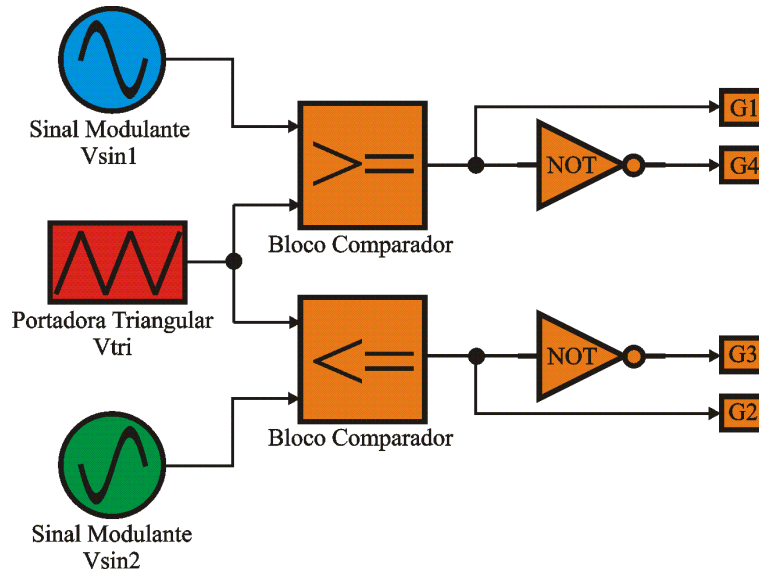


Figura 4.11 – Modulação PWM Unipolar com 1 Portadora – Diagrama em Blocos

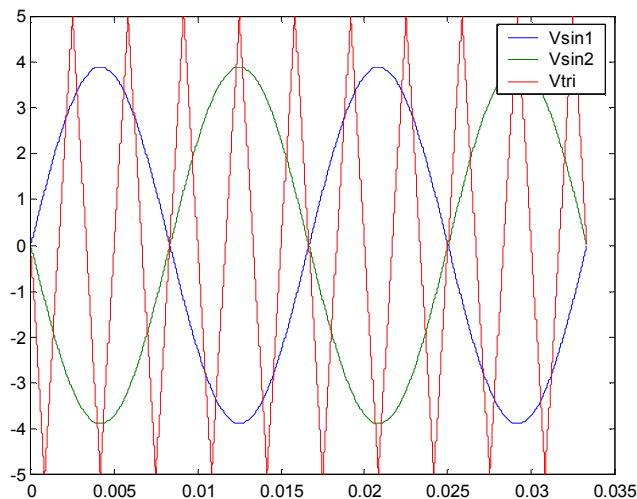


Figura 4.12 – Modulação PWM Unipolar com 1 Portadora – Sinal Modulante e Portadora

Nesta forma de modulação utiliza-se uma única portadora triangular e dois sinais modulantes de mesma amplitude e mesma frequência defasados de 180°. Este tipo de modulação tem muita utilidade quando se deseja gerar um sinal (após a demodulação) com parâmetros (amplitude, frequência e fase) que não precisem ser variados constantemente. Sendo assim, fica simples a determinação dos dois sinais modulantes.

Porém, para este projeto, deseja-se um sinal que deve variar o tempo todo, dependendo da tensão na carga. Dessa forma foi feita uma alteração no algoritmo original (**Figura 4.11 e Figura 4.12**), e o novo algoritmo de geração de pulsos pode ser visto nas **Figuras 4.13 e 4.14**.

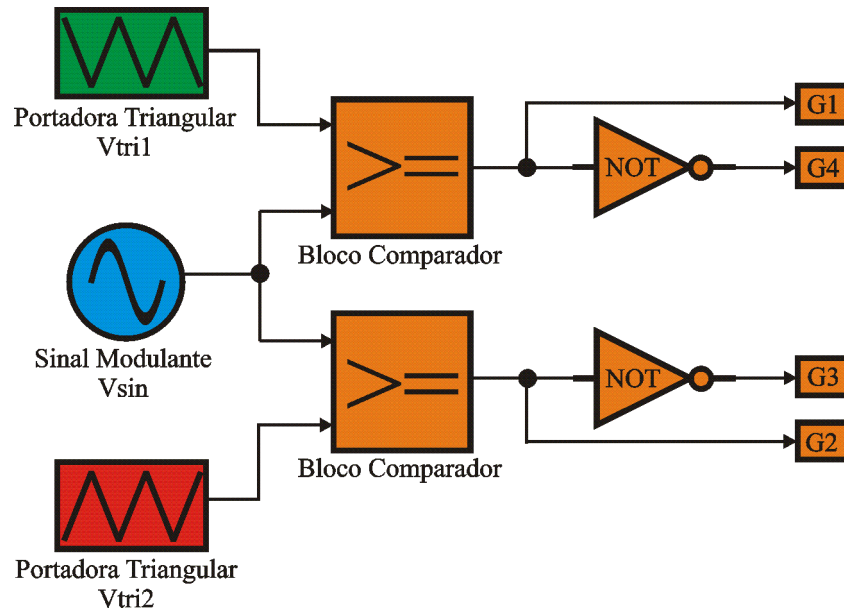


Figura 4.13 – Modulação PWM Unipolar com 2 Portadoras – Diagrama em Blocos

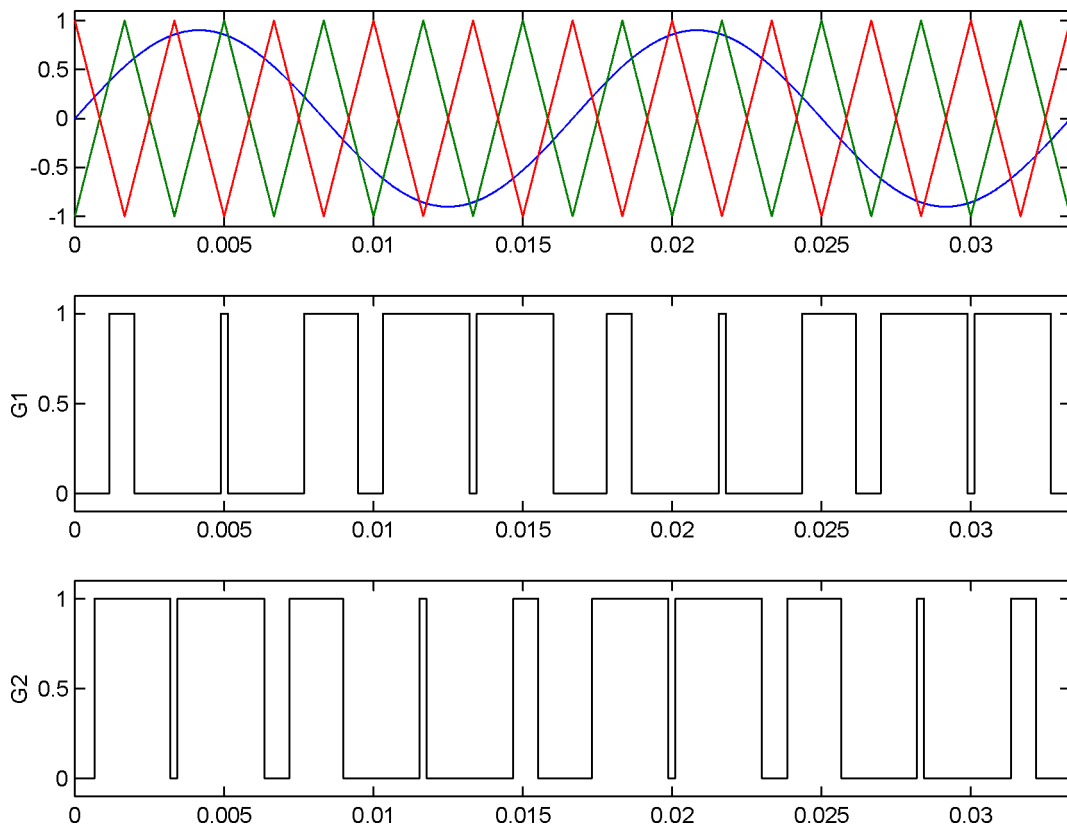


Figura 4.14 – Modulação PWM Unipolar com 2 Portadoras



O novo método utiliza apenas um sinal modulante (que será calculado instantaneamente, dependendo do valor da tensão na carga) e duas portadoras triangulares de mesma amplitude e mesma frequência defasados de  $180^\circ$ , que podem ser facilmente geradas.

Tanto o método da **Figura 4.11**, quanto o novo método da **Figura 4.13** apresentam os mesmos resultados com relação à forma como os IGBTs irão realizar o chaveamento. A **Figura 4.15** apresenta uma análise das possíveis situações de chaveamento.

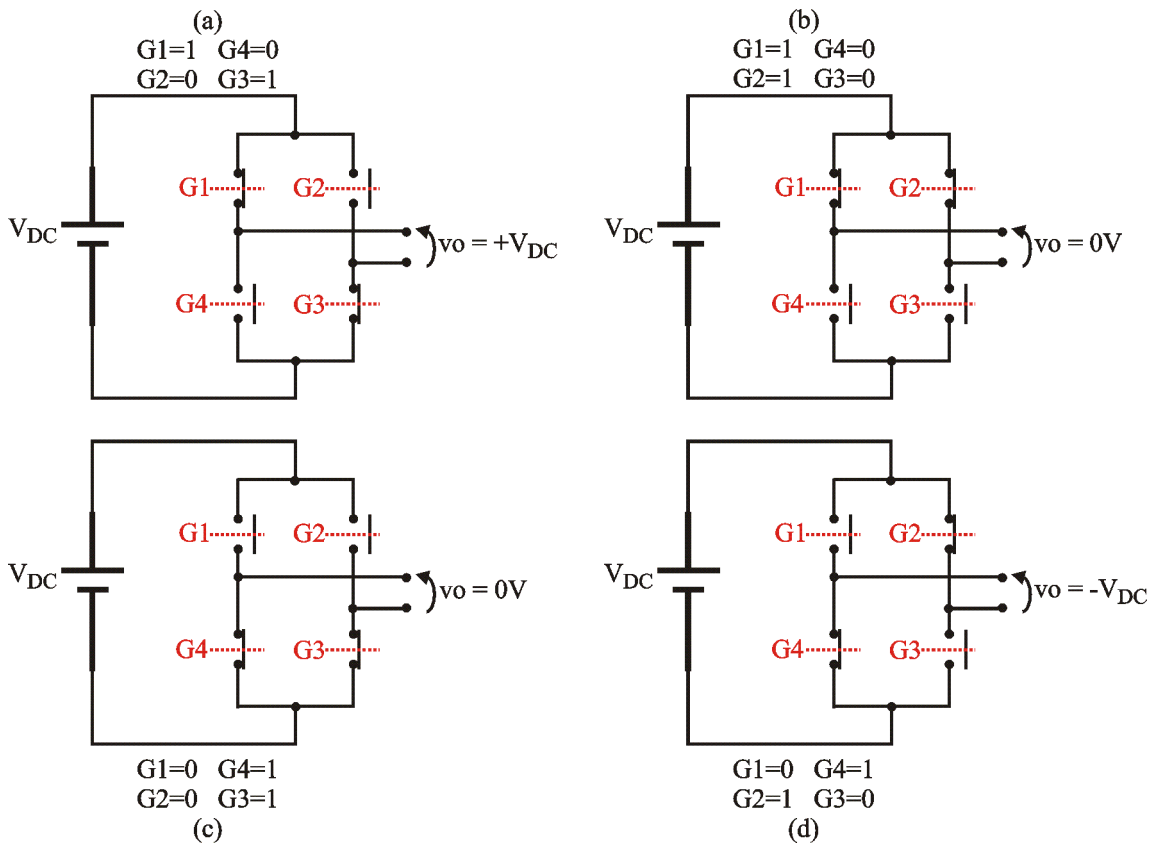


Figura 4.15 – Modulação PWM Unipolar – Situações possíveis de chaveamento

Na **Figura 4.15 (a)**, tem-se a situação em que  $G1=G3=1$  e  $G2=G4=0$ . Nesta situação tem-se  $+V_{DC}$  na saída do VSI. Na **Figura 4.15 (d)**, tem-se a situação em que  $G1=G3=0$  e  $G2=G4=1$ . Nesta situação tem-se  $-V_{DC}$  na saída do VSI. Estas duas situações são as duas situações possíveis que ocorriam na modulação bipolar.

Com a modulação unipolar surgem mais duas possibilidades. Na **Figura 4.15 (b)**, tem-se a situação em que  $G1=G2=1$  e  $G3=G4=0$ . Nesta situação tem-se  $0V$  na saída do VSI. Na **Figura 4.15 (c)**, tem-se a situação em que  $G1=G2=0$  e  $G3=G4=1$ . Nesta situação tem-se  $0V$ , novamente, na saída do VSI.

Essas duas novas situações de chaveamento possibilitam aos IGBTs gerar, na saída da ponte, uma tensão que pode excursionar de  $+V_{DC}$  para  $0V$ , de  $0V$  para  $-V_{DC}$ , de  $-V_{DC}$  para  $0V$  e de  $0V$  para  $+V_{DC}$ . Ou seja, a saída da ponte está sempre chaveando  $1V_{DC}$  a cada mudança de estado, enquanto que na modulação bipolar ela chaveava  $2V_{DC}$  a cada mudança de estado.

A **Figura 4.16** mostra a tensão de saída do VSI, a partir dos pulsos de gatilho.

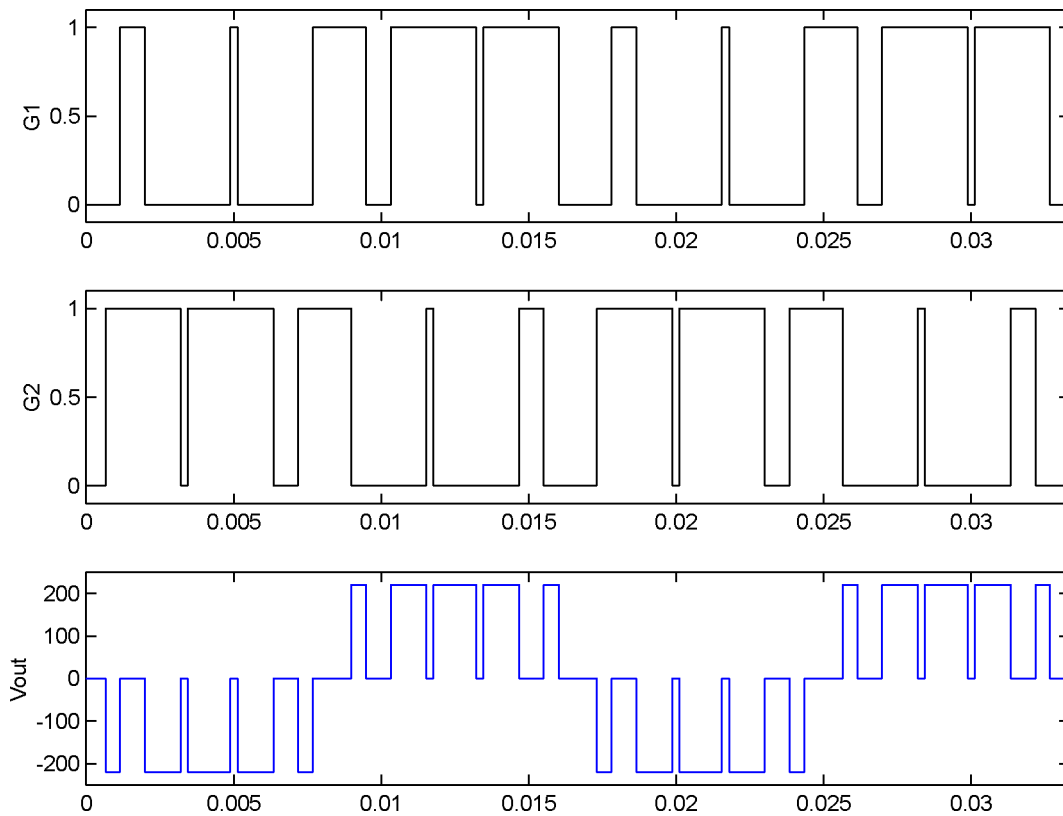


Figura 4.16 – Modulação PWM Unipolar – Saída do VSI

Conforme se pode observar na **Figura 4.16**, os IGBTs chaveiam entre apenas uma única polaridade do **link DC** por vez. Por isso é chamado **PWM Unipolar**.

A modulação Unipolar apresenta uma vantagem em relação à bipolar: maior facilidade na demodulação.

Conforme será discutido na **seção 4.4** (Filtro Passivo), quanto maior for a frequência de chaveamento dos IGBTs, mais fácil será a recuperação da informação pelo filtro. Porém, os IGBTs possuem um limite para a frequência de chaveamento e quanto maior esta frequência maiores serão as perdas de chaveamento.

Na modulação Bipolar, como os quatro IGBTs chaveiam ao mesmo tempo, o filtro passivo irá ver uma onda modulada com a própria frequência de chaveamento dos IGBTs. Isto pode ser observado na **Figura 4.10**.

Na modulação Unipolar apenas os dois IGBTs complementares [(1 e 4) ou (2 e 3)] chaveiam ao mesmo tempo. Os IGBTs das duas pernas opostas [(1 e 2) ou (4 e 3)] têm chaveamento intercalado. Dessa forma, o filtro passivo irá ver uma onda modulada com o dobro da frequência de chaveamento de cada IGBT. Isto pode ser observado na **Figura 4.16**.

Em outras palavras, na modulação Unipolar, os IGBTs continuam chaveando com a mesma frequência da modulação Bipolar, porém o efeito na tensão de saída é o de uma onda com o dobro da frequência, facilitando o processo de filtragem. Por isto a modulação Unipolar é mais eficiente do que a Bipolar.

Neste projeto utilizou-se modulação **Unipolar** com frequência de chaveamento de **10kHz**.

#### 4.3.1.3 – Modulador Desenvolvido

Implementou-se um modulador PWM unipolar, com 2 portadoras triangulares, utilizando um processo digital.

**A implementação do modulador PWM poderia ter sido feita de forma analógica. Para tanto, deveria ser necessário implementar um conversor D/A por fase, dois geradores de onda triangulares, e realizar as comparações com amplificadores operacionais.**

**O problema com este tipo de abordagem, é que sinais analógicos são muito mais sujeitos a contaminação por ruídos do que os sinais digitais. Além disto um projeto analógico bem feito requer muita experiência do projetista, pois são necessários muitos ajustes até que o resultado saia da forma esperada.**

**Sendo assim, optou-se pela implementação digital, pois esta não requer tantos cuidados.**

O modulador digital foi idealizado como um sistema modular, composto por 4 placas, cada uma com uma função específica. Sendo assim, a manutenção torna-se mais fácil, pois se pode isolar os defeitos de forma mais rápida.

E como cada placa tem uma função, pode-se re-projetar uma determinada placa, sem que seja necessário modificar as outras.

O modulador digital segue o diagrama da **Figura 4.13**, sendo que uma placa gera as portadoras triangulares, outra realiza uma comparação com o sinal modulante, e uma terceira realiza a interface entre o modulador e o VSI. Para que todas as placas se comuniquem entre si, uma quarta placa foi criada, sendo um barramento comum.

## 4.3.1.3.1 – Gerador de Portadora Triangular

Esta placa consiste basicamente de um contador, cuja contagem vai de 0 até 255, e, depois, de 255 até 0. Dessa forma tem-se uma onda triangular, cujo período dura 512 contagens. Para a geração da triangular complementar utilizam-se portas lógicas.

Esta placa também possui um gerador de clock interno, formado por 2 monoestáveis que se auto-gatilham indefinidamente. Os pulsos de clock gerados por este circuito oscilam numa frequência de 2.5MHz, para gerar as portadoras em 4.9kHz.

Como este foi um protótipo, esta placa possui um jumper para selecionar a fonte de clock para o contador. Este jumper seleciona clock interno, ou clock externo.

A **Figura 4.17** apresenta uma fotografia desta placa. O circuito esquemático desta placa encontra-se no **Apêndice D\_1** desta dissertação.

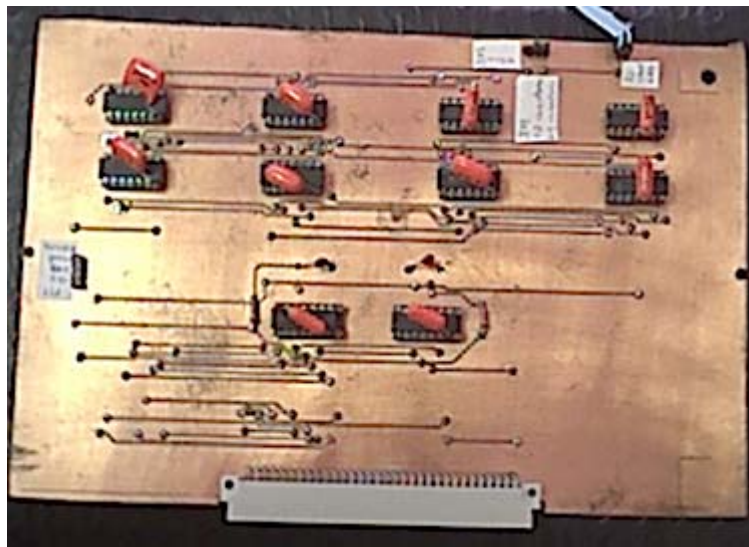


Figura 4.17 – Placa Gerador de Portadora Triangular

Inicialmente utilizou-se 4.9kHz como frequência das portadoras, mas depois foi necessário aumentar esta frequência para facilitar a filtragem (ver **seção 4.4**).

Aumentou-se a frequência das portadoras para 9.8kHz, sendo assim, foi preciso aumentar o clock do contador para 5MHz. Como a frequência de clock ficou alta demais, o circuito com monoestáveis não conseguiria gerá-la corretamente. Sendo assim criou-se uma placa com oscilador a cristal.

O correto seria confeccionar uma nova placa do gerador de portadoras com o oscilador a cristal, mas devido à necessidade de entregar o projeto rapidamente, configurou-se o jumper para clock externo; e confeccionou-se uma placa apenas com o oscilador a cristal, fornecendo o clock externo.

A **Figura 4.18** apresenta uma fotografia desta placa. O circuito esquemático encontra-se no **Apêndice D\_2** desta dissertação.

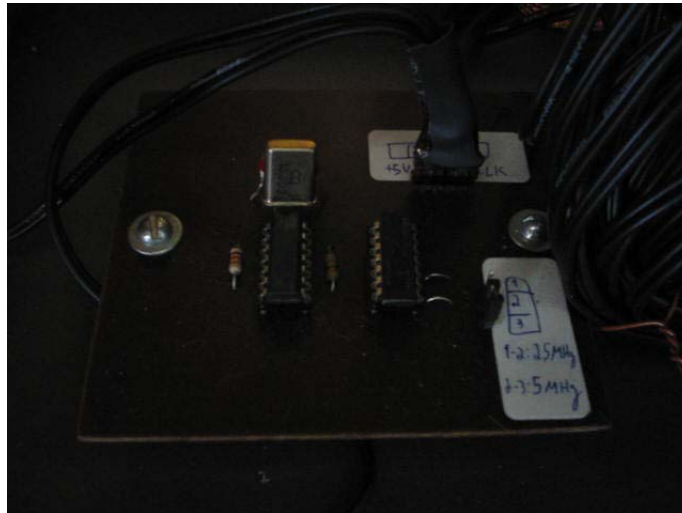


Figura 4.18 – Placa Oscilador a Cristal

Utilizou-se um cristal de 10MHz e um circuito com Flip-Flops para realizar uma divisão de frequências.

Esta placa possui um jumper que seleciona a frequência de clock, 5Mhz, ou 2.5MHz.

## 4.3.1.3.2 – Comparador Digital

Esta placa utiliza um arranjo de circuitos integrados do tipo 7485, que são comparadores digitais de 4 bits.

Como apenas 4 bits de resolução seriam muito pouco, utilizou-se um arranjo em cascata de 2 CIs 7485 para cada fase.

Então, a palavra de 8 bits da saída digital da PCM-3718H é comparada com a palavra de 8 bits da portadora triangular. Como resultado têm-se os pulsos de gatilho.

Tem-se apenas uma entrada de 8 bits para o sinal da PCM-3718H, sendo assim, esta informação deve ser demultiplexada para gerar 3 palavras de 8 bits, uma para cada fase. Esta demultiplexação é realizada com 3 CIs 74273. Cada um deles recebe um pulso de sincronismo, quando for a vez da palavra correspondente.

A **Figura 4.19** apresenta uma fotografia desta placa. O circuito esquemático desta placa encontra-se no **Apêndice D\_3** desta dissertação.

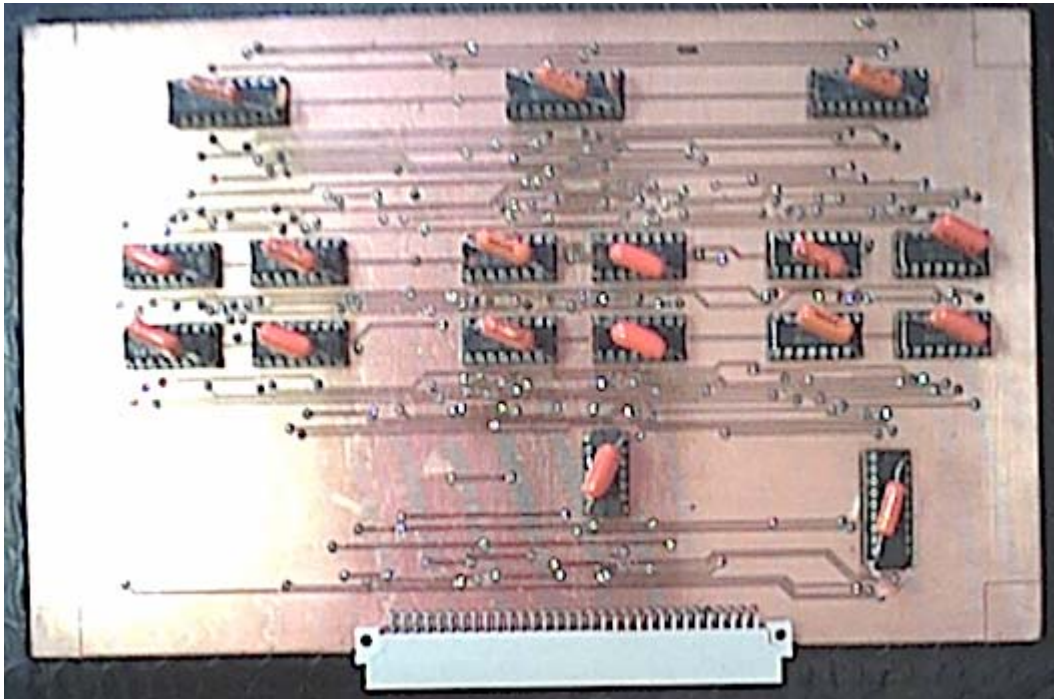


Figura 4.19 – Placa Comparador Digital

#### 4.3.1.3.3 – Interface PWM X VSI

Os pulsos de gatilho gerados pela Placa Comparador Digital apresentam níveis de tensão e correntes TTL. Porém os sinais do driver SKHI-22A, da Semikron trabalham com a seguinte representação de tensões: 15V para nível alto e 0V para nível baixo.

A função do Driver SKHI22A é amplificar os pulsos de gatilho, de forma que os IGBTs do VSI possam chavear corretamente. Este driver também possui uma outra função: proteção para os IGBTs.

Caso alguma condição de chaveamento seja considerada inválida pelo driver, o mesmo emitirá um sinal de erro e não permitirá a passagem de pulsos para os IGBTs. O sinal de erro permanecerá ativado, enquanto os pulsos não forem cortados.

O Driver SKHI22A deverá receber os pulsos de gatilho para os IGBTs, que vêm da PCM-3718H. Sendo assim, deve-se realizar uma conversão de TTL para 15V. Para realizar esta conversão, utilizou-se um buffer TTL não inversor de coletor aberto (7417), com pull-up ligado em 15V. O valor do resistor de pull-up utilizado foi de 10k $\Omega$ .

Em condições de operação normais o sinal de erro que o driver sinaliza deverá permanecer em 15V. Caso o driver detecte uma condição de chaveamento inválida, ele sinaliza erro enviando 0V pelo pino de erro.

A Placa de Interface PWM X VSI possui um circuito detector de erro. Originalmente, caso um erro fosse detectado, o inversor parava de funcionar e o operador do sistema, após eliminar a causa do erro, precisaria apertar um botão, para que o inversor voltasse a funcionar. A **Figura 4.20** apresenta uma fotografia desta placa. O circuito esquemático desta placa encontra-se no **Apêndice D\_4** desta dissertação.

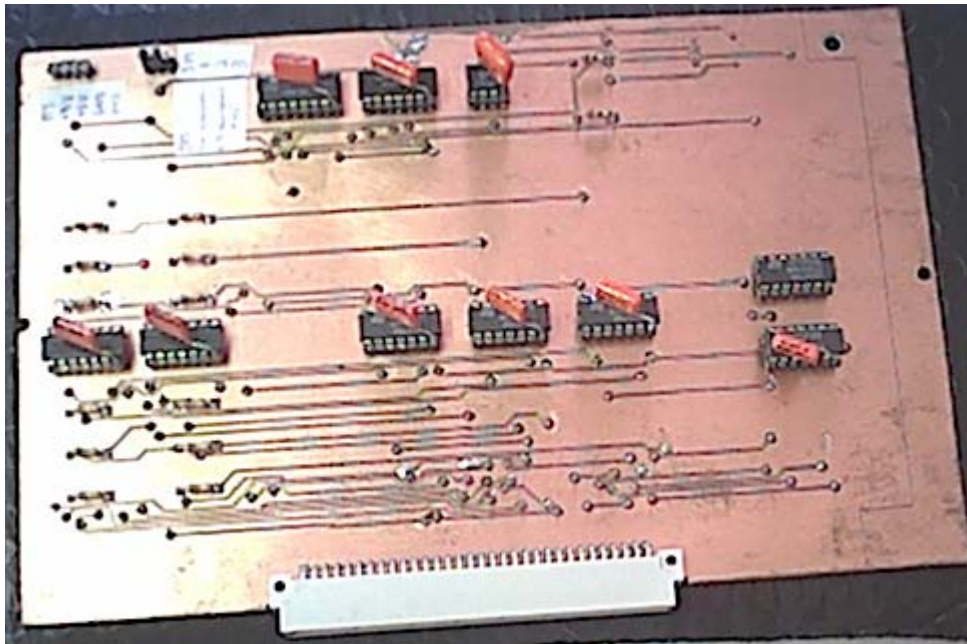


Figura 4.20 – Placa Interface PWM X VSI

Visando automatizar o processo de partida do equipamento, algumas mudanças foram necessárias nesta placa. Ao invés do operador pressionar um botão de reset nesta placa para religar o inversor em caso de erro, ele realizará esta operação através das botoeiras do painel de comando.

Sendo assim uma nova placa foi adicionada ao equipamento. Esta placa possui um circuito que lê o estado de uma botoeira do painel de comando e o envia para a CPU através da porta paralela. Um outro circuito recebe comandos da CPU (também pela porta paralela) para acionar o contator que liga o sistema.

A **Figura 4.21** apresenta uma fotografia desta placa. O circuito esquemático encontra-se no **Apêndice D\_5** desta dissertação.

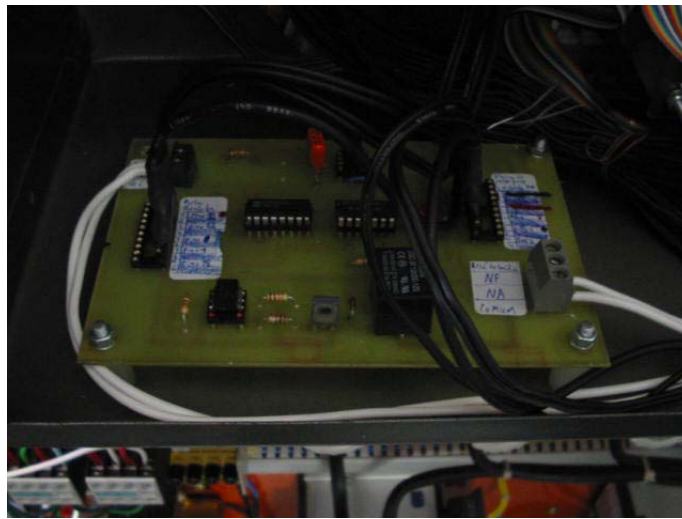


Figura 4.21 – Placa Interface CPU X Comando



#### 4.3.1.3.4 – Barramento

Optou-se por uma estrutura modular, sendo que cada placa possui uma função. A comunicação entre as placas ocorre através de um barramento comum. Sendo assim, caso seja necessário uma alteração de projeto em uma determinada função, apenas a placa relativa a esta função será modificada.

A conexão entre os cabos da saída digital da PCM-3718H e o modulador PWM ocorre nesta placa. Assim como a conexão entre os cabos dos inversores e o modulador. Os sinais são, então, distribuídos para as diferentes placas através de slots de 64 pinos.

A **Figura 4.22** apresenta uma fotografia desta placa. O circuito esquemático desta placa encontra-se no **Apêndice D\_6** desta dissertação.

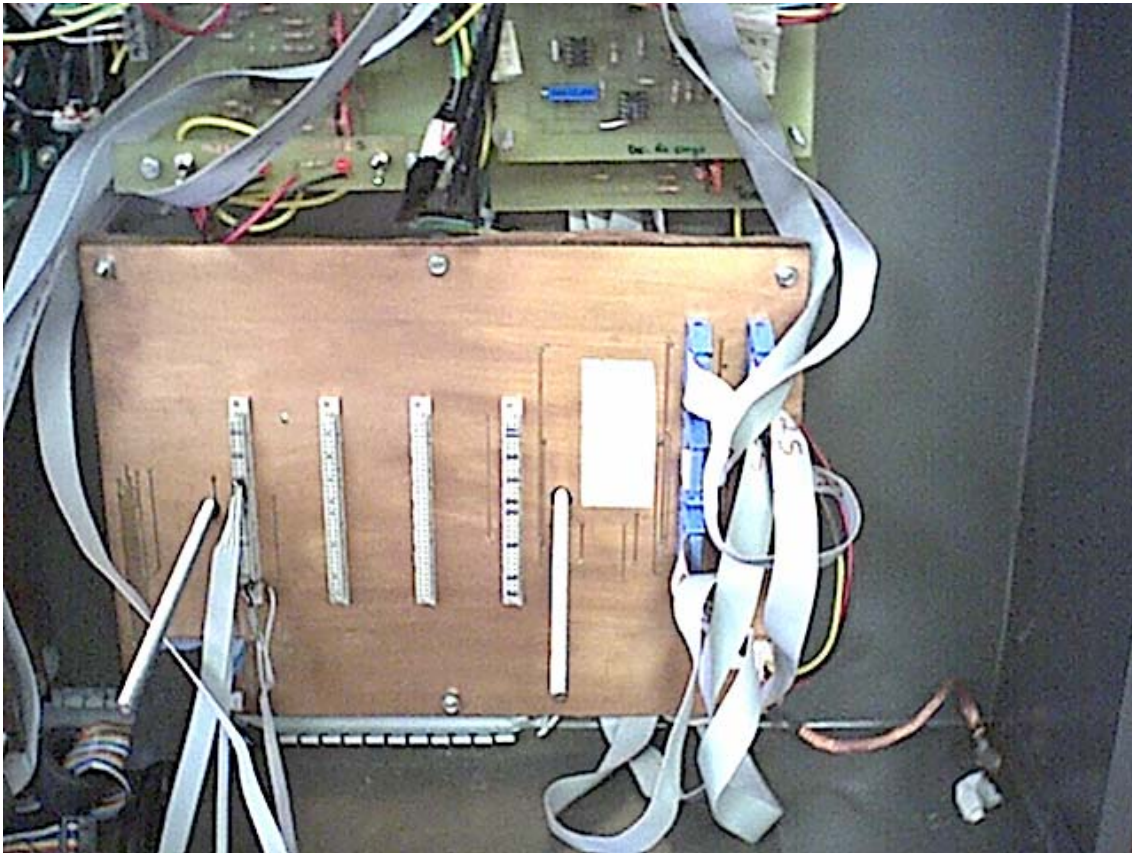


Figura 4.22 – Placa Barramento

A Figura 4.23 apresenta uma fotografia das placas descritas anteriormente, fixadas sobre a placa de barramento.

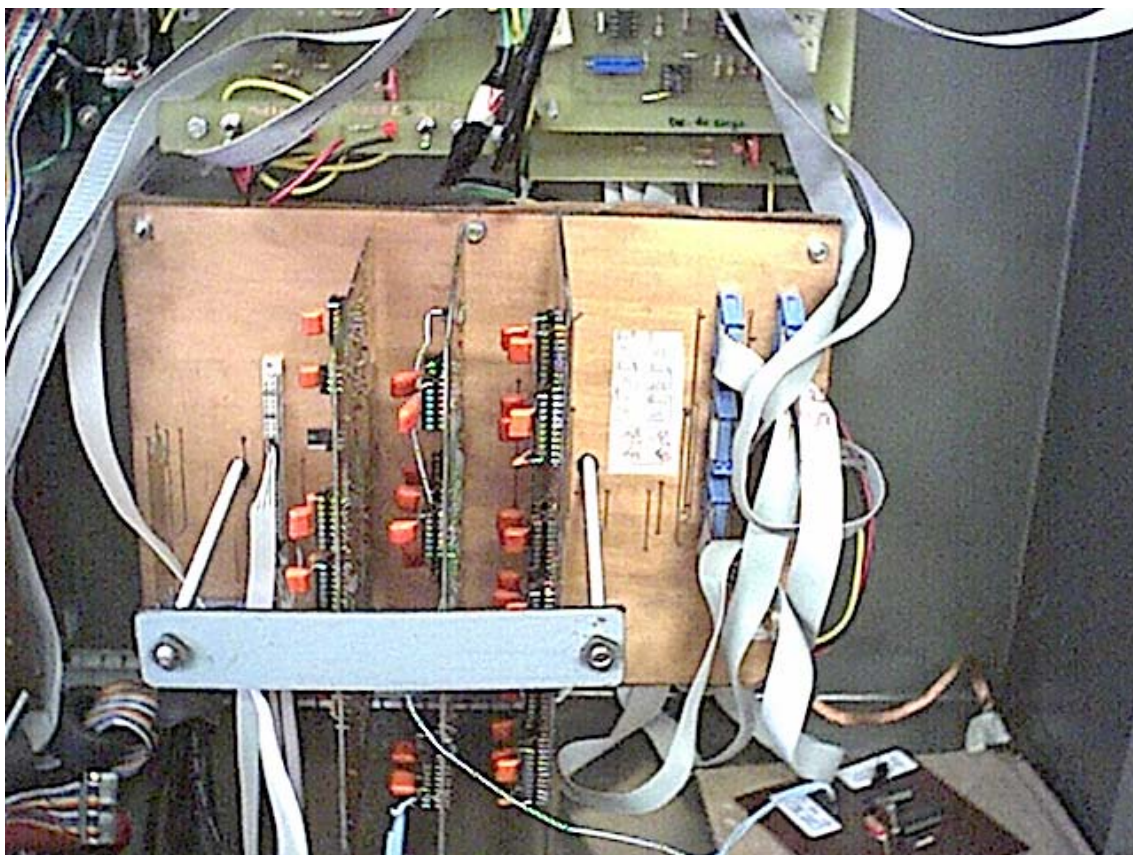


Figura 4.23 – Modulador PWM completo

#### 4.4 – Filtro Passivo

Tem-se na saída do VSI, conforme **Figura 4.16** da **seção 4.3**, uma onda modulada em largura de pulsos. Toda a informação sobre o sinal modulante encontra-se codificada na onda modulada. Para a recuperação do sinal modulante, deve-se realizar o processo reverso da modulação, ou seja, a demodulação do sinal.

As **Figuras 4.8, 4.10, 4.12, 4.14 e 4.16** da **seção 4.3** foram elaboradas utilizando uma frequência de chaveamento baixa. Isto foi feito apenas para que as figuras ficassem legíveis, demonstrando o processo de modulação. Em casos reais, a frequência de chaveamento deve ser muito maior do que a frequência do sinal modulante.

A **Figura 4.24** representa o espectro de frequências da tensão de saída do VSI para um sinal modulante senoidal com frequência de 60Hz e uma frequência de chaveamento de 10kHz.

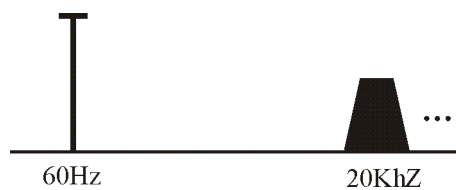


Figura 4.24 – Espectro de Frequências da onda de saída do Inversor

Nota-se, na **Figura 4.24**, uma raia em 60Hz, representando o sinal modulante; e outra em torno de 20kHz, representando o chaveamento. Conforme já explicado na **seção 4.3**, para a modulação Unipolar (caso descrito na **Figura 4.24**), a frequência da onda de saída é o dobro da frequência de chaveamento de cada IGBT.

Para a recuperação do sinal modulante deve-se eliminar todas as componentes de alta frequência do espectro, a maneira mais simples de se fazer isso é através de uma filtragem passiva do tipo passa-baixas.

Um filtro passa-baixas é um sistema cuja dinâmica não permite variações bruscas no sinal que passa através dele. Sendo assim, ele é utilizado para eliminar as componentes de altas frequências de um sinal, permitindo apenas a passagem das componentes de baixas frequências.

O filtro passivo pode ser construído a partir de diversas configurações. A configuração mais utilizada em aplicações com Inversores é a que utiliza um indutor e um capacitor como elementos armazenadores de energia, criando-se um sistema de 2ª ordem.

Tem-se na **Figura 4.25** o desenho esquemático de um filtro passivo passa-baixas de 2ª ordem.

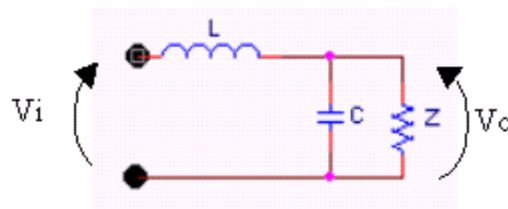


Figura 4.25 – Filtro Passivo Passa-Baixas

Após uma análise do circuito por meio de equações diferenciais e transformadas de Laplace tem-se a função de transferência (FT) da equação (Eq. 4.4):

$$\frac{Vo(s)}{Vi(s)} = \frac{1}{s^2 + s \cdot \frac{1}{Z \cdot C} + \frac{1}{L \cdot C}} \quad (\text{Eq. 4.4})$$

O denominador da FT pode ser comparado com o polinômio característico para sistemas dinâmicos de 2ª ordem:

$$s^2 + s \cdot (2 \cdot \zeta \cdot \omega_0) + \omega_0^2 \quad (\text{Eq. 4.5})$$

Tem-se então que:

$$\omega_0 = \frac{1}{\sqrt{L \cdot C}} \Rightarrow f_0 = \frac{1}{2 \cdot \pi \cdot \sqrt{L \cdot C}} \quad (\text{Eq. 4.6})$$

$$\zeta = \frac{\sqrt{L/C}}{2 \cdot Z} \quad (\text{Eq. 4.7})$$

Da equação (Eq. 4.6) pode-se perceber que a frequência de corte do filtro ( $f_0$ ) depende apenas do valor da indutância e da capacitância. Porém, da equação (Eq. 4.7), nota-se que o fator de amortecimento do circuito ( $\zeta$ ) depende, além da indutância e da capacitância, da impedância da carga.

O efeito do fator de amortecimento no sinal de saída do filtro é um pico de ressonância nas componentes de frequências próximas à frequência de corte. Na **Figura 4.26** tem-se a curva de resposta em frequência para um filtro sub-amortecido ( $\zeta < 1$ ). O pico de ressonância indicado na figura irá variar conforme a impedância de carga também variar. Esta variação se reflete numa variação de amplitude nas componentes de frequência compreendidas naquela região.

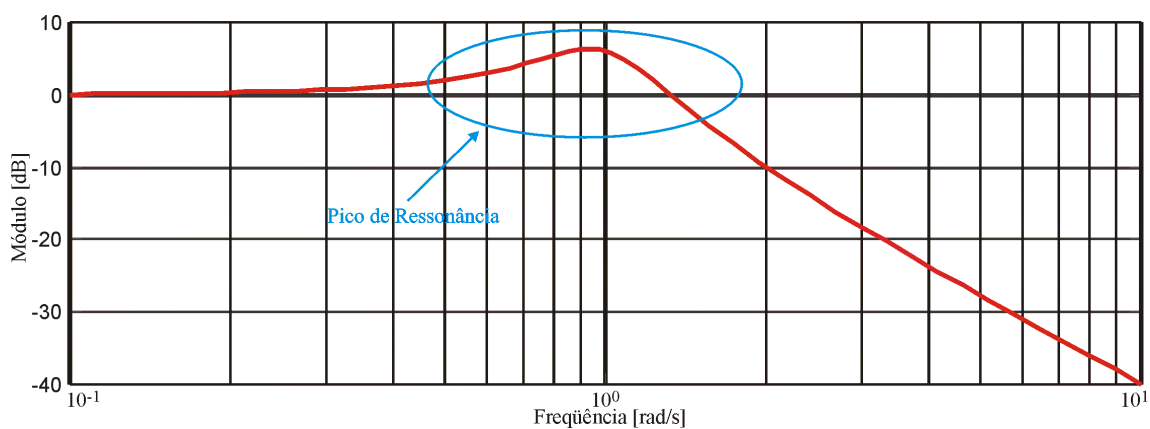


Figura 4.26 – Resposta em frequência do filtro

O ideal seria trabalhar com um amortecimento crítico ( $\zeta=1$ ), o problema é que não se tem controle sobre a Impedância da Carga, uma vez que esta impedância irá variar de acordo com a entrada e saída de carga do sistema. Sendo assim, deve-se encontrar alguma forma que a variação de carga não afete a tensão de saída do filtro.

Um outro problema dos filtros passivos é a defasagem que eles impõem ao sinal filtrado. Em aplicações de Compensadores Ativos não pode haver atraso entre a compensação calculada e a compensação que, de fato, é somada à tensão da concessionária.

A **Figura 4.27** ilustra a forma encontrada para solucionar estes dois problemas.

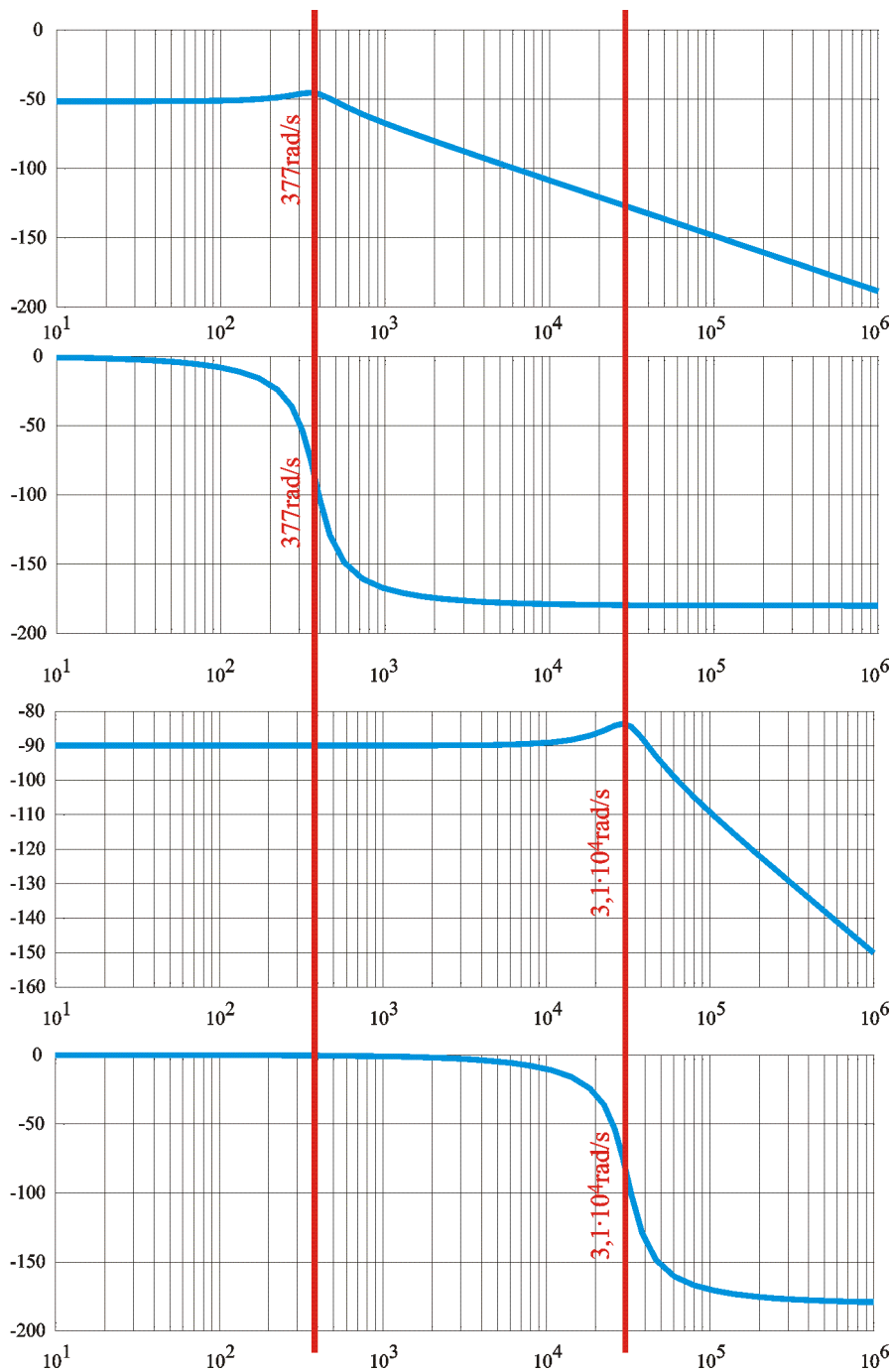


Figura 4.27 – Deslocamento da frequência de corte

A forma encontrada de solucionar estes dois problemas foi projetar o filtro para uma frequência de corte mais alta do que a necessária. Deseja-se eliminar apenas a alta frequência do chaveamento dos IGBTs; e permitir a passagem apenas das componentes fundamental e harmônicos de compensação. Sendo assim, projeta-se o filtro para corte numa frequência bem maior do que a maior frequência de compensação, por exemplo, 5kHz.

As duas curvas superiores da **Figura 4.27** representam o filtro passivo com corte em 60Hz ( $377\text{rad/s}$ ). Nota-se que o pico de ressonância afeta justamente a frequência que se deseja na saída do filtro. Nota-se, também, uma defasagem de  $90^\circ$  na frequência de corte.

As duas curvas inferiores da **Figura 4.27** representam o filtro passivo com corte em 5kHz ( $31.10^3\text{rad/s}$ ). Nota-se que o pico de ressonância continua na região da frequência de corte, porém a componente de 60Hz encontra-se na região de resposta plana. Sendo assim a saída do filtro não será afetada pela variação de carga. Nota-se, também, que a componente de 5kHz ( $31.10^3\text{rad/s}$ ) será defasada de  $90^\circ$ , porém a componente de 60Hz encontra-se em uma região de defasamento nulo.

Deve-se ressaltar que isto funciona bem para uma alta frequência de chaveamento. Pois, caso contrário, o filtro irá permitir a passagem de parte das altas frequências do chaveamento. Isto pode ser visto na **Figura 4.28**.

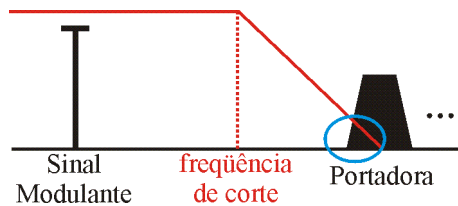


Figura 4.28 – Efeito do deslocamento da frequência de corte

A consequência da passagem de parte das altas frequências é uma ondulação no sinal de saída do filtro.

Utilizou-se neste projeto um indutor de valor em torno de  $75\mu\text{H}$  e um capacitor de  $10\mu\text{F}/660\text{V}$ . Isto representa (pela equação (Eq. 4.6)) uma frequência de corte em torno de  $5.8\text{kHz}$ .

Inicialmente utilizou-se modulação PWM Unipolar (com portadoras a  $5\text{kHz}$ ), sendo assim o filtro passivo via as componentes do chaveamento como frequências acima de  $10\text{kHz}$ . Porém  $10\text{kHz}$  está relativamente próxima de  $5.8\text{kHz}$ , sendo assim a compensação para a fundamental ocorria bem, mas a compensação para os harmônicos chegava atrasada.

Alterou-se, então, a frequência das portadoras para  $10\text{kHz}$  (ver seção 4.3.1.3.1), desta forma o filtro passivo passou a ver as componentes do chaveamento como frequências acima de  $20\text{kHz}$ . Permitindo uma boa compensação.

Deve-se ressaltar que quanto mais espaçamento houver entre a frequência de chaveamento e a maior frequência do sinal modulante, mais simples se torna o processo de filtragem. Porém os IGBTs possuem um limite, que caso ultrapassado, implica em perdas de chaveamento.

A Figura 4.29 apresenta uma fotografia dos filtros passivos montados no equipamento.



Figura 4.29 – Filtros Passivos no Equipamento Desenvolvido

## 4.5 – Fotografias do Equipamento Desenvolvido

O desenvolvimento deste equipamento teve início em maio de 2002.

Inicialmente foram realizados diversos testes, em partes isoladas do equipamento, tais como os sensores Hall, inversores e filtros passivos.

Por volta de junho de 2002, um primeiro protótipo foi finalizado. Tratava-se de um equipamento bastante “caseiro” e experimental. A idéia era simplesmente testar configurações e possibilidades de uso, antes da construção do equipamento definitivo.

Este primeiro protótipo utilizava o Real-Time Workshop (toolbox do Matlab) para implementar o algoritmo de controle. Na época utilizava-se um algoritmo bastante simples (e precário também). Este algoritmo não corrigia distorções harmônicas (corrigia apenas Voltage SAGs) e era em malha aberta.

O equipamento utilizava um transformador de acoplamento 1:1. Portanto os testes eram realizados apenas no 220V.

A **Figura 4.30** mostra uma fotografia deste primeiro protótipo

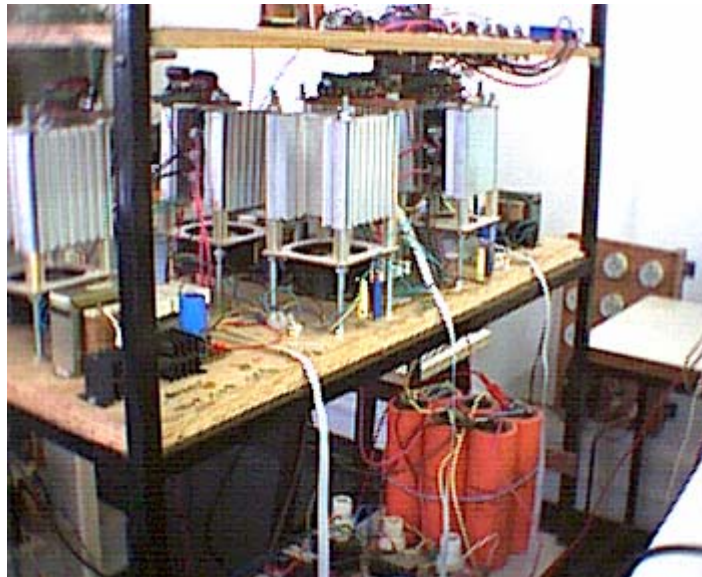


Figura 4.30 – Primeiro Protótipo

Após os testes iniciais no protótipo da **Figura 4.30**, iniciou-se a montagem do equipamento (que também é um protótipo) entregue a concessionária CEMAR.

Como software de controle, utilizou-se o mesmo algoritmo do primeiro protótipo, porém convertido para a linguagem ANSI C, rodando no Embedded PC. Conforme os testes foram sendo realizados, notaram-se dois problemas.

-Primeiro: Os ruídos na aquisição de dados eram transferidos para a carga.

-Segundo: A alta corrente de magnetização do transformador 220:13800 distorcia o sinal de compensação gerado.

A solução para estes dois problemas foi apresentada em detalhes no **Capítulo 3** e apenas será resumida aqui: Criou-se um algoritmo insensível a ruídos (com compensação de harmônicos) para solução do primeiro problema; e utilizou-se controle em malha fechada para a correção do segundo.



A **Figura 4.31** apresenta uma fotografia da parte frontal do equipamento desenvolvido.

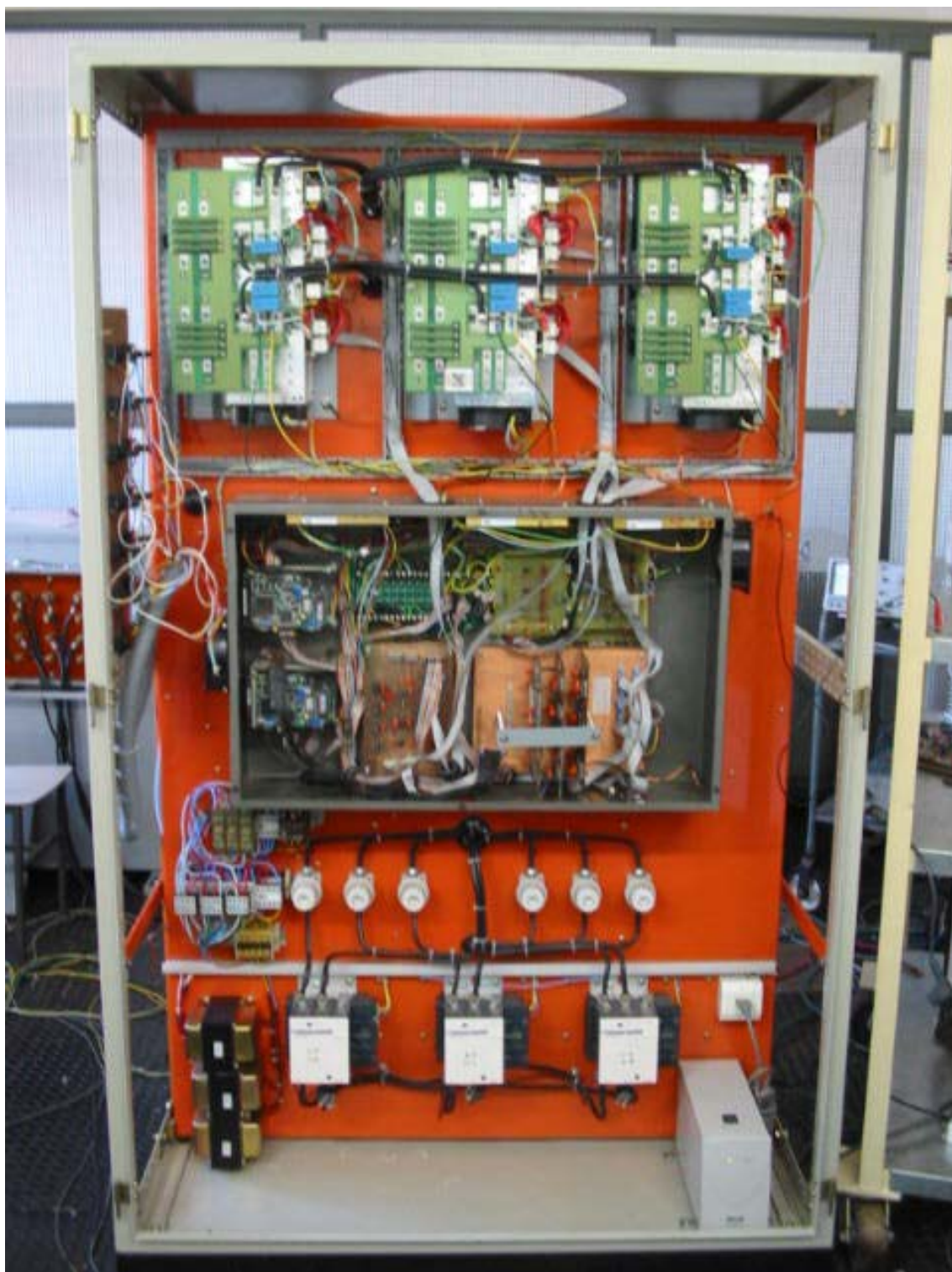


Figura 4.31 – Equipamento Desenvolvido – Vista Frontal

A **Figura 4.32** apresenta uma fotografia da parte traseira do equipamento desenvolvido.



Figura 4.32 – Equipamento Desenvolvido – Vista Traseira

Devido a falta de espaço, a parte traseira do equipamento teve de ser fotografada de cima do transformador de acoplamento, daí a **Figura 4.32** estar em perspectiva de cima para baixo.

Deve-se ressaltar que o equipamento das fotografias anteriores é composto por um Compensador Série e um Compensador Shunt. Sendo assim ele possui 6 módulos inversores, 2 CPUs, etc; tudo em dobro, de forma que o equipamento possa trabalhar de forma independente.

A **Figura 4.33** apresenta uma fotografia do equipamento desenvolvido com as portas dianteiras fechadas.



Figura 4.33 – Equipamento Desenvolvido – Portas Fechadas



# **Capítulo 5**

## **Resultados Experimentais**

<b>5.1 – Ambiente de Teste.....</b>	<b>5 - 3</b>
<b>5.2 – Resultados Obtidos .....</b>	<b>5 - 7</b>
<b>5.2.1 – Teste a vazio .....</b>	<b>5 - 7</b>
<b>5.2.2 – Teste com Carga Não Linear.....</b>	<b>5 - 11</b>



## 5.1 – Ambiente de Teste

O equipamento foi desenvolvido para compensar tensões no sistema de distribuição, em **13800V**. Sendo assim, foi preciso construir um ambiente de testes, que simulasse estas condições.

Como não havia modo de se trazer um barramento de 13800V para dentro do laboratório de testes, construiu-se, uma mini subestação 220-13800-220V. No laboratório havia disponível apenas o sinal trifásico em 220V. Sendo assim, utilizou-se um transformador (13800/220,  $\Delta$ -Y) para aumentar o nível de tensão de 220V para 13800V. Tem-se, então o barramento de 13800V.

Como carga foi utilizado um outro transformador (13800/220,  $\Delta$ -Y), este para abaixar a tensão para 220V, de forma que o 220V alimentasse um retificador trifásico. E como carga do retificador trifásico, utilizou-se uma “resistência líquida”. Esta “resistência líquida” é uma caixa d’água de 500 litros com um mecanismo de roldanas adaptado a ela. A **Figura 5.1** apresenta uma fotografia da “resistência líquida”.



Figura 5.1 - Resistência Líquida

Dentro da caixa d'água coloca-se água, e mistura-se barrilha. Forma-se, então, uma solução condutora de eletricidade. O sistema de roldanas controla a profundidade com que eletrodos, alimentados pelo DC do retificador, submergem na solução salina. Dessa forma, controla-se a corrente consumida pelo retificador trifásico e, por conseguinte, a corrente no barramento de 13800V. Desta forma é possível simular variações de carga.

Em série com cada fase, no barramento de 13800V, foi inserido um enrolamento do transformador de acoplamento. Este transformador é do tipo 13800/220V, porém nenhum tipo de conexão entre os enrolamentos. Ou seja, utilizou-se as três fases de forma independente.

Aqui, **nota-se um erro de projeto**. Para que o compensador consiga corrigir tensões desequilibradas é necessário que a tensão de compensação gerada para uma determinada fase seja diferente das outras. Os inversores geram esta tensão desbalanceada sem problemas. Mas essa tensão precisa passar pelo transformador de acoplamento.

Caso fossem utilizados três bancos monofásicos não haveria problemas. Pois cada banco possui um circuito magnético independente.

Como foi utilizado um transformador trifásico, a intenção de se remover a conexão entre as fases era a de poder realizar a compensação de forma independente. O problema é que, nesta configuração, o fluxo magnético que circulava nos 3 enrolamentos era o mesmo e com isto era impossível a compensação de forma independente.

Como o custo envolvido na troca do transformador trifásico por um banco de três monofásicos seria alto demais, decidiu-se por manter o transformador trifásico e não compensar faltas desequilibradas.

Em paralelo com o transformador da carga, foram instalados 3 **TPs** com o lado da alta (13800V) ligado em estrela com neutro aterrado e o com o secundário ( $115/\sqrt{3}$ ) ligado em delta

O circuito completo da mini subestação encontra-se ilustrado na **Figura 5.2**.



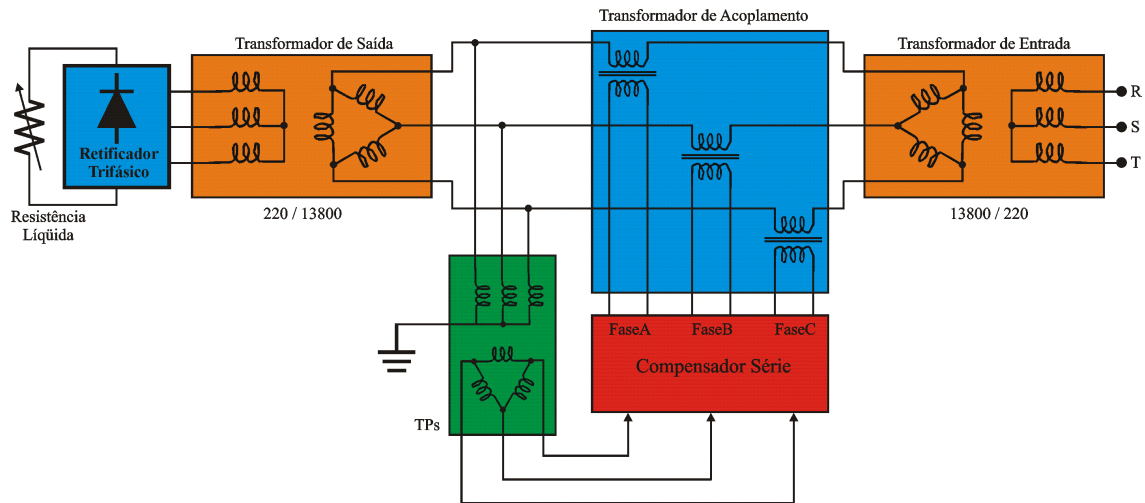


Figura 5.2 – Diagrama de ligação da mini subestação

As Figuras 5.3 e 5.4 apresentam duas fotografias da construção da mini subestação.



Figura 5.3 – Construção da mini Subestação – I



Figura 5.4 – Construção da mini Subestação – II

A **Figura 5.5** apresenta uma fotografia do retificador trifásico.



Figura 5.5 – Retificador Trifásico

## 5.2 – Resultados Obtidos

### 5.2.1 – Teste a vazio

Para a realização deste teste, desconectou-se o retificador trifásico do transformador de saída.

A **Figura 5.6** apresenta a oscilografia da tensão no TP da fase A antes do compensador entrar em funcionamento.

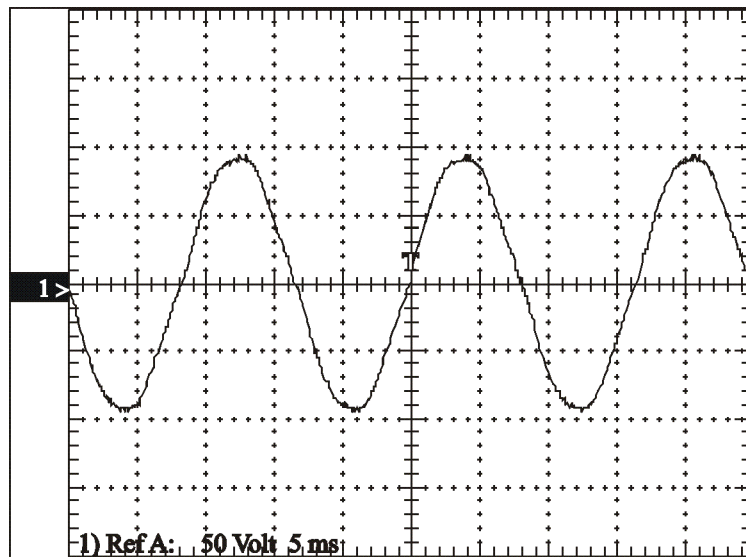


Figura 5.6 – Tensão no TP – Sem Compensação

Nota-se que a concessionária está fornecendo uma tensão distorcida, portanto contendo componentes harmônicas.

A **Figura 5.7** apresenta o espectro de frequências da tensão da **Figura 5.6**.

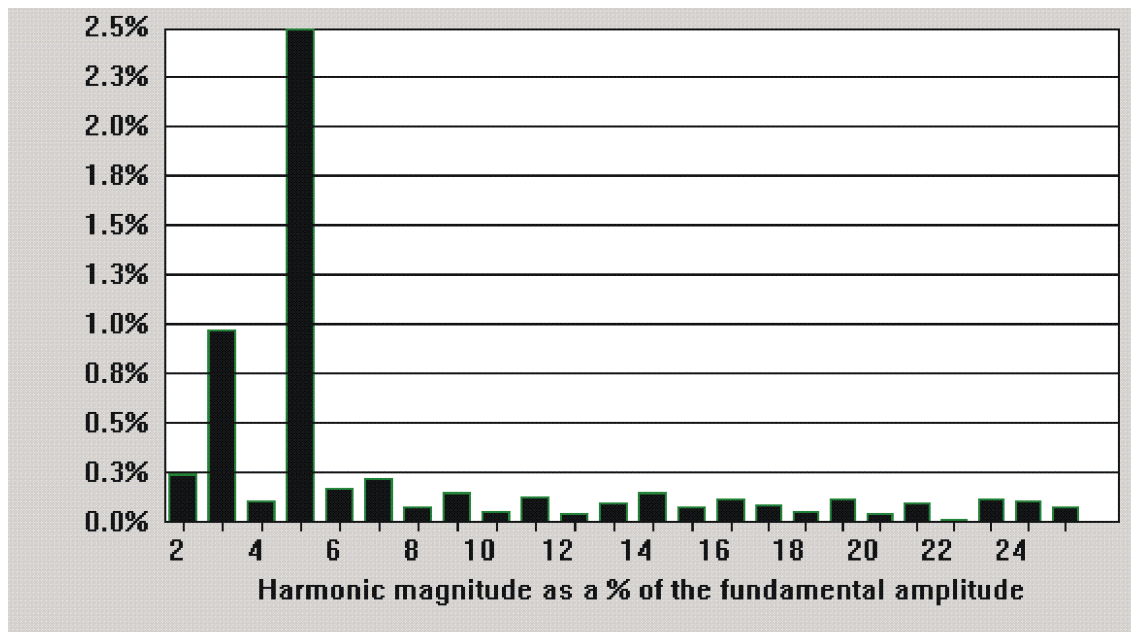


Figura 5.7 – Espectro de frequências da Tensão no TP – Sem Compensação

A **Tabela 5.1** apresenta uma descrição mais detalhada sobre cada componente harmônica do sinal da **Figura 5.6**.

Tabela 5.1  
Distorção Harmônica da Tensão no TP – Sem Compensação

	Freq	Tensão RMS	Porcentagem em relação à fundamental
Fundamental	60 Hz	65.7 V	100.00%
2ª harmônica	120 Hz	166 mV	0.25%
3ª harmônica	180 Hz	656 mV	1.00%
4ª harmônica	240 Hz	76.2 mV	0.12%
5ª harmônica	300 Hz	1.66 V	2.52%
6ª harmônica	360 Hz	122 mV	0.19%
7ª harmônica	420 Hz	156 mV	0.24%
8ª harmônica	480 Hz	62.1 mV	0.09%
9ª harmônica	540 Hz	104 mV	0.16%
10ª harmônica	600 Hz	44.9 mV	0.07%
11ª harmônica	660 Hz	93.5 mV	0.14%
12ª harmônica	720 Hz	36.6 mV	0.06%
13ª harmônica	780 Hz	72.2 mV	0.11%
14ª harmônica	840 Hz	106 mV	0.16%
15ª harmônica	900 Hz	61.4 mV	0.09%
16ª harmônica	960 Hz	85.8 mV	0.13%
17ª harmônica	1.02 kHz	67.7 mV	0.10%
18ª harmônica	1.08 kHz	44.6 mV	0.07%
19ª harmônica	1.14 kHz	88.5 mV	0.13%
20ª harmônica	1.2 kHz	41.1 mV	0.06%
21ª harmônica	1.26 kHz	74.5 mV	0.11%
22ª harmônica	1.32 kHz	16.6 mV	0.03%
23ª harmônica	1.38 kHz	88.8 mV	0.14%
24ª harmônica	1.44 kHz	79.2 mV	0.12%
25ª harmônica	1.5 kHz	57.4 mV	0.09%

Através de **Eq. 5.1** e dos dados da terceira coluna da **Tabela 5.1** pode-se calcular o THD (Distorção Harmônica Total) da tensão da concessionária.

$$\text{THD} = \frac{\sqrt{\sum_{h=2}^{\infty} V_h^2}}{V_f} \cdot 100\% \quad (\text{Eq. 5.1})$$

O valor calculado para o THD da tensão da concessionária é **2,78%**.

A **Figura 5.8** apresenta a oscilografia da tensão no TP da fase A depois do compensador entrar em funcionamento.

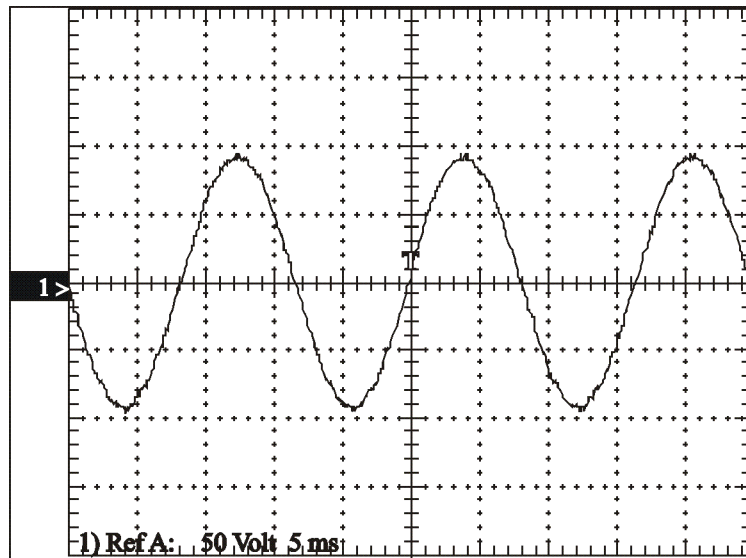


Figura 5.8 – Tensão no TP – Com Compensação

Comparando-se as **Figuras 5.6 e 5.8**, nota-se que o compensador cumpriu seu papel, eliminando os harmônicos e tornando a tensão no TP senoidal.

A **Figura 5.9** apresenta o espectro de freqüências da tensão da **Figura 5.8**.

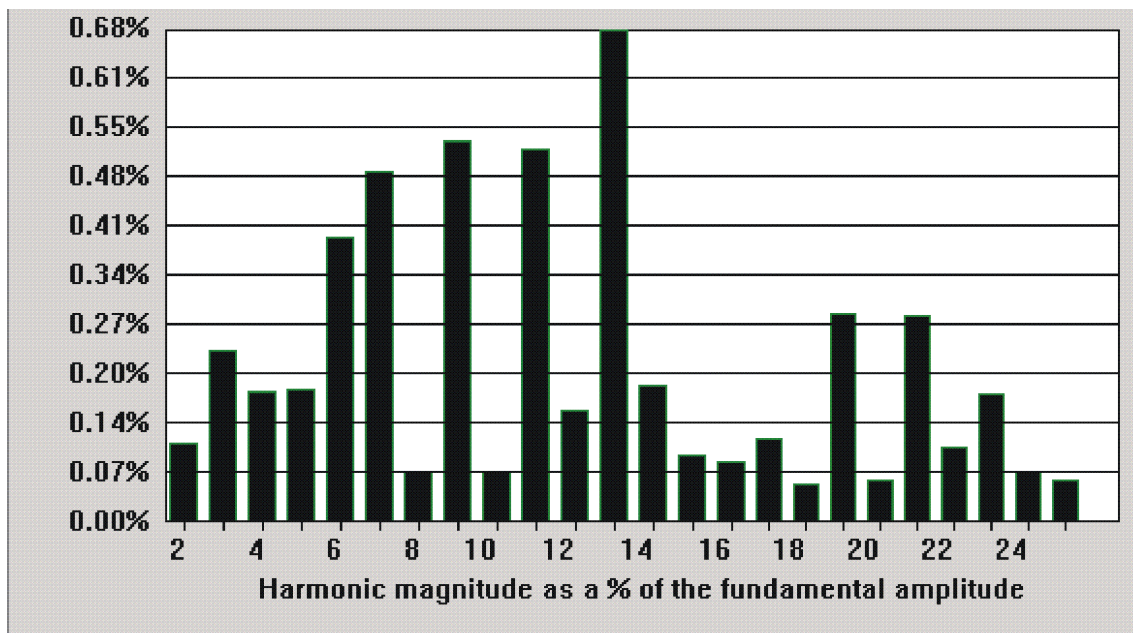


Figura 5.9 – Espectro de freqüências da Tensão no TP – Com Compensação

A **Tabela 5.2** apresenta uma descrição mais detalhada sobre cada componente harmônica do sinal da **Figura 5.8**.

Tabela 5.2  
Distorção Harmônica da Tensão no TP – Com Compensação

	Freq	Tensão RMS	Porcentagem em relação à fundamental
Fundamental	59.9 Hz	65 V	100.00%
2 <sup>a</sup> harmônica	120 Hz	72.4 mV	0.11%
3 <sup>a</sup> harmônica	180 Hz	157 mV	0.24%
4 <sup>a</sup> harmônica	240 Hz	120 mV	0.18%
5 <sup>a</sup> harmônica	300 Hz	121 mV	0.19%
6 <sup>a</sup> harmônica	359 Hz	257 mV	0.40%
7 <sup>a</sup> harmônica	419 Hz	317 mV	0.49%
8 <sup>a</sup> harmônica	479 Hz	46.4 mV	0.07%
9 <sup>a</sup> harmônica	539 Hz	344 mV	0.53%
10 <sup>a</sup> harmônica	599 Hz	46.4 mV	0.07%
11 <sup>a</sup> harmônica	659 Hz	338 mV	0.52%
12 <sup>a</sup> harmônica	719 Hz	102 mV	0.16%
13 <sup>a</sup> harmônica	779 Hz	443 mV	0.68%
14 <sup>a</sup> harmônica	839 Hz	124 mV	0.19%
15 <sup>a</sup> harmônica	899 Hz	62.7 mV	0.10%
16 <sup>a</sup> harmônica	959 Hz	57.1 mV	0.09%
17 <sup>a</sup> harmônica	1.02 kHz	77.6 mV	0.12%
18 <sup>a</sup> harmônica	1.08 kHz	36.6 mV	0.06%
19 <sup>a</sup> harmônica	1.14 kHz	190 mV	0.29%
20 <sup>a</sup> harmônica	1.2 kHz	40.4 mV	0.06%
21 <sup>a</sup> harmônica	1.26 kHz	188 mV	0.29%
22 <sup>a</sup> harmônica	1.32 kHz	68.5 mV	0.11%
23 <sup>a</sup> harmônica	1.38 kHz	117 mV	0.18%
24 <sup>a</sup> harmônica	1.44 kHz	46.8 mV	0.07%
25 <sup>a</sup> harmônica	1.5 kHz	39.3 mV	0.06%

A **Tabela 5.2** comprova o que já fora notado visualmente na comparação da **Figura 5.6** com a **Figura 5.8**.

Houve uma redução no terceiro harmônico de 1,00% para 0,24%, portanto uma redução de 76.00%.

E para o quinto harmônico houve uma redução de 2,52% para 0,19%, portanto uma redução de 92.46%.

Através de **Eq. 5.1** e dos dados da terceira coluna da **Tabela 5.2** pode-se calcular o THD da tensão compensada na carga.

O valor calculado para o THD da tensão compensada é **1,37%**.

Comparando-se o THD da tensão sem compensação (2,78%) com o THD da tensão com compensação (1,37%), nota-se que houve uma redução de 50,72% no THD.

### 5.2.2 – Teste com Carga Não Linear

Para a realização deste teste, utilizou-se como carga do transformador de saída um retificador trifásico alimentando a resistência líquida. Utilizando-se um medidor TRUE RMS de corrente, ajustou-se a corrente de carga (corrente AC, antes do retificador) para 50A.

A **Figura 5.10** apresenta a oscilografia da tensão no TP da fase A antes do compensador entrar em funcionamento.

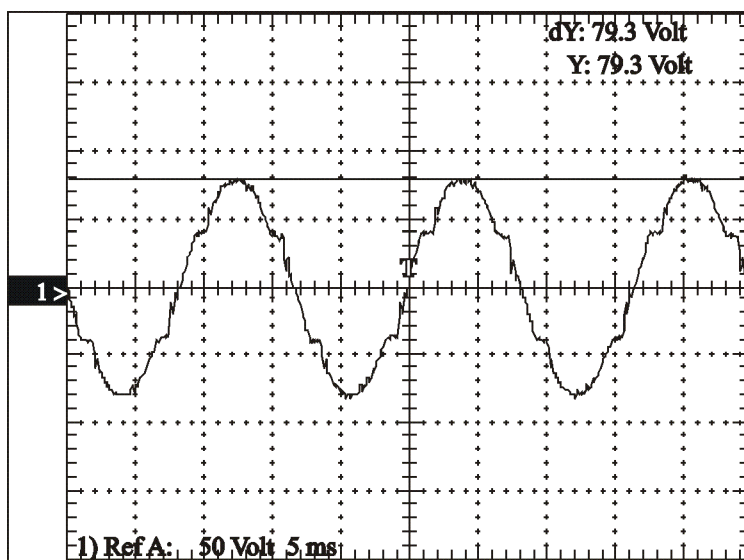


Figura 5.10 – Tensão no TP – Com Carga Não Linear – Sem Compensação

A **Figura 5.11** apresenta o espectro de freqüências da tensão da **Figura 5.10**.

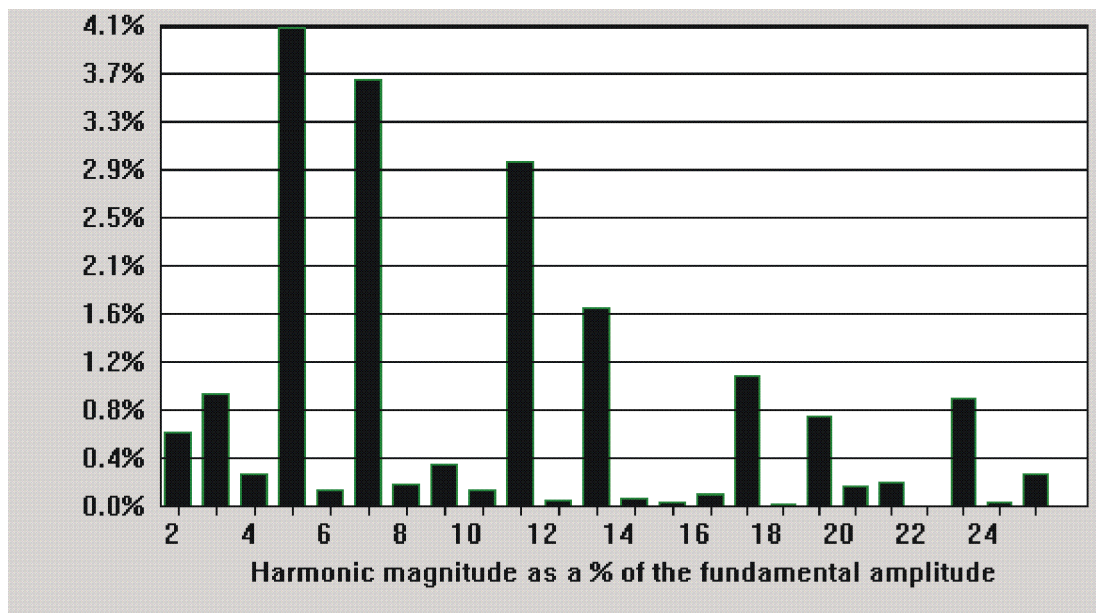


Figura 5.11 – Espectro de freqüências da Tensão no TP – Com Carga Não Linear – Sem Compensação

A **Tabela 5.3** apresenta uma descrição mais detalhada sobre cada componente harmônica do sinal da **Figura 5.10**.

Tabela 5.3

Distorção Harmônica da Tensão no TP – Com Carga Não Linear – Sem Compensação

	Freq	Tensão RMS	Porcentagem em relação à fundamental
Fundamental	59.9 Hz	55.2 V	100.00%
2 <sup>a</sup> harmônica	120 Hz	367 mV	0.66%
3 <sup>a</sup> harmônica	180 Hz	545 mV	0.99%
4 <sup>a</sup> harmônica	240 Hz	168 mV	0.30%
5 <sup>a</sup> harmônica	299 Hz	2.27 V	4.10%
6 <sup>a</sup> harmônica	359 Hz	92.7 mV	0.17%
7 <sup>a</sup> harmônica	419 Hz	2.02 V	3.66%
8 <sup>a</sup> harmônica	479 Hz	122 mV	0.22%
9 <sup>a</sup> harmônica	539 Hz	210 mV	0.38%
10 <sup>a</sup> harmônica	599 Hz	90.8 mV	0.16%
11 <sup>a</sup> harmônica	659 Hz	1.64 V	2.97%
12 <sup>a</sup> harmônica	719 Hz	42.2 mV	0.08%
13 <sup>a</sup> harmônica	778 Hz	950 mV	1.72%
14 <sup>a</sup> harmônica	838 Hz	54.6 mV	0.10%
15 <sup>a</sup> harmônica	898 Hz	28.6 mV	0.05%
16 <sup>a</sup> harmônica	958 Hz	70.7 mV	0.13%
17 <sup>a</sup> harmônica	1.02 kHz	631 mV	1.14%
18 <sup>a</sup> harmônica	1.08 kHz	22.6 mV	0.04%
19 <sup>a</sup> harmônica	1.14 kHz	436 mV	0.79%
20 <sup>a</sup> harmônica	1.2 kHz	110 mV	0.20%
21 <sup>a</sup> harmônica	1.26 kHz	131 mV	0.24%
22 <sup>a</sup> harmônica	1.32 kHz	10.5 mV	0.02%
23 <sup>a</sup> harmônica	1.38 kHz	524 mV	0.95%
24 <sup>a</sup> harmônica	1.44 kHz	37.1 mV	0.07%
25 <sup>a</sup> harmônica	1.5 kHz	164 mV	0.30%

Através de **Eq. 5.1** e dos dados da terceira coluna da **Tabela 5.3** pode-se calcular o THD da tensão não compensada na carga.

O valor calculado para o THD da tensão não compensada é **6,84%**.



A **Figura 5.12** apresenta a oscilografia da tensão no TP da fase A depois do compensador entrar em funcionamento.

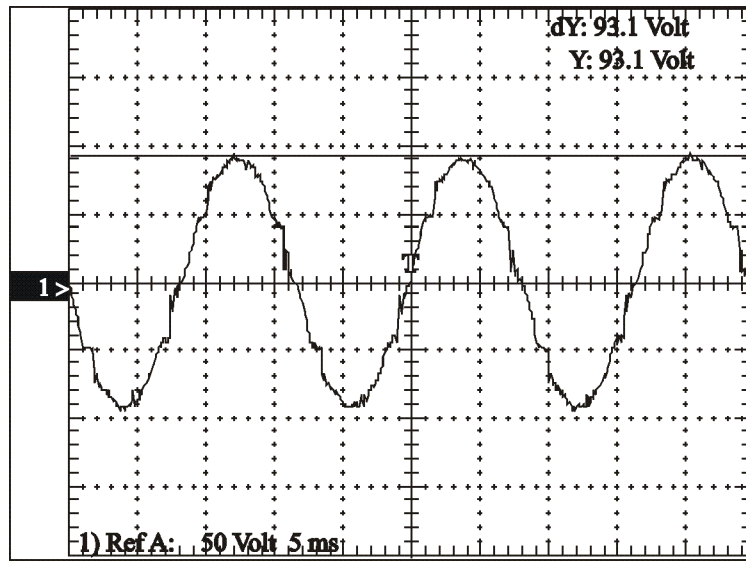


Figura 5.12 – Tensão no TP – Com Carga Não Linear – Com Compensação

A **Figura 5.13** apresenta o espectro de frequências da tensão da **Figura 5.12**.

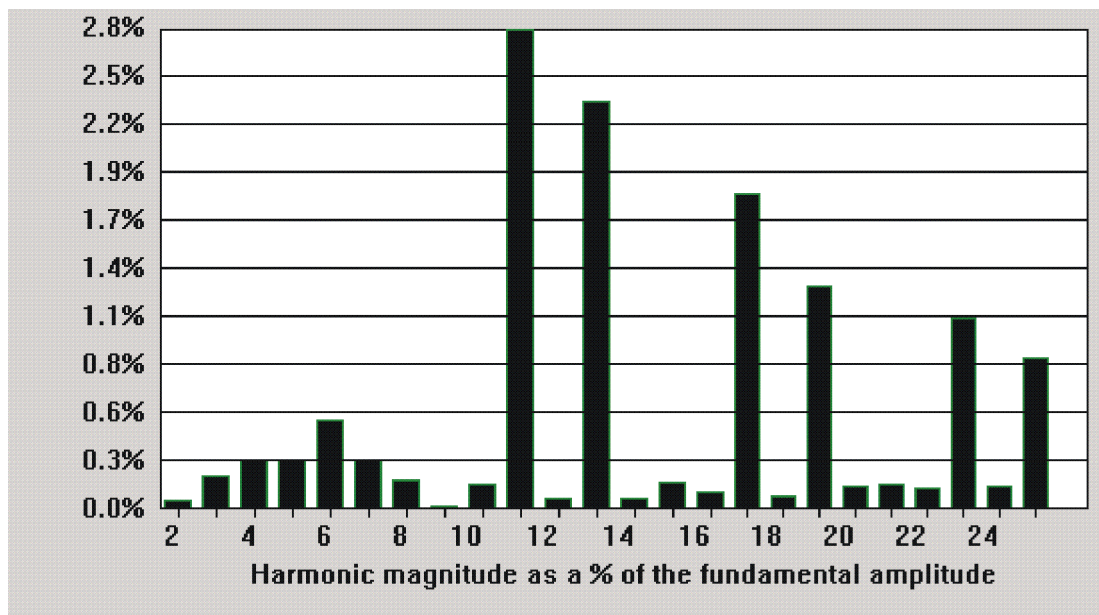


Figura 5.13 – Espectro de frequências da Tensão no TP – Com Carga Não Linear – Com Compensação

A **Tabela 5.4** apresenta uma descrição mais detalhada sobre cada componente harmônica do sinal da **Figura 5.12**.

Tabela 5.4

Distorção Harmônica da Tensão no TP – Com Carga Não Linear – Com Compensação

	Freq	Tensão RMS	Porcentagem em relação à fundamental
Fundamental	60 Hz	64.6 V	100.00%
2 <sup>a</sup> harmônica	120 Hz	37.2 mV	0.06%
3 <sup>a</sup> harmônica	180 Hz	127 mV	0.20%
4 <sup>a</sup> harmônica	240 Hz	187 mV	0.29%
5 <sup>a</sup> harmônica	300 Hz	186 mV	0.29%
6 <sup>a</sup> harmônica	360 Hz	337 mV	0.52%
7 <sup>a</sup> harmônica	420 Hz	189 mV	0.29%
8 <sup>a</sup> harmônica	480 Hz	112 mV	0.17%
9 <sup>a</sup> harmônica	540 Hz	15.9 mV	0.02%
10 <sup>a</sup> harmônica	600 Hz	96.9 mV	0.15%
11 <sup>a</sup> harmônica	660 Hz	1.78 V	2.76%
12 <sup>a</sup> harmônica	720 Hz	45 mV	0.07%
13 <sup>a</sup> harmônica	780 Hz	1.52 V	2.36%
14 <sup>a</sup> harmônica	840 Hz	50.3 mV	0.08%
15 <sup>a</sup> harmônica	900 Hz	108 mV	0.17%
16 <sup>a</sup> harmônica	960 Hz	70.4 mV	0.11%
17 <sup>a</sup> harmônica	1.02 kHz	1.17 V	1.82%
18 <sup>a</sup> harmônica	1.08 kHz	54.6 mV	0.08%
19 <sup>a</sup> harmônica	1.14 kHz	836 mV	1.30%
20 <sup>a</sup> harmônica	1.2 kHz	93 mV	0.14%
21 <sup>a</sup> harmônica	1.26 kHz	102 mV	0.16%
22 <sup>a</sup> harmônica	1.32 kHz	88.2 mV	0.14%
23 <sup>a</sup> harmônica	1.38 kHz	715 mV	1.11%
24 <sup>a</sup> harmônica	1.44 kHz	89.9 mV	0.14%
25 <sup>a</sup> harmônica	1.5 kHz	569 mV	0.88%

Através de **Eq. 5.1** e dos dados da terceira coluna da **Tabela 5.4** pode-se calcular o THD da tensão compensada na carga.

O valor calculado para o THD da tensão compensada é **4,57%**.

Nota-se nas **Figuras 5.10 e 5.12** a presença de alta distorção harmônica. Isto é devido ao fato da carga ser um retificador trifásico. Portanto a corrente que este retificador drena da concessionária apresenta harmônicos na forma ( $6 \cdot n \pm 1$ ), ou seja, 5<sup>o</sup> e 7<sup>o</sup>, 11<sup>o</sup> e 13<sup>o</sup>, 17<sup>o</sup> e 19<sup>o</sup>, etc... Esta corrente com harmônicos ao circular pelo transformador de acoplamento produz uma queda de tensão, proporcional à corrente, portanto contendo harmônicos. Como a impedância do enrolamento é muito alta, a queda de tensão também será muito alta e, conseqüentemente, a distorção harmônica da tensão.

A taxa de amostragem utilizada no equipamento foi 7kHz, isto representa 116 amostras em um período de 60Hz, mas apenas 16 amostras para o sétimo harmônico. Com esta taxa de amostragem, harmônicos de ordem superior a 7 não podem ser

compensados. Neste desenvolvimento, a taxa de amostragem não pôde ser aumentada acima de 7kHz, devido à falta de performance computacional da CPU.

Deve-se ressaltar que, a uma taxa de 7kHz, a CPU precisa finalizar todos os cálculos em um tempo menor do que 1/7000, ou seja 0,143ms.

Embora a CPU seja rápida (586 133Mhz, com performance equivalente à de um Intel 486 DX4-100MHz), ela não está apenas calculando o algoritmo do filtro. Além do algoritmo do filtro, ela precisa gerenciar outras tarefas do Sistema Operacional. Utilizou-se o MS-DOS (que não é um Sistema Operacional Real-Time, como seria necessário) como sistema operacional.

Devido a esta baixa frequência de amostragem, somente o 2º, 3º, 4º, 5º, 6º e 7º harmônicos puderam ser compensados.

Comparando-se a **Figura 5.10** com a **Figura 5.12** pouca diferença visual pode ser observada com relação à compensação de harmônicos. Porém, os espectros das **Figuras 5.11 e 5.13 e Tabelas 5.3 e 5.4** trazem mais informações.

Para o terceiro harmônico houve uma redução de 0,99% para 0,20%, portanto uma redução de 79,8%.

Para o quinto harmônico houve uma redução de 4,10% para 0,29%, portanto uma redução de 92,93%.

Para o sétimo harmônico houve uma redução de 3,66% para 0,29%, portanto uma redução de 92,08%.

Embora um ótimo resultado tenha sido obtido para os harmônicos entre a 2ª e a 7ª ordem, o software implementado nesta CPU não tem controle sobre os harmônicos de ordem superior a 7.

---

Devido a falta de instrumentação necessária não foi possível verificar em quanto o equipamento consegue corrigir um Voltage Sag. Porém, pode-se perceber, por comparação das **Figuras 5.8, 5.10 e 5.12**, que a regulação da tensão na carga, em regime permanente, está funcionando corretamente.

Embora não tenha sido possível a geração de um Voltage Sag controlado para os testes, nota-se que a **Figura 5.10** apresenta uma queda no valor de sua tensão. Isto ocorre devido à alta impedância dos enrolamentos do transformador de acoplamento.

A **Figura 5.8** apresenta a situação ideal de tensão nos TPs, portanto  $115/\sqrt{3}$  V RMS, ou 93,9V de pico.

Sem o funcionamento do Compensador, e com o retificador trifásico consumindo 50A TRUE RMS do transformador de saída, tem-se 79,3V de pico nos TPs, conforme **Figura 5.10**.

Com a entrada em funcionamento do Compensador, a tensão no TP volta para os 93,1V de pico (conforme **Figura 5.12**), que é a tensão nominal; comprovando, assim, a regulação de tensão.



# Conclusão

Este trabalho apresentou algoritmos para o controle de Compensadores Ativos. Esses algoritmos, com exceção da malha de correção de Voltage Sags, podem ser utilizados tanto para implementação de Compensadores Série, quanto para implementação de Compensadores Shunt. A compensação de Voltage Sags só tem sentido em Compensadores do tipo Série.

Mostrou-se que, em ambientes ruidosos, as técnicas convencionais de Compensadores Ativos transportam todo o ruído da aquisição de dados para a saída do sistema. Isto ocorre devido ao fato dos algoritmos decompor os sinais em uma parcela constante e em uma parcela oscilatória. Os algoritmos convencionais descartam a parcela constante e utilizam a parcela oscilatória para gerar as compensações. Quando existe ruído no sistema de aquisição de dados, esse ruído contamina justamente a parcela oscilatória; sendo assim, as técnicas convencionais tentam compensar algo que não existe na realidade, apenas na aquisição de dados.

Desenvolveu-se, então, uma nova técnica, que ao invés de descartar a parcela constante, descarta a parcela oscilatória; descartando, assim, o ruído do sistema de aquisição de dados. O inconveniente desta técnica é que ela utiliza muitos recursos computacionais, uma vez que é necessário o cálculo de uma transformação de Park para cada harmônico que se deseja compensar. Porém, hoje em dia, o custo de processadores mais potentes está cada vez acessível; sendo assim, deve-se pensar em uma relação custo/benefício adaptada às necessidades de cada projeto.

Implementou-se um Compensador Série para correção de Distorções Harmônicas de Tensão e compensação de Voltage Sags, em sistemas de distribuição em 13,8kV.

O equipamento desenvolvido pode ser utilizado para compensar a tensão de cargas que consumam até 75A. Isto é devido a uma limitação dos IGBTs utilizados neste projeto. Pode-se, porém, apenas com a substituição dos IGBTs por outros de maior capacidade de corrente, compensar cargas em uma potência mais elevada. Deve-se lembrar que este projeto foi apenas um protótipo, com o intuito de demonstrar que a compensação ativa em sistemas de distribuição é possível, por isso foram utilizados semicondutores de baixa potência (e baixo custo). Um *upgrade* do equipamento, para altas correntes, pode ser realizado facilmente, apenas com a modificação da parte de potência. A parte dos circuitos eletrônicos e software pode continuar a mesma.

Nota-se, portanto, a utilidade da construção do equipamento de forma modular. Dessa forma pode-se modificar a parte de potência sem a necessidade de modificação da parte eletrônica e vice-versa.

Bons resultados foram obtidos. O compensador conseguiu regular corretamente a tensão no barramento de 13.8kV, embora não fora possível registrar a duração do período transitório.

O compensador também conseguiu eliminar todas as componentes harmônicas a que ele estava habilitado a eliminar. Entretanto, observou-se que uma taxa de amostragem maior traria melhores resultados.

Tem-se duas alternativas para melhorar a performance do equipamento:

1 – A substituição do MS-DOS por um kernel Linux especialmente compilado para aplicações em tempo real. Esta alternativa não requer mudança de Hardware, e muito pouca alteração no Software de Controle precisaria ser feita. A grande desvantagem é que não existe um driver da placa de aquisição de dados para Linux, sendo assim este driver deveria ser escrito “do zero”, aumentando muito o tempo de desenvolvimento.

2 – A substituição do Hardware. O Embedded PC seria substituído por um microcontrolador do tipo DSP (Digital Signal Processor), que já possui instruções apropriadas para aplicações de controle digital em tempo real. Outra vantagem dos DSPs é que alguns deles já possuem os moduladores PWM integrados na mesma pastilha de silício que o processador, gerando assim uma simplificação muito grande no equipamento. Também apresentam Conversores A/D de vários canais e de alta taxa de aquisição na mesma pastilha de silício.

Deve-se ressaltar que o equipamento está utilizando baixa taxa de amostragem (7kHz), não por falha da placa de aquisição de dados (que suporta até 100kHz), mas pela falta de performance do conjunto CPU + MS-DOS, que não permite que muitos cálculos complexos de ponto flutuante sejam executados no intervalo entre duas amostras, se a taxa for maior do que 7kHz. Observa-se que 133MHz para o clock da CPU não é um valor baixo, o grande problema é que o MS-DOS gera interrupções periódicas (para, por exemplo, ler do teclado, escrever na tela, dentre outras tarefas) que não são possíveis de serem desabilitadas, e causam perda de performance em tempo real.

Outro ponto interessante é que mesmo com a utilização de uma taxa de amostragem extremamente alta, existe um limite para a ordem dos harmônicos a serem compensados. E este limite é dado pela máxima frequência de chaveamento dos IGBTs do inversor. Então, considerando um IGBT cuja máxima frequência de chaveamento seja 15kHz, pode-se compensar, satisfatoriamente, até o 19º harmônico. Pois em 15kHz tem-se 13 pontos em um período de 19º harmônico. Para harmônicos de ordem superior a 19, tem-se poucos pontos por período, não sendo suficientes para uma boa compensação.

---

Como fruto deste trabalho dois artigos foram aprovados para publicação nos anais de dois congressos internacionais, sendo eles:

[1] W.C.Santana, L.E.B.da Silva, L.E.L.de Oliveira, V.F.da Silva, G.L. Torres, “Noise-Tolerant Algorithm for Active Power Filters”, in *Proceedings of the IEEE International Symposium on Industrial Electronics – ISIE*, Ajaccio, France, May 2004;

[2] W.C.Santana, L.E.B.da Silva, L.E.L.de Oliveira, V.F.da Silva, G.L. Torres, “13.8kV Series Active Power Filter Implementation using a Noise-Tolerant Algorithm”, in *Proceedings of the IEEE Power Electronics Specialists Conference – PESC*, Aachen, Germany, June 2004.

# ***Apêndice A***

## ***Implementação do Algoritmo de Médias Móveis em uma S-Function***

<b>A.1 - Introdução.....</b>	<b>A - 3</b>
<b>A.2 – O Algoritmo de Médias Móveis.....</b>	<b>A - 3</b>
<b>A.3 – Conversão do Algoritmo para S-Function.....</b>	<b>A - 10</b>
<b>A.4 – Exemplo de utilização da S-Function <i>media_movel</i>.....</b>	<b>A - 13</b>
<b>A.5 - Referências.....</b>	<b>A - 16</b>





## A.1 - Introdução

Um algoritmo de Médias Móveis pode ser facilmente implementado, sob forma de equação recursiva, para ser utilizado dentro de um arquivo **.m** do **Matlab**, ou em qualquer linguagem de programação, como **C** ou **Fortran**.

Porém, as simulações dos algoritmos de controle para os Compensadores Ativos do **Capítulo 3** foram implementadas utilizando o **Simulink**. E o **Simulink** não possui um bloco que implemente as médias móveis, pelo menos não do jeito necessário para os algoritmos.

Por outro lado, o **Simulink** possui um bloco chamado **S-Function**. Com este bloco é possível colocar algoritmos inteiros, escritos em linguagem **C**, dentro deste bloco. Para isto é necessário seguir algumas regras, que estão descritas em [A.1].

## A.2 – O Algoritmo de Médias Móveis

Seja, por exemplo, o sinal periódico triangular (azul) da **Figura A.1** e sua média (vermelho).

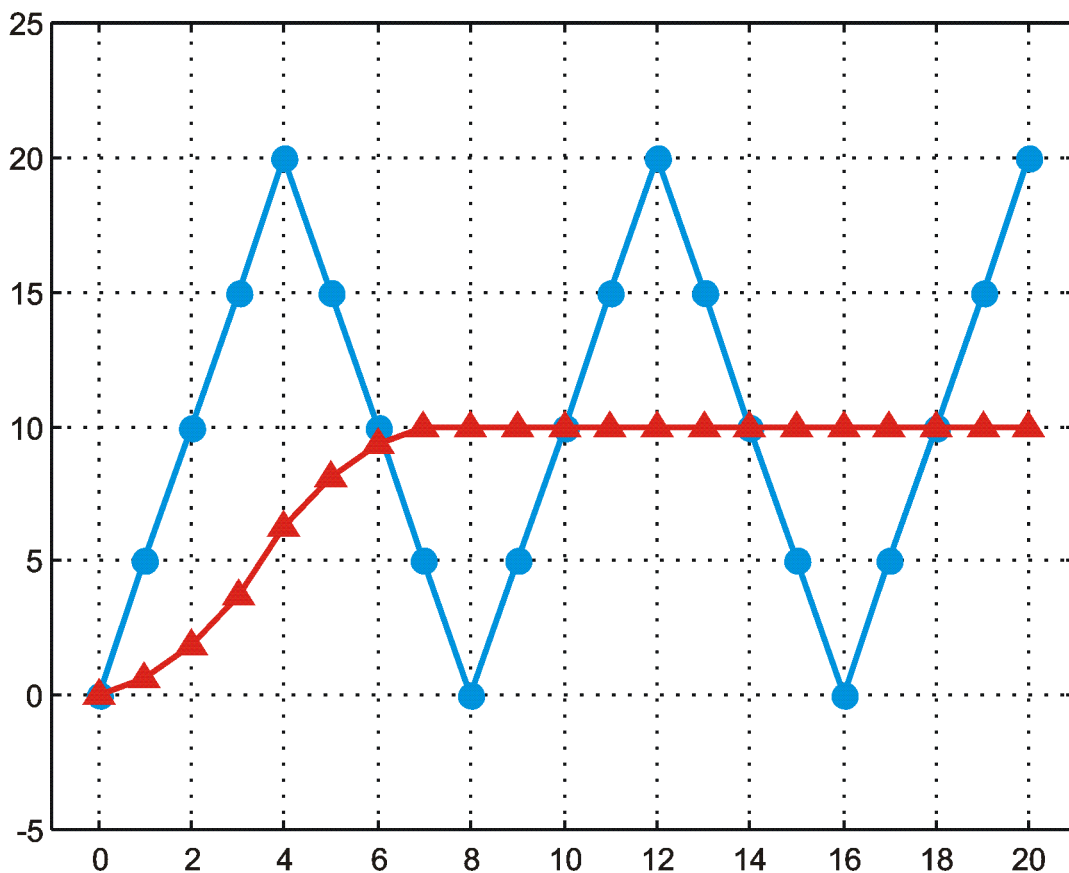


Figura A.1 – Onda Triangular Amostrada e Média Calculada

Uma primeira tentativa para o cálculo da média deste sinal seria somar todos os pontos e dividir a soma pelo número de pontos, ou seja:

$$\text{media} = \frac{0+5+10+15+20+15+ \dots +5+10+15+20}{21} = 10$$

Este tipo de técnica é a mais simples de ser implementada e apresenta a grande vantagem de não apresentar transitório. Ou seja, desde o instante **0**, sabe-se o valor correto da média. Porém, esta técnica só pode ser utilizada quando se possui todos os pontos do sinal de entrada, de forma que a média seja calculada de uma só vez. Como se pode notar, esta técnica não pode ser implementada em algoritmos que funcionem em *Real Time*. Como a essência do Compensador Ativo é a operação em *Real Time* para o cálculo das referências, esta técnica não pode ser utilizada.

Para o cálculo da média do sinal em *Real Time* deve-se utilizar uma média calculada dentro de uma janela que se movimenta acompanhando o deslocamento do tempo, a **Média-Móvel**. O tamanho da janela de tempo deve ser relativo ao número de amostras por período da onda de entrada.

Uma média é uma soma de **n** números, dividida pela quantidade de números que foram somados, ou seja, **n**. O sinal da **Figura A.1** apresenta 8 amostras em 1 período, portanto caso seja necessário retirar a informação do nível DC (a média) deste sinal, deve-se implementar uma média de 8 pontos. Portanto, neste caso, **n=8**.

A seguir têm-se três técnicas para implementação de médias móveis. Sendo que cada uma delas é uma evolução da anterior, visando economia de recursos computacionais.

A primeira técnica consiste na criação de um Buffer com **n** elementos. Este Buffer representa a janela de tempo. Todos os elementos do Buffer são somados e a soma é dividida por **n**. Os elementos do Buffer são sucessivamente deslocados a cada nova amostra, e o processo se repete. As **Figuras A.2** e **A.3** ilustram este processo.

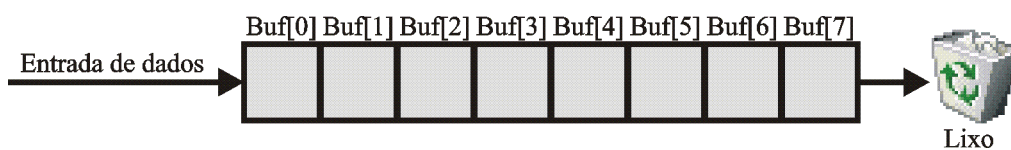


Figura A.2 – Deslocamento de amostras em Buffer com **n** elementos

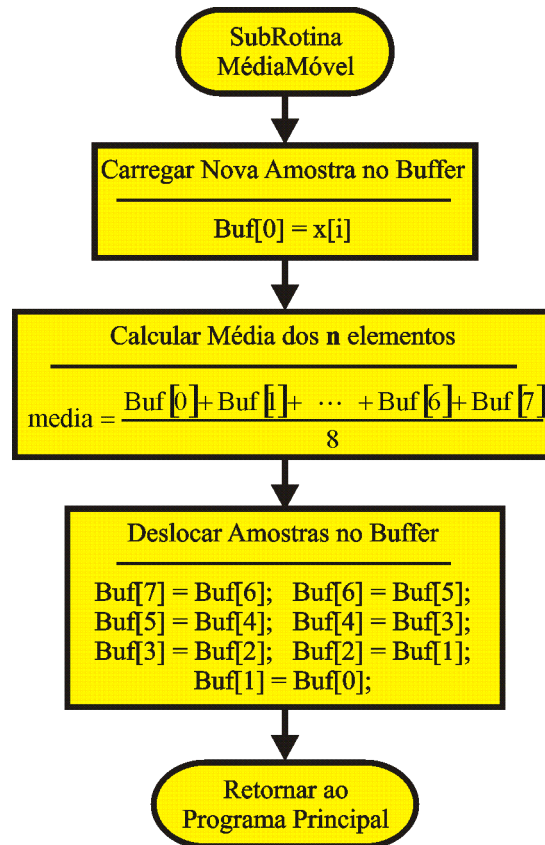


Figura A.3 – Fluxograma para a primeira técnica de Médias Móveis

Como se pode perceber, para se implementar esta técnica, é necessário realizar **n-1** somas, uma divisão e um deslocamento de posição para cada elemento do Buffer (portanto **n** deslocamentos); tudo isso a cada nova amostra. Nota-se que esta técnica consome muitos recursos computacionais. Deve-se ressaltar que a CPU, além do algoritmo de médias, deve realizar outros cálculos importantes, sendo assim, cada tarefa deve ser executada com o máximo de economia de recursos.

---

A evolução da primeira para a segunda técnica consiste em se reaproveitar a soma das amostras anteriores, criando um algoritmo recursivo. Dessa maneira, substitui-se as **n-1** somas por apenas 1 soma e 1 subtração, pois esta técnica adiciona à soma anterior o valor da nova amostra e retira o valor da última (a que ocorreu a **n+1** instantes passados). Deve-se, portanto criar um Buffer com **n+1** amostras.

Deve-se ressaltar que apenas o primeiro e o último elemento do Buffer são efetivamente utilizados nas operações, porém todos os **n+1** são necessários. Isto é necessário, pois o algoritmo precisa conhecer o histórico das amostras.

O algoritmo utiliza a amostra que ocorreu no instante atual (índice **0**) e a que ocorreu a **n+1** instantes passados (índice **n**); porém, na próxima rodada a amostra de índice **n** será a que no instante atual detém o índice **n-1**. E assim sucessivamente. Portanto, as amostras intermediárias (de **1** até **n-1**) precisam ser armazenadas.

As Figuras A.4 e A.5 ilustram este processo.

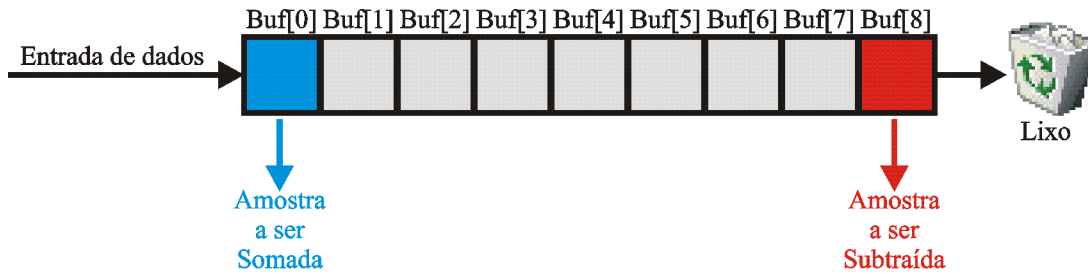


Figura A.4 - Deslocamento de amostras em Buffer com  $n+1$  elementos

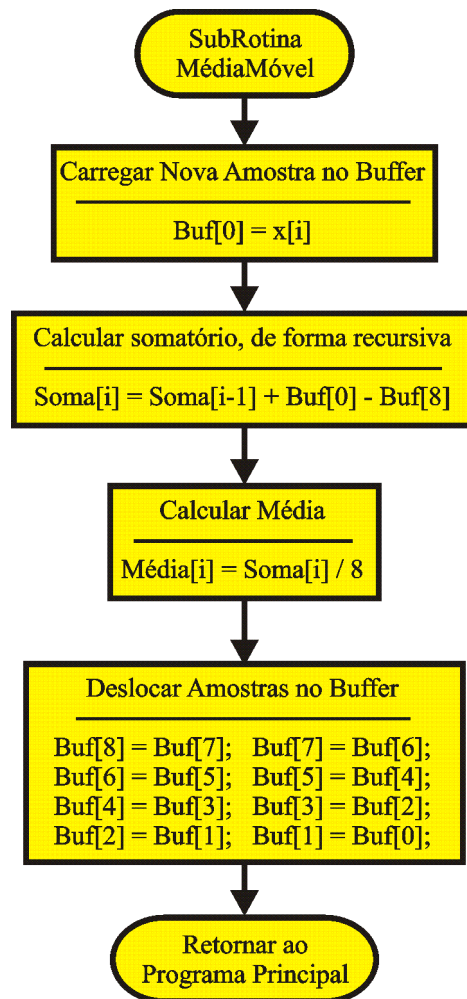


Figura A.5 - Fluxograma para a segunda técnica de Médias Móveis

A **Tabela A.1** mostra como as variáveis se modificariam ao longo da execução da segunda técnica de médias móveis, para o exemplo da **Figura A.1**.

Tabela A.1 – Execução do Segundo Algoritmo

i	x[i]	Buf[0]	Buf[1]	Buf[2]	Buf[3]	Buf[4]	Buf[5]	Buf[6]	Buf[7]	Buf[8]	Soma [i-1]	Soma[i]	Média[i]
0	0	0	0	0	0	0	0	0	0	0	0	0+0-0 = 0	0/8
1	5	5	0	0	0	0	0	0	0	0	0	0+0-5 = 5	5/8
2	10	10	5	0	0	0	0	0	0	0	5	5+10-0 = 15	15/8
3	15	15	10	5	0	0	0	0	0	0	15	15+15-0 = 30	30/8
4	20	20	15	10	5	0	0	0	0	0	30	30+20-0 = 50	50/8
5	15	15	20	15	10	5	0	0	0	0	50	50+15-0 = 65	65/8
6	10	10	15	20	15	10	5	0	0	0	65	65+10-0 = 75	75/8
7	5	5	10	15	20	15	10	5	0	0	75	75+5-0 = 80	80/8 = 10
8	0	0	5	10	15	20	15	10	5	0	80	80+0-0 = 80	80/8 = 10
9	5	5	0	5	10	15	20	15	10	5	80	80+5-5 = 80	80/8 = 10
10	10	10	5	0	5	10	15	20	15	10	80	80+10-10 = 80	80/8 = 10
11	15	15	10	5	0	5	10	15	20	15	80	80+15-15 = 80	80/8 = 10
12	20	20	15	10	5	0	5	10	15	20	80	80+20-20 = 80	80/8 = 10
13	15	15	20	15	10	5	0	5	10	15	80	80+15-15 = 80	80/8 = 10
14	10	10	15	20	15	10	5	0	5	10	80	80+10-10 = 80	80/8 = 10
15	5	5	10	15	20	15	10	5	0	5	80	80+5-5 = 80	80/8 = 10
16	0	0	5	10	15	20	15	10	5	0	80	80+0-0 = 80	80/8 = 10
17	5	5	0	5	10	15	20	15	10	5	80	80+5-5 = 80	80/8 = 10
18	10	10	5	0	5	10	15	20	15	10	80	80+10-10 = 80	80/8 = 10
19	15	15	10	5	0	5	10	15	20	15	80	80+15-15 = 80	80/8 = 10
20	20	20	15	10	5	0	5	10	15	20	80	80+20-20 = 80	80/8 = 10

Nota-se que as **n+1** somas foram eliminadas; entretanto o algoritmo ainda não atingiu sua eficiência máxima, pois ainda é necessário um deslocamento para cada amostra no Buffer (totalizando **n+1** deslocamentos). Isto foi resolvido com a terceira técnica, que será discutida a seguir.

A evolução da segunda para a terceira técnica consiste em não realizar deslocamentos de amostras dentro do Buffer, ao invés disto, mantém-se fixos os elementos e quem se desloca é o índice do Buffer, para indicar quais as amostras que devem ser subtraídas e somadas.

O índice do Buffer é um contador, que varia de **0** até **n**. Quando a contagem atinge o valor **n**, o próximo valor de índice é resetado para **0**.

A amostra atual é sempre armazenada na posição do Buffer indicada pelo índice. Esta amostra também é o valor que deve ser acrescentado à soma anterior.

O valor a ser subtraído é o que estiver armazenado na posição imediatamente superior da indicada pelo índice. Quando o valor do índice for igual a **n**, não existe posição superior; neste caso a amostra a ser subtraída será a que ocupa a posição **0**.

Resumindo, o elemento a ser subtraído é sempre aquele que, na próxima rodada, será substituído pela nova amostra.

As **Figuras A.6** e **A.7** ilustram este processo.

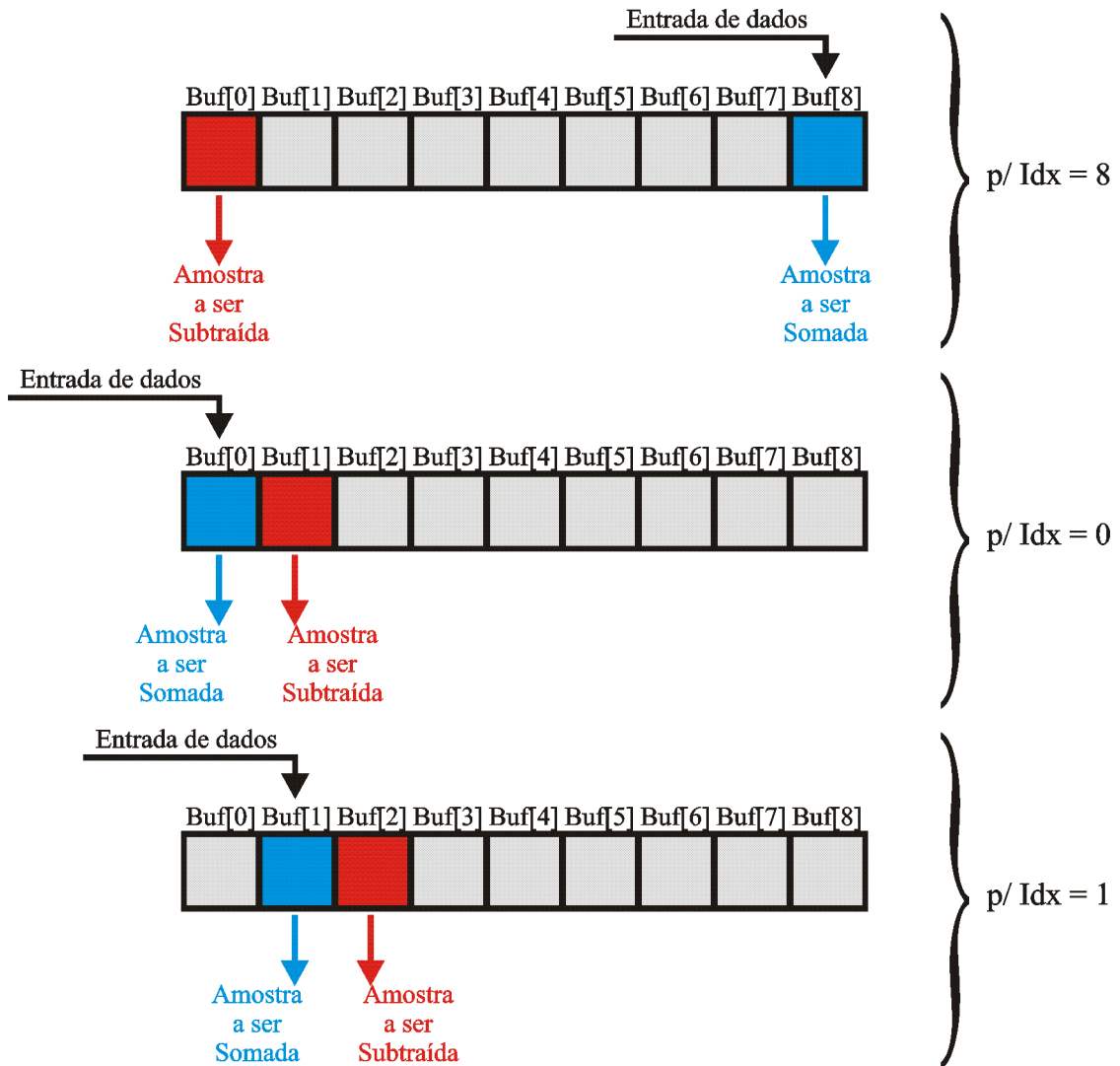


Figura A.6 - Deslocamento de índice em Buffer com  $n+1$  elementos

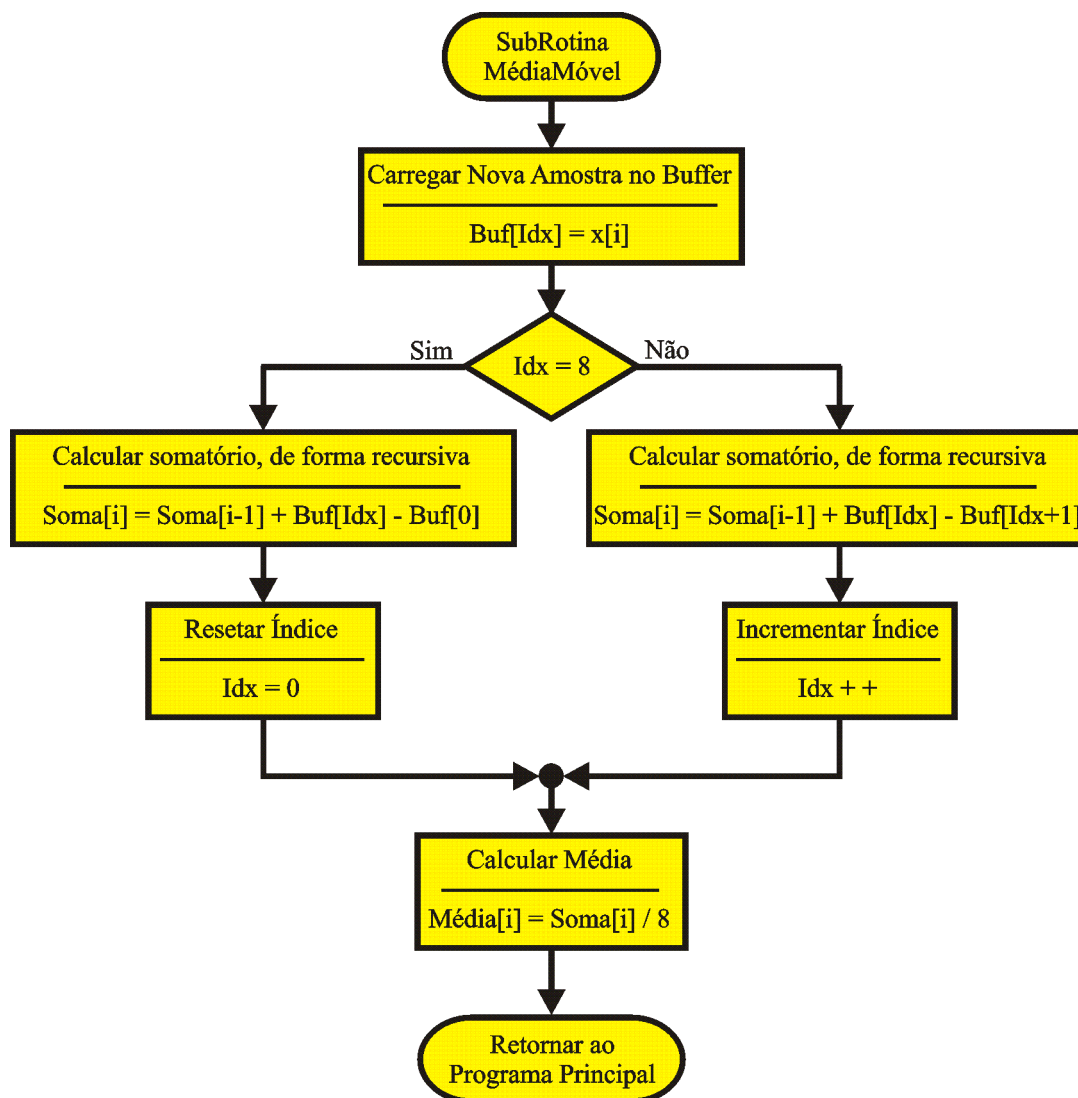


Figura A.7 - Fluxograma para a terceira técnica de Médias Móveis

A **Tabela A.2** mostra como as variáveis se modificariam ao longo da execução da terceira técnica de médias móveis, para o exemplo da **Figura A.1**.

Deve-se ressaltar que as três técnicas apresentadas para média móvel possuem exatamente a mesma saída, a onda na cor vermelho da **Figura A.1**. A diferença dos três métodos é a velocidade com que a CPU termina a execução dos cálculos.

Nota-se, na **Figura A.1**, um transitório de um período de sinal. Esta é uma desvantagem do método de médias móveis para execução em *Real Time*, e isto não pode ser evitado.

Este transitório será **SEMPRE** do tamanho da janela de tempo utilizada. Quanto maior o tamanho da janela maior será o transitório, entretanto a precisão do cálculo da média também será maior. Portanto há um compromisso entre a duração do transitório e qualidade do sinal filtrado na saída.

Tabela A.2 – Execução do Terceiro Algoritmo

i	x [i]	Idx	Buf [0]	Buf [1]	Buf [2]	Buf [3]	Buf [4]	Buf [5]	Buf [6]	Buf [7]	Buf [8]	Soma [i-1]	Soma[i]	Média[i]
0	0	0	0	0	0	0	0	0	0	0	0	0	0+0-0 = 0	0/8
1	5	1	0	5	0	0	0	0	0	0	0	0	0+0-5 = 5	5/8
2	10	2	0	5	10	0	0	0	0	0	0	5	5+10-0 = 15	15/8
3	15	3	0	5	10	15	0	0	0	0	0	15	15+15-0 = 30	30/8
4	20	4	0	5	10	15	20	0	0	0	0	30	30+20-0 = 50	50/8
5	15	5	0	5	10	15	20	15	0	0	0	50	50+15-0 = 65	65/8
6	10	6	0	5	10	15	20	15	10	0	0	65	65+10-0 = 75	75/8
7	5	7	0	5	10	15	20	15	10	5	0	75	75+5-0 = 80	80/8 = 10
8	0	8	0	5	10	15	20	15	10	5	0	80	80+0-0 = 80	80/8 = 10
9	5	0	5	5	10	15	20	15	10	5	0	80	80+5-5 = 80	80/8 = 10
10	10	1	5	10	10	15	20	15	10	5	0	80	80+10-10 = 80	80/8 = 10
11	15	2	5	10	15	15	20	15	10	5	0	80	80+15-15 = 80	80/8 = 10
12	20	3	5	10	15	20	20	15	10	5	0	80	80+20-20 = 80	80/8 = 10
13	15	4	5	10	15	20	15	15	10	5	0	80	80+15-15 = 80	80/8 = 10
14	10	5	5	10	15	20	15	10	10	5	0	80	80+10-10 = 80	80/8 = 10
15	5	6	5	10	15	20	15	10	5	5	0	80	80+5-5 = 80	80/8 = 10
16	0	7	5	10	15	20	15	10	5	0	0	80	80+0-0 = 80	80/8 = 10
17	5	8	5	10	15	20	15	10	5	0	5	80	80+5-5 = 80	80/8 = 10
18	10	0	10	10	15	20	15	10	5	0	5	80	80+10-10 = 80	80/8 = 10
19	15	1	10	15	15	20	15	10	5	0	5	80	80+15-15 = 80	80/8 = 10
20	20	2	10	15	20	20	15	10	5	0	5	80	80+20-20 = 80	80/8 = 10

Dessa forma, a terceira técnica analisada será o algoritmo utilizado para implementar as médias móveis no **Simulink**.

### A.3 – Conversão do Algoritmo para S-Function

A seguir tem-se um resumo dos passos para a conversão do algoritmo de médias móveis para uma **S-Function**. É altamente recomendável a leitura de [A.1], onde todos os detalhes e passos intermediários são explicados.

Primeiramente deve-se alterar o arquivo **sfuntmpl.c**, localizado em *matlabroot\simulink\src\*, sendo que *matlabroot* é o caminho para o diretório onde o **Matlab** se encontra instalado. Este arquivo é um *template* para criação de **S-Functions**, ou seja, ele já possui um “esqueleto básico”, e deve-se apenas acrescentar o algoritmo desejado.

Salvar este arquivo com outro nome, por exemplo, **media\_movel.c**.  
 Compilar este arquivo → No prompt de comando do Matlab digitar:

*mex media\_movel.c*

Como resultado da compilação será gerado um arquivo do tipo **.dll**, neste caso, **media\_movel.dll**. Este é o arquivo que será utilizado pelo bloco **S-Function** do **Simulink**.

A seguir tem-se a listagem do arquivo **media\_movel.c**, criado a partir do *template sfuntmpl.c*.



## Implementação do Algoritmo de Médias Móveis em uma S-Function

```
////////// Begin of media_movel.c ////////////////////////////////////////////
/*
 * File : media_movel.c
 * Abstract:
 * A C-file S-function for Moving Average calculation
 *
 * See simulink/src/sfunmpl.doc
 */

#define S_FUNCTION_NAME media_movel
#define S_FUNCTION_LEVEL 2
#define PARAM0(S) ssGetSFcnParam(S, 0)
#define NPARAMS 1
// #define NBuf 17

#include "simstruc.h"

#define MDL_CHECK_PARAMETERS
#if defined(MDL_CHECK_PARAMETERS) && defined(MATLAB_MEX_FILE)
/* Function: mdlCheckParameters =====
 * Abstract:
 * Validate our parameters to verify they are okay.
 */
static void mdlCheckParameters(SimStruct *S)
{
    /* Check 1st parameter: Lower Bound (LB) parameter */
    {
        if (!mxIsDouble(PARAM0(S)) ||
            mxGetNumberOfElements(PARAM0(S)) != 1) {
            ssSetErrorStatus(S, "1st parameter to S-function must be a "
                "scalar \"Numero de Pontos na Janela\" which limits "
                "the lower value of the integrator");
            return ;
        }
    }
}
#endif /* MDL_CHECK_PARAMETERS */

/* Function: mdlInitializeSizes =====
 * Abstract:
 * Setup sizes of the various vectors.
 */
static void mdlInitializeSizes(SimStruct *S)
{
    int_T NBuf;
    ssSetNumSFcnParams(S, NPARAMS); /* Number of expected parameters */
    #if defined(MATLAB_MEX_FILE)
        if (ssGetNumSFcnParams(S) == ssGetSFcnParamsCount(S))
        {
            mdlCheckParameters(S);
            if (ssGetErrorStatus(S) != NULL)
            {
                return ;
            }
        } else
        {
            return ; /* Parameter mismatch will be reported by Simulink */
        }
    #endif
    if (!ssSetNumInputPorts(S, 1)) return ;
    ssSetInputPortWidth(S, 0, DYNAMICALLY_SIZED);
    ssSetInputPortDirectFeedThrough(S, 0, 1);
    if (!ssSetNumOutputPorts(S, 1)) return ;
    ssSetOutputPortWidth(S, 0, DYNAMICALLY_SIZED);
    ssSetNumSampleTimes(S, 1);
    // NBuf = mxGetNumberOfElements(PARAM0(S));
    NBuf = *mxGetPr(PARAM0(S));
    ssSetNumRWork(S, NBuf+1);
    ssSetNumIWork(S, 1);
    /* Take care when specifying exception free code - see sfunmpl.doc */
    ssSetOptions(S, SS_OPTION_EXCEPTION_FREE_CODE);
}
}
```

```

/* Function: mdlInitializeSampleTimes =====
* Abstract:
* Specifiy that we inherit our sample time from the driving block.
*/
static void mdlInitializeSampleTimes(SimStruct *S)
{
    ssSetSampleTime(S, 0, INHERITED_SAMPLE_TIME);
    ssSetOffsetTime(S, 0, 0.0);
}

#define MDL_INITIALIZE_CONDITIONS
/* Function: mdlInitializeConditions =====
* Abstract:
* Initialize both continuous states to one.
*/
static void mdlInitializeConditions(SimStruct *S)
{
    int_T *IdxBuf = ssGetIWork(S);
    real_T *BufEntrada = ssGetRWork(S);
    int_T NBuf = *mxGetPr(PARAM0(S));
    int i;
    IdxBuf[0] = 0;
    for(i=0;i<NBuf+1;i++)
        BufEntrada[i]=0.0;
} /* end mdlInitializeConditions */

/* Function: mdlOutputs =====
* Abstract:
* Moving Average Calculation
*/
static void mdlOutputs(SimStruct *S, int_T tid)
{
    int_T i;
    InputRealPtrsType entrada = ssGetInputPortRealSignalPtrs(S,0);
    real_T *y = ssGetOutputPortRealSignal(S,0);
    int_T width = ssGetOutputPortWidth(S,0);
    int_T *IdxBuf = ssGetIWork(S);
    real_T *BufEntrada = ssGetRWork(S);
    int_T NBuf = *mxGetPr(PARAM0(S));
    for (i=0; i<width; i++)
    {
        BufEntrada[IdxBuf[0]]=*entrada[i];
        if (IdxBuf[0]==NBuf-1)
        {
            BufEntrada[NBuf]=BufEntrada[NBuf]+BufEntrada[IdxBuf[0]]-
            BufEntrada[0];
            IdxBuf[0]=0;
        }
        else
        {
            BufEntrada[NBuf]=BufEntrada[NBuf]+BufEntrada[IdxBuf[0]]-
            BufEntrada[IdxBuf[0]+1];
            IdxBuf[0]++;
        }
        *y+=BufEntrada[NBuf]/(NBuf-1);
    }
}

/* Function: mdlTerminate =====
* Abstract:
* No termination needed, but we are required to have this routine.
*/
static void mdlTerminate(SimStruct *S)
{
}

#ifdef MATLAB_MEX_FILE /* Is this file being compiled as a MEX-file? */
#include "simulink.c" /* MEX-file interface mechanism */
#else
#include "cg_sfun.h" /* Code generation registration function */
#endif
////////// End of media_movel.c //////////////////////////////////////

```

#### A.4 – Exemplo de utilização da S-Function *media\_movel*

Após a geração do arquivo **media\_movel.dll**, basta inserir um bloco do tipo **S-Function** em uma simulação do **Simulink**. A **Figura A.8** mostra a localização deste bloco no *menu* do **Simulink**.

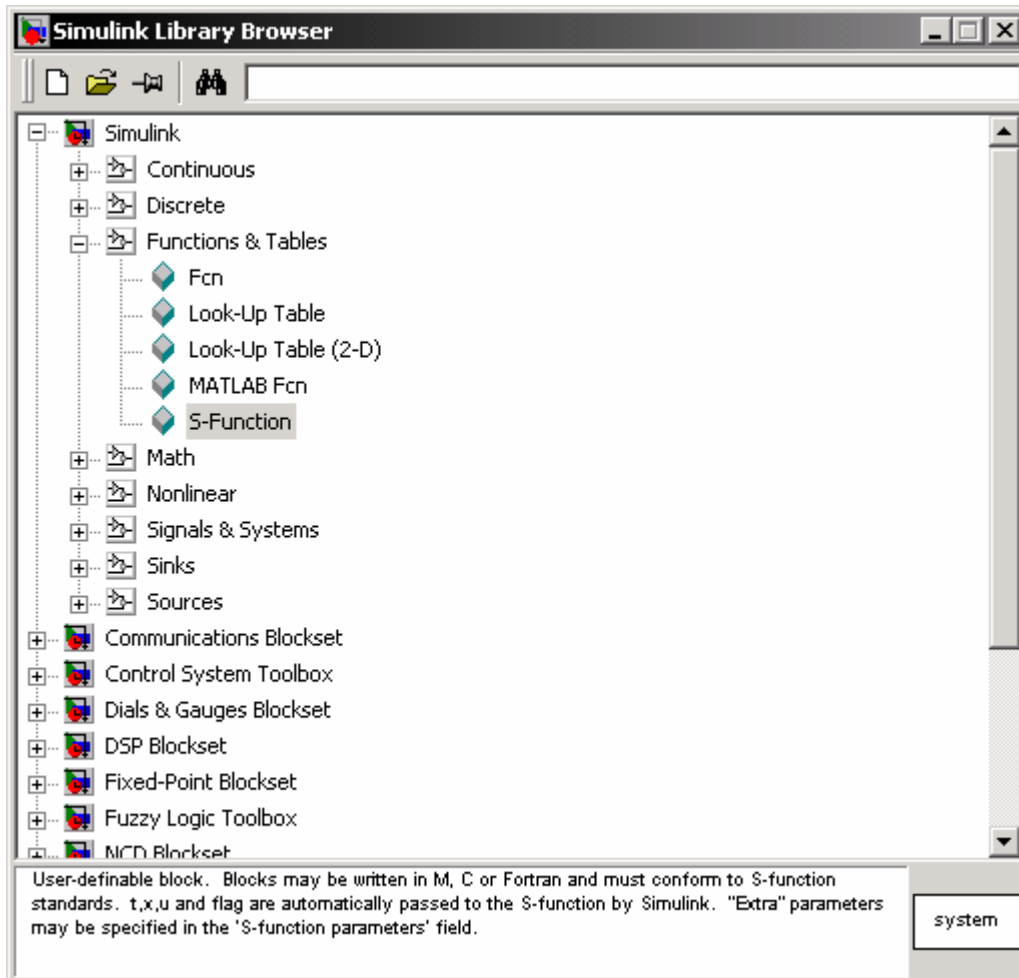


Figura A.8 – Localização do bloco **S-Function**

A **Figura A.9** mostra um exemplo de aplicação para este bloco

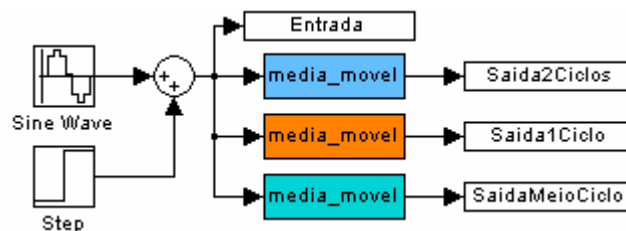


Figura A.9 – Simulação utilizando a S-Function *media\_movel*

Os bloco destacados, na **Figura A.9**, são os blocos **S-Function** inseridos na simulação. Um *Double-Click* nestes blocos resulta em um *menu* de opções, como o da **Figura A.10**.

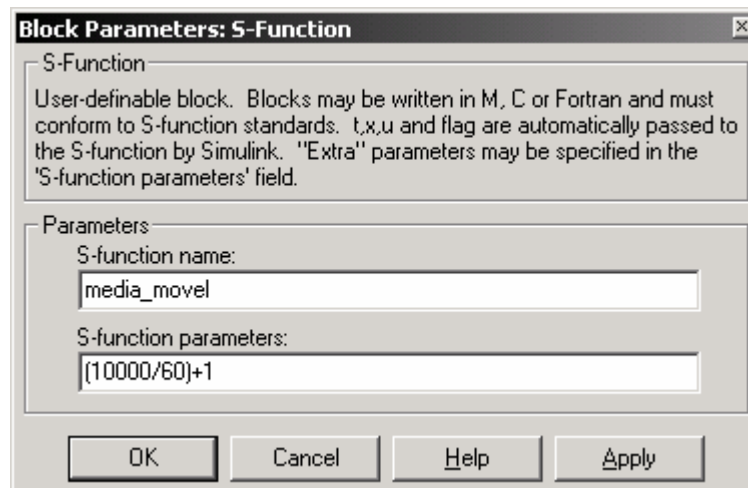


Figura A.10 – Opções para o bloco S-Function

No campo *S-function name* deve-se digitar o nome do arquivo **.dll** gerado no processo de compilação. Neste caso, **media\_movel**.

No campo *S-function parameters* deve-se digitar a lista de parâmetros, que é específica para cada tipo de **S-Function**. No caso das médias móveis esta lista é composta por um único elemento e representa o tamanho do Buffer da média.

O sinal de entrada da simulação da **Figura A.9** é uma onda de 60Hz sobreposta a um degrau. Deseja-se eliminar a parte oscilatória, sendo assim, conforme discutido na **seção A.2**, deve-se utilizar um Buffer com **n+1** pontos, sendo que **n** deve ser o número de pontos em 1 período. A frequência de amostragem do sinal de entrada é 10kHz, portanto em um período de 60Hz tem-se **10000/60** pontos. Para uma filtragem correta do sinal de entrada, com o menor transitório possível, o parâmetro no campo *S-function parameters* deve conter o valor **(10000/60)+1**, ou seja **n+1**.

A simulação da **Figura A.9** utiliza 3 blocos de médias móveis configuradas com tamanhos de Buffers diferentes, de forma que uma comparação possa ser feita com relação à qualidade da filtragem e à duração do transitório.

O bloco na cor azul está configurado com **(2\*10000/60)+1**, ou seja, Buffer com tamanho de 2 ciclos de 60Hz.

O bloco na cor laranja está configurado com **(1\*10000/60)+1**, ou seja, Buffer com tamanho de 1 ciclo de 60Hz.

O bloco na cor verde está configurado com **(0.5\*10000/60)+1**, ou seja, Buffer com tamanho de  $\frac{1}{2}$  ciclo de 60Hz.

Como resultado desta simulação, gerou-se a **Figura A.11**.

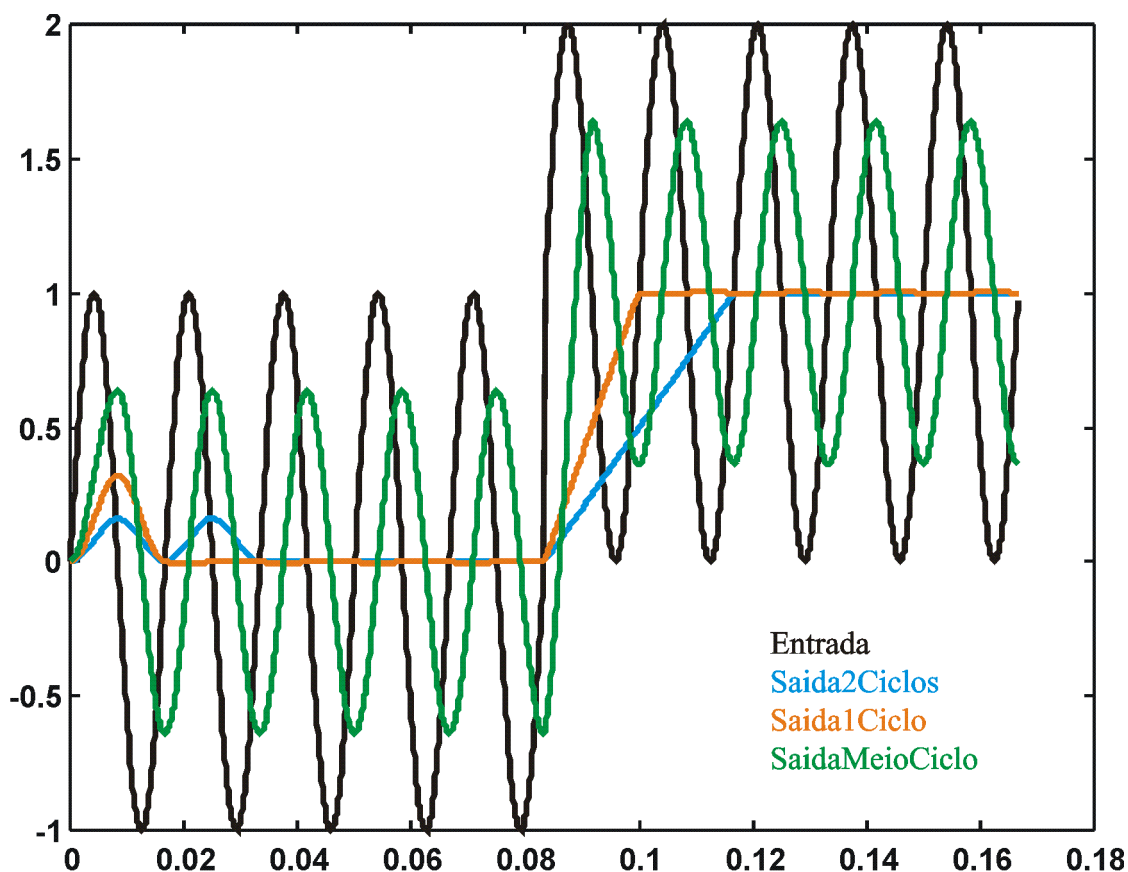


Figura A.11 – Efeito do tamanho da janela de tempo sobre as médias móveis

Da **Figura A.11** pode-se notar que o sinal na cor azul apresenta o maior transiênto (2 ciclos de 60Hz); enquanto que o sinal na cor verde não apresenta filtragem satisfatória, pois o tamanho de buffer é insuficiente (1/2 ciclo de 60Hz). Nota-se que o bloco configurado para Buffer de 1 ciclo foi o que apresentou a melhor performance, pois conseguiu filtrar as oscilações com o menor transiênto (1 ciclo).

Este exemplo utilizou como entrada um sinal oscilatório de apenas 1 componente de frequência. Nas aplicações de Compensadores Ativos os sinais são compostos por várias frequências. Portanto, deve-se utilizar um Buffer grande o suficiente para filtrar a componente com o maior período (frequência mais baixa).

## **A.5 - Referências**

[A.1] “Writing S-Functions”, The MathWorks Inc, USA, October, 1998;

# ***Apêndice B***

## ***PLL – Phase Locked Loop***

<b>B.1 - Introdução</b> .....	B - 3
<b>B.2 - PLL Trifásica</b> .....	B - 4
<b>B.2.1 - Cálculo dos ganhos do Controlador PI</b> .....	B - 6
<b>B.3 - Simulação Computacional</b> .....	B - 7
<b>B.4 - Referências</b> .....	B - 10





## B.1 - Introdução

O Compensador precisa inserir um sinal de compensação que esteja perfeitamente sincronizado com a onda a ser compensada. Sendo assim, necessita-se um sistema que detecte, precisamente, a frequência e a fase desta onda. Este sistema é uma PLL (Phase Locked Loop).

Uma PLL é uma malha de controle cuja função é manter dois sinais sincronizados. Isto é possível analisando-se a diferença de fase entre os dois sinais e forçando este erro até zero. A **Figura B.1** ilustra este processo.

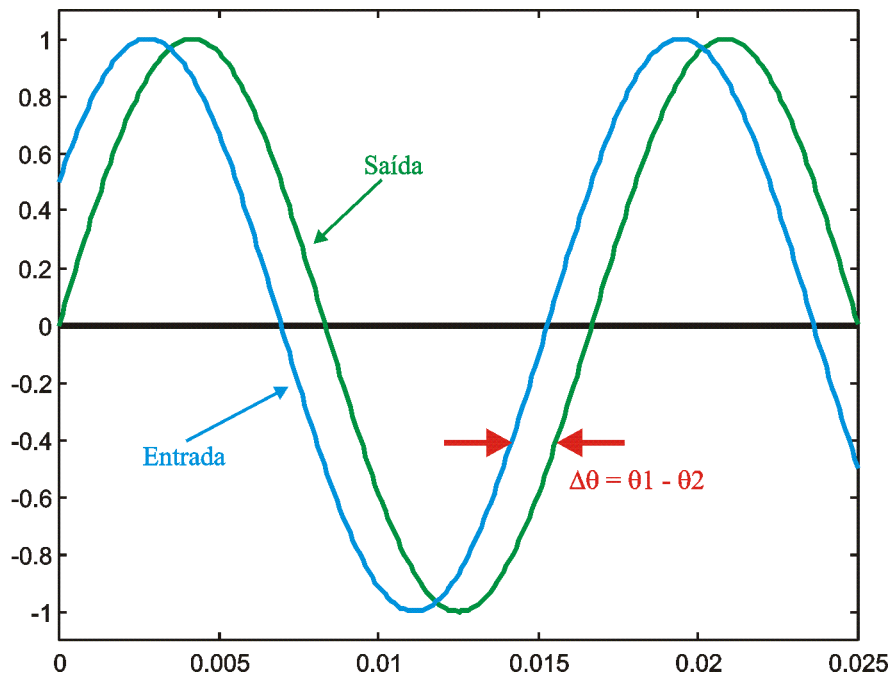


Figura B.1 – Defasagem entre os sinais de entrada e saída da PLL

Quando o erro  $\Delta\theta$  for zero significa que os sinais de entrada e de saída estão sincronizados em frequência e em fase.

Circuitos com PLL são muito utilizados em sistemas de comunicação, sendo que diversos textos foram elaborados para estas aplicações, dentre eles o livro [B.1]. Entretanto os circuitos descritos nestes textos são monofásicos; e, portanto não apresentam o desempenho desejado em circuitos trifásicos.

Uma PLL monofásica até poderia ser utilizada no compensador, como foi feito em [B.2], porém a confiabilidade desta técnica é baixa; uma vez que a PLL sincroniza os sinais de saída baseando-se apenas em uma entrada. Caso ocorra uma falta na fase que a PLL está monitorando, perde-se o sincronismo do sistema inteiro.

Deve-se, portanto, utilizar uma PLL trifásica. A referência [B.3] trata deste assunto.

## B.2 - PLL Trifásica

O método de PLL trifásica proposto em [B.3] utiliza o mesmo conceito utilizado nos algoritmos do **Capítulo 3**, a transformação de coordenadas.

Este método será descrito aqui novamente, porém de forma resumida.

Tem-se como sinal de entrada o sistema trifásico, representado pelas equações **(Eq. B.1)**, **(Eq. B.2)** e **(Eq. B.3)**.

$$va_+ = U_+ \sin(\omega t + \theta_1) \quad (\text{Eq. B.1})$$

$$vb_+ = U_+ \sin(\omega t + \theta_1 - 120^\circ) \quad (\text{Eq. B.2})$$

$$vc_+ = U_+ \sin(\omega t + \theta_1 + 120^\circ) \quad (\text{Eq. B.3})$$

Aplica-se a transformação de Clarke ao sistema trifásico e tem-se como resultado 2 vetores ortogonais estáticos (**alfa** e **beta**). Aos sinais **alfa** e **beta** é aplicada a transformação de Park e como resultado tem-se 2 vetores ortogonais síncronos com a frequência da rede elétrica (**d<sub>+</sub>** e **q<sub>+</sub>**). Os passos intermediários para se chegar às equações **(Eq. B.4)** e **(Eq. B.5)** podem ser vistos no **Capítulo 3**.

$$vd_+ = U_+ \cdot \sqrt{\frac{3}{2}} \cdot \sin(\theta_1 - \theta_2) \quad (\text{Eq. B.4})$$

$$vq_+ = -U_+ \cdot \sqrt{\frac{3}{2}} \cdot \cos(\theta_1 - \theta_2) \quad (\text{Eq. B.5})$$

Nas equações **(Eq. B.4)** e **(Eq. B.5)** tem-se uma subtração de dois ângulos **θ<sub>1</sub>** e **θ<sub>2</sub>**. **θ<sub>1</sub>** é o ângulo de referência do sistema trifásico, e **θ<sub>2</sub>** é o ângulo usado no cálculo na Transformação de Park. **θ<sub>2</sub>** é o ângulo que determina a velocidade angular com que os vetores **d<sub>+</sub>** e **q<sub>+</sub>** rotacionam. Deseja-se que os vetores rotacionem com velocidade angular igual a da rede elétrica. Sendo assim a taxa de variação de **θ<sub>1</sub>** deve ser igual a taxa de variação de **θ<sub>2</sub>**. Isto é conseguido utilizando o ângulo estimado pela PLL como **θ<sub>2</sub>**.

A PLL irá trabalhar para fazer com que **θ<sub>2</sub>** se iguale a **θ<sub>1</sub>**, tornando o erro igual a zero, e conseqüentemente sincronizando a saída com a entrada.

Nota-se nas equações **(Eq. B.4)** e **(Eq. B.5)** que quando **θ<sub>1</sub>** for igual a **θ<sub>2</sub>** **vd<sub>+</sub>** será igual a zero e **vq<sub>+</sub>** será igual a uma constante. Deve-se, portanto implementar uma malha de controle que atue no sinal de saída forçando **vd<sub>+</sub>** se aproximar de zero. Quando isto ocorrer, a saída estará sincronizada com a entrada. Esta malha de controle é justamente a PLL.

A malha poderia ser construída em torno de **vq<sub>+</sub>**, porém ela ficaria dependente da amplitude dos sinais de entrada, ou seja, da tensão da rede elétrica. Como as tensões da rede estão sujeitas a diversos problemas de qualidade de energia (conforme **Capítulo 2**), a malha não ficaria muito estável. Sendo assim constrói-se a malha em torno de **vd<sub>+</sub>**, que na condição **θ<sub>1</sub>=θ<sub>2</sub>** não depende das tensões de entrada.

Tem-se na **Figura B.2** um esquema para implementação da PLL.

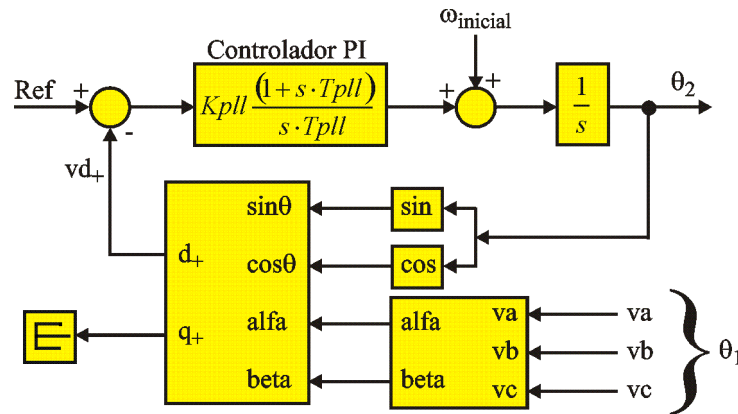


Figura B.2 – Malha de controle da PLL

Nota-se na **Figura B.2** uma entrada de referência **Ref**. Esta entrada permite setar uma determinada defasagem entre os sinais de entrada e saída. Esta defasagem pode ser calculada pela equação (**Eq. B.4**). Determina-se o valor da defasagem e calcula-se, via (**Eq. B.4**), o valor de  $vd_+$  que irá estabilizar a malha. Isto não possui muito sentido para aplicações de Compensadores Ativos, pois se deseja que a defasagem seja sempre zero. Sendo assim a entrada **Ref** estará sempre setada como zero.

Faz-se uma comparação entre **Ref** e  $vd_+$ . O resultado é o erro que será processado pelo controlador PI. A função do controlador PI é atuar na frequência angular da saída, de forma a minimizar o erro. Os parâmetros do controlador PI determinam a forma como o erro será minimizado. A **seção B.2.1** mostra como esses parâmetros podem ser calculados. O cálculo desses parâmetros é muito importante, pois eles determinam a velocidade e a precisão com que o erro será minimizado. Como saída do controlador PI tem-se a frequência angular estimada do sinal de entrada. Esta frequência angular é integrada e como resultado tem-se o ângulo estimado do sinal de entrada. Este ângulo é utilizado na Transformação de Park (**Eq. B.4**), gerando-se um novo sinal  $vd_+$ . O ciclo, então, se repete, forçando  $\theta_2$  se igualar a  $\theta_1$ .

Para acelerar o processo de convergência (e determinar em qual frequência deseja-se estabilizar, no caso de uma entrada composta por múltiplas frequências), pode-se somar à saída do controlador PI um valor inicial de frequência angular. Dessa forma, o controlador precisará apenas corrigir pequenos desvios, tornando o processo muito mais rápido.

### B.2.1 - Cálculo dos ganhos do Controlador PI

Em [B.3] utilizou-se o método *symmetrical optimum* para o cálculo dos ganhos do controlador PI. A técnica pode ser resumida nas equações (Eq. B.6) e (Eq. B.7):

$$T_{pll} = \alpha^2 \cdot T_s \quad (\text{Eq. B.6})$$

$$K_{pll} = \left( \frac{1}{\alpha \cdot U_+ \cdot \sqrt{3/2}} \right) \quad (\text{Eq. B.7})$$

sendo que  $T_s$  é o intervalo de tempo entre cada amostra (inverso da taxa de amostragem);  $U_+$  é a amplitude do sinal trifásico de entrada; e  $\alpha$  é um fator de normalização, que esta relacionado com a resposta dinâmica do sistema em malha fechada.

Quanto menor for o fator  $\alpha$ , mais rápida será a convergência do ângulo estimado para o ângulo correto. Isto ocorre, pois a banda passante do controlador fica mais aberta. Entretanto, na presença de distorções harmônicas no sinal de entrada, o controlador permite que o ângulo de saída siga a trajetória do ângulo de entrada, ocasionando um sinal de saída também distorcido.

Utilizando-se um fator  $\alpha$  de valor mais elevado, a banda passante do controlador fica mais estreita, conseqüentemente ele ficará mais seletivo e o sinal de saída não conterá distorções, mesmo que o sinal de entrada as tenha. Entretanto a convergência é mais lenta.

Deve-se, portanto, encontrar um valor de  $\alpha$  que apresente imunidade satisfatória às distorções da entrada dentro de um intervalo de tempo razoável. A seção B.3 apresenta uma simulação computacional com dois valores diferentes para  $\alpha$ , para que os resultados possam ser comparados.

### B.3 - Simulação Computacional

No software **Matlab 5.3 / Simulink**, montou-se a simulação das **Figuras B.3 e B.4**.

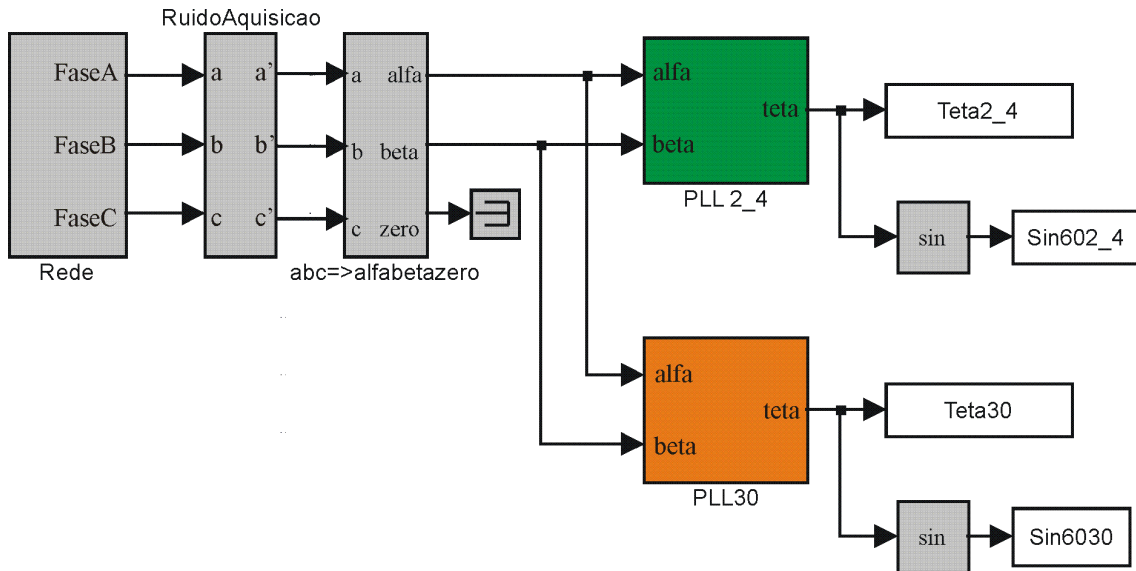


Figura B.3 – Simulação da PLL para dois valores diferentes de  $\alpha$

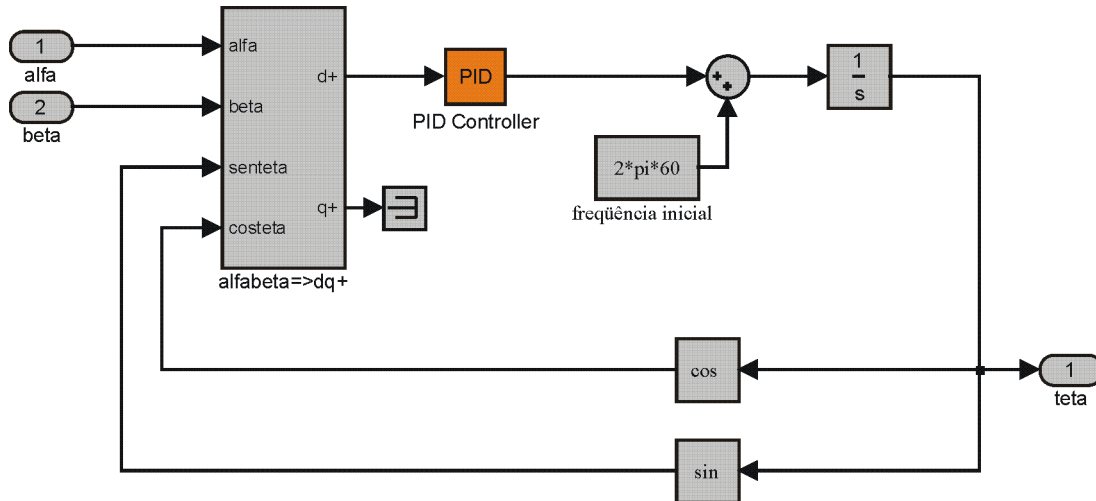


Figura B.4 – Detalhe do bloco da PLL

O bloco em verde na **Figura B.3** contém uma malha de PLL (como a da **Figura B.4**) com os ganhos do controlador PID ajustados para um  $\alpha$  igual a 2.4.

O bloco em laranja na **Figura B.3** contém uma malha de PLL (como a da **Figura B.4**) com os ganhos do controlador PID ajustados para um  $\alpha$  igual a 30.

A taxa de amostragem utilizada foi 10kHz, portanto  $T_s$  será igual a 100us.

Utilizou-se como sinal de entrada 3 ondas senoidais defasadas de  $120^\circ$  com amplitude igual a 1. Portanto  $U_+$  será igual a 1. Acrescentou-se ao sinal de entrada distorção harmônica, na proporção dada na **Tabela B.1**.

Tabela B.1  
Distorção Harmônica de Entrada

	Fase A	Fase B	Fase C
5 <sup>o</sup> harmônico	0.10%	0.15%	0.05%
7 <sup>o</sup> harmônico	0.05%	0.10%	0.15%
11 <sup>o</sup> harmônico	0.08%	0.03%	0.05%
13 <sup>o</sup> harmônico	0.05%	0.03%	0.08%

Além da distorção harmônica, foi acrescentado ruído branco ao sinal de entrada, simulando ruído no sistema de aquisição de dados.

Para  $T_s = 100\mu s$ ,  $U_+ = 1$  e  $\alpha = 2.4$ , pelas equações (Eq. B.6) e (Eq. B.7), tem-se:

$$\mathbf{K_{pll}} = 3402 \quad \text{e} \quad \mathbf{T_{pll}} = 5.76 \times 10^{-4} \text{ [s]}$$

Para  $T_s = 100\mu s$ ,  $U_+ = 1$  e  $\alpha = 30$ , pelas equações (Eq. B.6) e (Eq. B.7), tem-se:

$$\mathbf{K_{pll}} = 272 \quad \text{e} \quad \mathbf{T_{pll}} = 0.09 \text{ [s]}$$

O Simulink já possui pronto um bloco que implementa um controlador do tipo PID, trata-se do bloco Additional Linear / PID Controller. Basta informar o valor dos ganhos Proporcional (**P**), Integral (**I**) e Derivativo (**D**). Para tanto é necessário converter **K<sub>pll</sub>** e **T<sub>pll</sub>** em **P** e **I**, conforme o processo a seguir:

$$K_{pll} \frac{(1 + s \cdot T_{pll})}{s \cdot T_{pll}} = P + \frac{I}{s} = \frac{s \cdot P + I}{s} \quad (\text{Eq. B.8})$$

dividindo a equação (Eq. B.8) por **T<sub>pll</sub>**:

$$\frac{\left( \frac{K_{pll}}{T_{pll}} + s \cdot K_{pll} \right)}{s} = \frac{s \cdot P + I}{s} \quad (\text{Eq. B.9})$$

Tem-se, portanto, que:

$$\mathbf{P} = \mathbf{K_{pll}} \quad \text{e} \quad \mathbf{I} = \mathbf{K_{pll}} / \mathbf{T_{pll}}$$

Logo, para  $\alpha = 2.4$ , **P** = 3402 e **I** = 5906370; e para  $\alpha = 30$ , **P** = 272 e **I** = 3024.

Para melhor comparar o desempenho dos dois tipos de configuração, foi dado um degrau de frequência de 60 para 50 Hz durante a simulação. Conforme pode-se notar na **Figura B.5**.

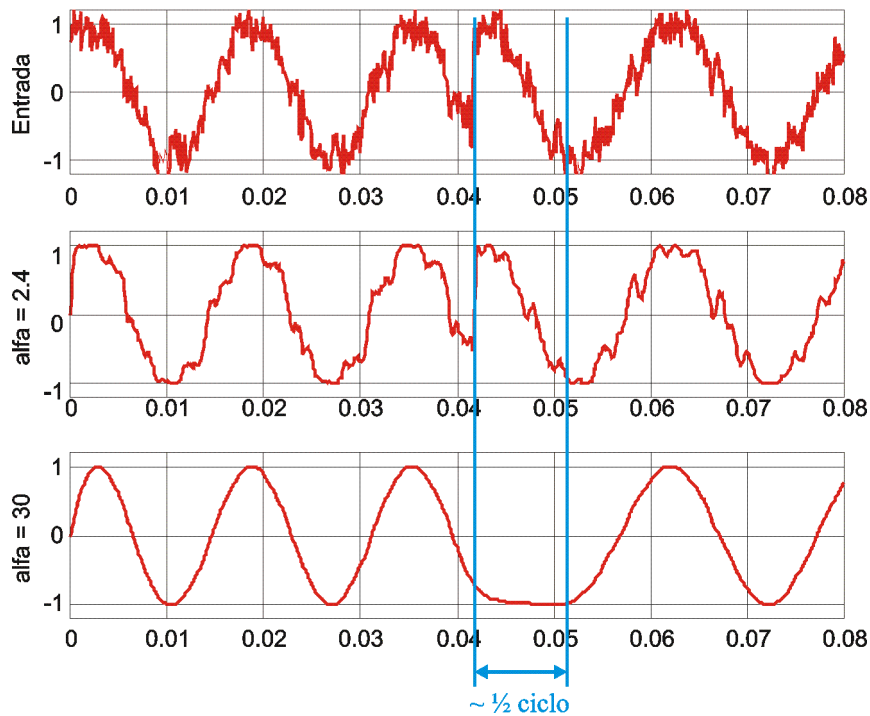


Figura B.5 – Desempenho da PLL

Conforme explicado anteriormente, para  $\alpha$  igual a **2.4**, a PLL, quase que instantaneamente, encontrou o sincronismo com o sinal de entrada após o degrau de frequência. Entretanto, nota-se que a distorção da entrada contaminou a saída.

Para  $\alpha$  igual a **30**, a PLL cumpriu bem seu papel de seguir a informação da frequência fundamental, mesmo que a entrada esteja distorcida. Entretanto, após o degrau de frequência, a PLL demorou aproximadamente meio ciclo para conseguir sincronizar sua saída com o sinal de entrada. Deve-se ressaltar que um degrau de frequência desta ordem **JAMAIS** irá ocorrer no sistema elétrico, porém é sob condições estressantes que melhor se verifica a estabilidade dos algoritmos. Mesmo assim, sob condições extremas, a PLL conseguiu o sincronismo em meio ciclo, que é um tempo satisfatório.

## **B.4 - Referências**

[B.1] R.E.Best, “Phase Locked Loops – Theory, Design and Applications”, McGraw-Hill Book, USA, 1984;

[B.2] M.E.Oliveira Filho, “Estratégias de controle de corrente para filtros ativos de potência”, Dissertação de Mestrado, Escola Federal de Engenharia de Itajubá, 1998;

[B.3] V.Kaura, V.Blasko, “Operation of a phase locked loop system under distorted utility conditions”, IEEE Trans. Ind. Appl., vol. 33, pp 58-63, 1997;



# Apêndice C

## Software de Controle

Escreveu-se em **linguagem C** um software que implementa o algoritmo de controle em malha fechada para compensação de Voltage SAGs e harmônicos de tensão. Neste Apêndice, será transcrito o software que acompanhou o equipamento desenvolvido para a concessionária de energia CEMAR. Este projeto foi finalizado em agosto de 2003. Posteriormente o algoritmo evoluiu para o que foi descrito no **Capítulo 3**. Porém a idéia central é exatamente a mesma.

```

/*
*****
* Program : serie.C
* PCM-3718H pacer trigger A/D
* conversion with Block Scan data transfer.
* amostragem ciclica em 4980Hz
* separa os 3 canais de entrada
* canal 0 -> tensao de entrada na fase A
* canal 1 -> tensao de entrada na fase B
* canal 2 -> tensao de entrada na fase C
*
* saida digital
* D[7:0] : dados
* D8 : clkA
* D9 : clkB
* D10 : clkC
* D11 : NC
* D12 : start
* D[15:13] : reservados para o futuro
*****
*/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <dos.h>
#include <math.h>
extern pcm3718(int, unsigned int *);
unsigned int param[60];
unsigned int data[354]; /* Conversion data buffer */
unsigned int far * dat;
#define NBuf 118 //numero de amostras em 1 ciclo de 60Hz
#define NBufMed 119 //tamanho do buffer das medias moveis (NBuf+1)
#define NHarm 8
#define Ki 50 //ganho integral do compensador PI
#define Kp 1 //ganho proporcional do compensador PI
//mascaras para set e reset de bits do Digital Port
#define BASE 0x210 //endereço da placa
#define clk_A 0x01
#define clk_B 0x02
#define clk_C 0x04
#define START 0x10
//#define DP14 0x20
//#define DP15 0x40
//#define DP16 0x80
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

////////////////////////////////////
void main(void )
{
  unsigned char dados,controle,dadosC;
  float SaidaA=0,SaidaB=0,SaidaC=0;
  float faseA=0,faseB=0,faseC=0;
  float va=0,vb=0,vc=0,senTeta=0,cosTeta=0;
  float Alfa=0,Beta=0,AlfaRef=0,BetaRef=0;
  float *dPos,*qPos,*dNeg,*qNeg;
  float dPosF=0,qPosF=0,dNegF=0,qNegF=0;
  float Soma_dPos=0,Soma_qPos=0,Soma_dNeg=0,Soma_qNeg=0;
  float Soma_dNegHarm=0,Soma_qNegHarm=0;
  float AlfaComp=0,BetaComp=0;
  float AlfaCompFund=0,BetaCompFund=0;
  float **BufAlfaCompHarm,**BufBetaCompHarm,*AlfaCompHarm,*BetaCompHarm;
  float dRef=0,qRef=0;
  float dCompPos=0,dCompNeg=0,qCompPos=0,qCompNeg=0;
  float ErrodPos=0,ErrodNeg=0,ErroqPos=0,ErroqNeg=0;
  float ErrodPos1=0,ErrodNeg1=0,ErroqPos1=0,ErroqNeg1=0;
  float kdf=0,kdfantigo=0,pospi=0;
  float wpll=0,wpllantigo=0,Teta=0,tetaantigo=0;
  unsigned int
  IdxConv=0,IdxAtual=0,IdxPassado=0,Idx83=0,Idx3600=0,IdxBufMed=0;
  float *TetaTable,*cosTable,*sinTable,InvNBuf=0,KM=0,Km=0;
  int i,j;
  //malha de saida
  float dPosHarm=0,qPosHarm=0,dPosHarmF=0,qPosHarmF=0;
  float dNegHarm=0,qNegHarm=0,dNegHarmF=0,qNegHarmF=0;
  float *dCompPosHarm,*qCompPosHarm,*dCompNegHarm,*qCompNegHarm;
  float Soma_dPosHarm=0,Soma_qPosHarm=0;
  float *VaComp,*VbComp,*VcComp;
  float *tHarm,**senHarm,**cosHarm;
  float AlfCosHarm=0,BetCosHarm=0,AlfSenHarm=0,BetSenHarm=0;
  float AlfCosFund=0,BetCosFund=0,AlfSenFund=0,BetSenFund=0;
  unsigned int IdxHarm=2;
  float FeedA=0,FeedB=0,FeedC=0;
  //Sinais do Circuito de Comando
  unsigned char LigaIGBT=0,ErroSemikron=0;
  unsigned int ContLiga=0;
  //////////////////////////////////////

  //////////////////////////////////////
  ///
  //alocacao dinamica de memoria
  TetaTable=(float *)calloc(3600,sizeof (float ));
  if(TetaTable==NULL)
  {
    printf("Erro ao alocar vetor TetaTable\n");
    exit(1);
  }
  cosTable=(float *)calloc(3600,sizeof (float ));
  if(cosTable==NULL)
  {
    printf("Erro ao alocar vetor cosTable\n");
    exit(1);
  }
  sinTable=(float *)calloc(3600,sizeof (float ));
  if(sinTable==NULL)
  {
    printf("Erro ao alocar vetor sinTable\n");
    exit(1);
  }
}

```

```
dCompPosHarm=(float *)calloc(NHarm,sizeof (float ));
if(dCompPosHarm==NULL)
{
    printf("Erro ao alocar vetor dCompPosHarm\n");
    exit(1);
}
qCompPosHarm=(float *)calloc(NHarm,sizeof (float ));
if(qCompPosHarm==NULL)
{
    printf("Erro ao alocar vetor qCompPosHarm\n");
    exit(1);
}
dCompNegHarm=(float *)calloc(NHarm,sizeof (float ));
if(dCompNegHarm==NULL)
{
    printf("Erro ao alocar vetor dCompNegHarm\n");
    exit(1);
}
qCompNegHarm=(float *)calloc(NHarm,sizeof (float ));
if(qCompNegHarm==NULL)
{
    printf("Erro ao alocar vetor qCompNegHarm\n");
    exit(1);
}
tHarm=(float *)calloc(3600,sizeof (float ));
if(tHarm==NULL)
{
    printf("Erro ao alocar vetor tHarm\n");
    exit(1);
}
VaComp=(float *)calloc(NBuf,sizeof (float ));
if(VaComp==NULL)
{
    printf("Erro ao alocar vetor VaComp\n");
    exit(1);
}
VbComp=(float *)calloc(NBuf,sizeof (float ));
if(VbComp==NULL)
{
    printf("Erro ao alocar vetor VbComp\n");
    exit(1);
}
VcComp=(float *)calloc(NBuf,sizeof (float ));
if(VcComp==NULL)
{
    printf("Erro ao alocar vetor VcComp\n");
    exit(1);
}
senHarm=(float **)calloc(NHarm,sizeof (float *));
if(senHarm==NULL)
{
    printf("Erro ao alocar matriz senHarm\n");
    exit(1);
}
for(i=0;i<NHarm;i++)
{
    senHarm[i]=(float *)calloc(3600,sizeof (float ));
    if(senHarm[i]==NULL)
    {
        printf("Erro ao alocar matriz senHarm, linha[%d]\n",i);
        exit(1);
    }
}
}
```

```
cosHarm=(float **)calloc(NHarm,sizeof (float *));
if(cosHarm==NULL)
{
    printf("Erro ao alocar matriz cosHarm\n");
    exit(1);
}
for(i=0;i<NHarm;i++)
{
    cosHarm[i]=(float *)calloc(3600,sizeof (float ));
    if(cosHarm[i]==NULL)
    {
        printf("Erro ao alocar matriz cosHarm, linha[%d]\n",i);
        exit(1);
    }
}
dPos=(float *)calloc(NBufMed,sizeof (float ));
if(dPos==NULL)
{
    printf("Erro ao alocar vetor dPos\n");
    exit(1);
}
dNeg=(float *)calloc(NBufMed,sizeof (float ));
if(dNeg==NULL)
{
    printf("Erro ao alocar vetor dNeg\n");
    exit(1);
}
qPos=(float *)calloc(NBufMed,sizeof (float ));
if(qPos==NULL)
{
    printf("Erro ao alocar vetor qPos\n");
    exit(1);
}
qNeg=(float *)calloc(NBufMed,sizeof (float ));
if(qNeg==NULL)
{
    printf("Erro ao alocar vetor qNeg\n");
    exit(1);
}
BufAlfaCompHarm=(float **)calloc(NHarm,sizeof (float *));
if(BufAlfaCompHarm==NULL)
{
    printf("Erro ao alocar matriz BufAlfaCompHarm\n");
    exit(1);
}
for(i=0;i<NHarm;i++)
{
    BufAlfaCompHarm[i]=(float *)calloc(NBuf,sizeof (float ));
    if(BufAlfaCompHarm[i]==NULL)
    {
        printf("Erro ao alocar matriz BufAlfaCompHarm,linha[%d]\n",i);
        exit(1);
    }
}
```

```

BufBetaCompHarm=(float **)calloc(NHarm,sizeof (float *));
if (BufBetaCompHarm==NULL)
{
    printf("Erro ao alocar matriz BufBetaCompHarm\n");
    exit(1);
}
for(i=0;i<NHarm;i++)
{
    BufBetaCompHarm[i]=(float *)calloc(NBuf,sizeof (float ));
    if (BufBetaCompHarm[i]==NULL)
    {
        printf("Erro ao alocar matriz BufBetaCompHarm,linha[%d]\n",i);
        exit(1);
    }
}
AlfaCompHarm=(float *)calloc(NBuf,sizeof (float ));
if (AlfaCompHarm==NULL)
{
    printf("Erro ao alocar vetor AlfaCompHarm\n");
    exit(1);
}
BetaCompHarm=(float *)calloc(NBuf,sizeof (float ));
if (BetaCompHarm==NULL)
{
    printf("Erro ao alocar vetor BetaCompHarm\n");
    exit(1);
}
/////////////////////////////////////////////////////////////////
InvNBuf=1.0/NBuf;
KM=.0000706*Ki+Kp;
Km=.0000706*Ki-Kp;
/////////////////////////////////////////////////////////////////
//LookUp Table de seno e cosseno
//gerar vetor de angulo com precisao de 0.1 grau
//em 1 periodo tem-se 3600 pontos
for(i=0;i<3600;i++)
    TetaTable[i]=i*.001745;
//gerar vetor de seno e cosseno
for(i=0;i<3600;i++)
{
    cosTable[i]=cos(TetaTable[i]);
    sinTable[i]=sin(TetaTable[i]);
}
/////////////////////////////////////////////////////////////////
//gerar senos e cossenos das transformadas de Park das harmonicas
//vetor de tempo
for(i=0;i<3600;i++)
    tHarm[i]=i/(60.0*3600);
//vetores de senos e cossenos de 2a ateh 7a harmonica
for(j=0;j<NHarm;j++)
    for(i=0;i<3600;i++)
    {
        senHarm[j][i]=sin(376.99*j*tHarm[i]);
        cosHarm[j][i]=cos(376.99*j*tHarm[i]);
    }
/////////////////////////////////////////////////////////////////

```

```

////////////////////////////////////
//zera buffers
for(i=0;i<NBuf;i++)
{
    VaComp[i]=0;VbComp[i]=0;VcComp[i]=0;
    AlfaCompHarm[i]=0;BetaCompHarm[i]=0;
}
for(i=0;i<NBufMed;i++)
{
    dPos[i]=0;qPos[i]=0;dNeg[i]=0;qNeg[i]=0;
}
for(i=0;i<NHarm;i++)
{
    dCompPosHarm[i]=0;qCompPosHarm[i]=0;
    dCompNegHarm[i]=0;qCompNegHarm[i]=0;
    for(j=0;j<NBuf;j++)
    {
        BufAlfaCompHarm[i][j]=0;
        BufBetaCompHarm[i][j]=0;
    }
}
////////////////////////////////////

////////////////////////////////////
//liga oscilador 2,5Mhz da placa moduladora PWM
controle|=START;
outportb(BASE+11,controle);//seta START
controle&=~START;
outportb(BASE+11,controle);//reseta START
controle|=START;
outportb(BASE+11,controle);//seta START
////////////////////////////////////
dat = data;
clrscr();
param[0] = 0;          /* Board number */
param[1] = 0x210;     /* Base I/O address */
param[2] = 3;         /*DMA */
param[4] = 3;         /* IRQ level : IRQ3 */
param[5] = 353;       /* Pacer rate=10M/(353*4)=7082.15Hz */
param[6] = 4;
param[7] = 0;         /* Trigger mode,0 :pacertrigger */
param[8] = 1;         /* cyclic */
param[10] = FP_OFF(dat); /* Offset of A/D data buffer A */
param[11] = FP_SEG(dat); /* Segment of A/D data buffer A */
param[12] = 0;        /* Data buffer B address, if not used, */
param[13] = 0;        /* must set to 0. */
param[14] = 354;      /* A/D conversion number: 118pontos X 3canais*/
param[15] = 0;        /* A/D conversion start channel */
param[16] = 2;        /* A/D conversion stop channel */
param[17] = 0;        /* Overall gain code,0 :+/-5V */
/* param[18] = FP_OFF(gain_array);
param[19] = FP_SEG(gain_array); */
/* param[45] : Error code
param[46] : Return value 0
param[47] : Return value 1 */
////////////////////////////////////

```

```
////////////////////////////////////  
pcm3718(3, param); /* Func 3 : Hardware initialization */  
if (param[45] != 0) {  
    printf("\nDRIVER INITIALIZATION FAILED !");  
    exit(1);  
}  
  
pcm3718(100, param); /* Func 100 : A/D initialization */  
if (param[45] != 0) {  
    printf("\nA/D INITIALIZATION FAILED !");  
    exit(1);  
}  
  
pcm3718(105, param); /* Func 105 : Pacer trigger A/D conversion */  
if (param[45] != 0) { /* with block scan data transfer */  
    printf("\nA/D PACER TRIGGER WITH block scan DATA TRANSFER FAILED  
!");  
    exit(1);  
}  
  
IdxPassado=354;  
  
controle&=~clk_A;  
outportb(BASE+11,controle); //reseta clk_A  
dados=(0+5)*256/10;  
outportb(BASE+3,dados);  
controle|=clk_A;  
outportb(BASE+11,controle); //seta clk_A  
  
controle&=~clk_B;  
outportb(BASE+11,controle); //reseta clk_B  
dados=(0+5)*256/10;  
outportb(BASE+3,dados);  
controle|=clk_B;  
outportb(BASE+11,controle); //seta clk_B  
  
controle&=~clk_C;  
outportb(BASE+11,controle); //reseta clk_C  
dados=(0+5)*256/10;  
outportb(BASE+3,dados);  
controle|=clk_C;  
outportb(BASE+11,controle); //seta clk_C  
////////////////////////////////////
```

```

////////////////////////////////////
do
{
volta:
pcm3718(106, param); /* Func 106 : Check interrupt status */
IdxAtual=param[47];
if((IdxAtual==IdxPassado+3)|| (IdxAtual<IdxPassado))//entao mudou de canal
{
    IdxPassado=IdxAtual;
    //////////////////////////////////////

    //////////////////////////////////////
    va = data[0+3*IdxConv] & 0xFFFF;
    va =(.00244*va)-5;
    // vaxn =( (5 - (-5)) * va / 4096) + (-5);
    // (5 - (-5)) : A/D input range (-5V to 5V)
    // 4096 : Full scale 12 bit A/D data
    // DataBuf : A/D input data
    // (-5) : A/D input range "-5" V
    //////////////////////////////////////
    vb = data[1+3*IdxConv] & 0xFFFF;
    // vbxn =( (5 - (-5)) * vb / 4096) + (-5);
    vb =(.00244*vb)-5;
    //////////////////////////////////////
    vc = data[2+3*IdxConv] & 0xFFFF;
    // vcxn =( (5 - (-5)) * vc / 4096) + (-5);
    vc =(.00244*vc)-5;
    //////////////////////////////////////
    //////////////////////////////////////
    va+=FeedA;vb+=FeedB;vc+=FeedC;
    //////////////////////////////////////
    //////////////////////////////////////
    //leitura do sinal LigaIGBT
    LigaIGBT=(inportb(0x379)&0x40);
    //leitura do sinal de Erro
    ErroSemikron=(inportb(0x379)&0x20);
    //////////////////////////////////////

    //////////////////////////////////////

    //////////////////////////////////////
    //calculos do filtro
    //////////////////////////////////////

    //PLL trifasica
    //transf. Clarke da entrada
    //alfa=sqrt(2/3)*va-[sqrt(2/3)/2]*(vb+vc)
    //beta=sqrt(2/3)*[sqrt(3)/2]*(vb-vc)
    Alfa=.8165*va-.4082*(vb+vc);
    Beta=.7071*(vb-vc);
    //simplificacao:
    //ao inves de 8 multiplicacoes de floating point
    //faz apenas 4;
    AlfCosFund=Alfa*cosTeta;
    BetCosFund=Beta*cosTeta;
    AlfSenFund=Alfa*senTeta;
    BetSenFund=Beta*senTeta;
    //trans. Park seq. pos. da entrada
    dPos[IdxBufMed]=AlfCosFund+BetSenFund;
    qPos[IdxBufMed]=BetCosFund-AlfSenFund;
    //trans. Park seq. neg. da entrada
    dNeg[IdxBufMed]=AlfCosFund-BetSenFund;
    qNeg[IdxBufMed]=BetCosFund+AlfSenFund;
}
}

```



```
//controlador PI(calculados para fs=4980Hz)
//kdf calculado para U=3.8Vp (ganho do Hall)
// kdf=776.55*d2; //alfa=2.4
// pospi=pospiantigo+1.0868*kdf-0.9132*kdfantigo; //alfa=2.4
// kdf=124.25*d2; //alfa=15
// pospi=pospiantigo+1.0022*kdf-0.9978*kdfantigo; //alfa=15
kdf=62.12*dPos[IdxBufMed]; //alfa=30
pospi+=(1.00056*kdf-0.99944*kdfantigo); //alfa=30
kdfantigo=kdf;
wpll=pospi+376.99111843;//wref=2*pi*60
// wpll=pospi+376.99;//wref=2*pi*60
//integrador
//trapezoidal
Teta=tetaantigo+0.0000706*(wpll+wpllantigo);//fs=10020.04Hz
wpllantigo=wpll;
////////////////////////////////////
////////////////////////////////////
//Protecao contra Overflow do registrador que guarda teta
if(Teta>6.283188)
Teta=Teta-6.283188;
////////////////////////////////////
tetaantigo=Teta;
Idx3600=Teta*572.9575;
cosTeta=cosTable[Idx3600];
senTeta=sinTable[Idx3600];
//calculo de alfa e beta de referencia
// amplitude de referencia: 3.8
// fator de transformacao da faseA para alfa,
// estando o sistema simetrico e equilibrado: 1.2247
// 4.65386=3.8*1.2247
AlfaRef=4.65386*senTeta;
BetaRef=-4.65386*cosTeta;
////////////////////////////////////
////////
////////////////////////////////////
////////
//sincronizar amostras
if(Teta<.05325)
    Idx83=0;
else if(Teta>=6.2831)
    Idx83=117;
else
    // Idx83=Teta*118/6.2831;
    Idx83=Teta*18.78;
////////////////////////////////////
////////////////////////////////////
// Malha de controle para Voltage SAGs
// seno e cosseno das transformacoes em 60Hz

// calculo da trans. Park seq. pos. da entrada
//jah foi feito no bloco da PLL

// calculo da trans. Park seq. neg. da entrada
//jah foi feito no bloco da PLL

// calculo da trans. Park seq. pos. da referencia
dRef=AlfaRef*cosTeta+BetaRef*senTeta;
qRef=BetaRef*cosTeta-AlfaRef*senTeta;
```

```

// extracao das componentes DC dos sinais
//d e q da entrada
if (IdxBufMed==NBuf)
{
    Soma_dPos+=(dPos[IdxBufMed]-dPos[0]);
    Soma_dNeg+=(dNeg[IdxBufMed]-dNeg[0]);
    Soma_qPos+=(qPos[IdxBufMed]-qPos[0]);
    Soma_qNeg+=(qNeg[IdxBufMed]-qNeg[0]);
    IdxBufMed=0;
}
else
{
    Soma_dPos+=(dPos[IdxBufMed]-dPos[IdxBufMed+1]);
    Soma_dNeg+=(dNeg[IdxBufMed]-dNeg[IdxBufMed+1]);
    Soma_qPos+=(qPos[IdxBufMed]-qPos[IdxBufMed+1]);
    Soma_qNeg+=(qNeg[IdxBufMed]-qNeg[IdxBufMed+1]);
    IdxBufMed++;
}
dPosF=Soma_dPos*InvNBuf;//InvNBuf=1/NBuf
dNegF=Soma_dNeg*InvNBuf;
qPosF=Soma_qPos*InvNBuf;
qNegF=Soma_qNeg*InvNBuf;

//controladores PI da malha para Voltage SAGs
//y[n]=y[n-1]+KM*X[n]+Km*X[n-1]
//KM=Ki*Ts/2+Kp
//Km=ki*Ts/2-Kp
ErrodPos=dRef-dPosF;ErroqPos=qRef-qPosF;
ErrodNeg=-dNegF;ErroqNeg=-qNegF;
dCompPos+=(KM*ErrodPos+Km*ErrodPos1);
qCompPos+=(KM*ErroqPos+Km*ErroqPos1);
dCompNeg+=(KM*ErrodNeg+Km*ErrodNeg1);
qCompNeg+=(KM*ErroqNeg+Km*ErroqNeg1);
ErrodPos1=ErrodPos;ErrodNeg1=ErrodNeg;
ErroqPos1=ErroqPos;ErroqNeg1=ErroqNeg;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Malha de controle para Harmonicos
// seno e cosseno das transformacoes em IdxHarm*60Hz
// Compensa um harmonico a cada ciclo de 60Hz
// 2,3,4,5,6,7,2,3,4,...
//simplificacao:
//ao inves de 8 multiplicacoes de floating point
//faz apenas 4;
AlfCosHarm=Alfa*cosHarm[IdxHarm][Idx3600];
BetCosHarm=Beta*cosHarm[IdxHarm][Idx3600];
AlfSenHarm=Alfa*senHarm[IdxHarm][Idx3600];
BetSenHarm=Beta*senHarm[IdxHarm][Idx3600];

//trans. Park seq. pos. da parcela harmonica
dPosHarm=AlfCosHarm+BetSenHarm;
qPosHarm=BetCosHarm-AlfSenHarm;
Soma_dPosHarm+=dPosHarm;
Soma_qPosHarm+=qPosHarm;
//trans. Park seq. neg. da parcela harmonica
dNegHarm=AlfCosHarm-BetSenHarm;
qNegHarm=BetCosHarm+AlfSenHarm;
Soma_dNegHarm+=dNegHarm;
Soma_qNegHarm+=qNegHarm;

```

```

//transf. inversa Park da parcela fundamental
AlfaCompFund = (dCompPos + dCompNeg) * cosTeta + (qCompNeg -
qCompPos) * senTeta;
BetaCompFund = (qCompPos + qCompNeg) * cosTeta + (dCompPos -
dCompNeg) * senTeta;
//transf. inversa Park da parcela harmonica
AlfaCompHarm[Idx83] -= BufAlfaCompHarm[IdxHarm][Idx83];
BetaCompHarm[Idx83] -= BufBetaCompHarm[IdxHarm][Idx83];
BufAlfaCompHarm[IdxHarm][Idx83] = (dCompPosHarm[IdxHarm] +
dCompNegHarm[IdxHarm]) * cosHarm[IdxHarm][Idx3600] +
(qCompNegHarm[IdxHarm] - qCompPosHarm[IdxHarm]) *
senHarm[IdxHarm][Idx3600];
BufBetaCompHarm[IdxHarm][Idx83] = (qCompPosHarm[IdxHarm] +
qCompNegHarm[IdxHarm]) * cosHarm[IdxHarm][Idx3600] +
(dCompPosHarm[IdxHarm] - dCompNegHarm[IdxHarm]) *
senHarm[IdxHarm][Idx3600];
AlfaCompHarm[Idx83] += BufAlfaCompHarm[IdxHarm][Idx83];
BetaCompHarm[Idx83] += BufBetaCompHarm[IdxHarm][Idx83];

//soma dos alfas e betas de compensacao
AlfaComp=AlfaCompFund+AlfaCompHarm[Idx83];
BetaComp=BetaCompFund+BetaCompHarm[Idx83];

//transf. inversa de Clarke da compensacao
VaComp[Idx83]=.8165*AlfaComp;
VbComp[Idx83]=.7071*BetaComp-.4082*AlfaComp;
VcComp[Idx83]=-(.7071*BetaComp+.4082*AlfaComp);

SaidaA=VaComp[Idx83];
SaidaB=VbComp[Idx83];
SaidaC=VcComp[Idx83];

//atualiza Indice dos Harmonicos
if (IdxConv==117)
{
    //pos
    //XF=X/118=X*.008475
    dPosHarmF=Soma_dPosHarm*.008475;
    qPosHarmF=Soma_qPosHarm*.008475;
    Soma_dPosHarm=0;
    Soma_qPosHarm=0;
    //neg
    dNegHarmF=Soma_dNegHarm*.008475;
    qNegHarmF=Soma_qNegHarm*.008475;
    Soma_dNegHarm=0;
    Soma_qNegHarm=0;

    //Controlador PI
    if ((dPosHarmF>.1) || (dPosHarmF<-.1))
        dCompPosHarm[IdxHarm]-=.5*dPosHarmF;
    else if (dPosHarmF<-.001)
        dCompPosHarm[IdxHarm]+=.001;
    else if (dPosHarmF>.001)
        dCompPosHarm[IdxHarm]-=.001;
    if ((dNegHarmF>.1) || (dNegHarmF<-.1))
        dCompNegHarm[IdxHarm]-=.5*dNegHarmF;
    else if (dNegHarmF<-.001)
        dCompNegHarm[IdxHarm]+=.001;
    else if (dNegHarmF>.001)
        dCompNegHarm[IdxHarm]-=.001;
    if ((qPosHarmF>.1) || (qPosHarmF<-.1))
        qCompPosHarm[IdxHarm]-=.5*qPosHarmF;
    else if (qPosHarmF<-.001)
        qCompPosHarm[IdxHarm]+=.001;
}

```

```

else if(qPosHarmF>.001)
    qCompPosHarm[IdxHarm]-=.001;
if((qNegHarmF>.1)|| (qNegHarmF<-.1))
    qCompNegHarm[IdxHarm]-=.5*qNegHarmF;
else if(qNegHarmF<-.001)
    qCompNegHarm[IdxHarm]+=.001;
else if(qNegHarmF>.001)
    qCompNegHarm[IdxHarm]-=.001;
if(IdxHarm<(NHarm-1))
    IdxHarm++;
else
    IdxHarm=2;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//SE Driver indicando erro e usuario desligou IGBTs
//entao resetar erro
if(LigaIGBT&&(!ErroSemikron))
    outportb(0x378,0x00);//reseta erro do driver
else if(!LigaIGBT&&ErroSemikron)//operacao normal
{
    // SaidaParalela|=HabilitaSemikron;//habilita os pulsos para o
    driver
    // SaidaParalela|=LigaContator;//liga Contator
    // outportb(0x378,SaidaParalela);
    outportb(0x378,0xFF);
    // if(ContLiga<=7082)//timer de 1 segundo
    if(ContLiga<=35410)//timer de 5 segundos
    // if(ContLiga<65535)//timer de 9.25 segundos
    {
        //modo off-line
        FeedA=SaidaA;FeedB=SaidaB;FeedC=SaidaC;
        faseA=0;faseB=0;faseC=0;
        ContLiga++;
    }
    else
    {
        // FeedA=SaidaA;FeedB=SaidaB;FeedC=SaidaC;
        //modo on-line
        FeedA=0;FeedB=0;FeedC=0;
        faseA=SaidaA;faseB=SaidaB;faseC=SaidaC;
    }
}
else
{
    ContLiga=0;
    //modo off-line
    FeedA=SaidaA;FeedB=SaidaB;FeedC=SaidaC;
    faseA=0;faseB=0;faseC=0;
    // SaidaParalela&=~LigaContator;//desliga Contator
    // SaidaParalela&=~HabilitaSemikron;//corta os pulsos para o
    driver
    // outportb(0x378,SaidaParalela);
    outportb(0x378,0x40);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

/////////////////////////////////////////////////////////////////
//Saida para inversor A
// dados=(faseA+5)*256/10;
dados=(faseA+5)*25.6;
outportb(BASE+3,dados);
controle&=~clk_A;
outportb(BASE+11,controle); //reseta clk_A
controle|=clk_A;
outportb(BASE+11,controle); //seta clk_A
/////////////////////////////////////////////////////////////////
//Saida para inversor B
dados=(faseB+5)*25.6;
outportb(BASE+3,dados);
controle&=~clk_B;
outportb(BASE+11,controle); //reseta clk_B
controle|=clk_B;
outportb(BASE+11,controle); //seta clk_B
/////////////////////////////////////////////////////////////////
//Saida para inversor C
dados=(faseC+5)*25.6;
outportb(BASE+3,dados);
controle&=~clk_C;
outportb(BASE+11,controle); //reseta clk_C
controle|=clk_C;
outportb(BASE+11,controle); //seta clk_C
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
Idx83++;
if(Idx83>117) //83 pontos: de 0 ateh 82
    Idx83-=118;
if(IdxConv==117) //83 pontos: de 0 ateh 82
    IdxConv=0;
else
    IdxConv++;
} //fim if mudou de canal

}while (!kbhit());
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
//libera memoria alocada
free(TetaTable); free(cosTable); free(sinTable);
free(dCompPosHarm); free(dCompNegHarm);
free(qCompPosHarm); free(qCompNegHarm);
free(tHarm);
for(i=0; i<NHarm; i++)
    free(senHarm[i]);
free(senHarm);
for(i=0; i<NHarm; i++)
    free(cosHarm[i]);
free(cosHarm);
free(VaComp); free(VbComp); free(VcComp);
free(dPos); free(dNeg); free(qPos); free(qNeg);
for(i=0; i<NHarm; i++)
    free(BufAlfaCompHarm[i]);
free(BufAlfaCompHarm);
for(i=0; i<NHarm; i++)
    free(BufBetaCompHarm[i]);
free(BufBetaCompHarm);
free(AlfaCompHarm);
free(BetaCompHarm);
/////////////////////////////////////////////////////////////////

```

```
////////////////////////////////////  
//finaliza driver  
pcm3718(107,param);  
  
}//fim main  
  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////  
////////////////////////////////////
```

# ***Apêndice D***

## ***Placas Confeccionadas***

**Apêndice D\_1 – Placa Gerador de Portadora Triangular**

**Apêndice D\_2 – Placa Oscilador a Cristal**

**Apêndice D\_3 – Placa Comparador Digital**

**Apêndice D\_4 – Placa Interface PWM X VSI**

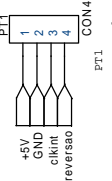
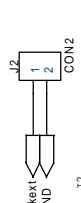
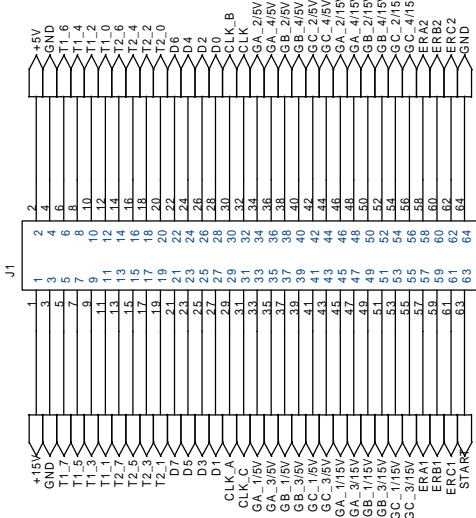
**Apêndice D\_5 – Placa Interface CPU X Comando**

**Apêndice D\_6 – Placa Barramento**

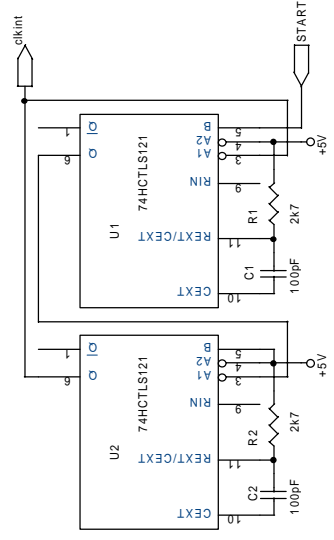
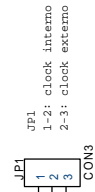
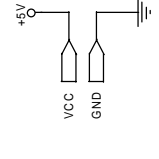
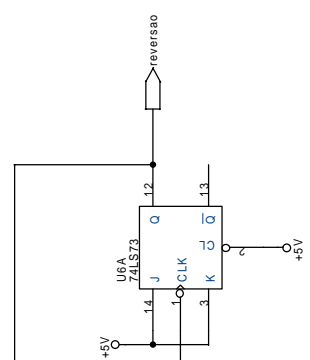
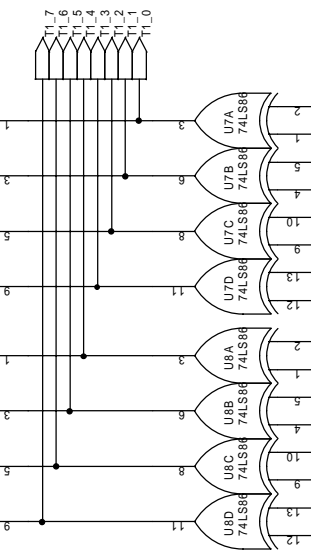
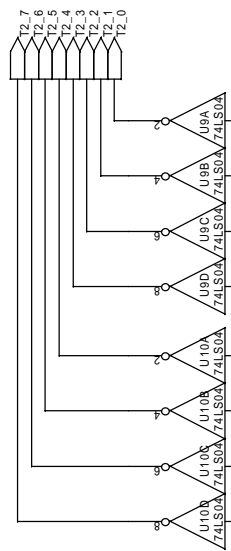




J1  
Conector DIN41612 de 64Vias,  
macho, TipoB, 90graus

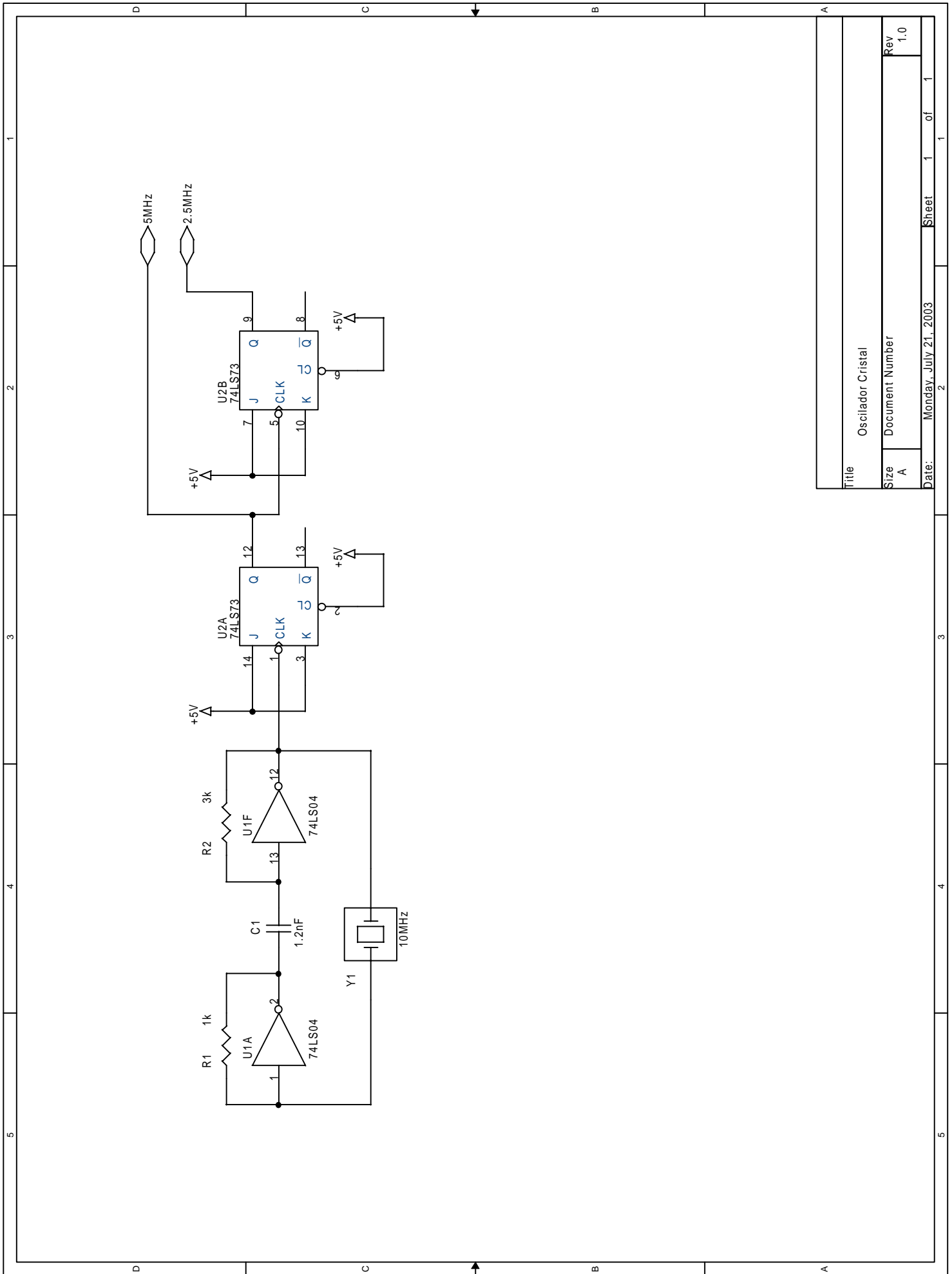


macho\_64



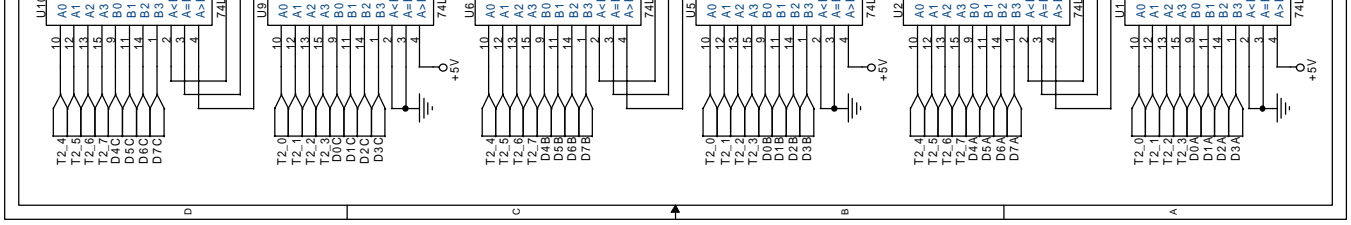
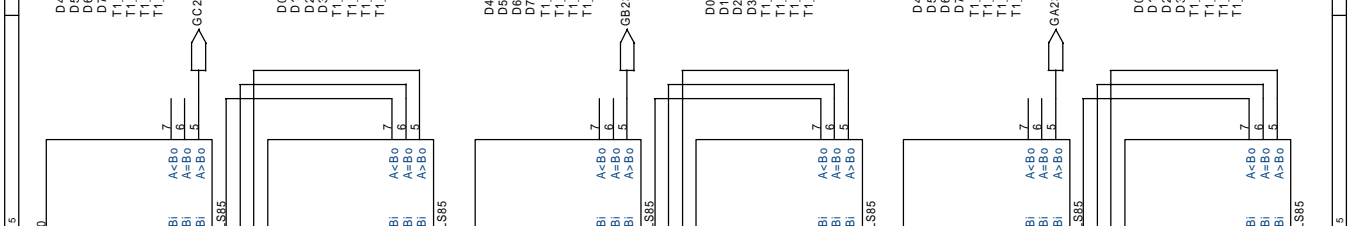
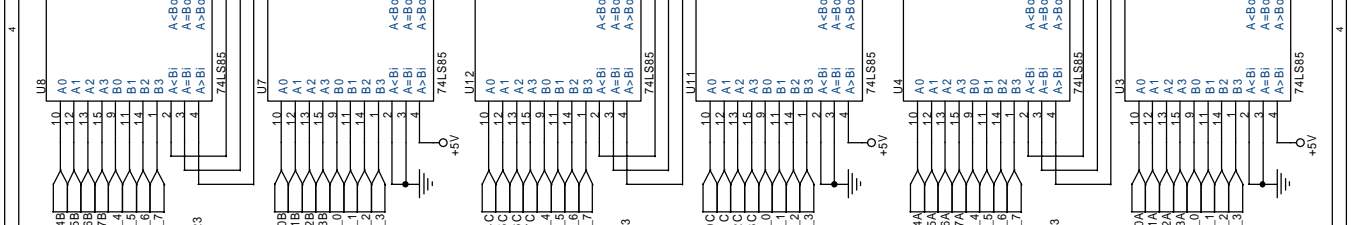
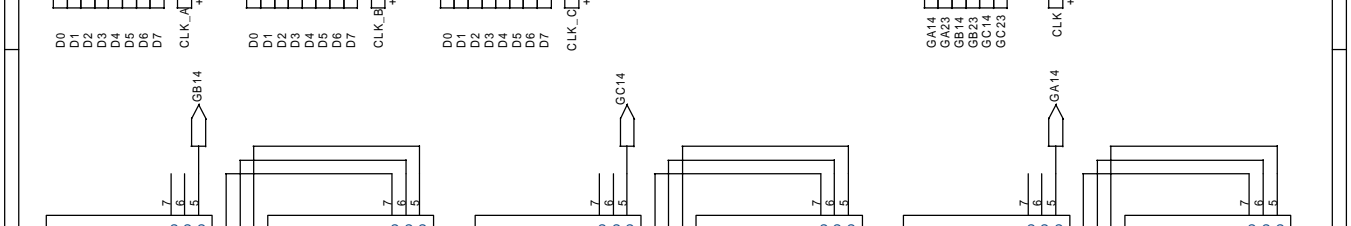
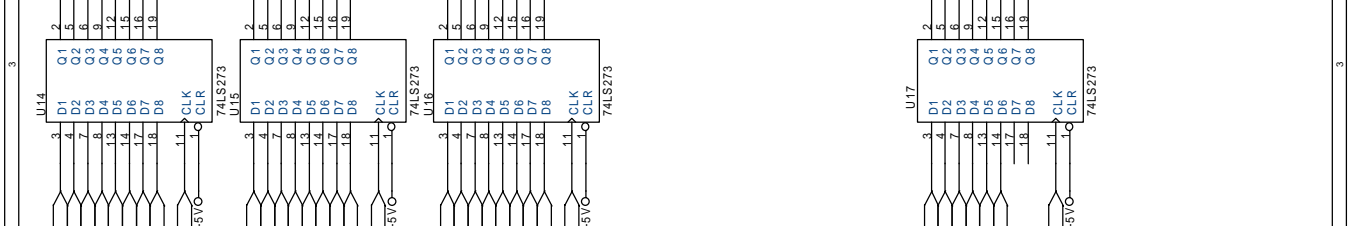
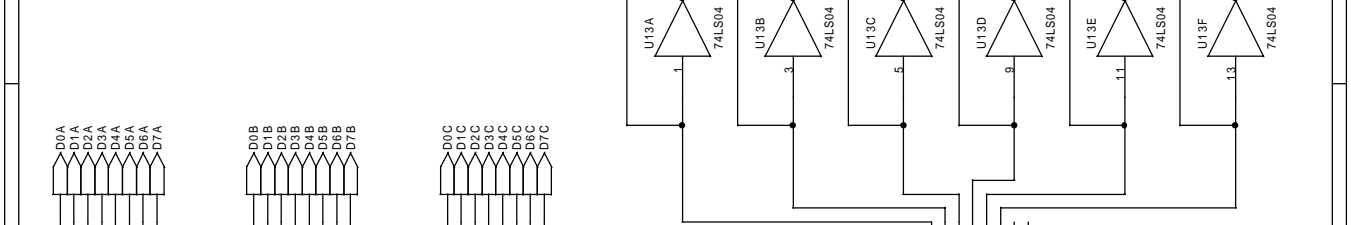
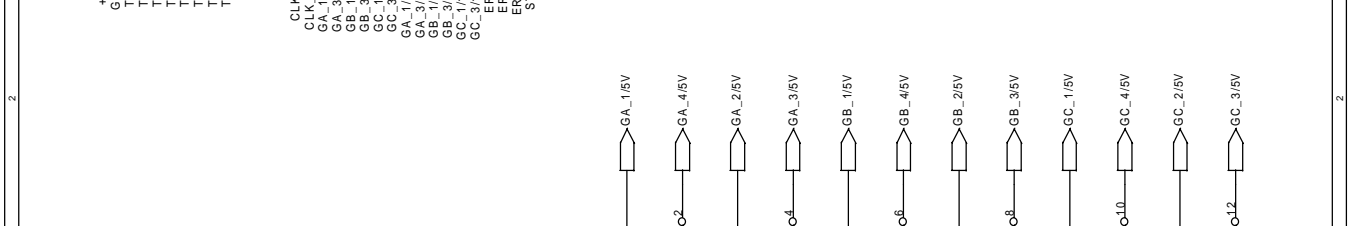
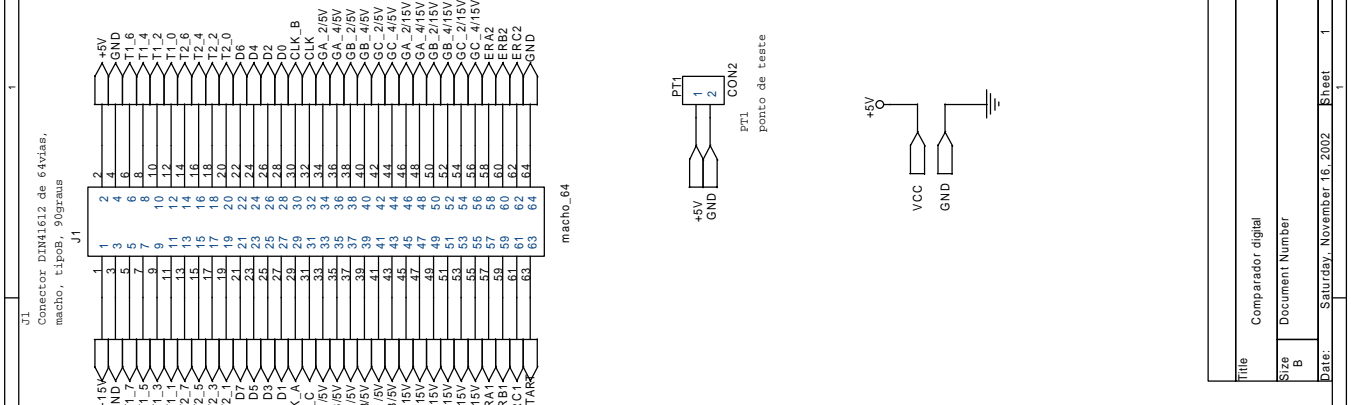
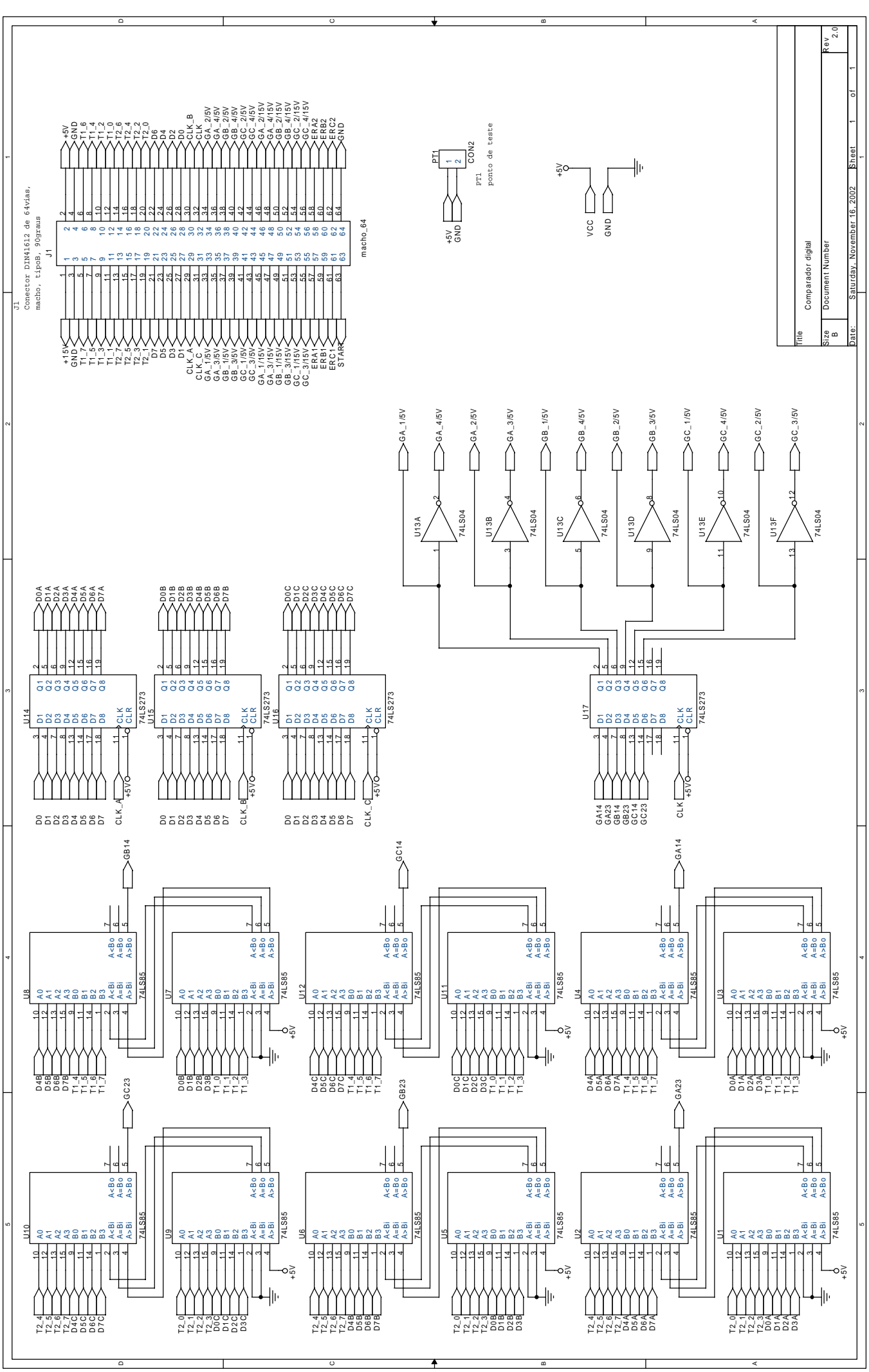
Title: gerador triangular digital  
Size: B  
Date: Saturday, November 16, 2002  
Sheet: 1 of 1  
Rev: 2.0



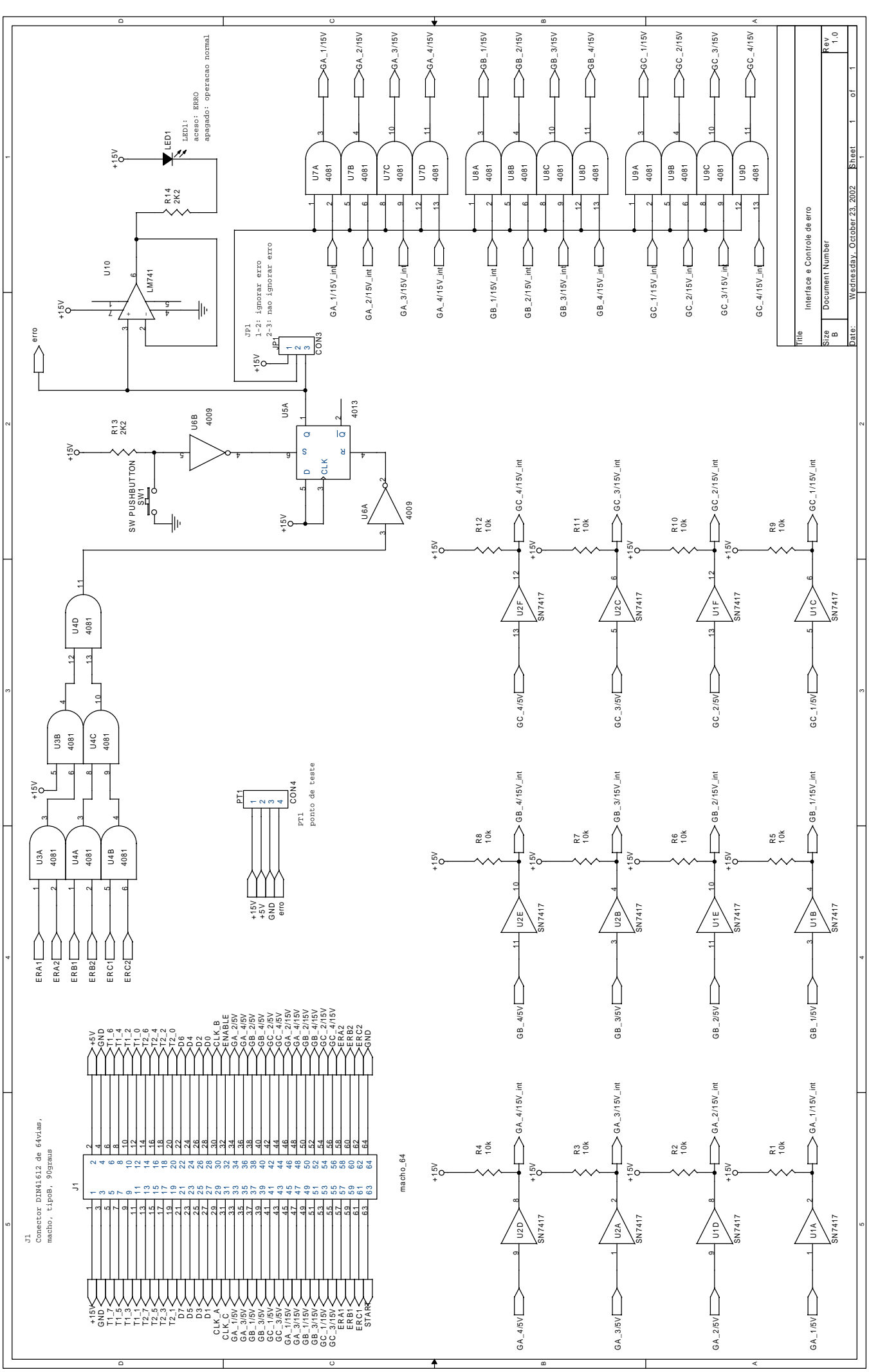


Title		Oscilador Cristal	
Size	A	Document Number	
Rev	1.0	Date:	Monday, July 21, 2003
Sheet 1 of 1		2	









Title	Interface e Controle de erro		
Size	B	Document Number	
Date:	Wednesday, October 23, 2002	Sheet	1 of 1
Rev	1.0		

J1  
Conector DIN41612 de 64vias,  
macho, tipoB, 90graus

PT1  
ponto de teste

JP1  
1-2: ignorar erro  
JP2 2-3: nao ignorar erro

LED1:  
aceso: ERRO  
apagado: operacao normal

erro

macho\_64

CON4

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

CON3

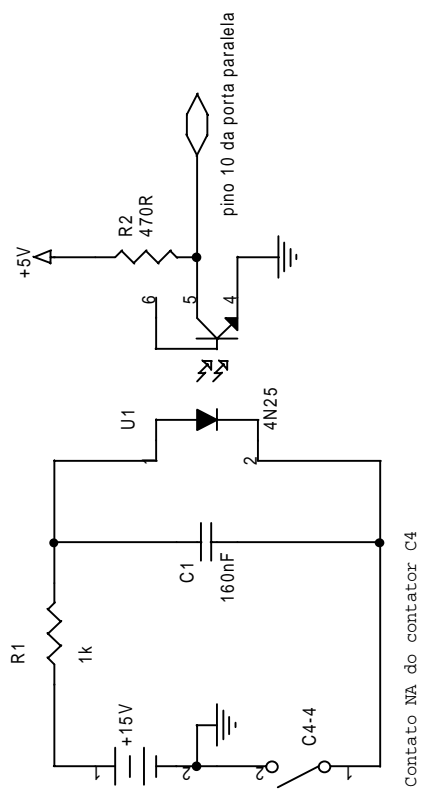
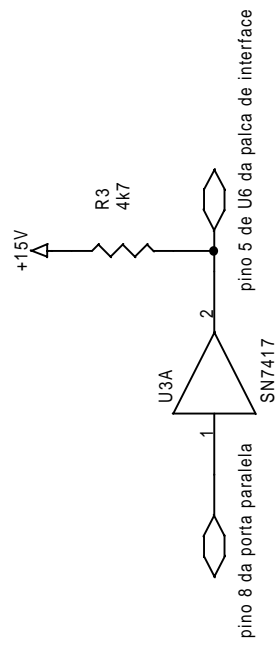
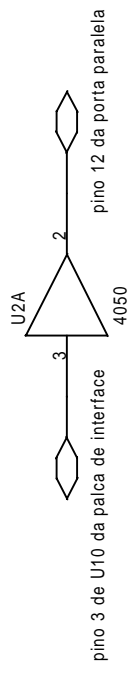
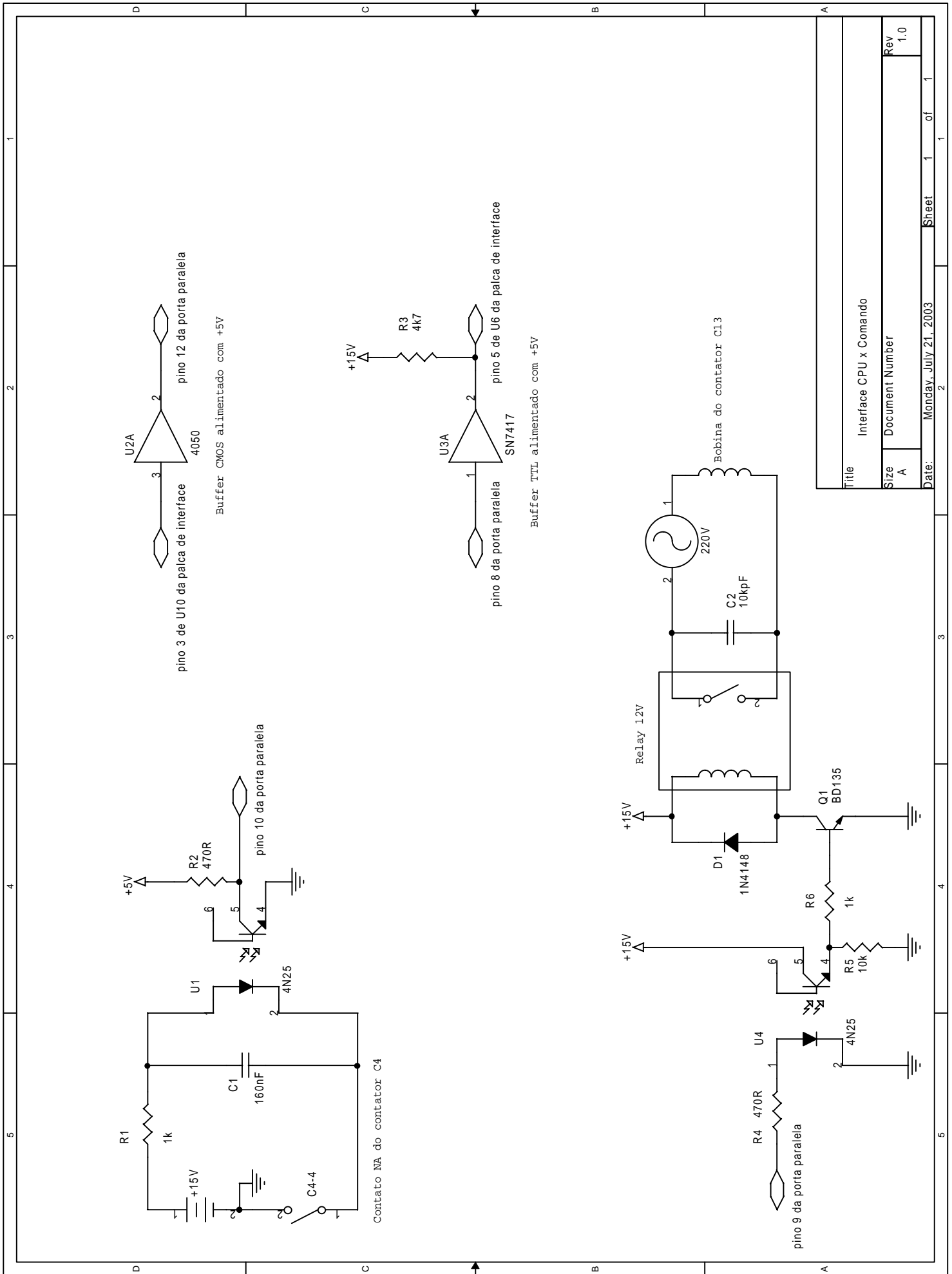
CON3

CON3

CON3







Title		Interface CPU x Comando	
Size	A	Document Number	
Rev	1.0	Date	Monday, July 21, 2003
Sheet 1 of 1		Sheet 1 of 1	





