



UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

DEPARTAMENTO DE SISTEMAS E CONTROLE DE ENERGIA

IMPLEMENTAÇÃO DO MÉTODO DIRETO DO CONTROLE  
VETORIAL DE UM MOTOR DE INDUÇÃO, COM ORIENTAÇÃO DE  
FLUXO DO ESTATOR UTILIZANDO DSP

*JOSÉ ALBERTO TORRICO ALTUNA*

Orientador : Prof. Dr. Edson Bim

Co-orientador: Prof. Dr. Jaime Szajner

Este exemplar corresponde à redação final da tese  
defendida por JOSÉ ALBERTO TORRICO ALTUNA

e aprovada pela Comissão

Juizadores em 26/02/97

  
Orientador

Dissertação apresentada à  
Faculdade de Engenharia  
Elétrica e de Computação da  
Universidade Estadual de  
Campinas como parte dos  
requisitos exigidos para a  
obtenção do título de *Mestre  
em Engenharia Elétrica*.

Fevereiro/1997

UNIDADE	BC		
N.º CHAMADA:	UNICAMP		
	T637i		
V.	Ex.		
TOMBO BC/	30560		
PROC.	28497		
C	<input type="checkbox"/>	D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00		
DATA	24/05/97		
N.º CPD			

CM-0009312-1

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

T637i      Torrico Altuna, José Alberto  
Implementação do método direto do controle vetorial de um motor de indução, com orientação de fluxo do estator utilizando DSP / José Alberto Torrico Altuna.--Campinas, SP: [s.n.], 1997.

Orientador: Edson Bim.

Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Motores elétricos de indução. 2. Processamento de sinais - Técnicas digitais. 3. Sistema de controle digital. 4. Controle automático. I. Bim, Edson. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

*A Maria, minha avó*

# AGRADECIMENTOS

Agradeço ..

Ao Prof. Dr. Edson Bim pela sua excelente orientação deste trabalho. Sua constante preocupação, incentivo e pronta disponibilidade até o final foram determinantes para a conclusão desta dissertação.

Ao Prof. Dr. Jaime Szajner e ao Prof. Dr. Márcio Luiz de Andrade Netto pelo apoio no laboratório.

À minha família no Perú, pelo apoio constante nos anos de formação pessoal e acadêmica, eles sempre me incentivaram e ajudaram desde criança.

A Alina, pela sua paciência e apoio, a constante preocupação e sacrificio foram fundamentais para o desenvolvimento deste projeto.

Aos meus amigos Rodolfo Moreno e Victor Bravo por sua importante colaboração e apoio constante.

A Andrés pela informação fornecida.

A Eduardo Garcia por oferecer-me sua amizade.

A Lino pela companhia nos momentos mais importantes deste projeto.

À Faculdade de Engenharia Elétrica e de Computação, em especial ao Departamento de Computação e Automação, pela oportunidade dada à realização do curso de Mestrado.

Ao CNPq pelo suporte financeiro.

# SUMÁRIO

**RESUMO**

**ABSTRACT**

## **CAPÍTULO 1**

*INTRODUÇÃO*.....1

## **CAPITULO 2**

*TEORIA DO CONTROLE VETORIAL DA MÁQUINA DE INDUÇÃO POR ORIENTAÇÃO DO FLUXO DE ESTATOR*

2.1 Introdução .....5

2.2 Uma Revisão Bibliográfica do Método Direto.....6

2.3 Modelo Dinâmico da Máquina de Indução.....9

2.4 Equações do Controle Vetorial por Orientação do Fluxo de Estator..... 12

2.5 Projeto do Desacoplador..... 13

2.6 Transformação Entre Sistemas de Coordenadas..... 15

## **CAPITULO 3**

*ESTRUTURA E DESCRIÇÃO FUNCIONAL DO SISTEMA IMPLEMENTADO*

3.1 Introdução..... 18

3.2 Transformação Trifásico-Bifásico..... 18

3.3 Estimação .....20

3.3.1 Equações de estimação de fluxo .....20

3.3.2 Estimação de velocidade .....22

3.4 Sistema de Controle.....	24
3.4.1 Equações do algoritmo do controlador.....	24
3.4.2 Implementação digital do controlador PI.....	25
3.5 Sistema Conversor de Potência.....	27
3.5.1 Modulador vetorial PWM realimentado em corrente.....	27
3.5.2 Representação da tensão de saída do inversor .....	28
3.5.3 Estratégias do controle da corrente.....	31
3.5.4 Comportamento do vetor espacial de corrente.....	34
3.5.5 Geração da seqüência de chaveamento.....	37

## **CAPITULO 4**

### *DESCRIÇÃO DA IMPLEMENTAÇÃO EXPERIMENTAL UTILIZANDO DSP*

4.1 Introdução.....	43
4.2 Descrição do Sistema Implementado.....	45
4.3 Hardware do Sistema.....	46
4.3.1 Circuito de Aquisição.....	46
4.3.2 Circuito de geração de correntes e tensões no sistema de referência $\alpha$ - $\beta$ .....	49
4.3.3 Interface de conversão A/D e D/A.....	50
4.4 Software do sistema.....	51
4.4.1 Programa de inicialização.....	51
4.4.2 Temporização do sistema.....	53
4.4.3 Lógica de chaveamento do inversor.....	55
4.4.4 Rotina de aquisição de dados.....	57
4.4.5 Geração de correntes de referência.....	57
4.4.6 Software de estimação de fluxo.....	59
4.4.7 Rotina de controle vetorial.....	61

## **CAPITULO 5**

### *RESULTADOS EXPERIMENTAIS*

5.1 Introdução.....	64
5.2 Verificação da Operação do Sistema Implementado.....	65
5.3 Resultados referentes ao método direto de controle.....	70

## **CAPITULO 6**

<i>CONCLUSÕES</i> .....	73
-------------------------	----

<b>Apêndice A</b>	Determinação de parâmetros do motor de indução.....	76
-------------------	---	----

<b>Apêndice B</b>	Descrição do DSP56001.....	78
-------------------	----------------------------	----

B.1	Diagrama de blocos do DSP56001.....	78
-----	-------------------------------------	----

B.2	Espaços de memória.....	79
-----	-------------------------	----

B.3	Registros do DSP56001 .....	82
-----	-----------------------------	----

<b>Apêndice C</b>	Descrição do sistema de desenvolvimento.....	85
-------------------	--	----

<b>Bibliografia</b> .....	89
---------------------------	----

## RESUMO

É desenvolvida uma metodologia de implementação digital do controle vetorial por orientação de campo do motor de indução trifásico. É estudado o método direto de orientação baseado no fluxo de estator. A partir da medição das tensões terminais do estator e integração digital das mesmas obtém-se o vetor fluxo de estator e a frequência elétrica. Desde que existe um acoplamento entre o fluxo e a componente em quadratura da corrente de estator neste método, um termo desacoplador é utilizado para eliminar este inconveniente. O Processador Digital de Sinais DSP56001 da Motorola processa os sinais de tensão e corrente adquiridos por sensores e executa algoritmos que tem as seguintes funções: determinar o vetor fluxo de estator, estimar velocidade, realizar o controle de velocidade ou de torque, e gerar os sinais de chaveamento para um inversor com modulação vetorial PWM controlado por corrente. São apresentados resultados de testes experimentais que mostram o potencial do controle vetorial, método direto, orientado, segundo o fluxo de estator.

## Abstract

A digital implementation of stator field oriented control is studied. The direct computation of stator flux vector is performed using measured stator currents and voltages. The integration of the stator voltage minus the drop in stator resistance results in the stator flux which is used to compute the electrical frequency and slip frequency of a squirrel cage induction motor. From these data the motor speed is calculated. Since there is a coupling between the flux and torque in this control method, a decoupling term is applied to eliminate this inconvenient. The speed and torque control are obtained choosing the adequate feedback loop. The implementation of the control algorithm is based on a Digital Signal Processor (DSP 56001) of the Motorola, which processes the measured current and voltage signals. The implemented system uses a low harmonic content current controlled PWM inverter which is operated by employing concurrent routines with the control program. Experimental results are obtained and analyzed in view of high and low operation frequencies.

# *CAPÍTULO 1*

## *INTRODUÇÃO*

A alta competitividade no meio produtivo faz com que as indústrias tornem-se, a cada dia, mais exigentes quanto a qualidade e quantidade de seus produtos. Com isto, na área de automação industrial são exigidos acionamentos de alto desempenho. O motor de indução de rotor em gaiola, devido às suas características naturais de robustez e baixos custos de produção, é, atualmente, muito usado em acionamentos à velocidade variável.

O surgimento da técnica do controle vetorial por orientação de fluxo permitiu que a máquina de indução fosse controlada de maneira semelhante às máquinas de corrente contínua. Nesta técnica, a corrente de estator é decomposta em duas componentes: uma responsável pelo estabelecimento do fluxo (componente de campo) e outra diretamente ligada ao conjugado eletromagnético (componente de conjugado).

Existem, basicamente, dois métodos de controle por orientação de campo da máquina de indução: o indireto e o direto. No método indireto, a posição do vetor fluxo é determinada a partir do modelo do motor de indução. No método direto o fluxo é medido ou computado a partir das grandezas terminais medidas (tensões e correntes).

A orientação de campo na máquina de indução pode ser realizada utilizando a orientação segundo o fluxo de rotor, fluxo de estator e o de entreferro.

A orientação de campo indireta, segundo o fluxo de rotor, é a mais estudada por ser de fácil implementação e possibilitar um controle desacoplado. A sua principal desvantagem está no fato de ser sensível à variação dos parâmetros, pois a relação do escorregamento, que é fundamental na determinação da posição espacial do fluxo do rotor é fortemente dependente da resistência do rotor, e da indutância de magnetização. Nesta estratégia, a velocidade do rotor e as componentes da corrente do estator constituem-se nas variáveis de entrada do sistema de controle.

A orientação de campo direta permite que a posição do fluxo seja determinada diretamente a partir da colocação na máquina de bobinas exploratórias, sensores de fluxo de efeito Hall ou através da medição das grandezas terminais do estator (tensões e correntes). A vantagem do método direto que utiliza as grandezas terminais do estator está no fato de ser sensível apenas à resistência do estator, embora exija uma estrutura computacional complexa. Neste trabalho será realizada a implementação do método direto segundo o fluxo de estator.

O desenvolvimento dos microprocessadores e dos semicondutores de potência na década de 80, deu um forte impulso à implementação do controle por orientação de campo do motor de indução, permitindo que se alcançasse um desempenho dinâmico do motor de indução comparável ao motor de corrente contínua. Com isto o motor de indução pode substituir o motor de corrente contínua em acionamentos que exigem alto desempenho e robustez. Para conseguir este bom desempenho do motor é preciso utilizar ferramentas de implementação digital (hardware e software) que permitam a execução, em tempo real, das operações do sistema.

O desempenho de um sistema digital de controle de motor, não somente depende da estrutura do processador, capacidades de computação e velocidade de execução senão também dos algoritmos de controle implementados. Estes fatores relacionam-se mutuamente e, portanto, devem ser considerados conjuntamente no projeto do sistema.

Microprocessadores, com alta capacidade de processamento e de velocidade de execução, levam a projetos de sistemas de controle de alto desempenho dinâmico. Como consequência, incrementa-se a velocidade de amostragem que resulta em menor ruído de quantização e atraso computacional, melhoram-se a largura de banda do controle, a estabilidade e a rejeição ao distúrbio de carga. Também, pode-se implementar algoritmos de controle avançado como: realimentação de estados, controle adaptativo e controles neural e fuzzy.

Nos anos recentes os processadores digitais de sinais (DSPs) tem sido implementados com características de desempenho muito superiores comparados aos microprocessadores convencionais.

A maior parte dos DSPs são construídos com arquitetura Harvard, onde dados e instruções ocupam memórias separadas e usam barramentos separados. Devido a esta estrutura de barramento dual, o processador pode buscar simultaneamente uma instrução e um operando. Operação paralela de instruções e transferência de dados é possível, resultando em uma velocidade de execução de instruções mais alta. O paralelismo pode ser de dois a quatro níveis, dependendo da arquitetura.

Nos atuais DSPs, a velocidade de execução tem sido melhorada usando algumas unidades independentes, múltiplos conjuntos de barramentos, e unidades adicionais como memória "cache" de instruções, arquivo de registros e memórias de duplo acesso. Outra característica importante do DSP é que a maioria das suas instruções são executadas em um ciclo de instrução. Estas vantagens tem sido complementadas com a diminuição dos custos de hardware e os sistemas de desenvolvimento para DSPs, devido ao processo de globalização.

O melhor entendimento da utilização de DSPs e do controle vetorial é alcançado realizando a implementação experimental de um sistema digital. Sendo assim, é possível investigar as características de operação de um sistema de controle vetorial utilizando um DSP. O presente projeto tem como motivação principal a realização desta implementação segundo a orientação do fluxo de estator.

A estrutura geral deste trabalho é descrita nos parágrafos seguintes:

No capítulo 2 faz-se uma revisão dos estudos realizados sobre o método direto de orientação de campo, estabelecem-se as equações do modelo dinâmico da máquina de indução, obtém-se as equações do controle vetorial por orientação de fluxo de estator e por último são apresentadas as equações de transformação entre diferentes sistemas de coordenadas.

No capítulo 3 é apresentado o diagrama de blocos do sistema implementado com a descrição detalhada do mesmo. São analisadas as equações que governam a estimação do fluxo de estator a partir das grandezas terminais (tensões e correntes) e, é projetado o algoritmo do controlador PI. São apresentadas a descrição e a análise do inversor implementado, que consiste em um modulador vetorial controlado por corrente. Um algoritmo de chaveamento gera a seqüência de vetores de corrente que são aplicados ao circuito de potência do inversor.

O capítulo 4 centraliza os objetivos deste projeto de pesquisa, fazendo a descrição em detalhe da parte experimental do sistema. São apresentados os circuitos e interfaces usados na implementação do sistema proposto.

No capítulo 5 são apresentados os resultados experimentais obtidos que verificam o desempenho do controle vetorial direto com orientação do fluxo de estator.

As conclusões a respeito da utilização do DSP na implementação do controle bem como sobre a orientação do fluxo de estator são apresentadas no capítulo 6.

## CAPÍTULO 2

# TEORIA DO CONTROLE VETORIAL DA MÁQUINA DE INDUÇÃO POR ORIENTAÇÃO DO FLUXO DE ESTATOR

### 2.1 INTRODUÇÃO

Embora o princípio do controle vetorial se aplique aos motores síncronos e de indução, é neste último que reside o maior interesse, pois ele é o mais utilizado, por ser de baixo custo, robusto e de fácil construção. O conceito de orientar o fluxo consiste em conhecer a sua posição espacial e então alinhar os *eixos d-q* de tal forma que um dos três fluxos ( de estator, de rotor ou de entreferro) coincida com o eixo direto. O sucesso desta orientação depende do conhecimento da posição do fluxo para o qual a orientação é desejada. Dois métodos existem para determinar a posição do fluxo.[1]

**1- Orientação indireta do campo:** em geral escolhe-se o fluxo de rotor na orientação e tem sido o mais adotado por levar a um completo desacoplamento do torque e fluxo magnético. Nesta estratégia, é de fundamental importância a relação do escorregamento para estimar a posição do fluxo.

A relação do escorregamento depende dos parâmetros resistência de rotor e da indutância de magnetização, representando este fato uma desvantagem, pois leva a um

acoplamento entre o torque e fluxo magnético. Simplicidade de implementação e o desacoplamento inerente, se o sistema estiver bem sintonizado, constituem as suas vantagens.

**2- Orientação direta do campo:** Na literatura encontramos dois esquemas: a) o esquema pelo qual a posição do fluxo é diretamente medida através de bobinas especiais de inspeção e b) aquele que estima a posição do fluxo a partir das medições das correntes e tensões de estator. Enquanto o uso de bobinas exploratórias tem o inconveniente de exigir alterações no motor e portanto na linha industrial de produção, o segundo método se aplica aos motores de objetivos gerais e exige processadores rápidos para estimação do fluxo. A partir do início dos anos 90, o interesse pelo controle vetorial do motor de indução por orientação direta, em particular aquele baseado no fluxo de estator, tem sido estudado graças ao fato dos processadores digitais de sinais apresentarem cada vez mais custos declinantes.

O controle de máquinas elétricas por orientação de campo (*FOC-field oriented control*) possui uma estrutura complexa e só passou a ser largamente utilizado a partir do momento em que se teve disponível microprocessadores com velocidade e precisão razoáveis para implementar os algoritmos necessários.

Neste capítulo são apresentados uma breve revisão bibliográfica do método direto de determinação de fluxo e as equações do modelo matemático do motor de indução sob controle vetorial por orientação de fluxo do estator.

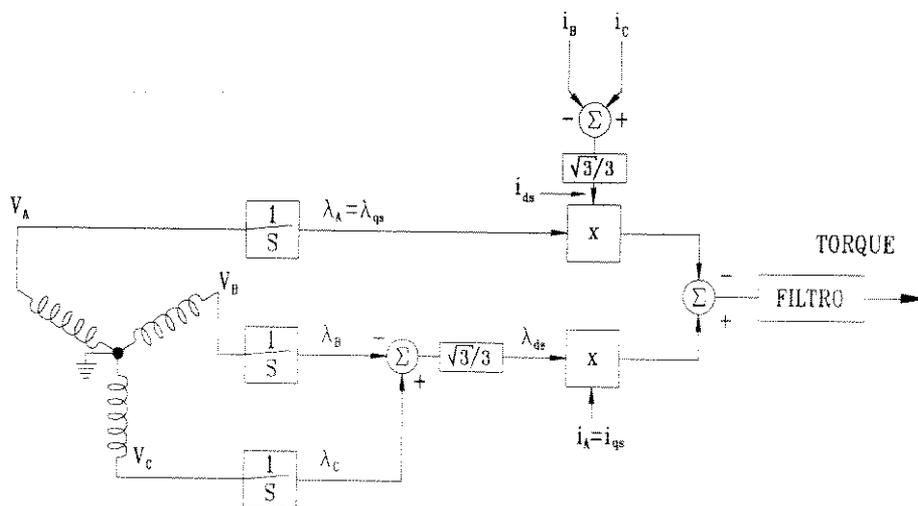
## **2.2 REVISÃO BIBLIOGRÁFICA SOBRE O MÉTODO DIRETO DE ORIENTAÇÃO DO FLUXO**

Nesta seção com o intuito de apresentar um histórico do método direto comentamos as principais propostas de sistemas que fazem uso do método direto de determinação de fluxo em máquinas de indução.

O método direto de orientação foi introduzido, em 1972, por Blaschke [2] que, através de sensores de efeito Hall colocados no entreferro, determinava o fluxo de entreferro e, a partir deste, calculava o vetor fluxo de rotor, implementando, desta maneira,

a orientação direta segundo o fluxo de rotor. Como as máquinas não são projetadas para acomodar estes sensores, este esquema não é prático, sendo, portanto, não aconselhável às aplicações industriais.

Plunkett [3], em 1977, propõe a determinação do fluxo magnético de entreferro para um motor de indução em operação, através da integração das tensões geradas nas bobinas de inspeção, especialmente colocadas no estator. Com as correntes de estator medidas e transformadas ao sistema de referência  $d-q$ , o conjugado foi calculado como ilustrado na Figura 2.1. Lipo[4], em 1977, projetou um enrolamento especial de estator para permitir adquirir o fluxo magnético, sendo que este enrolamento foi otimizado objetivando minimizar os harmônicos espaciais e os erros de saturação. Comparado com os sensores de efeito Hall, as bobinas de fluxo não funcionam bem em velocidade zero pelo fato de utilizarem integradores analógicos na obtenção do fluxo.

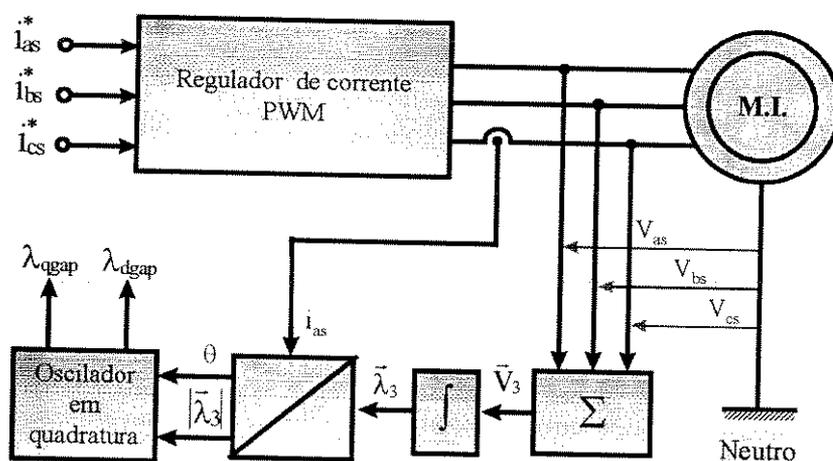


**Figura 2.1-Sistema implementado por Plunkett para determinação direta do fluxo de entreferro e do conjugado eletromagnético.**

Para eliminar a colocação de bobinas de inspeção e, assim evitar modificações no motor ou o projeto de motores especiais, Lipo e Chang [5], em 1986, desenvolveram um enrolamento de estator com "taps" que permitia a aquisição do fluxo. Este esquema foi implementado por Zinger et al. [6] em um controle vetorial do motor de indução

baseado em microprocessador. Como no caso de bobinas de inspeção, esta técnica não funcionou bem em baixas frequências.

Moreira [7] sem se utilizar de bobinas especiais de estator e explorando o fato de que, na conexão Y do estator, a soma das tensões terminais resulta na sua componente de 3<sup>a</sup> harmônica, gerada pelo correspondente harmônico de fluxo de entreferro, implementou um sistema que fazia a medição da componente de 3<sup>a</sup> harmônica de tensão e, a partir desta, determinava a posição e a amplitude da componente fundamental do fluxo de entreferro. A desvantagem deste método está no fato de ser aplicável, somente para máquinas com o estator conectado em Y. A Figura 2.2 ilustra o referido sistema.



**Figura 2.2-Sistema de Moreira para cálculo da posição e da magnitude do fluxo de entreferro utilizando a 3<sup>a</sup> harmônica da tensão de estator.**

Enquanto Moreira a partir das tensões terminais de estator determinou o fluxo de entreferro, Xu e Novotny [8] utilizaram estas mesmas tensões para determinar a posição do fluxo de estator e, assim, implementar o controle direto por orientação de fluxo de estator. Como esta implementação exige grande potência de cálculo do processador devido à necessidade de realizar um observador de fluxo a partir das tensões e correntes medidas através de sensores que necessitam ser transformadas para sistemas bifásicos, utilizou-se um sistema baseado no DSP. As vantagens desta implementação estão em observar diretamente o fluxo de estator através de grandezas terminais da máquina

(tensões e correntes) como também a possibilidade de fazer um controle sem a necessidade de sensores de velocidade, embora o método seja sensível à resistência de estator.

Para resolver o problema do processo de integração em baixas frequências, apresentado na implementação digital, Bose et. al. [9] implementaram um sistema com filtros, em cascata, e que possuíam constantes de tempo programáveis. Outra característica interessante desta implementação é que o controlador era híbrido: em baixas frequências (velocidades próximas de zero) se utilizava o método indireto de orientação e para velocidades maiores era adotado o método direto.

### 2.3 MODELO DINÂMICO DA MÁQUINA DE INDUÇÃO

As equações de tensão da máquina de indução trifásica com rotor gaiola de esquilo é representada no sistema de coordenadas  $d$ - $q$  girando na velocidade síncrona  $\omega_e$ , pelas seguintes equações [10]:

- *equações de tensão do circuito de estator*

$$v_{qs}^e = r_s i_{qs}^e + p \lambda_{qs}^e + \omega_e \lambda_{ds}^e \quad (2.1)$$

$$v_{ds}^e = r_s i_{ds}^e + p \lambda_{ds}^e - \omega_e \lambda_{qs}^e \quad (2.2)$$

- *equações de tensão do circuito de rotor:*

$$0 = v_{qr}^e = r_r i_{qr}^e + p \lambda_{qr}^e + (\omega_e - \omega_r) \lambda_{dr}^e \quad (2.3)$$

$$0 = v_{dr}^e = r_r i_{dr}^e + p \lambda_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e \quad (2.4)$$

- *equações do fluxo concatenado pelas bobinas de estator e de rotor*

$$\lambda_{qs}^e = L_s i_{qs}^e + L_m i_{qr}^e \quad (2.5)$$

$$\lambda_{ds}^e = L_s i_{ds}^e + L_m i_{dr}^e \quad (2.6)$$

$$\lambda_{qr}^e = L_m i_{qs}^e + L_r i_{qr}^e \quad (2.7)$$

$$\lambda_{dr}^e = L_m i_{ds}^e + L_r i_{dr}^e \quad (2.8)$$

- *conjugado eletromagnético gerado*

$$T = \frac{3 P L_m}{2 L_r} (\lambda_{dr}^e i_{qs}^e - \lambda_{qr}^e i_{ds}^e) \quad (2.9)$$

nas quais,  $P$  é número de pólos,  $r_s$  e  $r_r$  são as resistências de estator e rotor, respectivamente,  $L_s, L_r$  e  $L_m$  representam as indutâncias de estator, rotor e de magnetização, respectivamente.

Para que a máquina seja descrita por equações expressas em função das correntes de estator, dos vetores fluxo de rotor e fluxo de estator, as componentes de corrente  $i_{dr}^e$  e  $i_{qr}^e$  precisam ser eliminadas das equações (2.3)-(2.9). Rescrevendo as equações 3.7 e 3.8 como:

$$i_{qr}^e = \frac{1}{L_r} (\lambda_{qr}^e - L_m i_{qs}^e) \quad (2.10)$$

$$i_{dr}^e = \frac{1}{L_r} (\lambda_{dr}^e - L_m i_{ds}^e) \quad (2.11)$$

e substituindo-as em (2.3), (2.4), (2.5) e (2.6) obtém-se, após algumas manipulações algébricas, as seguintes equações:

$$(1 + T_r p) \lambda_{dr}^e = L_m i_{ds}^e + \omega_{slip} T_r \lambda_{qr}^e \quad (2.12)$$

$$(1 + T_r p) \lambda_{qr}^e = L_m i_{qs}^e - \omega_{slip} T_r \lambda_{dr}^e \quad (2.13)$$

$$\lambda_{qr}^e = \frac{L_r}{L_m} (\lambda_{qs}^e - \sigma L_s i_{qs}^e) \quad (2.14)$$

$$\lambda_{dr}^e = \frac{L_r}{L_m} (\lambda_{ds}^e - \sigma L_s i_{ds}^e) \quad (2.15)$$

nas quais:

$$T_r = \frac{L_r}{R_r} \text{ é a constante de tempo elétrica do rotor}$$

$\sigma = 1 - \frac{L_m^2}{L_s L_r}$  é o fator de dispersão total que depende da forma geométrica das ranhuras e do comprimento do entreferro.

Este fator de dispersão pode ser expresso também em função dos fatores de dispersão de estator  $\sigma_s$  e de rotor  $\sigma_r$  [10]. Definindo,

$$L_s = (1 + \sigma_s) L_m \quad (2.16)$$

$$L_r = (1 + \sigma_r) L_m \quad (2.17)$$

escreve-se que

$$\sigma = 1 - \frac{1}{(1 + \sigma_s)(1 + \sigma_r)} \quad (2.18)$$

Como, neste trabalho, é proposta a orientação segundo o fluxo de estator utilizando as correntes e tensões terminais do motor, é conveniente expressar as equações do modelo em função destas variáveis. Na verdade, como o sistema de controle opera com estas variáveis definidas no sistema de referência  $d-q$ , devemos escrever as equações em função das componentes  $d-q$  dos vetores fluxo e corrente de estator. Substituindo as equações (2.14) e (2.15) em (2.12) e (2.13), obtém-se:

$$-(1 + T_r p) \lambda_{qs}^e + (1 + \sigma T_r p) L_s i_{qs}^e - \omega_{slip} T_r (\lambda_{ds}^e - \sigma L_s i_{ds}^e) = 0 \quad (2.19)$$

$$\omega_{slip} T_r \lambda_{qs}^e - (1 + T_r p) \lambda_{ds}^e + (1 + \sigma T_r p) L_s i_{ds}^e - \omega_{slip} T_r \sigma L_s i_{qs}^e = 0 \quad (2.20)$$

A equação do torque, dada por (2.9), torna-se, em função das componentes da corrente de estator e do vetor de fluxo do estator, em

$$T_e = \frac{3P}{4} (\lambda_{ds}^e i_{qs}^e - \lambda_{qs}^e i_{ds}^e) \quad (2.21)$$

## 2.4 EQUAÇÕES DO CONTROLE VETORIAL POR ORIENTAÇÃO DO FLUXO DE ESTATOR [12]

Para que o sistema de coordenadas girante  $d$ - $q$  esteja corretamente orientado segundo o fluxo de estator, o eixo direto positivo deste deve coincidir, em qualquer instante, com vetor de fluxo de estator  $\vec{\lambda}_s$ , ou, em outras palavras, a componente em quadratura do fluxo de estator deve ser nula,  $\lambda_{qs} = 0$ . Portanto, levando em consideração esta condição, as equações (2.19) e (2.20) reduzem-se à:

$$(1 + T_r p) \lambda_{ds}^e = (1 + \sigma T_r p) L_s i_{ds}^e - \omega_{slip} T_r \sigma L_s i_{qs}^e \quad (2.22)$$

$$(1 + \sigma T_r p) L_s i_{qs}^e - \omega_{slip} T_r (\lambda_{ds}^e - \sigma L_s i_{ds}^e) = 0 \quad (2.23)$$

Ao contrário da orientação segundo o fluxo de rotor, a orientação do fluxo de estator apresenta um acoplamento entre a componente de torque  $i_{qs}^e$  e o fluxo de estator  $\lambda_{ds}^e$ , como pode ser observado nas equações (2.22) e (2.23).

No caso, de se utilizar o método indireto, é necessário encontrar a relação do escorregamento. A partir da equação (2.23) pode-se escrever que a frequência de escorregamento do motor como uma função do fluxo e as correntes do estator é dada por:

$$\omega_{slip} = \frac{(1 + \sigma T_r p) L_s i_{qs}^e}{T_r (\lambda_{ds}^e - \sigma L_s i_{ds}^e)} \quad (2.24)$$

Se  $\lambda_{qs} = 0$ , a expressão do conjugado eletromagnético torna-se

$$T_e = \frac{3P}{4} \lambda_{ds}^e i_{qs}^e \quad (2.25)$$

indicando que o torque pode ser regulado pelo controle do fluxo  $\lambda_{ds}$  e/ou pela componente da corrente de estator  $i_{qs}$ .

## 2.5 PROJETO DO DESACOPLADOR

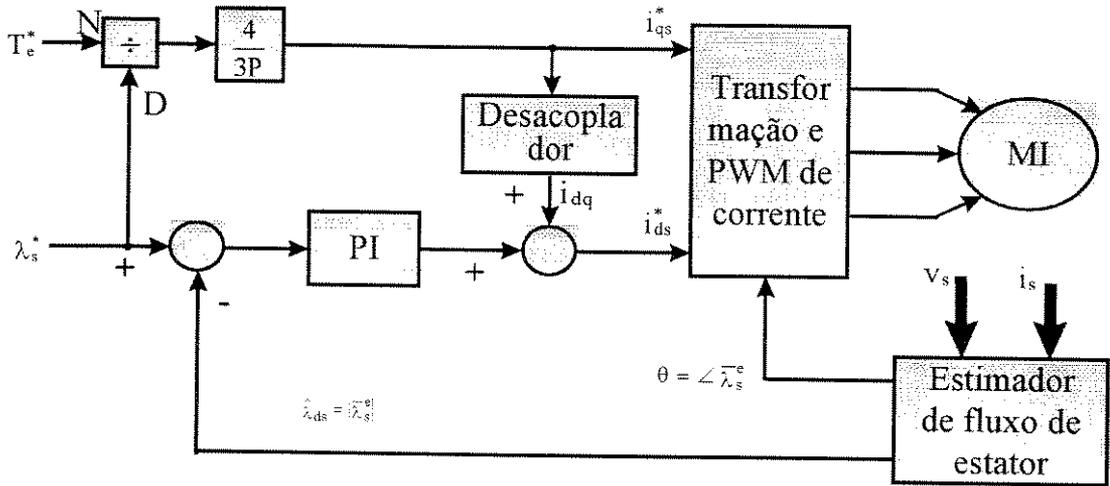
A equação (2.22) mostra que qualquer mudança na corrente de torque  $i_{qs}^e$ , mantendo  $i_{ds}^e$  constante, produzirá um transitório no fluxo de estator. Contudo, para conseguir uma resposta de torque rápida,  $\lambda_{ds}$  deve ser mantido constante. Isto pode ser realizado pelo regulador de fluxo da Figura 2.3. Neste esquema, existem dois sinais de referência  $T_e^*$ , para o controle do torque e  $\lambda_{ds}^*$  para o controle do fluxo. A malha de realimentação de fluxo permite, empregando um regulador PI manter a componente de fluxo  $\lambda_{ds}^e$  constante. O bloco desacoplador é projetado considerando que qualquer incremento no valor da corrente  $i_{qs}^e$  deve ser compensado com um incremento em  $i_{ds}^e$ . Desta forma, a corrente de referência  $i_{ds}^*$  deve obedecer a seguinte relação:

$$i_{ds}^* = G_f(p)(\lambda_{ds}^* - \lambda_{ds}) + i_{dq} \quad (2.26)$$

na qual,  $G_f(p) = K_{pf} + \frac{K_{if}}{p}$  representa a função de transferência do regulador de fluxo e  $i_{dq}$  é o elemento desacoplador.

Substituindo o valor da corrente  $i_{ds}^*$  expressa por (2.26) na equação (2.22), obtém-se:

$$(1 + T_r p) \lambda_{ds}^e = (1 + \sigma T_r p) L_s G_f(p) (\lambda_{ds}^* - \hat{\lambda}_{ds}) + (1 + \sigma T_r p) L_s i_{dq}^e - \omega_{slip} T_r \sigma L_s i_{qs}^e \quad (2.27)$$



**Figura 2.3-**Esquema de orientação do fluxo de estator com rede de desacoplamento

Para que o desacoplamento seja obtido, os dois últimos termos da equação (2.27) devem cancelar-se, ou seja,

$$(1 + \sigma T_r p) L_s i_{dq}^e - \omega_{slip} T_r \sigma L_s i_{qs}^e = 0 \quad (2.28)$$

e, com isto, o termo desacoplador deve ser dado por:

$$i_{dq} = \frac{\omega_{slip} T_r \sigma i_{qs}^e}{(1 + \sigma T_r p)} \quad (2.29)$$

Se a expressão da frequência de escorregamento, dada por (2.24), for substituída em (2.29), obtém-se:

$$i_{dq} = \frac{\sigma L_s (i_{qs}^e)^2}{(\lambda_{ds}^e - \sigma L_s i_{ds}^e)} \quad (2.30)$$

Como se observa nesta equação, o termo desacoplador, tem como entradas o fluxo de estator  $\lambda_{ds}$  e as componente de corrente  $i_{ds}$  e  $i_{qs}$ . O desacoplador depende também, do fator de dispersão total  $\sigma$  e indutância do estator  $L_s$ . Esta forma de compensação se configura como sendo "Feedforward", pois a alteração na componente de corrente  $i_{qs}$  é,

imediatamente, alimentada à referência  $i_{ds}^*$ , antes da correção do fluxo, melhorando, desta maneira, a resposta dinâmica do sistema.

Quando o sistema de controle está sintonizado, é gerada corretamente a corrente  $i_{ds}^e$  para uma corrente  $i_{qs}^e$  dada, como consequência  $\lambda_{ds}^e$  não é alterado com as mudanças na referência de torque  $T_e^*$ . Esta característica pode ser apreciada ao rescrever a equação (2.27) com  $i_{dq}$  determinado pela equação (2.29):

$$(1 + T_r p) \lambda_{ds}^e = (1 + \sigma T_r p) L_s G_f(p) (\lambda_{ds}^* - \hat{\lambda}_{ds}) \quad (2.31)$$

o que mostra que  $\lambda_{ds}$  não depende de  $i_{qs}$ , ou seja, eles estão desacoplados.

## 2.6 TRANSFORMAÇÃO ENTRE SISTEMAS DE COORDENADAS

Nas implementações do controle vetorial por orientação de campo, as variáveis elétricas tais como tensão, corrente, fluxo concatenado, geralmente são definidas no sistema de referência estacionário ou síncrono. No primeiro, as variáveis são alternadas, enquanto que no segundo elas se tornam grandezas contínuas. As medições são realizadas no sistema trifásico  $abc$  e o sistema de controle, normalmente, opera no sistema bifásico definido no sistema estacionário (eixos  $\alpha\text{-}\beta$ ) ou síncrono ( $d\text{-}q$ ).

A seguir apresentamos uma síntese destas transformações de coordenadas [10], segundo os sistemas mostrados na Figura 2.4, na qual  $\theta$  representa a posição angular instantânea do eixo direto  $d$  em relação ao eixo  $\alpha$ . Nas relações que estabelecem as transformações,  $f_{an}$ ,  $f_{bn}$  e  $f_{cn}$  representam as grandezas de fase da máquina de indução e  $f_0$  a componente de seqüência zero.

- *transformação abc para  $\alpha\text{-}\beta$ :*

$$f_{\alpha\beta} = \frac{2}{3} (f_{an} + \bar{a} f_{bn} + \bar{a}^2 f_{cn}) \quad (2.32)$$

$$\text{na qual } \bar{a} = e^{j120^\circ} = -\frac{1}{2} + j\frac{\sqrt{3}}{2}, \quad \bar{a}^2 = e^{j240^\circ} = -\frac{1}{2} - j\frac{\sqrt{3}}{2}.$$

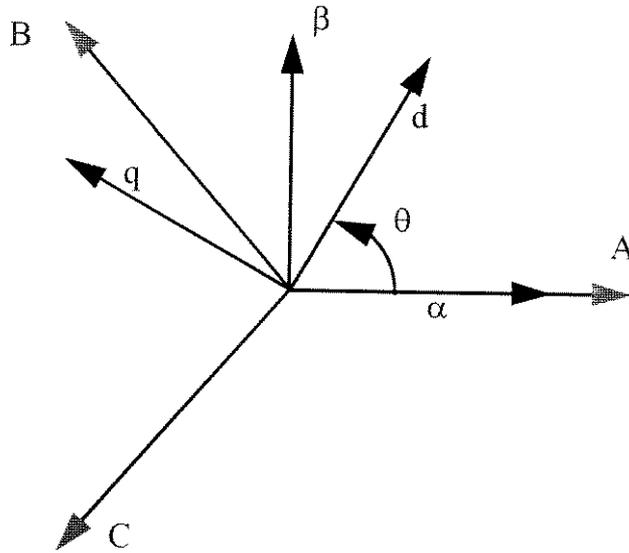


Figura 2.4- Sistema de eixos ABC,  $\alpha\beta$  e dq.

Pode-se, então, escrever após algumas manipulações que:

$$f_{\alpha} = \frac{2}{3} \left( f_{an} - \frac{1}{2} f_{bn} - \frac{1}{2} f_{cn} \right) \quad (2.33)$$

$$f_{\beta} = \frac{1}{\sqrt{3}} (f_{bn} - f_{cn}) \quad (2.34)$$

Se  $f_{an} + f_{bn} + f_{cn} = 0$ , estas relações se resumem à:

$$f_{\alpha} = f_{an} \quad (2.35)$$

$$f_{\beta} = \frac{1}{\sqrt{3}} (f_{bn} - f_{cn}) \quad (2.36)$$

- transformação  $\alpha\beta$  para abc:

$$f_{an} = f_{\alpha} + \frac{1}{\sqrt{2}} f_0 \quad (2.37)$$

$$f_{bn} = \frac{1}{2} (\sqrt{3} f_{\beta} - f_{\alpha}) + \frac{1}{\sqrt{2}} f_0 \quad (2.38)$$

$$f_{cn} = -\frac{1}{2} (f_{\alpha} + \sqrt{3} f_{\beta}) + \frac{1}{\sqrt{2}} f_0 \quad (2.39)$$

*- transformação abc para dq:*

$$f_d = \frac{2}{3} [f_{an} \cos\theta + f_{bn} \cos(\theta - 2\pi/3) + f_{cn} \cos(\theta + 2\pi/3)] \quad (2.40)$$

$$f_q = \frac{2}{3} [f_{an} (-\sin\theta) + f_{bn} (-\sin(\theta - 2\pi/3)) + f_{cn} (-\sin(\theta + 2\pi/3))] \quad (2.41)$$

$$f_0 = \frac{2}{\sqrt{3}} [f_{an} \frac{1}{\sqrt{2}} + f_{bn} \frac{1}{\sqrt{2}} + f_{cn} \frac{1}{\sqrt{2}}] \quad (2.42)$$

*- transformação dq para abc:*

$$f_{an} = f_d \cos\theta + f_q (-\sin\theta) + f_0 \frac{1}{\sqrt{2}} \quad (2.43)$$

$$f_{bn} = f_d \cos(\theta - 2\pi/3) + f_q (-\sin(\theta - 2\pi/3)) + f_0 \frac{1}{\sqrt{2}} \quad (2.44)$$

$$f_{cn} = f_d \cos(\theta + 2\pi/3) + f_q (-\sin(\theta + 2\pi/3)) + f_0 \frac{1}{\sqrt{2}} \quad (2.45)$$

*transformação  $\alpha\beta$  para dq:*

$$f_d = f_\alpha \cos\theta + f_\beta \sin\theta \quad (2.46)$$

$$f_q = -f_\alpha \sin\theta + f_\beta \cos\theta \quad (2.47)$$

*transformação dq para  $\alpha\beta$ :*

$$f_\alpha = f_d \cos\theta - f_q \sin\theta \quad (2.48)$$

$$f_\beta = f_d \sin\theta + f_q \cos\theta \quad (2.49)$$

## **CAPÍTULO 3**

### **ESTRUTURA E DESCRIÇÃO FUNCIONAL DO SISTEMA IMPLEMENTADO**

#### **3.1 INTRODUÇÃO**

A implementação do sistema proposto para o controle vetorial pode ser estruturado por diagramas de blocos, como mostrado na Figura 3.1, sendo que cada diagrama de bloco está associado a um circuito ou algoritmo com uma função específica. Podemos identificar quatro partes principais: Transformação trifásico-bifásico, estimação, sistema de controle e sistema conversor de potência. Nas seções subseqüentes são dadas a descrição funcional de cada uma dessas partes.

#### **3.2 TRANSFORMAÇÃO TRIFÁSICO-BIFÁSICO**

As correntes e tensões medidas no motor são transformadas para o sistema de coordenadas  $\alpha$ - $\beta$ , estacionário em relação ao estator. São tomadas duas correntes de estator  $i_a$  e  $i_b$ , que transformadas resultam nas correntes  $i_\alpha$  e  $i_\beta$ , através das equações dadas no capítulo 2 e que são rescritas, por conveniência, a seguir:

$$i_\alpha = i_a \tag{3.1}$$

$$i_\beta = \frac{1}{\sqrt{3}}(i_a + 2i_b) \tag{3.2}$$

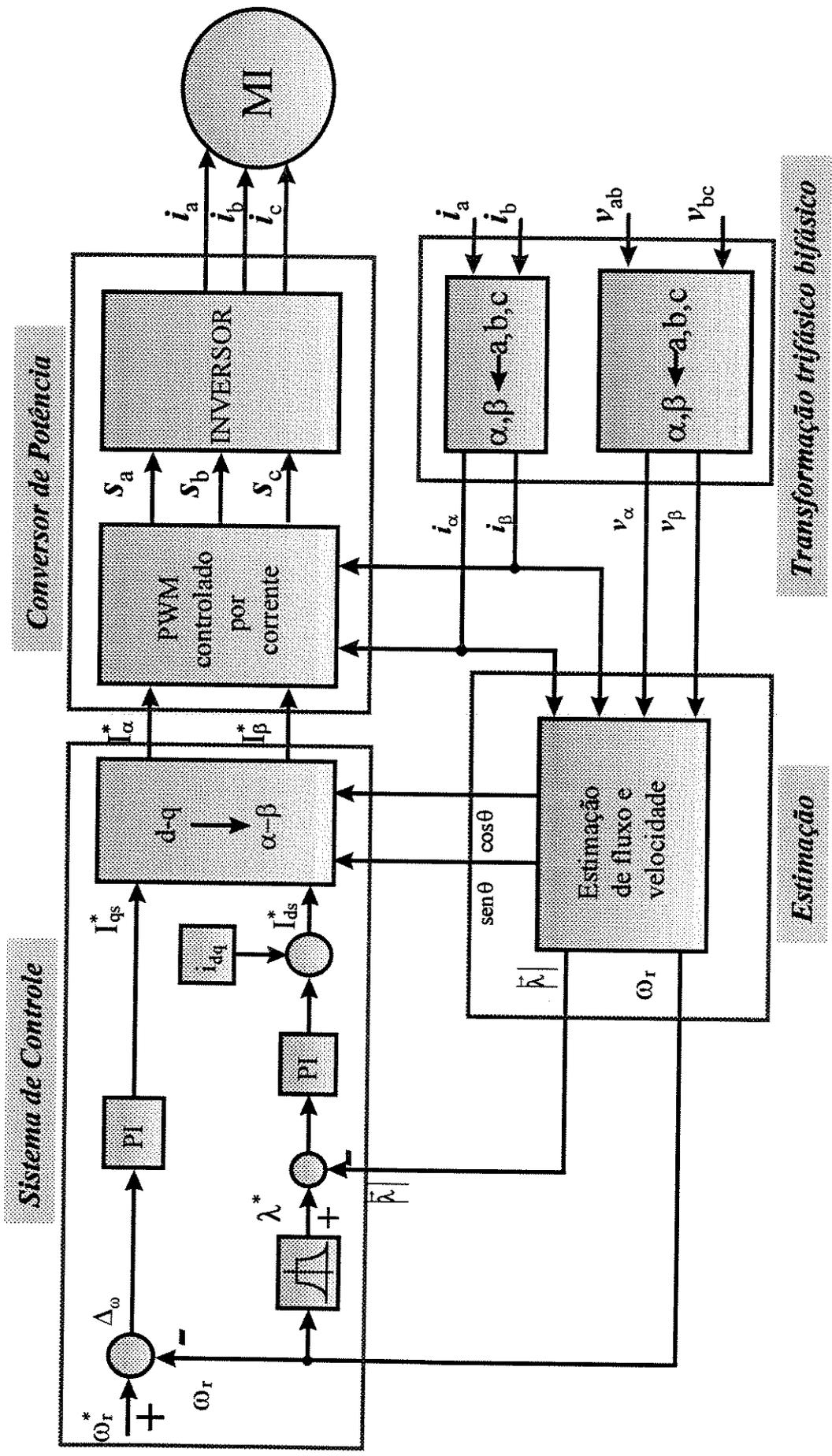


Figura 3.1-Diagrama esquemático do controle de velocidade com orientação de fluxo no estator.

De maneira semelhante ao que foi realizado com as correntes, as tensões de linha medidas no motor  $V_{ab}$  e  $V_{bc}$  são transformadas ao sistema de referência  $\alpha$ - $\beta$ . As equações que governam esta transformação são as seguintes:

$$V_{\alpha} = \frac{1}{3}(2 V_{ab} + V_{bc}) \quad (3.3)$$

$$V_{\beta} = \frac{1}{\sqrt{3}} V_{bc} \quad (3.4)$$

Os valores de corrente e tensões transformados são enviados para o bloco estimador.

### 3.3 ESTIMAÇÃO [19,20,21,22,23,24,25,26]

Este bloco recebe os sinais de tensões e correntes terminais do motor, referenciadas no sistema  $\alpha$ - $\beta$ , e fornece como saída o módulo e o ângulo de fluxo de estator

Assumindo conhecida a resistência do estator, o fluxo do estator é estimado a partir das quantidades terminais, segundo o modelo de tensão do motor. Uma vez determinados o módulo e a posição espacial do fluxo de estator, são implementados os algoritmos de rotação de coordenadas do sistema síncrono d-q para o sistema estacionário  $\alpha$ - $\beta$  e vice-versa.

A frequência elétrica do fluxo do estator pode ser calculada a partir da derivação do ângulo do fluxo em relação ao tempo. Considerando as equações do motor no sistema de referência síncrono d-q, a frequência de escorregamento pode ser obtida a partir das correntes, o módulo de fluxo e os parâmetros do motor. A velocidade mecânica do eixo do motor é calculada a partir da diferença destas duas últimas variáveis.

#### 3.3.1 Equações de estimação de fluxo

A Figura 3.2 mostra o diagrama de blocos do estimador de fluxo de estator. O algoritmo de estimação de fluxo realizado em tempo real fornece as componentes do vetor de fluxo de estator segundo os eixos  $\alpha$  e  $\beta$ , a partir das tensões e correntes de estator dadas no sistema de referência  $\alpha$ - $\beta$ , utilizando as seguintes equações:

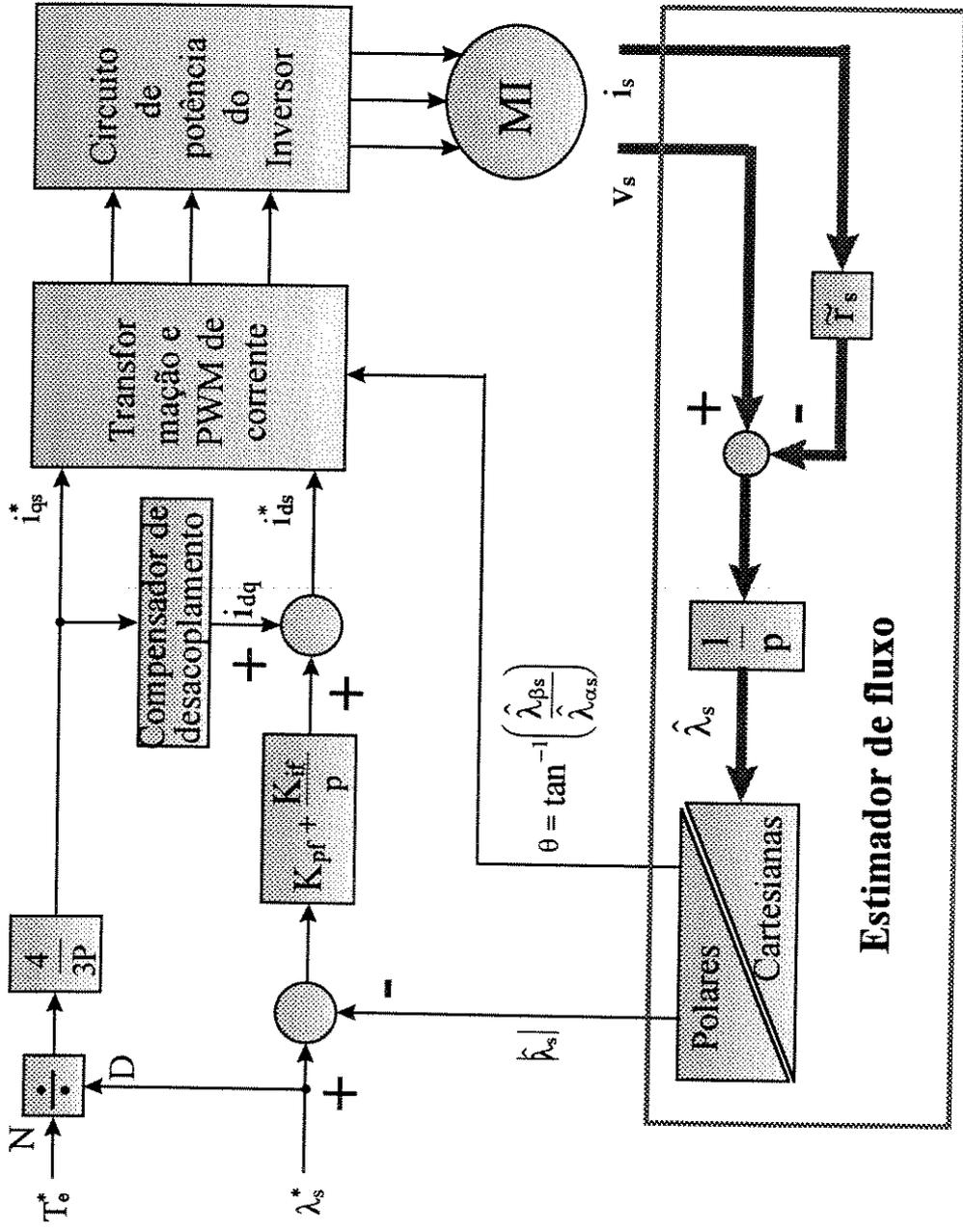


Figura 3.2-Esquema funcional do controle por orientação de fluxo de estator

$$\hat{\lambda}_{\alpha s} = \int (v_{\alpha s} - \hat{r}_s i_{\alpha s}) dt \quad (3.5)$$

$$\hat{\lambda}_{\beta s} = \int (v_{\beta s} - \hat{r}_s i_{\beta s}) dt \quad (3.6)$$

De acordo com as equações 3.5 e 3.6, a estimação do fluxo de estator depende do conhecimento da resistência de estator. Exceto em baixas frequências (alguns Hertz), o efeito da resistência de estator é desprezível comparado com a força contraeletromotriz da máquina. Como as variações da resistência são lentas pois são devidas a mudanças na temperatura dos enrolamentos, é possível empregar algoritmos adaptativos que atualizem o seu valor.

Uma vez determinadas as componentes  $\alpha$  e  $\beta$  do fluxo do estator, são calculados o módulo e os valores do seno e do cosseno do ângulo do fluxo de estator através das seguintes equações, dadas no capítulo 2:

$$|\hat{\lambda}_s| = \sqrt{(\hat{\lambda}_{\alpha s}^2 + \hat{\lambda}_{\beta s}^2)} \quad (3.7)$$

$$\text{sen } \hat{\theta} = \frac{\hat{\lambda}_{\beta s}}{|\hat{\lambda}_s|} \quad (3.8)$$

$$\text{cos } \hat{\theta} = \frac{\hat{\lambda}_{\alpha s}}{|\hat{\lambda}_s|} \quad (3.9)$$

Com os valores do seno e cosseno do ângulo é realizada a transformação das correntes  $\alpha$ - $\beta$  para o sistema síncrono d-q.

### 3.3.2 Estimação de velocidade

Um algoritmo de estimação de velocidade do motor é obtido através do conhecimento da frequência elétrica do sistema  $\omega_e$  e a frequência de escorregamento  $\omega_{\text{slip}}$ . A frequência elétrica é calculada com a seguinte equação:

$$\hat{\omega}_e = \frac{d\hat{\theta}}{dt} = \frac{d}{dt} \left( \tan^{-1} \left( \frac{\hat{\lambda}_{\beta s}}{\hat{\lambda}_{\alpha s}} \right) \right) \quad (3.10)$$

Levando em consideração que  $\hat{\theta} = \tan^{-1} \left( \frac{\hat{\lambda}_{\beta s}}{\hat{\lambda}_{\alpha s}} \right)$ , substituindo esta relação em 3.10 e

realizando a operação de derivação obtemos:

$$\omega_e = \frac{(p\hat{\lambda}_{\beta s})\hat{\lambda}_{\alpha s} - (p\hat{\lambda}_{\alpha s})\hat{\lambda}_{\beta s}}{\hat{\lambda}_{\alpha s}^2 + \hat{\lambda}_{\beta s}^2} \quad (3.11)$$

As expressões referentes à derivada das componentes do fluxo de estator são obtidas das equações 3.5 e 3.6:

$$p\hat{\lambda}_{\alpha s} = v_{\alpha s} - \hat{r}_s i_{\alpha s} \quad (3.12)$$

$$p\hat{\lambda}_{\beta s} = v_{\beta s} - \hat{r}_s i_{\beta s} \quad (3.13)$$

Substituindo as equações 3.12 e 3.13 em 3.11 obtém-se:

$$\omega_e = \frac{(v_{\beta s} - \hat{r}_s i_{\beta s})\hat{\lambda}_{\alpha s} - (v_{\alpha s} - \hat{r}_s i_{\alpha s})\hat{\lambda}_{\beta s}}{|\hat{\lambda}_s|^2} \quad (3.14)$$

A expressão da frequência de escorregamento  $\omega_{slip}$  deduzida no capítulo 2 para o caso do controle vetorial direto é rescrita por conveniência a seguir:

$$\omega_{slip} = \frac{L_s + L_\sigma T_r p}{T_r (\lambda_{ds} - L_\sigma i_{ds})} i_{qs} \quad (3.15)$$

Assim, obtém-se a velocidade do rotor empregando as equações 3.14 e 3.15, como mostrado a seguir:

$$\hat{\omega}_r = \hat{\omega}_e - \hat{\omega}_{slip} \quad (3.16)$$

Na equação 3.14 é suposto que as componentes de fluxo, de tensão e de corrente sejam ondas senoidais, embora as tensões terminais da máquina sejam obtidas através

do chaveamento de uma fonte de corrente contínua. Isto pode levar a resultados errados de frequência estimada. Este inconveniente é resolvido com uma filtragem prévia dos sinais de tensão adquiridos para eliminar as componentes harmônicas de frequências altas que têm influência na estimação da velocidade.

### 3.4 SISTEMA DE CONTROLE

Este bloco recebe informações a respeito do módulo e do ângulo do vetor fluxo de estator como também da velocidade de referência e fornece como saída as referências das correntes no sistema estacionário.

Nesta etapa são gerados os sinais de controle necessários para os reguladores PI e o bloco desacoplador é adicionado ao sinal de comando.

#### 3.4.1 Equações do algoritmo do controlador

As equações envolvidas neste processo são:

- Termo Desacoplador

$$i_{dq} = \frac{\hat{\omega}_{slip} T_r L_\sigma i_{qs}^*}{(L_s + L_\sigma T_r p)} \quad (3.17)$$

- Controlador PI

$$i_{ds}^* = G_f(p)(\lambda_{ds}^* - \lambda_{ds}) + i_{dq} \quad (3.18)$$

$$i_{qs}^* = G_\omega(p)(\omega_r^* - \omega_r) \quad (3.19)$$

nas quais  $G_f(p) = K_{pf} + \frac{K_{if}}{p}$  e  $G_\omega(p) = K_{p\omega} + \frac{K_{i\omega}}{p}$  são as funções de transferência dos reguladores de fluxo e velocidade respectivamente. A equação 3.18 gera o sinal de comando  $i_{ds}^*$  para o inversor.

Como o controle de velocidade do motor é feito através da componente de torque  $i_{qs}^*$ , o algoritmo PI da equação 3.19 deve ser implementado para satisfazer este

eixo direto deve permanecer constante, enquanto que na zona de enfraquecimento de campo, o fluxo deve ser inversamente proporcional à frequência de operação do sistema. Uma vez geradas as correntes de referência  $i_{ds}^*$  e  $i_{qs}^*$ , estas são transformadas ao sistema  $\alpha$ - $\beta$  e impostas como referências para a lógica de controle do inversor.

### 3.4.2 Implementação digital do controlador PI

O controle vetorial direto exige freqüentemente o cálculo em tempo real de integrais e derivadas e, na literatura, tem sido encontrados diversos métodos numéricos para resolver o processo de integração. Levando em consideração os fatores precisão, estabilidade e velocidade de execução, tem sido desenvolvidos os métodos de: Euler de primeira ordem, trapezoidal e o método de segunda ordem de dois passos [13].

Dados os requerimentos de tempo real críticos do sistema de controle e assumindo um passo de integração muito pequeno, foi escolhido o método de Euler de primeira ordem por sua alta velocidade de execução e boa precisão.

Pelo método explícito de Euler de derivada de primeira ordem, de um passo, a integral pode ser calculada como:

$$y_{n+1} = y_n + h x_n \quad (3.20)$$

na qual,  $h$  é o passo de integração.  $x_n$  é o valor do integrando no instante  $n$  e  $y_n$  é o valor da integral no instante  $n$ .

O resultado dado na equação 3.20 dá uma boa aproximação da integral  $y(t)$  para as condições de operação anteriormente exigidas. Outros algoritmos também podem ser empregados, mas uma grande parte deles consomem muito tempo da CPU.

A Figura 3.3 mostra o diagrama de blocos do PI, cuja equação pode ser representada no domínio da frequência por:

$$Y(S) = K_p E + \frac{K_i}{S} E \quad (3.21)$$

na qual  $K_p$  é a constante proporcional,  $K_i$  é a constante integral,  $Y$  é a variável de saída e  $E$  a entrada do controlador.

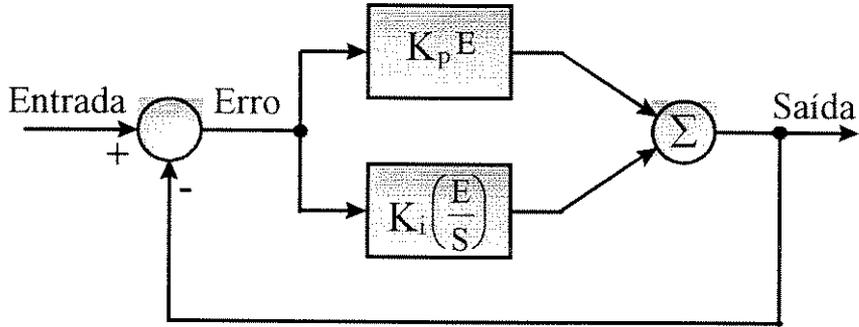


Figura 3.3-Controlador PI

Rearranjando a expressão convenientemente e efetivando a sua discretização obtém-se:

$$Y_{n+1} = Y_n + (K_p + K_i T_a) E_{n+1} - K_p E_n \quad (3.22)$$

na qual  $T_a$  representa o período de amostragem.

Para a solução dessa equação é necessário o conhecimento da entrada  $E$  em dois tempos consecutivos,  $E_n$  e  $E_{n+1}$  e, do estado anterior da saída do PI ( $Y_n$ ). Analisando a equação 3.22 e a Figura 3.3, pode-se concluir que se o erro  $E$  aumenta constantemente então a saída  $Y$  também aumentará, fazendo com que o erro diminua até chegar a zero, em regime permanente. Um processo similar é realizado no caso do erro for negativo por vários ciclos de amostragem.

A Figura 3.4 mostra o fluxograma do PI usado na implementação do sistema de controle vetorial, o operador de atraso unitário  $z^{-1}$  faz com que seja armazenado na memória o estado anterior de ambas as variáveis  $E$  e  $Y$ .

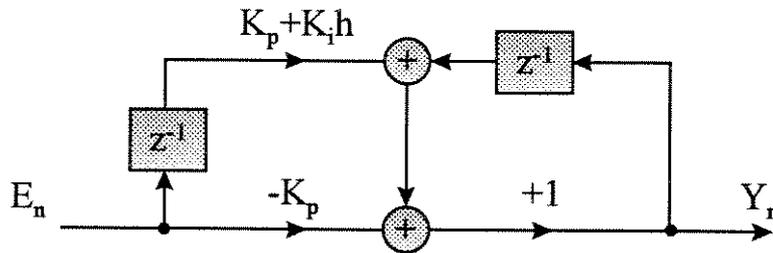


Figura 3.4-Fluxograma do PI

### 3.5 SISTEMA CONVERSOR DE POTÊNCIA

O motor de indução com rotor bobinado é alimentado a partir de um inversor trifásico que utiliza IGBTs, com os pulsos de disparo do inversor gerados por uma lógica de PWM vetorial baseada em realimentação de corrente. Um algoritmo é encarregado de fornecer as correntes de referência para o inversor.

#### 3.5.1 Modulador vetorial PWM realimentado em corrente

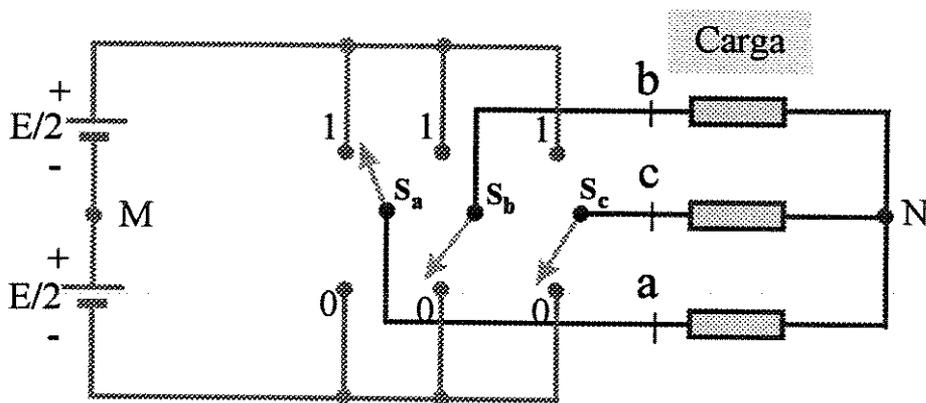
Os moduladores vetoriais PWM de tensão são muito usados em acionamentos de potências médias e baixas devido ao fato de apresentarem bom desempenho, operarem com baixa frequência de chaveamento e poderem ser otimizados para diminuir harmônicos. Embora tenham esses atributos, apresentam uma razoável complexidade na implementação, consomem muito tempo de processamento da CPU e no caso de implementação por "hardware" os circuitos de controle são de grande complexidade.

O inversor de laço de histerese é frequentemente adotado para controles de motor de indução, devido às vantagens de simples implementação e desempenho dinâmico excelente. Contudo, para eliminar os harmônicos das correntes, tanto quanto seja possível, a faixa de histerese deve ser a mais estreita possível e isto resulta em altas frequências de chaveamento, com o conseqüente aumento das perdas do inversor. Impossibilidade de se obter vetores de tensão zero, o fato do processo de modulação gerar componentes subharmônicas são outras desvantagens que essa técnica apresenta. Os inversores preditivos de corrente resolvem em grande parte estes problemas mas, com o aumento razoável de complexidade computacional.

O esquema proposto e implementado no trabalho de Ching-Tsai e Ting-Yu[14] foi escolhido por ser de fácil implementação, rápida resposta em corrente e baixa frequência de chaveamento; além disso, está efetivamente integrado com o algoritmo de controle vetorial. Este inversor adota o mesmo conceito dos inversores preditivos mas elimina os complicados cálculos. Nos próximos itens faz-se uma descrição dos conceitos e critérios usados na implementação do controle do inversor.

### 3.5.2 Representação da tensão de saída do inversor

O diagrama do inversor é mostrado na Figura 3.5, onde  $S_a$ ,  $S_b$  e  $S_c$  são as chaves das fases a, b e c, respectivamente.



**Figura 3.5-Circuito do Inversor**

Separando a fonte de tensão DC de valor  $E$  volts em duas fontes imaginárias de valor  $E/2$ , então o ponto médio  $M$  pode ser usado como uma terra virtual, sendo  $N$  o neutro da carga. Os estados 1 e 0 são associados às chaves  $S_a$ ,  $S_b$  e  $S_c$ : o estado 1 refere-se à situação em que o terminal positivo da fonte está conectado à fase em questão, sendo o oposto para o estado 0. Supondo que os pontos  $M$  e  $N$  não estejam conectados, as tensões das fases em relação a  $M$  são escritas como:

$$V_{aM} = \frac{E}{2} s_a^* = V_{aN} + V_{NM} \quad (3.23)$$

$$V_{bM} = \frac{E}{2} s_b^* = V_{bN} + V_{NM} \quad (3.24)$$

$$V_{cM} = \frac{E}{2} s_c^* = V_{cN} + V_{NM} \quad (3.25)$$

Somando estas três equações obtém-se a tensão do neutro da máquina em relação a M:

$$V_{NM} = \frac{E}{6} (s_a^* + s_b^* + s_c^*) \quad (3.26)$$

As tensões de fase do motor são determinadas substituindo (3.26) em (3.23)-(3.25):

$$V_{aN} = \frac{E}{6} (2s_a^* - s_b^* - s_c^*) \quad (3.27)$$

$$V_{bN} = \frac{E}{6} (2s_b^* - s_a^* - s_c^*) \quad (3.28)$$

$$V_{cN} = \frac{E}{6} (2s_c^* - s_a^* - s_b^*) \quad (3.29)$$

Se  $s_i^*$  for definido como um operador associado ao estado da chave  $S_i$  ( $i = a, b$  ou  $c$ ), pode-se escrever que:

$$s_i^* = \begin{cases} 1 & \text{se } s_i = 1 \\ -1 & \text{se } s_i = 0 \end{cases} \quad (3.30).$$

Esta expressão (3.30) pode ser escrita de uma maneira mais simples como:

$$s_i^* = 2s_i - 1; \quad i = a, b \text{ ou } c \quad (3.31).$$

Levando em consideração (3.31) em (3.27)-(3.29) obtém-se:

$$V_{aN} = \frac{E}{3} (2s_a - s_b - s_c) \quad (3.32)$$

$$V_{bN} = \frac{E}{3}(2s_b - s_a - s_c) \quad (3.33)$$

$$V_{cN} = \frac{E}{3}(2s_c - s_a - s_b) \quad (3.34)$$

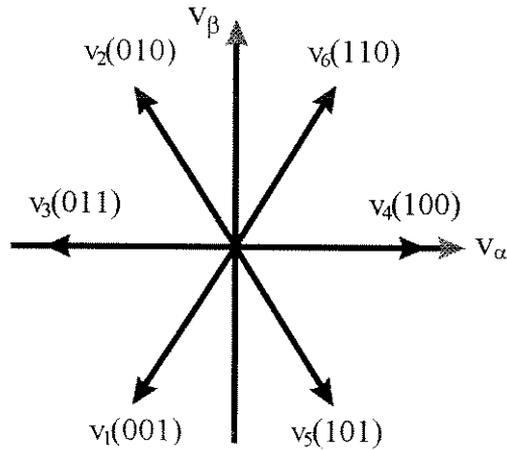
Os estados das chaves produzem 8 combinações possíveis e com a condição de onda senoidal nos sinais  $V_{an}$ ,  $V_{bn}$  e  $V_{cn}$  obtém-se os vetores espaciais de tensão como mostrados na tabela 3.1 e a Figura 3.6.

**Tabela 3.1-Valores instantâneos dos vetores na carga**

$S_a \ S_b \ S_c$	$\bar{V} = V_p e^{j(\theta)}$
0 0 0	$\bar{V}_0 = 0$
0 0 1	$\bar{V}_1 = (2E / 3)e^{j(240^\circ)}$
0 1 0	$\bar{V}_2 = (2E / 3)e^{j(120^\circ)}$
0 1 1	$\bar{V}_3 = (2E / 3)e^{j(180^\circ)}$
1 0 0	$\bar{V}_4 = (2E / 3)e^{j(0^\circ)}$
1 0 1	$\bar{V}_5 = (2E / 3)e^{j(-60^\circ)}$
1 1 0	$\bar{V}_6 = (2E / 3)e^{j(60^\circ)}$
1 1 1	$\bar{V}_7 = 0$

Pode ser observado que cada um dos oito estados do conjunto  $S_a$ ,  $S_b$  e  $S_c$  é representado por um só vetor, o que significa dizer que o mesmo representa as tensões instantâneas de fase  $V_{an}$ ,  $V_{bn}$  e  $V_{cn}$  em um particular instante. Por exemplo, a projeção

do vetor  $V_6$  nos eixos  $\alpha$  e  $\beta$  que resulta nas componentes  $V_\alpha$  e  $V_\beta$  que se transformadas ao sistema trifásico reproduzem as tensões instantâneas nas três fases.



**Figura 3.6-Vetores espaciais da tabela 3.1**

Então, o comportamento do sistema trifásico pode ser observado no plano  $\alpha$ - $\beta$ , se for representado neste mesmo diagrama as componentes de correntes  $i_\alpha$  e  $i_\beta$ . Deve-se notar que as posições dos vetores no plano  $\alpha$ - $\beta$  só dependem dos estados das chaves; então, para cada estado de  $s_a$ ,  $s_b$  e  $s_c$  existe um único vetor possível. Os vetores  $\bar{v}_0$  e  $\bar{v}_7$  são chamados vetores zero, ou seja, nenhuma tensão externa é aplicada à carga.

### 3.5.3 Estratégia do controle da corrente [29,30,31,32]

O controlador de corrente do inversor é implementado com os vetores espaciais de fluxo, tensão e corrente referenciados ao sistema de coordenadas  $\alpha$ - $\beta$ , o que significa dizer que esses vetores giram na velocidade síncrona definida pela frequência elétrica das tensões instantâneas.

A Figura 3.7 apresenta o diagrama de blocos deste inversor, que possui a estratégia de modulação vetorial controlada por corrente. As referências de entradas do inversor  $i_\alpha^*$  e  $i_\beta^*$  são ondas senoidais defasadas de  $90^\circ$  e a realimentação permite que as

correntes  $i_\alpha$  e  $i_\beta$  tenham a mesma forma de onda que as primeiras, mantendo os erros  $\Delta_\alpha$  e  $\Delta_\beta$  dentro de uma determinada faixa.

Quatro comparadores de histerese recebem os valores  $\Delta_\alpha$  e  $\Delta_\beta$  e geram os sinais D, DX, Q e QX que apontam a uma tabela armazenada na memória que, por sua vez, fornece o vetor apropriado ao circuito de potência do inversor. Os sinais  $i_a$  e  $i_b$  medidos devem ser transformados em  $i_\alpha$  e  $i_\beta$ , para poderem ser realimentados ao sistema de controle.

Este esquema mostrado na Figura 3.7 foi implementado utilizando um DSP da Motorola com baixo consumo de tempo de CPU. Esta tarefa foi executada concorrentemente com o algoritmo de controle vetorial.

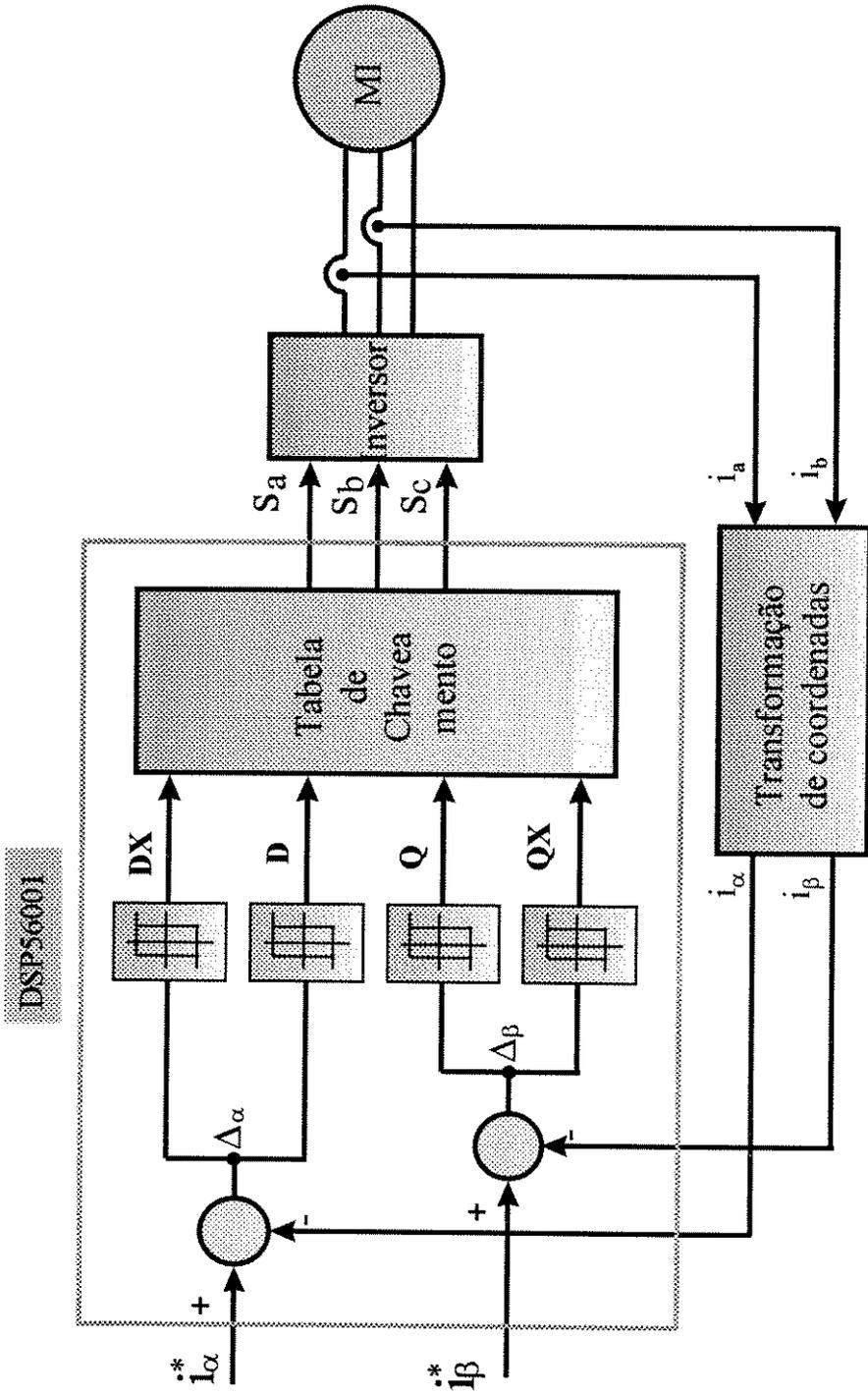
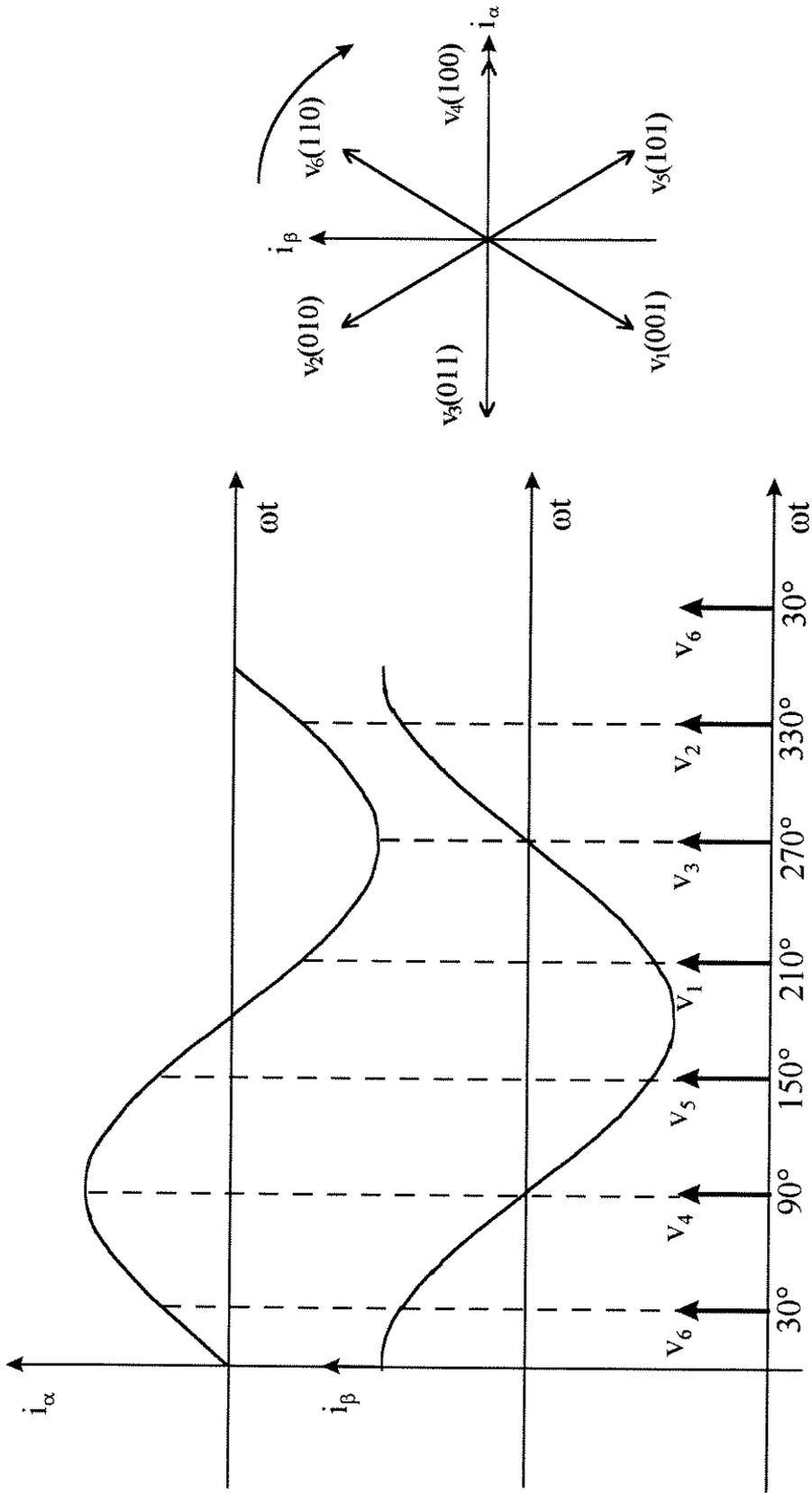


Figura 3.7-Diagrama de blocos do inversor

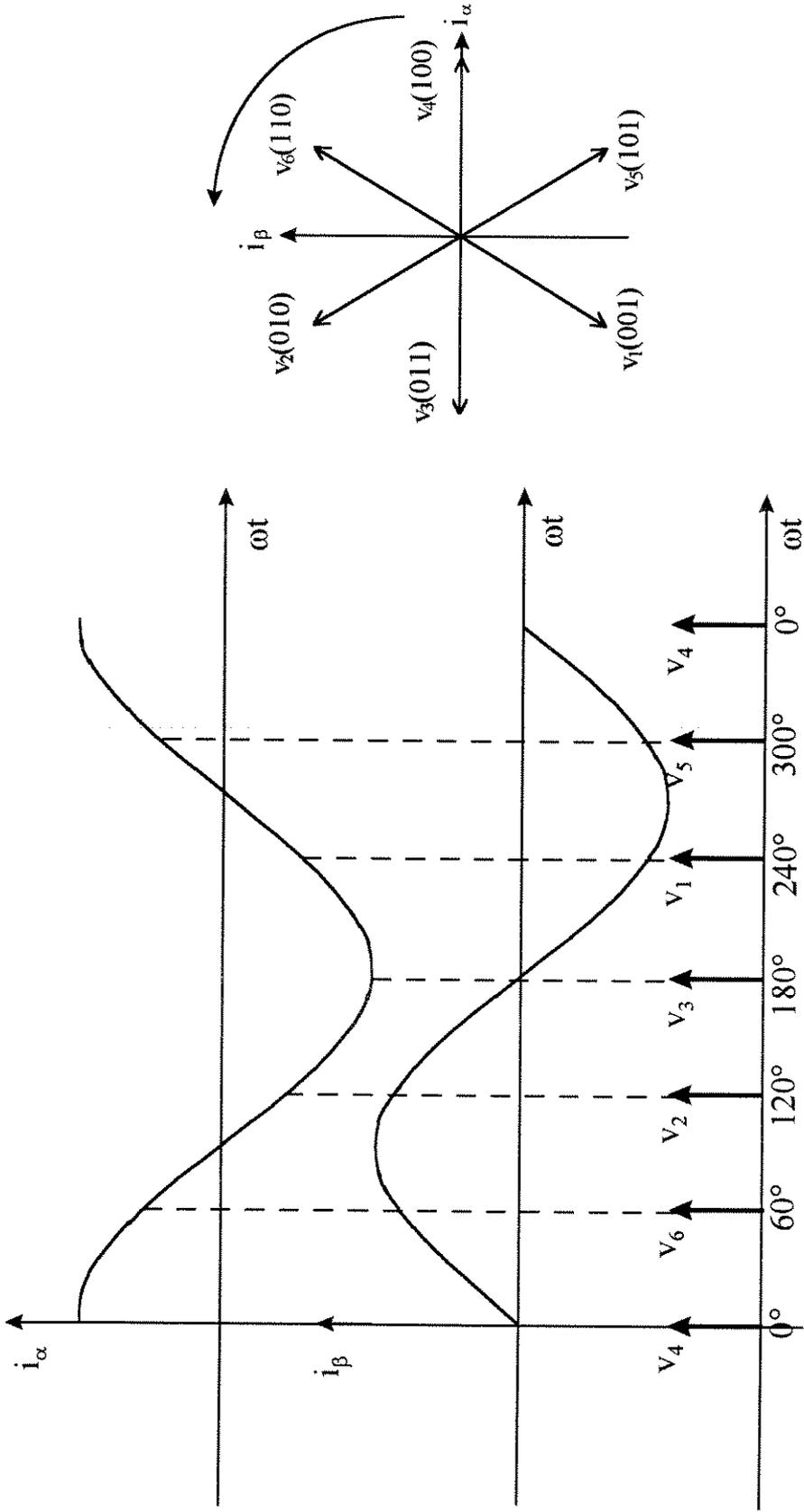
### 3.5.4 Comportamento do vetor espacial de corrente

A trajetória do vetor espacial no plano  $\alpha$ - $\beta$  pode ser controlada mudando as amplitudes, frequências e defasagem dos sinais de referência de entrada  $i_{\alpha}^*$  e  $i_{\beta}^*$ , como mostrado na Figura 3.8, na qual a corrente  $i_{\alpha}$  está atrasada em relação a  $i_{\beta}$ . Nesta situação o vetor gira no sentido horário.

Na Figura 3.9 é ilustrado o caso contrário, ou seja, a corrente  $i_{\alpha}$  adianta-se de  $i_{\beta}$  e como consequência o vetor espacial girará no sentido anti-horário. Esta propriedade do vetor foi testada e confirmada experimentalmente com o inversor alimentando um motor de 2KVA.



**Figura 3.8-Vetor espacial no sentido horário**



**Figura 3.9-Vetor espacial no sentido antihorário**

### 3.5.5 Geração da seqüência de chaveamento

A implementação da lógica de chaveamento é realizada considerando que as correntes de referência podem ser crescentes ou decrescentes conforme mostrado na Figura 3.10.

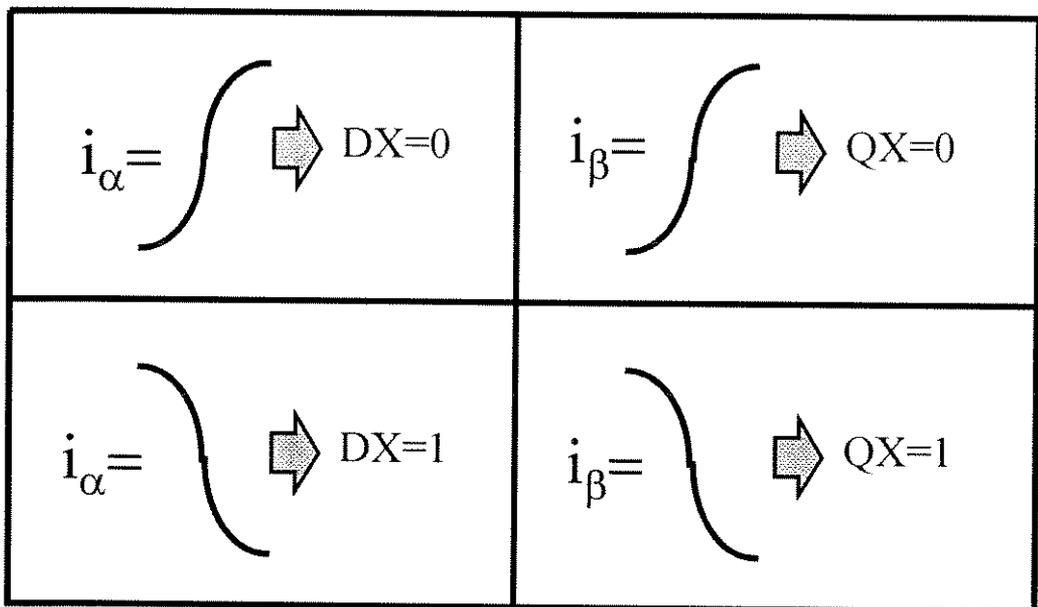


Figura 3.10-Estados de  $i_{\alpha}$  e  $i_{\beta}$

Os estados da Figura 3.10 são representados por valores associados (1 ou 0) à DX e QX. Quando o vetor espacial gira em sentido anti-horário, a seqüência para DXQX será escrita na forma binária 10, 11, 01, 00, 10... que é aproveitada para projetar a rotina de chaveamento no DSP.

Após ter os estados definidos de DX e QX, os valores de D e Q conseguem apontar ao vetor desejado para corrigir as correntes reais. Por exemplo se  $DX=0$  e  $QX=0$ ,  $i_{\alpha}$  e  $i_{\beta}$  estão na zona ascendente. Se assumirmos que D indica que o erro chegou aos limites de histerese de  $\Delta_{\alpha}$  da seguinte maneira:

$$D=0 \Rightarrow \Delta_{\alpha} = \text{limite inferior}$$

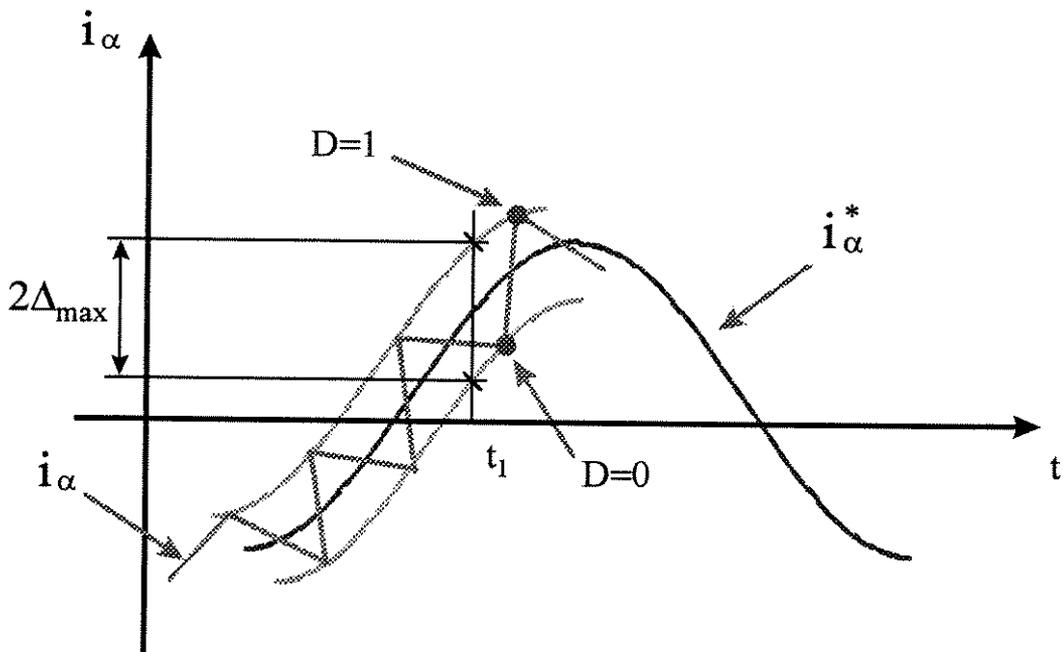
$$D=1 \Rightarrow \Delta_{\alpha} = \text{limite superior}$$

igualmente para Q:

$$Q=0 \Rightarrow \Delta_{\beta} = \text{limite inferior}$$

$$Q=1 \Rightarrow \Delta_{\beta} = \text{limite superior}$$

Se  $\Delta_{\alpha}$  alcança o limite superior, tem que gerar-se o vetor que consiga que  $i_{\alpha}$  seja aproximadamente constante ou  $i_{\alpha}$  diminua até alcançar o limite inferior. Um análise similar é feito no lado do  $\Delta_{\beta}$ . Na Figura 3.11 pode-se apreciar que a corrente  $i_{\alpha}$  varia ao redor da corrente de referência e está limitada pela faixa  $2 \cdot D_{\max}$ .

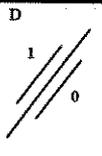
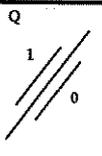
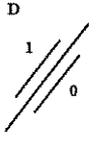
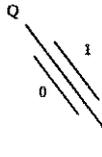
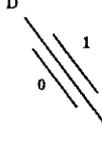
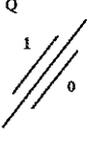
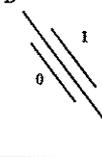
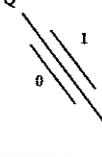


**Figura 3.11-Faixa de variação para  $i_{\alpha}$**

Fazendo a análise para as 4 regiões e os limites  $D=0$  e  $D=1$  obtemos a Tabela 3.2.

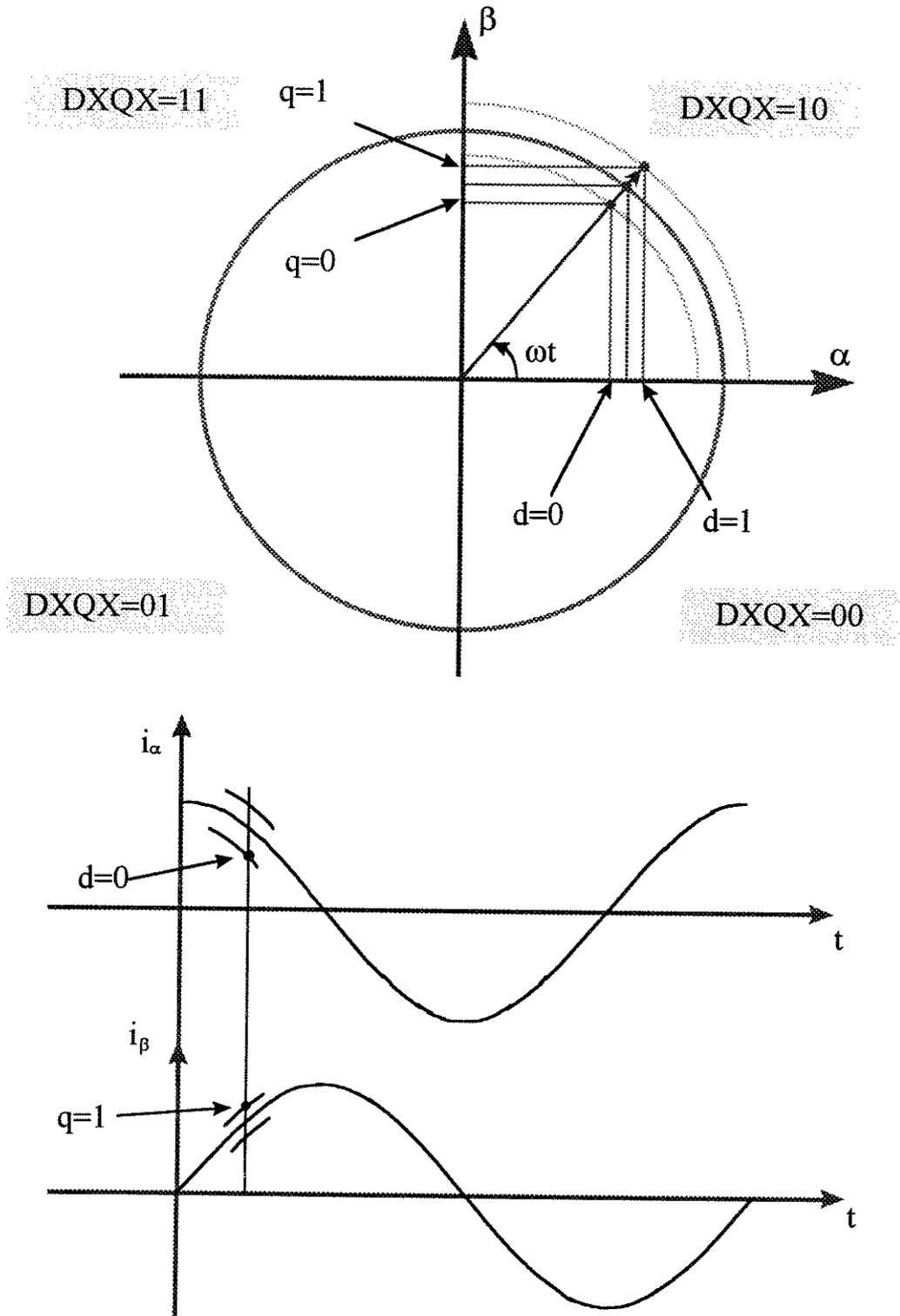
A partir da informação obtida do análise anterior, podemos escrever a Tabela 3.3, que mostra os estados das chaves para cada combinação das variáveis de saída do comparador de histerese. Nas colunas de  $v_\alpha$  e  $v_\beta$  está colocado o sinal (+) no caso que a componente  $\alpha$  ou  $\beta$  tenha que aumentar, o sinal (-) quando a componente do vetor deva diminuir e (0) quando a componente não tenha tensão aplicada. No caso que tem mais de uma solução, escolhe-se o vetor que gere menos chaveamento.

**Tabela 3.2-Regiões de operação da lógica de chaveamento**

<i>Região</i>	<i>DX</i>	<i>QX</i>	$v_\alpha$	$v_\beta$
<b>A</b>	0	0		
<b>B</b>	0	1		
<b>C</b>	1	0		
<b>D</b>	1	1		

**Tabela 3.3- Chaveamento**

<i>Estado</i>	<i>DX</i>	<i>QX</i>	<i>D</i>	<i>Q</i>	<i>S<sub>a</sub></i>	<i>S<sub>b</sub></i>	<i>S<sub>c</sub></i>	<i>V<sub>α</sub></i>	<i>V<sub>β</sub></i>
<b>0</b>	0	0	0	0	1	1	0	+	+
<b>1</b>	0	0	0	1	1	0	0	+	0-
<b>2</b>	0	0	1	0	0	1	0	0-	+
<b>3</b>	0	0	1	1	0	0	0	0-	0-
<b>4</b>	0	1	0	0	1	0	0	+	0-
<b>5</b>	0	1	0	1	1	0	1	+	-
<b>6</b>	0	1	1	0	0	0	0	0-	0+
<b>7</b>	0	1	1	1	0	0	1	0-	-
<b>8</b>	1	0	0	0	1	1	0	0+	+
<b>9</b>	1	0	0	1	0	0	0	0+	0-
<b>10</b>	1	0	1	0	0	1	0	-	+
<b>11</b>	1	0	1	1	0	1	1	-	0-
<b>12</b>	1	1	0	0	0	0	0	0+	0+
<b>13</b>	1	1	0	1	1	0	1	0+	-
<b>14</b>	1	1	1	0	0	1	1	-	0+
<b>15</b>	1	1	1	1	0	0	1	-	-



**Figura 3.12-Geração dos dados da tabela de chaveamento**

Pode-se observar que no estado zero o sistema está na região A da Tabela 3.2 e, como D e Q tem valor zero, então  $v_\alpha$  e  $v_\beta$  devem ser positivos para poder incrementar

negativo ou zero; quando  $D=1$  e  $Q=1$ , os vetores  $V_\alpha$  e  $V_\beta$  podem ser zeros ou negativos, então do ponto de vista de desempenho é conveniente colocar um vetor para o menor chaveamento.

A Figura 3.12 mostra esta mesma lógica em outro ponto de operação. o estado de  $DXQX=10$  estabelece a operação no primeiro quadrante no plano  $\alpha$ - $\beta$ . Para o ponto desenhado  $d=0$  na corrente  $\alpha$  precisa-se de um vetor que aumente ou mantenha  $i_\alpha$  a mesma e para  $q=1$  precisa-se de um vetor que diminua ou mantenha constante a corrente  $i_\beta$ ; então, o vetor apropriado para menor chaveamento é o vetor zero. Logo após estabelecido o vetor correto, os sinais  $S_a$ ,  $S_b$  e  $S_c$  e seus respectivos sinais negados, são aplicados ao inversor para gerar o PWM de tensão.

## *CAPÍTULO 4*

### *DESCRIÇÃO DA IMPLEMENTAÇÃO EXPERIMENTAL UTILIZANDO "DSP"*

#### **4.1 INTRODUÇÃO**

O controle vetorial direto exige uma grande complexidade computacional devido aos algoritmos de estimação do fluxo e de cálculo da posição espacial do mesmo a partir das quantidades terminais da máquina. Atualmente, existem duas alternativas principais de implementação: a) microcontroladores e b) processadores digitais de sinal (DSP).

Os microcontroladores são no fundo processadores com funções adicionais que estão integrados no mesmo "chip". Estas funções são, por exemplo: memória para dados e programas, entradas e saídas digitais, entradas e saídas analógicas, portas de comunicação serial, temporizador/contador, hardware de interrupções, etc. A maioria de microcontroladores atuais utilizam a arquitetura Von-Neuman que consiste principalmente no uso de um barramento para dados e instruções e outro para endereços. Devido a estas características, o desenvolvimento de projetos com microcontroladores tornou-se simples e rápido.

Os DSPs são microprocessadores com uma arquitetura dedicada ao processamento digital de sinais analógicos previamente digitalizados, possuindo instruções otimizadas para o processamento de algoritmos tais como FFT, filtros IIR e FIR, etc. Estas instruções são geralmente realizadas em um ciclo de instrução e, para poder atender este requisito, os DSPs usam 3 principais estratégias: multiplicador em hardware, encadeamento de instruções ("pipeline") e arquitetura Harvard. Nos primeiros DSPs lançados em 1980 o multiplicador em hardware ocupava mais de 20% da área do chip, e atualmente ocupam menos de 3%. Na estrutura "pipeline" a instrução é dividida em várias operações, cada uma realizada em um ciclo de relógio; por exemplo o DSP56001 de Motorola possui um "pipeline" de 3 estágios. Na arquitetura Harvard as instruções e os dados são armazenados em memórias separadas, cada uma com seu barramento próprio. Esta arquitetura permite a sobreposição das operações de busca de instrução com operações de execução de instrução, e as operações de entrada/saída são realizadas através do barramento de memória de dados.

Para definir o processador adequado ao presente projeto as seguintes especificações foram consideradas:

1. Execução concorrente das rotinas de controle, chaveamento do inversor e aquisição de dados de correntes e tensões.
2. Resposta rápida a interrupções.
3. Sistema de desenvolvimento flexível.

Inicialmente foi testado o microcontrolador 87C196KD de 20 MHz obtendo-se uma resposta à interrupções de  $10\mu\text{seg}$ , e com isto a execução das 3 rotinas concorrentes foi realizada lentamente. Além disso o ciclo de instrução de 4 períodos de relógio e a arquitetura interna do chip não permitiam operações de multiplicação e divisão rápidas. O DSP56001 mostrou ter uma resposta à interrupção de 600 nseg, realizando as operações de multiplicação e iteração de divisão em um ciclo de instrução de 100 nseg. Além disso os custos de "hardware" e "software" de suporte para o DSP são comparáveis com os do microcontrolador. Recentemente a Intel colocou no mercado o 80C296 que possui características de DSP, embora esteja na etapa de avaliação.

O DSP56001 da Motorola foi escolhido para a implementação do sistema de controle proposto pois, os testes realizados com ele, satisfizeram as especificações do sistema.

Nas seções seguintes serão abordados o "hardware" e "software" do sistema implementado.

#### 4.2 DESCRIÇÃO DO SISTEMA IMPLEMENTADO.

A Figura 4.1 mostra o diagrama de blocos geral do sistema implementado. O motor de indução é alimentado por um inversor trifásico a IGBTs de 10KVA de potência; os circuitos de comando dos transistores de potência são fornecidos pelos circuitos integrados SKHI 22 da Semikron que gera sinais de comando de +15 e -15 volts. Estes circuitos integrados tem proteção contra curtos-circuitos nos IGBTs e estão isolado dos sinais de comando gerados pelo DSP.

Sensores de efeito Hall são usados para adquirir tensões e correntes do motor. Esses sinais são enviados aos blocos de transformação trifásico-bifásico, cujos circuitos baseados em amplificadores operacionais geram os sinais  $i_\alpha$ ,  $i_\beta$  e  $v_\alpha$ ,  $v_\beta$ . Com as correntes  $i_\alpha$ ,  $i_\beta$  e conhecendo a resistência de estator, o bloco estimador de fluxo calcula o fluxo  $\vec{\lambda}$  estatórico em módulo e ângulo. Derivando este último obtém-se a frequência elétrica  $\hat{\omega}_e$ .

As correntes síncronas  $i_d$  e  $i_q$  são obtidas a partir de  $i_\alpha$ ,  $i_\beta$  e o ângulo de fluxo calculado. Com esta informação e parâmetros adicionais do motor o programa calcula a frequência de escorregamento.

O fluxo de referência  $\lambda^*$  é mantido constante até a frequência nominal do motor, para frequências maiores o campo é diminuído inversamente proporcional à frequência.

O bloco PI de fluxo gera um sinal  $y_{PI}$  que é somado com a saída do bloco desacoplador para gerar a referência  $i_d^*$ . Uma vez geradas as correntes de comando  $i_d^*$

e  $i_q^*$  no sistema síncrono, o DSP através de um algoritmo faz a conversão destas ao sistema estacionário, determinando as referências de corrente  $i_\alpha^*$  e  $i_\beta^*$  do inversor.

Com a informação destas correntes o algoritmo que controla o inversor gera o vetor correspondente ( $S_a, S_b, S_c$ ) no plano  $\alpha$ - $\beta$ . O bloco inversor amplifica estes sinais e alimenta o motor de indução MI.

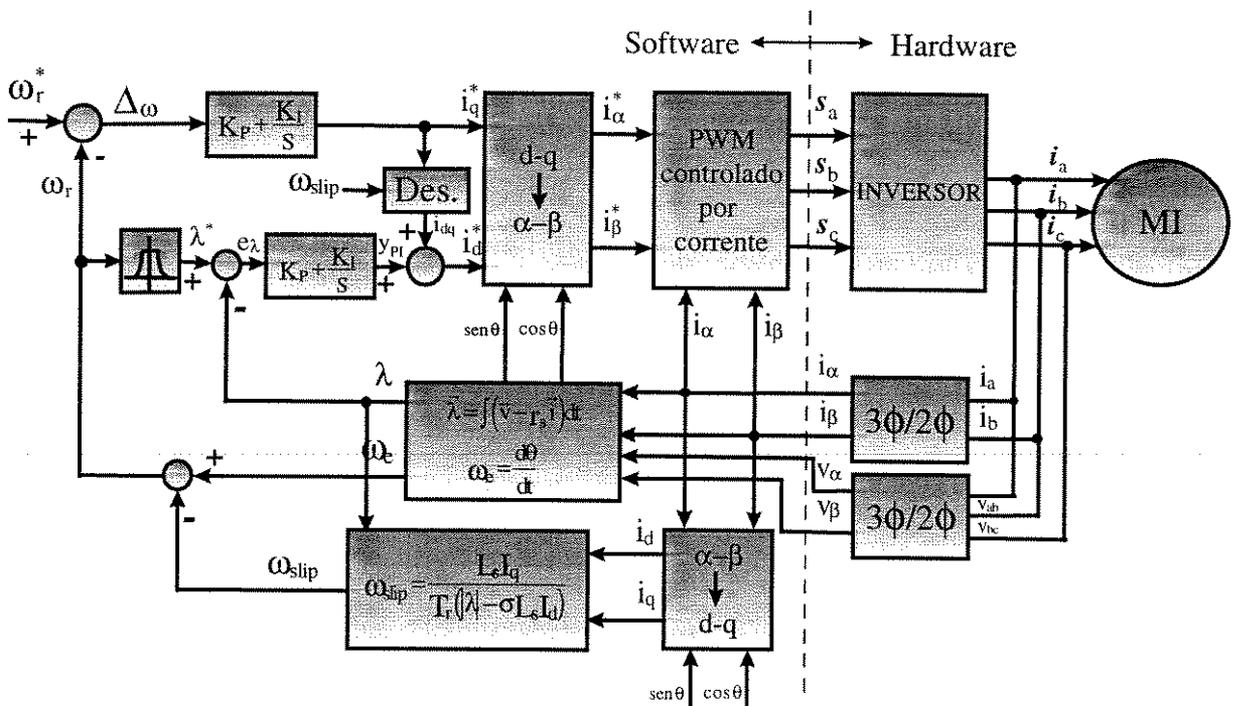


Figura 4.1-Diagrama de blocos do sistema implementado

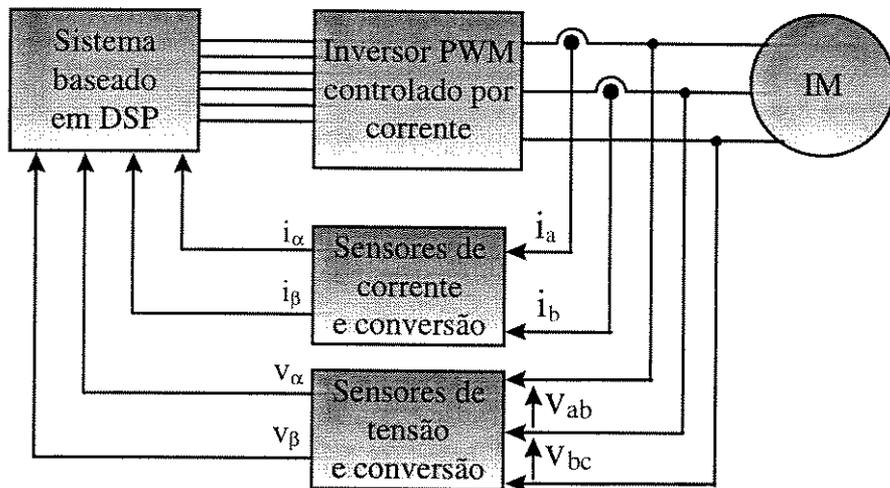
### 4.3 HARDWARE DO SISTEMA

Nos itens seguintes é feita a descrição das partes principais do "hardware" desenvolvido no projeto.

#### 4.3.1 Circuito de aquisição

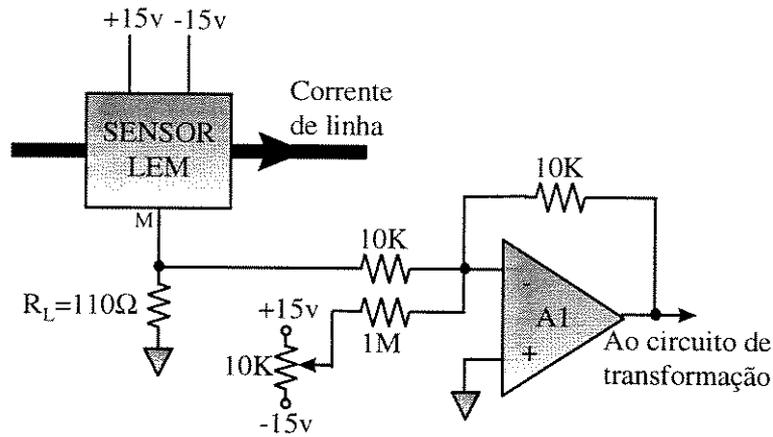
Para adquirir as tensões e correntes do motor foram usados sensores de efeito Hall que apresentaram ter uma boa resposta em frequência; a linearidade dos sensores foi testada com correntes no lado primário de 1 a 50 A, apresentando erros menores que 0,5%.

Os sensores de efeito Hall possuem isolamento elétrico entre o circuito de potência e o sistema de controle. Na Figura 4.2 está mostrado o diagrama de blocos do esquema de aquisição e pode ser observado que são utilizados 4 sensores de efeito Hall para medir  $i_a$ ,  $i_b$ ,  $v_{ab}$  e  $v_{bc}$ .



**Figura 4.2-Componentes do sistema experimental**

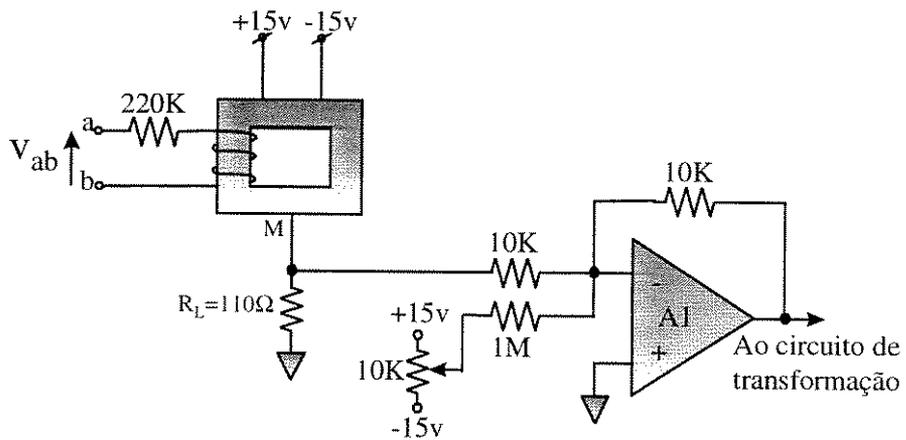
A Figura 4.3 apresenta o circuito usado na aquisição das correntes do motor. No pino M o sensor gera uma corrente que circula pela carga  $R_L$ , que tem um valor adequado de  $110\Omega$  para uma corrente nominal de 9 A. no motor. Sem corrente no lado primário o sensor gera uma tensão DC de aproximadamente 50mV que varia com a carga  $R_L$  colocada. Este erro é compensado pelo amplificador operacional A1 e, seu circuito associado.



**Figura 4.3-Circuito de aquisição de corrente**

Para adquirir tensão foram usados sensores de corrente com uma bobina especial de 500 espiras, para produzir uma tensão na resistência  $R_L$  de não máximo 5 volts.

Também neste caso é usado o amplificador A1 para eliminar o OFFSET DC do sensor. Este esquema é mostrado na Figura 4.4.

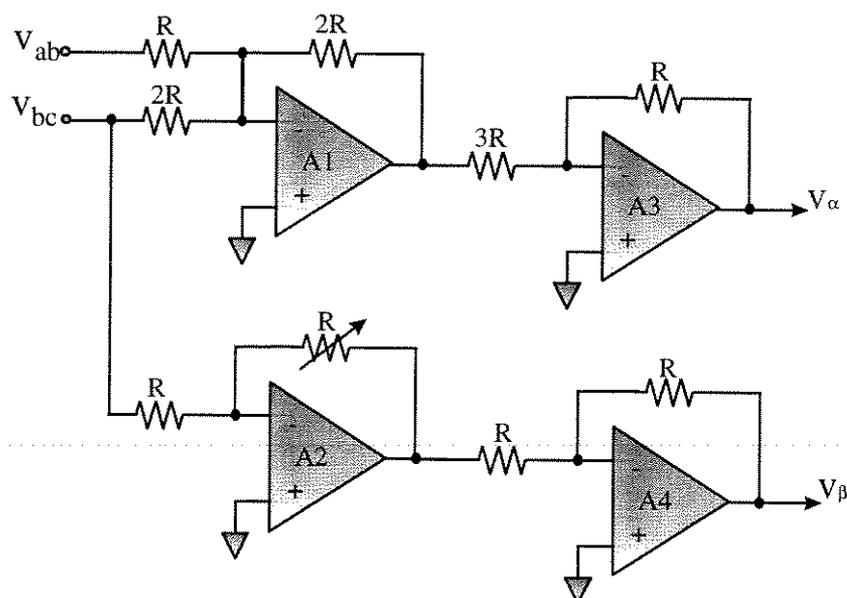


**Figura 4.4-Circuito de aquisição de tensão**

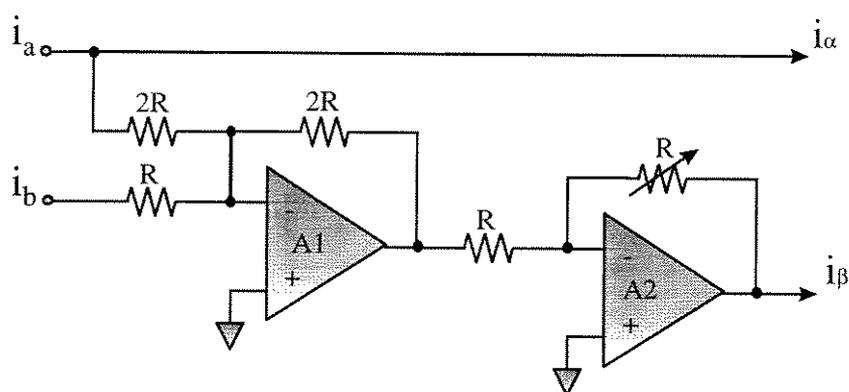
### 4.3.2 Circuito de geração de correntes e tensões no sistema de referência $\alpha$ - $\beta$

Para realizar a transformação trifásica-bifásica foram projetados circuitos com operacionais TL084. Os amplificadores TL084 foram escolhidos por possuírem entrada FET e um "slew rate" de 13 volts/ $\mu$ seg.

As equações que governam as transformações de tensões e correntes são aquelas dadas no capítulo 2. Nas Figuras 4.5a e 4.5b são mostrados os circuitos de tensão e de corrente, respectivamente.



(a)



(b)

**Figura 4.5-Circuitos analógicos utilizados na transformação (a) de tensões e (b) de correntes**

### 4.3.3 Interface de conversão A/D e D/A

A Figura 4.6 mostra o diagrama de blocos da interface de aquisição de dados. O AD7828 de Analog Devices foi escolhido por ter as seguintes características: rapidez de conversão 2,33  $\mu$ seg, "bus" de dados de 8 bits, multiplexador de 8 canais analógicos e fácil interconexão com qualquer microprocessador.

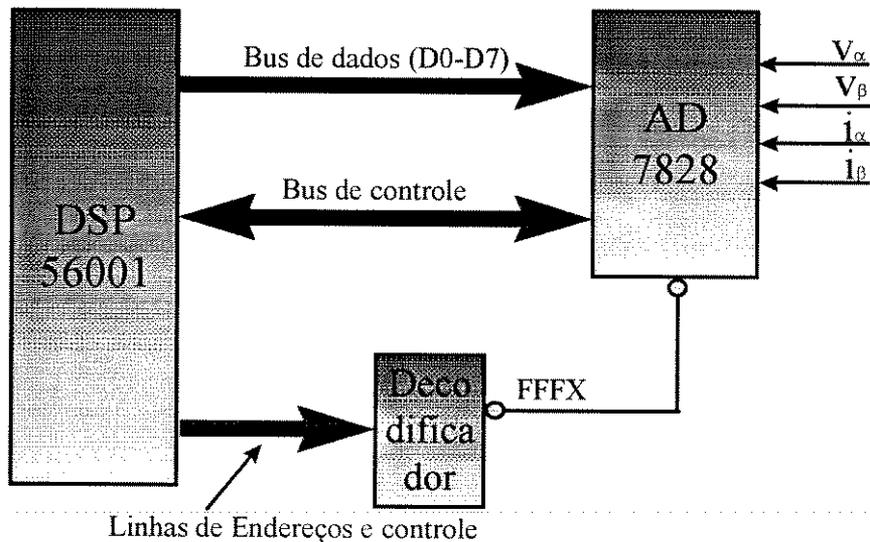


Figura 4.6-Circuito de interface do A/D

Os endereços decodificados para o A/D são mostrados na Tabela 4.1.

Tabela 4.1-ENDEREÇOS DO A/D

Endereço de memória Y	Canal analógico
FFF0h	1
FFF1h	2
FFF2h	3
FFF3h	4

O processo de aquisição de correntes e tensões é iniciado no final da rotina de chaveamento. A rotina do A/D lê seqüencialmente o dado correspondente a cada canal até completar as aquisições dos 4 canais.

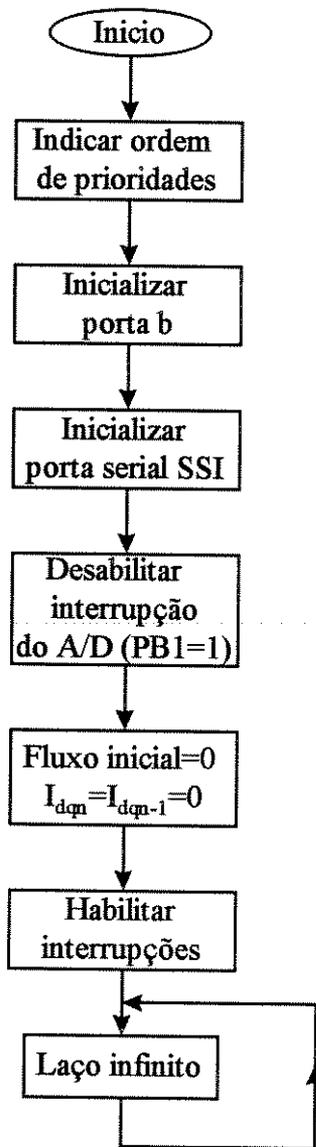
#### **4.4 SOFTWARE DO SISTEMA**

O programa do sistema foi implementado em linguagem "assembler" e ocupa 1/2K de endereços de memória de programa. As variáveis do sistema foram armazenadas nas memórias de dados X e ocupam 76 endereços, as constantes foram armazenadas nos primeiros 30 endereços de memória Y. Também foram reservadas 9 endereços de memória de 48 bits para serem usados como fileira do sistema. Nos próximos itens, será feita uma descrição das rotinas empregadas na implementação.

##### **4.4.1 Programa de inicialização**

Está encarregado de estabelecer a inicialização de todas as variáveis e parâmetros necessários para as rotinas do sistema. Este programa estabelece uma ordem de prioridades das rotinas de interrupção, inicializa as portas e periféricos usados no controle e desabilita a interrupção do A/D, conforme mostrado na Figura 4.7.

Após desabilitar a interrupção do A/D, o fluxo e a variável de saída  $I_{dq}$  do desacoplador são inicializados com valores zero. Em seguida são habilitados por "software" as interrupções IRQA, IRQB e TIMER SCI. Após isto, o programa entra em um laço infinito de espera às interrupções habilitadas.



**Figura 4.7-Programa de inicialização**

#### 4.4.2 Temporização do sistema

A Figura 4.8 mostra o diagrama de temporização do sistema. Como se observa após a inicialização, o sistema executa em tempo real 4 rotinas concorrentes conforme mostradas na Tabela 4.2.

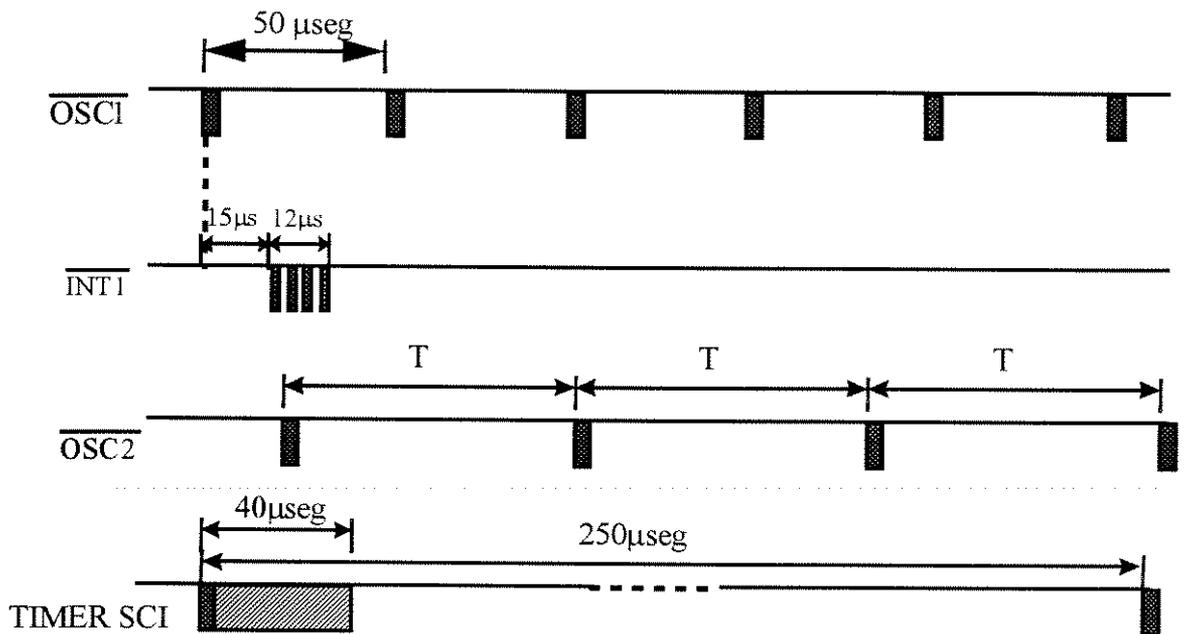


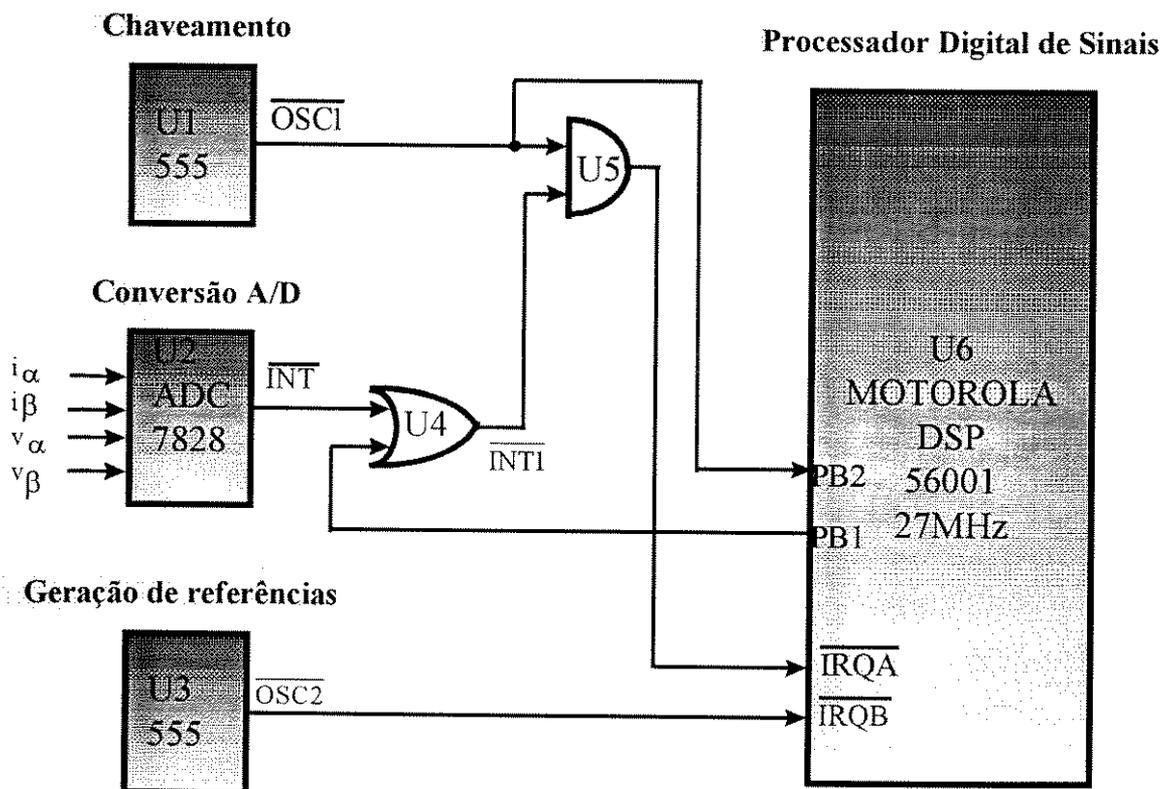
Figura 4.8-Temporização do sistema

O programa de inicialização configura as interrupções com a seguinte ordem de prioridades:  $\overline{\text{IRQA}} > \overline{\text{IRQB}} > \text{TIMER - SCI}$ . Por isto a primeira rotina ao ser executada é à referente ao chaveamento ( $\overline{\text{IRQA}}$ ), conforme mostrado na Figura 4.9.

**Tabela 4.1 ROTINAS DO NÚCLEO DE TEMPO REAL**

Rotina	Interrupção no DSP	Ordem de prioridades	Função
$\overline{\text{OSCI}}$	$\overline{\text{IRQA}}$	3(mais alta)	Chaveamento e estimação de fluxo
$\overline{\text{INT}}$	$\overline{\text{IRQA}}$	2	Aquisição de correntes e tensões
$\overline{\text{OSC2}}$	$\overline{\text{IRQB}}$	1	Geração de referências do inversor
TIMER SCI	Timer SCI	0(mais baixa)	Rotina de controle vetorial

A temporização começa com a interrupção gerada por  $\overline{\text{OSCI}}$ , cuja frequência é controlada pelo temporizador U1 fixado em 20Khz (50 $\mu$ seg ). Ao final da rotina de chaveamento, o conversor A/D realiza 4 conversões analógico digitais e os dados  $i_\alpha$ ,  $i_\beta$ ,  $v_\alpha$ ,  $v_\beta$  são colocados em memória. Os sinais  $\overline{\text{OSCI}}$  e  $\overline{\text{INTI}}$  estão sempre sincronizados. A frequência do sinal  $\overline{\text{OSC2}}$  é controlada pelo temporizador U3.



**Figura 4. 9-Hardware de interrupções**

O temporizador SCI do DSP56001 foi programado para interromper a cada 250  $\mu$  seg (4KHz) e opera de maneira assíncrona com a rotina associada à  $\overline{\text{OSC2}}$ .

#### 4.4.3 Lógica de chaveamento do inversor

Como foi descrito na seção 3.5, o inversor implementado é do tipo PWM controlado por corrente. O objetivo deste inversor é manter os sinais erro $_{\alpha}$  e erro $_{\beta}$  dentro das faixas limites M e m ( $M > m$ ), como mostrado nas Figuras 3.11 e 3.12. Estes valores são armazenados na área de memória RAM de dados e podem ser modificados livremente para minimizar o conteúdo de harmônicos.

A Figura 4.10 mostra a lógica de chaveamento do laço  $\alpha$  do inversor, sendo que a mesma estrutura é utilizada para o laço  $\beta$ .

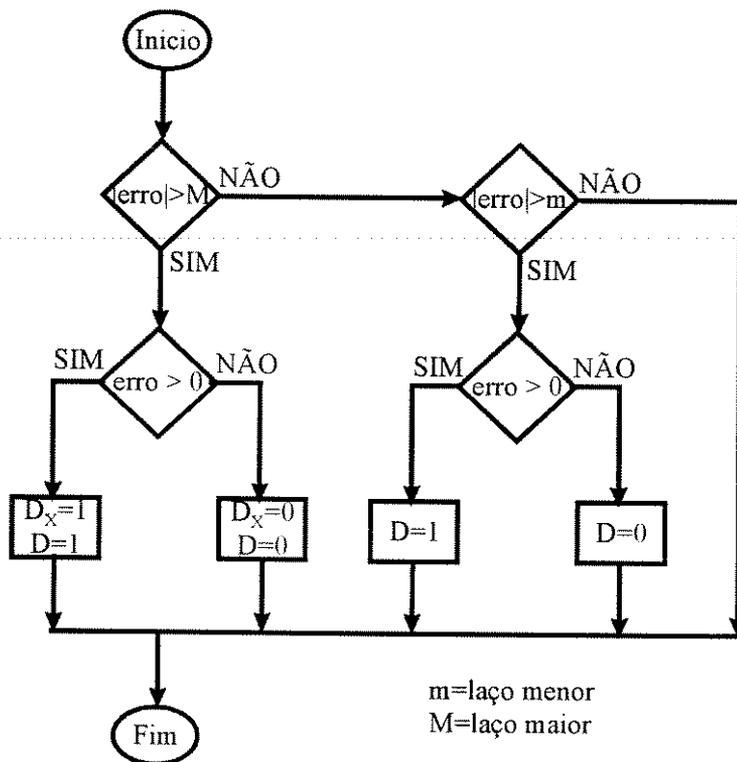
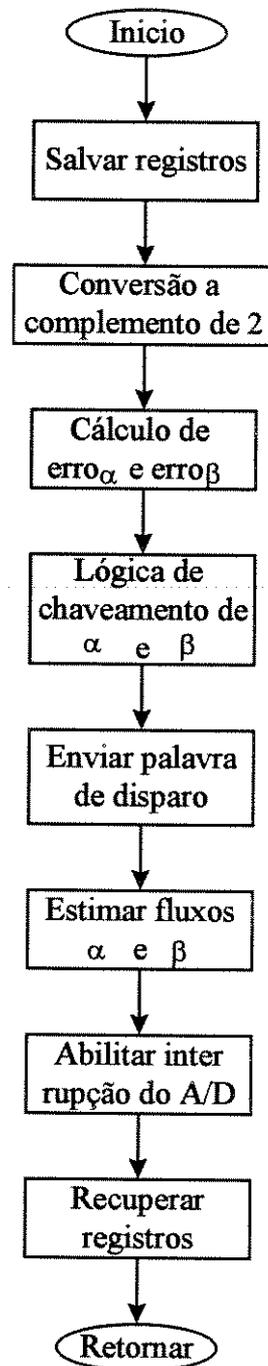


Figura 4.10-Lógica de chaveamento do laço  $\alpha$  do inversor

O objetivo destas rotinas de chaveamento é calcular as variáveis D, DX, Q e QX como definidas no capítulo 3. Uma palavra de 4 bits é usada como apontador para a uma tabela em memória RAM. Os dados da tabela fornecem o vetor de corrente

apropriado para o inversor. Para implementar esta lógica é usada a instrução de comparação com valor absoluto que faz parte do conjunto de instruções do DSP56001.

A Figura 4.11 mostra o fluxograma da rotina de chaveamento. Como foi mencionado anteriormente, este programa é realizado a cada 50  $\mu$ seg e tem a mais alta prioridade. Por esta razão a integral de estimação de fluxo é executada nesta rotina.



**Figura 4.11-Rotina de chaveamento e estimação de fluxo**

O programa começa salvando registros em memória para, na seqüência, converter a complemento de 2 os dados de corrente e tensão. Em seguida, são calculados os erros entre as correntes de referência e as correntes reais e com essa informação é executada a lógica de chaveamento. De posse do vetor apropriado de corrente do inversor, este é enviado aos circuitos de comando dos IGBTs, para depois estimar o fluxo. Uma vez realizada esta estimação, é habilitada a interrupção do A/D para a aquisição de dados. O algoritmo termina recuperando os registros inicialmente salvados.

#### **4.4.4 Rotina de aquisição de dados**

A Figura 4.12 mostra o fluxograma de aquisição de correntes e tensões. A conversão analógica digital do primeiro canal (tensão  $v_\alpha$ ) é iniciada no final da rotina de chaveamento, na seguinte execução do algoritmo é lido o segundo canal ( $i_\alpha$ ), isto continua até a leitura do quarto canal, momento no qual, o conversor A/D é desabilitado. As 4 aquisições levam aproximadamente 12 $\mu$ seg.

#### **4.4.5 Geração de correntes de referência**

Antes de carregar o programa do sistema em memória RAM, tabelas de senos e cossenos são armazenadas nas memórias de dados X e Y. Estas tabelas consistem de 360 pontos na faixa de 0 a 360<sup>o</sup>, sendo que o cosseno é armazenado a partir do endereço de memória X:0100h e o seno a partir de Y:0100h. A leitura dos dois dados é realizada numa só instrução, aproveitando o paralelismo do DSP. Estes eventos são sincronizados com a interrupção gerada por OSC2 (IRQB). A seqüência de operações realizadas para a geração das correntes de referência é mostrada na Figura 4.13

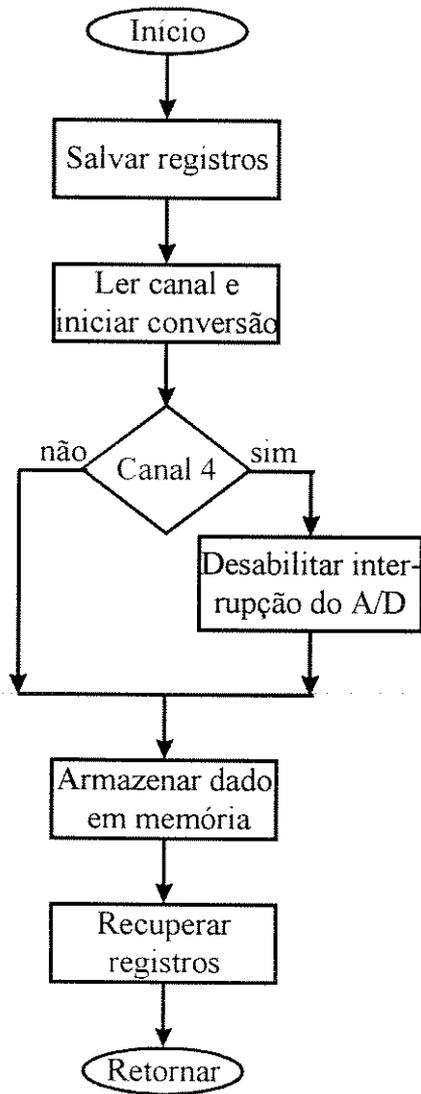
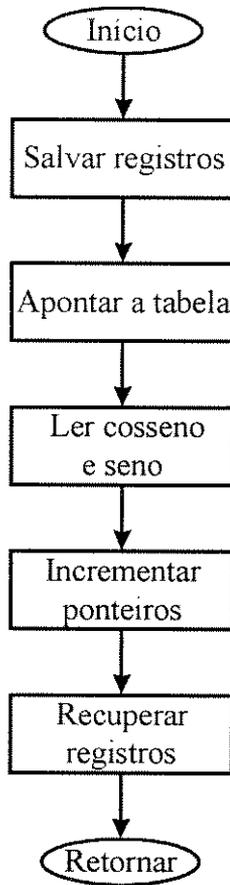


Figura 4.12-Fluxograma de aquisição de dados



**Figura 4.13-Rotina de leitura de correntes de referência do inversor**

#### 4.4.6 Software de estimação de fluxo

Com o valor de  $r_s$  obtido no início do processo, a equação 4.1 mostra o cálculo necessário para estimar a componente  $\alpha$  do fluxo, sendo que um processo similar é realizado para a componente  $\beta$ :

$$\lambda_\alpha = \int (v_\alpha - r_s i_\alpha) dt \quad (4.1)$$

O integrando é conhecido como força contra-eletromotriz, ou seja:

$$fem_\alpha = v_\alpha - r_s i_\alpha \quad (4.2)$$

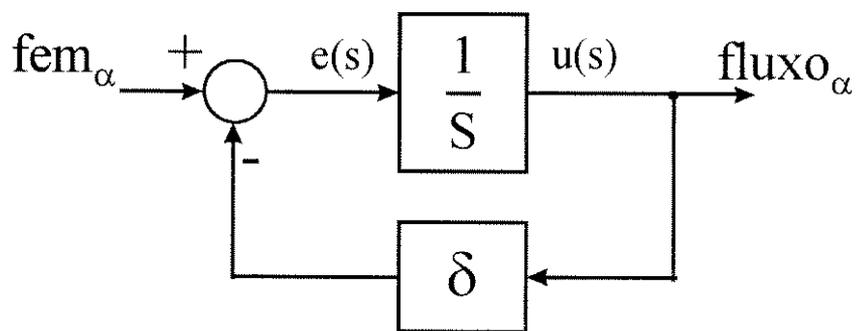
Do ponto de vista de implementação o fluxo não pode ser calculado por uma integração simples, já que a componente DC do fluxo é amplificada com ganho infinito. Devido a este fato, a saída da integral tende a aumentar positiva ou

negativamente até exceder o número de bits suportados pelo processador, este fenômeno é conhecido como efeito de saturação da integral.

Para resolver este problema Xu [12] implementou uma versão modificada da integral, tomando a integral como sendo a função de transferência:

$$F(j\omega) = \frac{1}{j\omega + \delta} \quad (4.3)$$

Se o termo  $\delta$  for pequeno comparado com a frequência de operação, ou seja,  $|j\omega| \gg \delta$ , a função de transferência dada por 4.3 representa a integral. Se por outro lado, a frequência é muito menor que  $\delta$  ( $\delta \gg |j\omega|$ ) então a função de transferência apresenta um ganho limitado em baixas frequências. Este comportamento é semelhante ao de um filtro passa baixas com uma frequência de corte pequena. A Figura 4.14 mostra o diagrama de blocos da função de transferência implementada, sendo que a rotina de chaveamento escolhida para implementar o estimador possui alta prioridade e é executada a uma frequência de 20 KHz, o que significa erros pequenos no processo de estimação.



**Figura 4.14-Estimador de fluxo para a coordenada  $\alpha$**

O bloco integrador  $1/S$  da Figura 4.14 foi implementado por um método de integração simples: foram somados retângulos com período igual a  $50 \mu\text{seg}$ . O algoritmo utilizado é dado pela seguinte equação a diferenças:

$$u_n = u_{n-1} + T e_n \quad (4.4)$$

na qual  $u_n$  é o fluxo atual,  $u_{n-1}$  é o fluxo anterior,  $T$  é o passo de integração e  $e_n$  é o erro atual. Como pode ser observado, esta equação não exige muito tempo de processamento da CPU.

A função de transferência da equação 4.3 implementada tem o inconveniente de deslocar a onda de saída da onda de entrada, por isso o ganho  $\delta$  deve ser escolhido considerando a frequência mais baixa de operação do sistema. No caso do sistema implementado, foram obtidas boas respostas no controle com frequências de até 2 Hz, com valores de  $\delta=0,2$  rad./seg.

#### **4.4.7 Rotina de controle vetorial**

A rotina de controle é a parte do processamento que mais exige da CPU. A Figura 4.15 esquematiza a seqüência de operações executadas dentro de um ciclo de 250  $\mu$ seg. Esta rotina tem a mais baixa prioridade devido ao fato dela gerar sinais de saída de baixa frequência para o inversor, como por exemplo a frequência de 60 Hz para as correntes do motor. As referências  $i_\alpha^*$  e  $i_\beta^*$  são atualizadas a cada 5,4 graus isto significa que ela é atualizada 66 vezes em um período de 16.66 ms. À medida que a frequência de operação das correntes do motor diminui, o número de atualizações é aumentado no mesmo período.

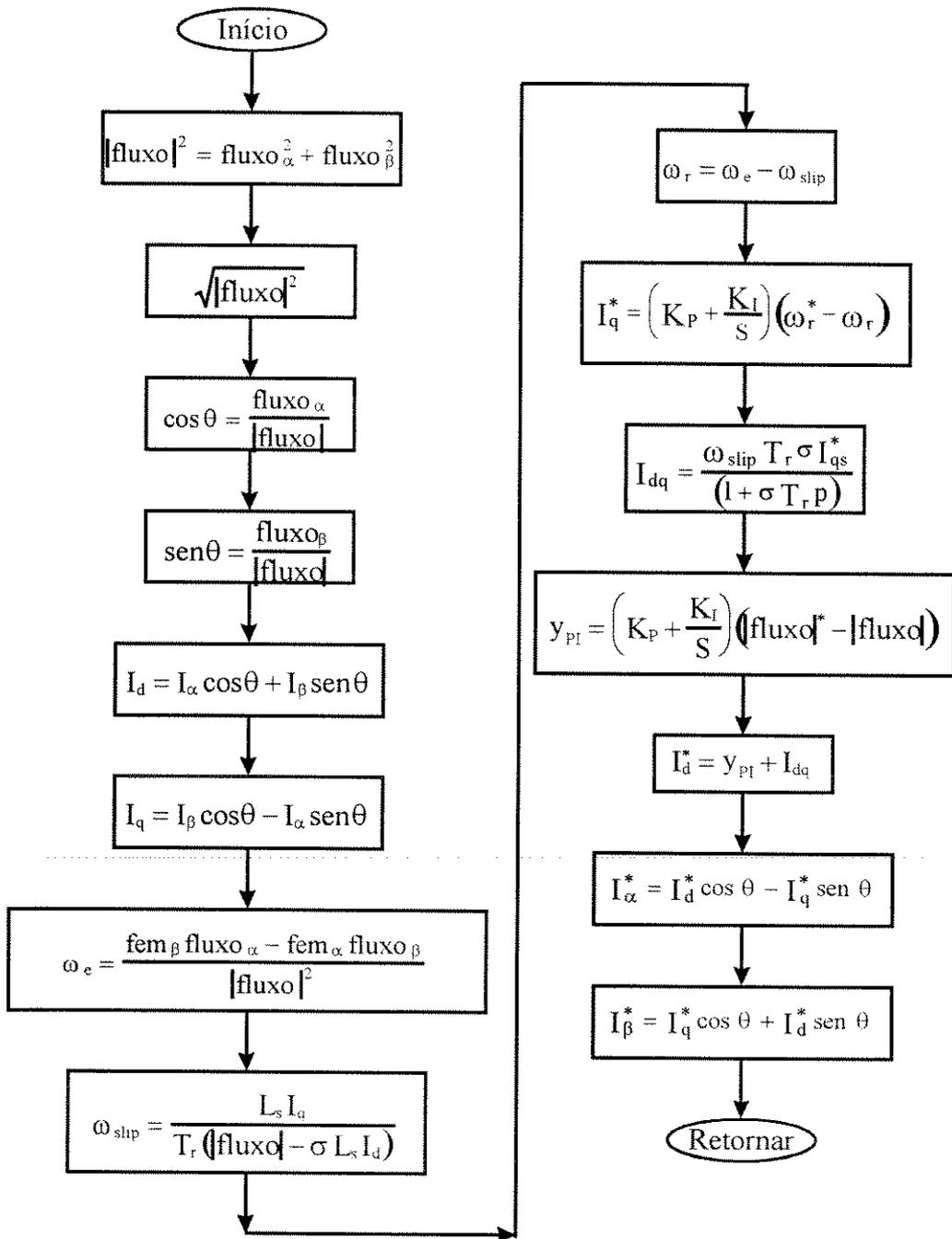


Figura 4.15-Rotina de controle

Como esta rotina não interrompe nenhuma outra, devido a sua baixa prioridade, não é necessário salvar inicialmente os registros por ela usados. Sendo assim, o programa começa calculando o módulo do fluxo ao quadrado a partir das componentes estimadas na rotina de chaveamento. Um algoritmo de raiz quadrada de 23 bits é usado para calcular o módulo do vetor de fluxo, a partir deste juntamente

com as componentes  $\alpha$  e  $\beta$  são calculados os valores seno e cosseno do ângulo de fluxo.

Uma vez calculados o seno e o cosseno é realizada a transformação das correntes  $\alpha$  e  $\beta$  ao sistema d-q. A frequência elétrica é calculada usando a derivada do ângulo de fluxo que está em função das componentes de fluxo e força contra-eletromotriz (fem).

De posse dos parâmetros indutância de estator, constante de tempo elétrica de rotor e indutância de dispersão, todos eles armazenados como constantes em memória, é calculada a frequência de escorregamento que é utilizada para a determinação da velocidade do eixo do motor. A saída do PI  $i_q^*$  é somada com a saída do bloco desacoplador para conseguir o sinal de comando de corrente  $i_d^*$ . O sinal de comando  $i_q^*$  é lido em um endereço de memória, cujo valor é controlado externamente através de um potenciômetro se o objetivo for operar o motor com controle de torque. Para o caso de controle de velocidade a referência  $i_q^*$  é obtida a partir da saída do PI de velocidade.

Os sinais de comando  $\alpha$  e  $\beta$  necessários para o inversor são produzidos a partir da rotação dos sinais de comando d-q gerados pelo algoritmo de controle. O tempo total de execução da rotina foi de 40  $\mu\text{seg}$  no caso de controle de torque e 45  $\mu\text{seg}$  no controle de velocidade. Já que o sistema apresentou estas características, foram testados tempos de interrupção do "timer" SCI de até 125  $\mu\text{seg}$  conseguindo-se respostas mais rápidas nos sinais de controle gerados e um melhor desempenho geral do sistema.

As variáveis estimadas  $\omega_e$  e  $\omega_{\text{slip}}$  apresentam um "ripple" significativo que pode ser minimizado quando o sistema está corretamente sintonizado, mas no caso de um transitório no sistema o estimador gera oscilações iniciais. Isto é resolvido por um filtro associado a cada uma destas variáveis. O inconveniente desta técnica é o consumo de tempo da CPU e o atraso gerado pelos filtros.

# *CAPÍTULO 5*

## *RESULTADOS EXPERIMENTAIS*

### **5.1 INTRODUÇÃO**

Neste capítulo são apresentados resultados experimentais referentes ao sistema implementado para orientação direta de fluxo de estator, utilizando um DSP para executar os algoritmos de controle e de aquisição de sinais. Um computador pessoal Pentium 100 Mhz faz o gerenciamento do sistema com as funções de edição, compilação e depuração dos algoritmos projetados. Neste capítulo, são apresentados, inicialmente, resultados de testes efetuados com o objetivo de validar o desempenho das diversas partes componentes do sistema. Uma vez testados os circuitos e o "software", um motor de indução, em vazio, é alimentado com referências de corrente impostas e, são tomados resultados para o sistema de controle com e sem orientação de fluxo.

As curvas apresentadas foram tomadas utilizando o osciloscópio digital TDS 340 da Tektronix de 100 Mhz. Os sinais adquiridos pelo osciloscópio são transmitidos ao PC, utilizando o "software" de aquisição do osciloscópio.

Embora, fosse um dos objetivos deste trabalho apresentar resultados de um estimador de velocidade, projetado a partir da posição espacial do fluxo de estator,

pelo fato dele não operar adequadamente, as curvas de velocidade não foram incluídas. Isto se deve ao fato do sinais de velocidade apresentarem um "ripple" pronunciado. O desenvolvimento deste estimador de velocidade exige um maior tempo, o que não foi, infelizmente, possível para o presente trabalho.

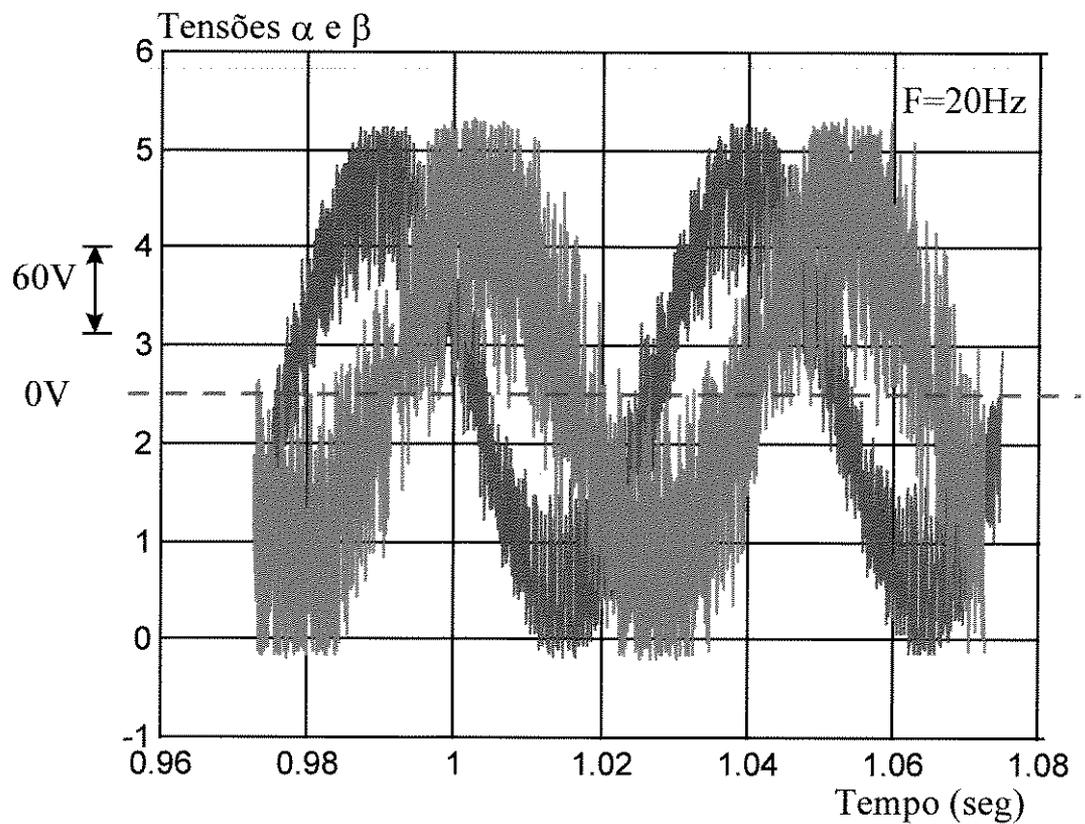
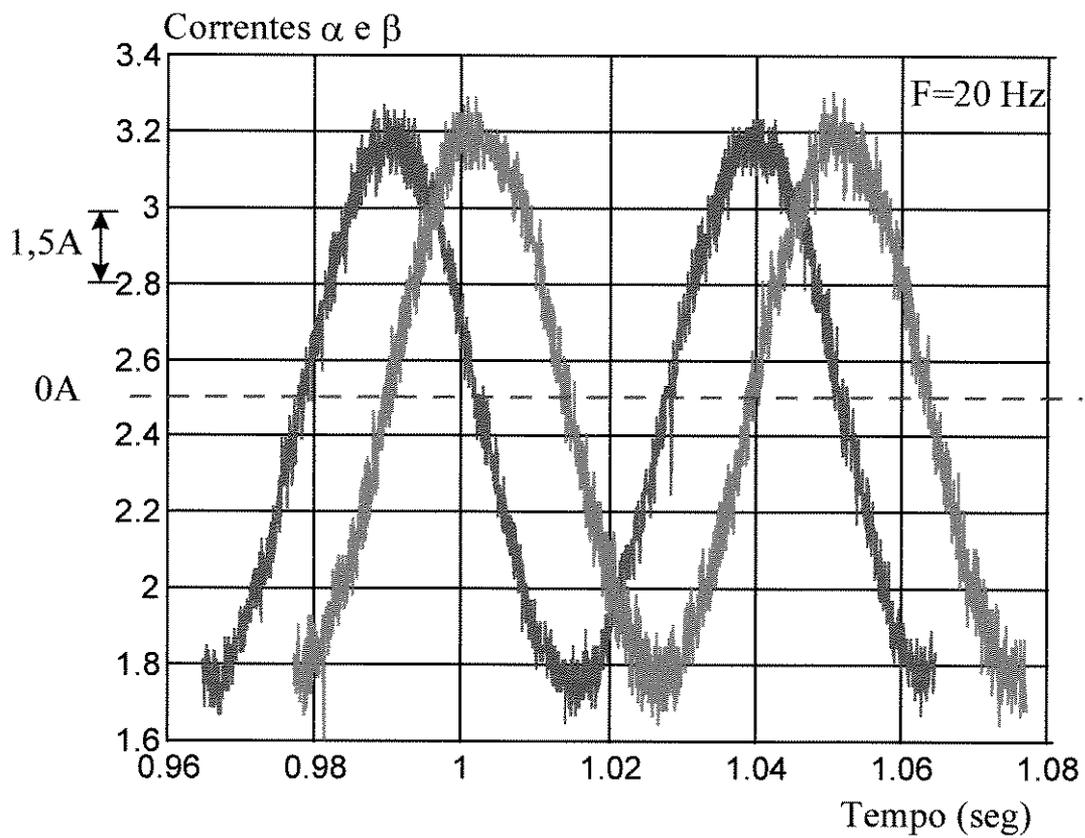
O inversor utilizado é de 10KVA e se utiliza de seis "IGBTs"

## **5.2 VERIFICAÇÃO DA OPERAÇÃO DO SISTEMA IMPLEMENTADO**

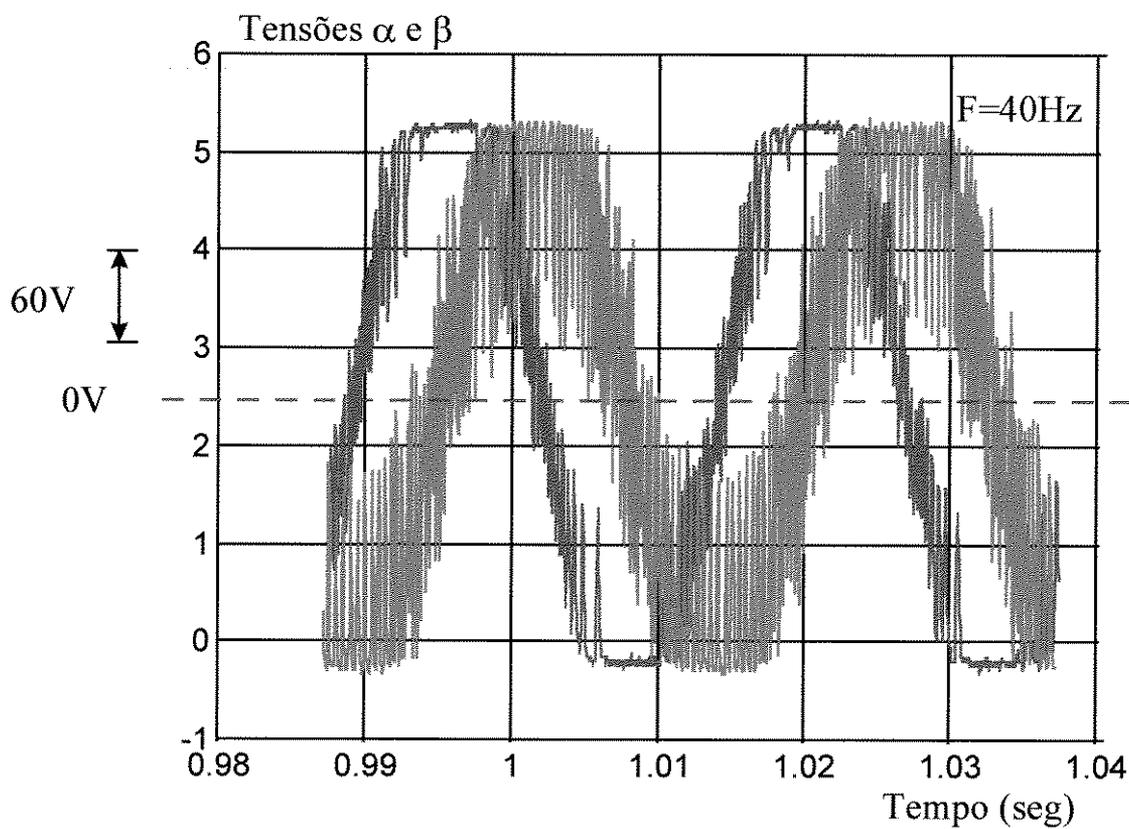
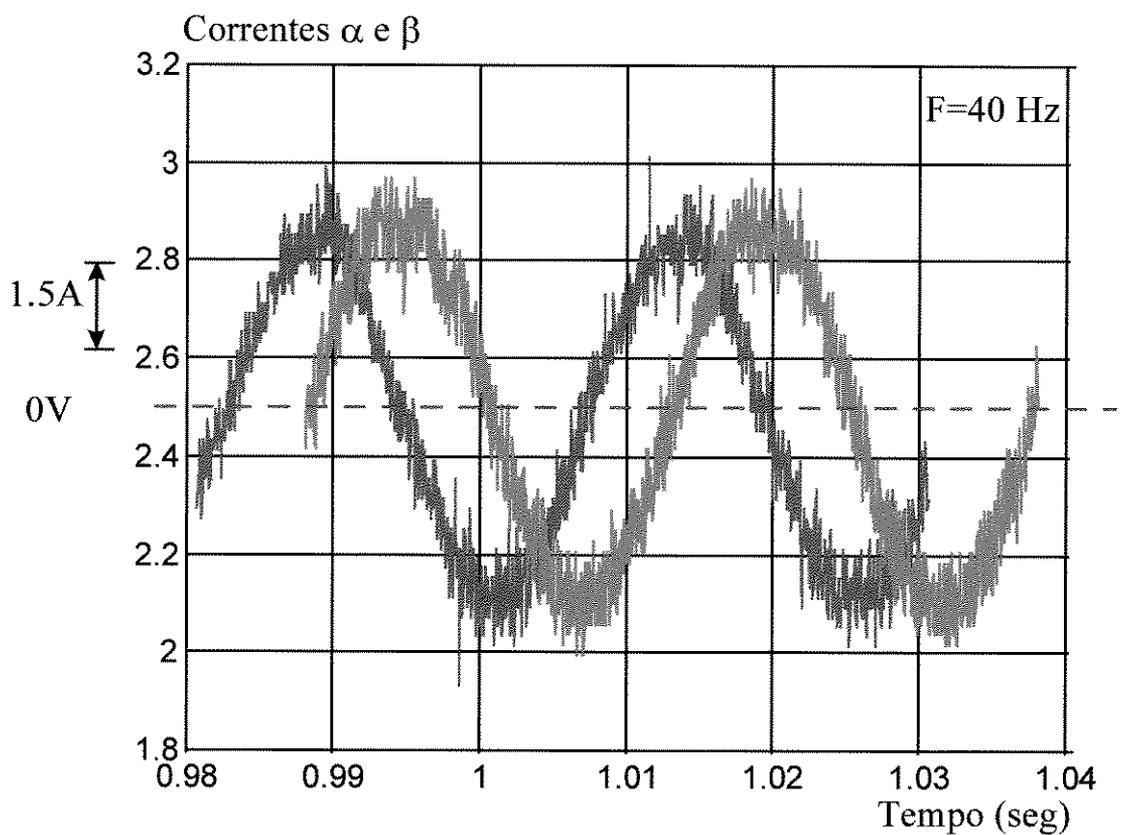
Para verificar a aquisição das correntes e tensões de estator e a correspondente transformação ao sistema de referência estacionário  $\alpha$ - $\beta$ , bem como a estimação do fluxo de estator, foram realizados testes adotando três valores de frequências de alimentação: 20, 40 e 60 Hz. A tensão de entrada do retificado foi mantida constante no valor de 220 Volts eficazes.

As curvas experimentais relativas às correntes e tensões estão mostradas nas Figuras 5.1, 5.2 e 5.3, enquanto o resultado da estimação do fluxo é colocado na Figura 5.4. Como se observa, tanto as correntes como as tensões  $\alpha$ - $\beta$ , estão em quadratura, demonstrando que o sistema de aquisição, de transformação das correntes e tensões e o algoritmo de controle do inversor está operando corretamente.

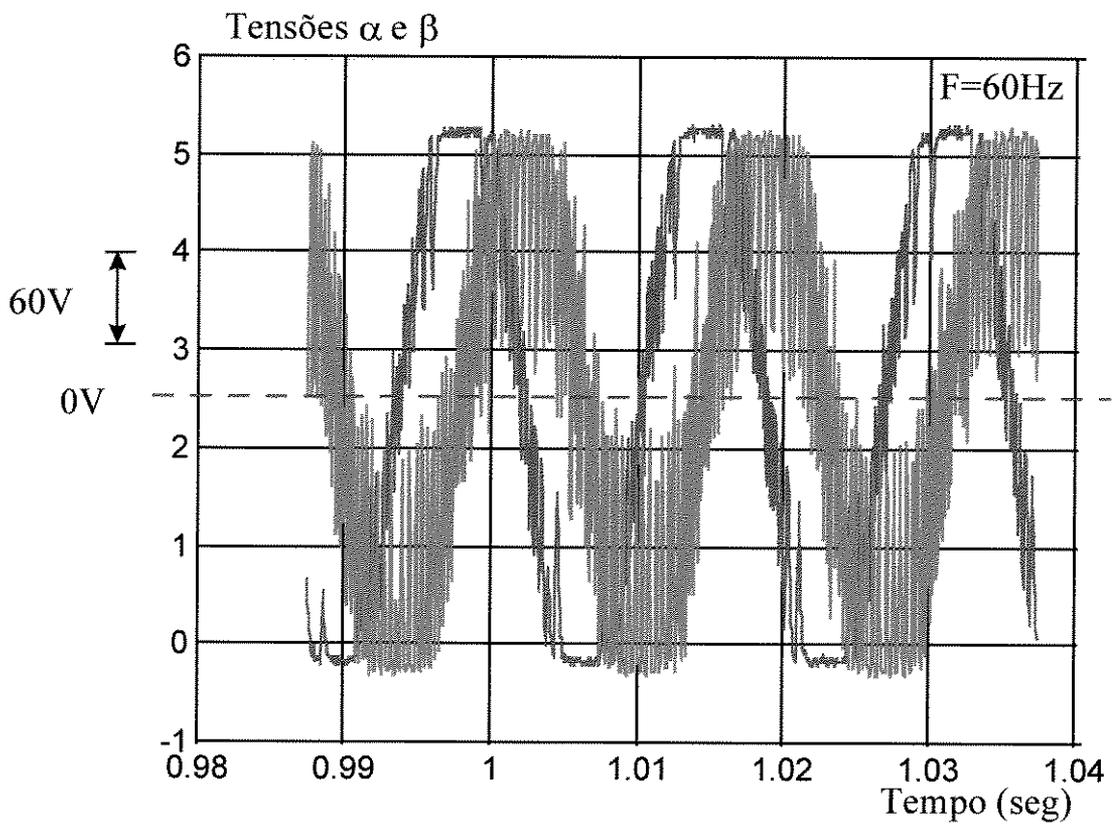
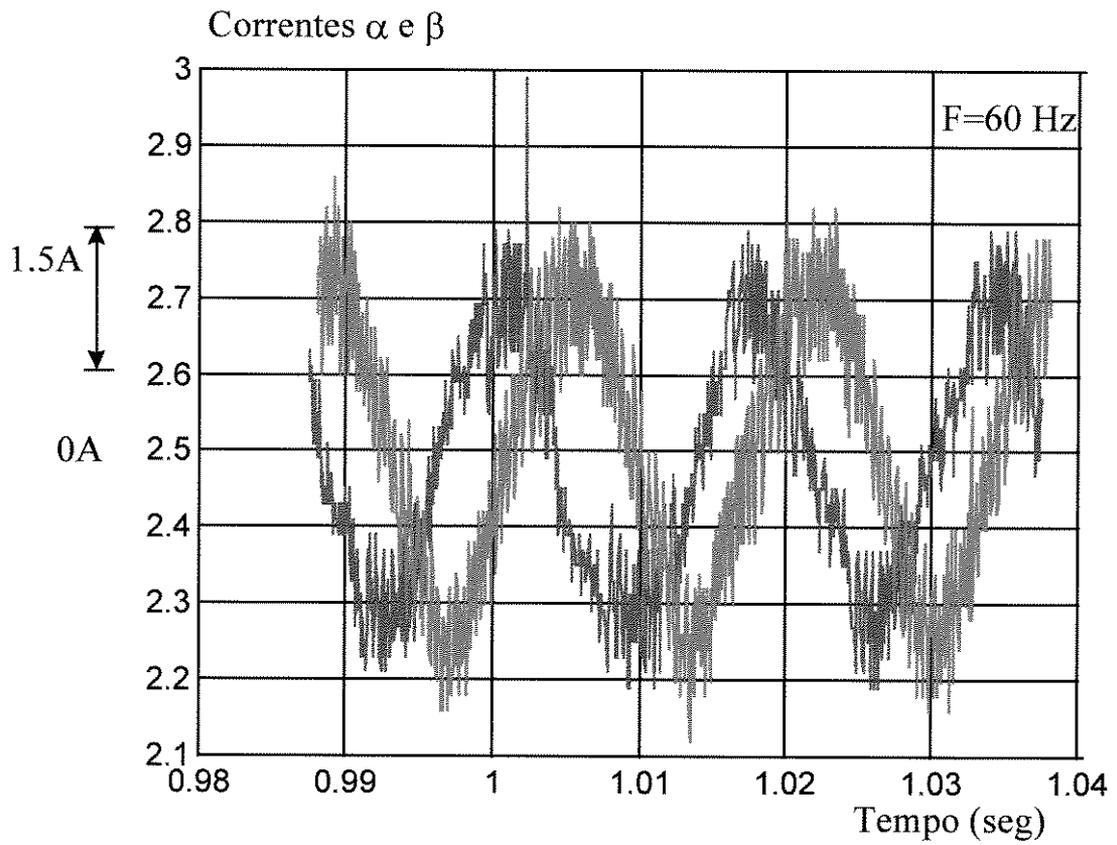
Inicialmente, o algoritmo estimador de fluxo foi projetado usando o método de integração de Euler de primeira ordem. Este estimador foi abandonado por apresentar problemas de saturação da integral. Adotou-se, então, uma versão modificada da integral empregando um filtro corta baixo com uma frequência de corte de 2 Hz. Os resultados do fluxo estimado, mostrado na Figura 5.4, não só possui baixo conteúdo harmônico como também as suas componentes  $\alpha$  e  $\beta$  estão em quadratura. Isto indica que o algoritmo estimador de fluxo está operando corretamente.



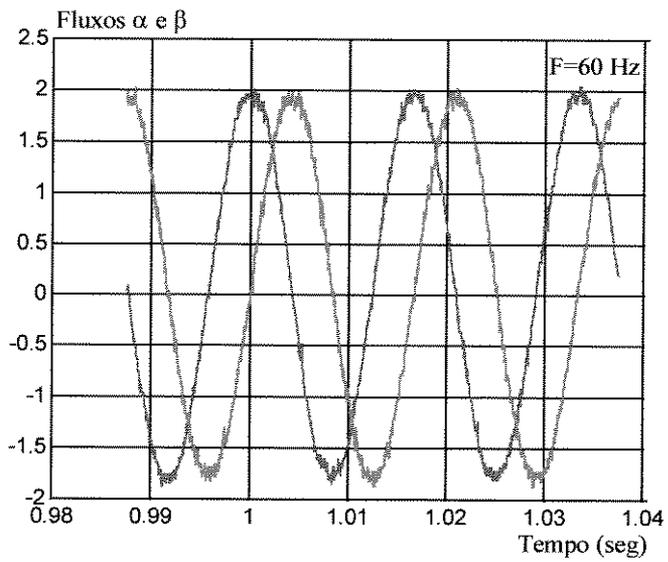
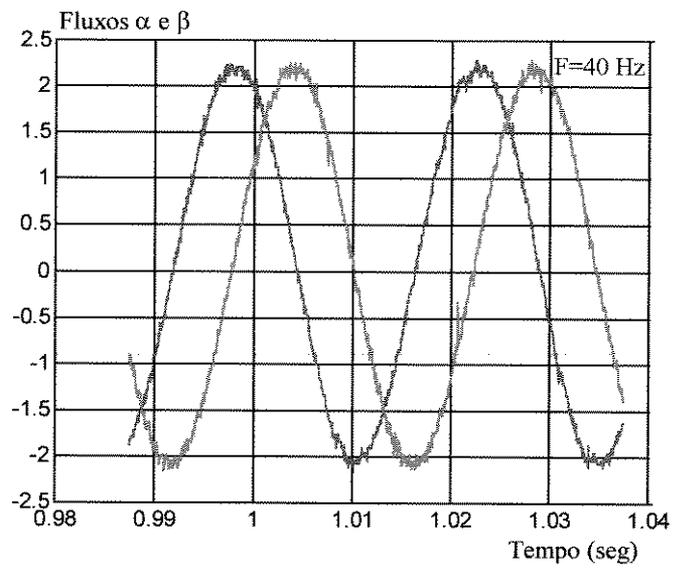
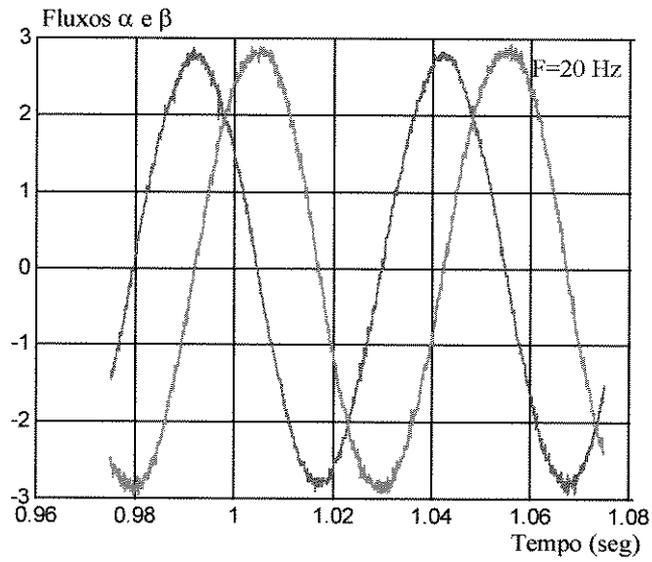
**Figura 5.1-Curvas experimentais de correntes e tensões a 20 Hz,  
para o motor em vazio**



**Figura 5.2-Curvas experimentais de correntes e tensões a 40 Hz,  
para o motor em vazio**



**Figura 5.3-Curvas experimentais de correntes e tensões a 60 Hz,  
para o motor em vazio**



**Figura 5.4-Curvas experimentais do estimador de fluxo**

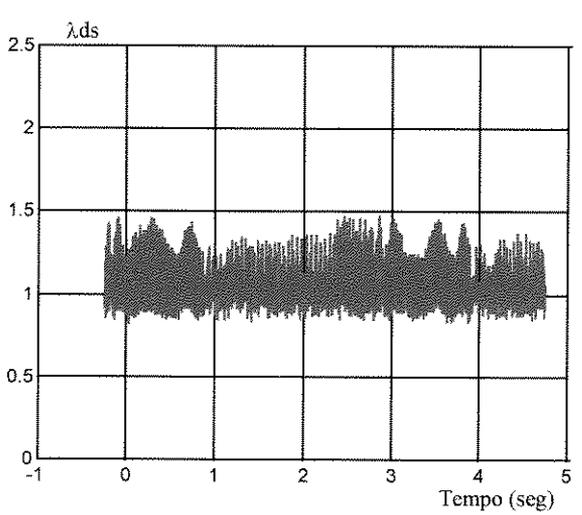
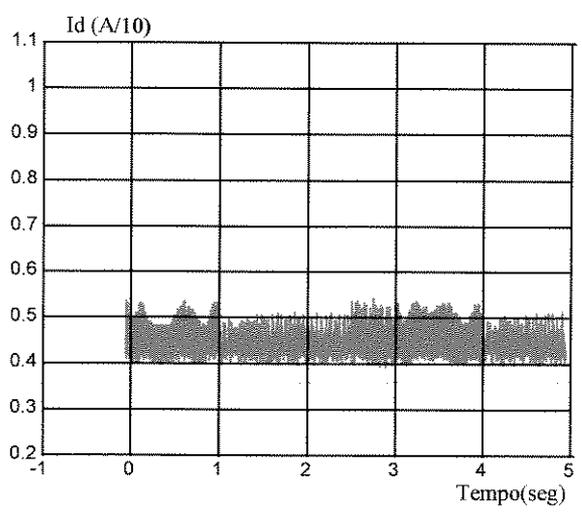
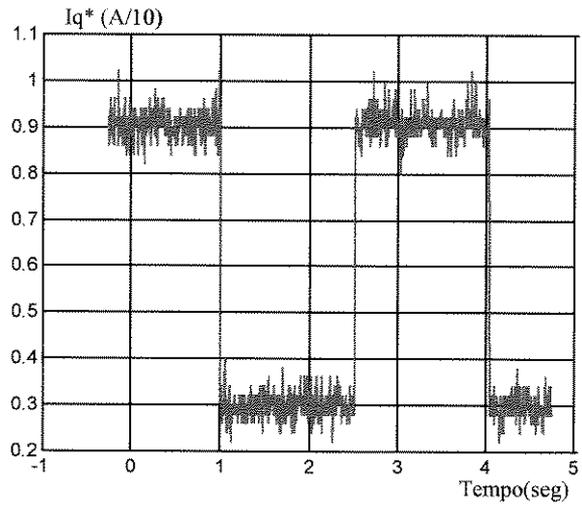
### 5.3 RESULTADOS REFERENTES AO MÉTODO DIRETO DE CONTROLE VETORIAL

Foram feitos vários testes para avaliar o desempenho do sistema de controle vetorial tanto em regime como no estado transitório.

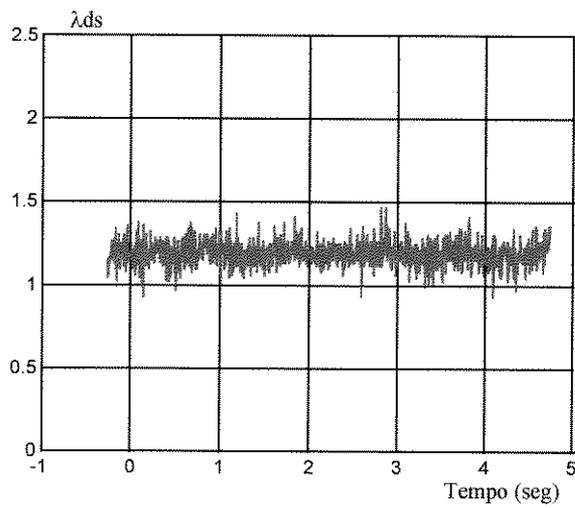
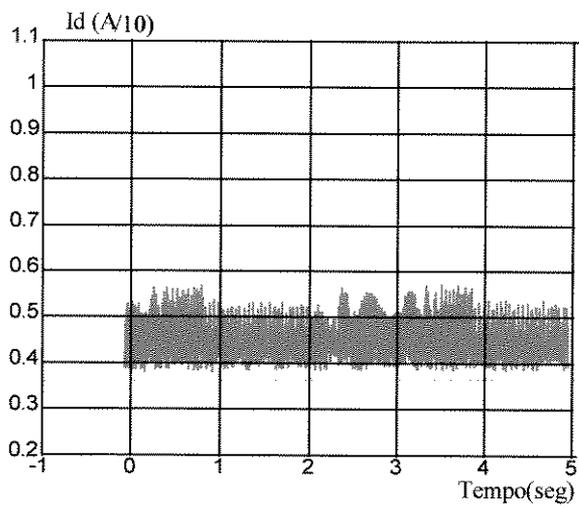
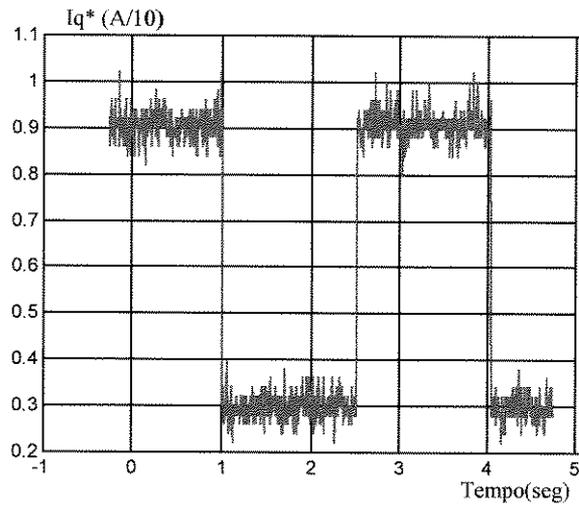
Os testes em regime, foram feitos com e sem o bloco desacoplador. Foi notado que para baixos valores de correntes de referência  $I_{qs}^*$ , o conjugado produzido apresentou fortes componentes oscilatórias; a frequência elétrica nesta situação apresentou baixo valor e, o estimador de fluxo não operou adequadamente.

Nos testes experimentais realizados para verificar a resposta ao degrau da corrente  $I_{qs}$ , colocados na Figura 5.5, mostram a existência do acoplamento entre a componente  $I_{qs}$ ,  $I_{ds}$  e  $\lambda_{ds}$ . Observa-se, ainda que a corrente  $I_{ds}$  altera-se de maneira síncrona com a componente de referência  $I_{qs}$ , enquanto que a variação no fluxo  $\lambda_{ds}$  é pequena e isto é devido a atuação do controlador PI.

Em relação aos testes com orientação e com o elemento desacoplador, os resultados da Figura 5.6 mostram que o elemento desacoplador conseguiu diminuir levemente a variação do fluxo  $\lambda_{ds}$ , ocorrida para o caso sem o desacoplador. Para o motor sobre testes, o elemento desacoplador teve uma participação muito pequena, não nos permitindo concluir a sua eficácia.



**Figura 5.5-Resposta da corrente  $I_d$  e o fluxo  $\lambda_{ds}$  ao degrau de  $I_q^*$ , sistema com orientação mas sem o elemento desacoplador  $I_{dq}$ .**



**Figura 5.6- Resposta da corrente  $I_d$  e o fluxo  $\lambda_{ds}$  ao degrau de  $I_q^*$ , sistema com orientação e com o elemento desacoplador  $I_{dq}$ .**

## *CAPÍTULO 6*

### *CONCLUSÕES*

Neste trabalho foi feita a implementação digital do controle direto mediante a orientação do fluxo de estator, utilizando um processador digital de sinais (DSP). A geração de procedimentos sobre o uso do DSP, sobre o controle por orientação do fluxo de estator como também a construção de uma infra-estrutura que permita futuros desenvolvimentos e estudos na área de controle vetorial de máquinas elétricas constituíram-se nas principais metas deste trabalho.

Para facilitar o acesso aos sinais das correntes transformadas  $\alpha$ - $\beta$ , optou-se por uma implementação por "hardware", embora estas transformações pudessem ser realizadas por "software". Testes preliminares, com algoritmos projetados para realizar a transformação de correntes e utilizando um DSP, indicaram o tempo de 500 nseg para executar a referida transformação.

Os sensores de tensão utilizados, na verdade sensores de corrente adaptados, apresentaram uma resposta em frequência de 20 KHz. Para futuras implementações recomenda-se o uso de um sensor de maior resposta em frequência.

A técnica utilizada no controle do inversor foi a modulação PWM vetorial controlada por corrente. O emprego dos vetores espaciais zero de corrente permitiu a diminuição do chaveamento dos "IGBT", o que significou um baixo conteúdo de harmônicos na corrente. O chaveamento do inversor foi sincronizado com uma onda de 20KHz, ou seja, cada chaveamento só pode ser realizado em múltiplos de 50  $\mu$ seg. O algoritmo de controle do inversor foi implementado integralmente no DSP, que o executava em 15  $\mu$ s e, este baixo tempo da CPU permitiu a execução, de maneira concorrente, do algoritmo do sistema de controle por orientação de campo.

O sistema de controle do inversor verificou ter um bom desempenho para o sistema de controle vetorial implementado e, como consequência, não foi preciso utilizar algoritmos de otimização para diminuir a quantidade de harmônicos gerados pelo modulador vetorial PWM.

No que se refere ao algoritmo de estimação de fluxo do estator, verificou-se que o processo de filtragem provocou uma defasagem de 0,1 graus entre a componente fundamental de tensão terminal do motor e a onda filtrada. A frequência de corte do filtro implementado para a onda de tensão medida foi de 4KHz. Dependendo da defasagem apresentada, pode ser necessário o uso de algoritmos mais sofisticados ou mesmo de circuitos de correção do ângulo de fase.

O algoritmo do sistema de controle vetorial foi executado em intervalos de 250  $\mu$ seg, com um tempo de execução de aproximadamente 40 $\mu$ s.

O elemento desacoplador projetado e implementado permitiu diminuir o acoplamento entre a componente de fluxo  $\lambda_{ds}$  e a componente de corrente  $I_{qs}$ , embora não seja possível concluir-se sobre a sua eficácia. Seria importante fazer testes com motores de potência maiores para concluir sobre o seu desempenho. É importante ressaltar que embora a componente  $I_{ds}$  se altere diante da alteração da componente  $I_{qs}$ , o fluxo de estator, segundo o eixo direto, altera-se muito pouco.

O método direto de controle vetorial, por orientação de fluxo de estator, exige grande complexidade computacional; além do mais, com o advento das técnicas modernas de estimação, identificação e controle adaptativo tais como lógica fuzzy e redes neurais artificiais aplicadas ao controle das máquinas elétricas fazem dos processadores digitais de sinais, a alternativa mais natural às implementações digitais.

A implementação de estimadores de velocidade, dispensando o uso de "encoders", de estimadores de resistência de estator constituem-se em interessantes projetos a ser executados no futuro.

# APÊNDICE A

## DETERMINAÇÃO DE PARÂMETROS DO MOTOR DE INDUÇÃO

### A.1 DADOS DE PLACA DO MOTOR

Potência nominal : 2,25 KW

Velocidade Nominal: 1700 RPM

Tensão Nominal: 230 Volts

Corrente Nominal: 9 A

### A.2 ENSAIOS DO MOTOR

Feitos os ensaios de vazio e de rotor bloqueado no motor de indução foram encontrados os seguintes parâmetros:

Indutância do estator :  $L_s=100.32$  mH

Indutância do rotor :  $L_r=100.32$  mH

Indutância de magnetização:  $L_m=94$  mH

Resistência do estator:  $R_s=0.6765$   $\Omega$

Resistência do rotor :  $R_r=1.93$   $\Omega$

Constante de tempo elétrica do rotor:  $T_r=0.051813$  seg

Fator de dispersão :  $\sigma=0.12202$

### **A.3 CÁLCULO DA RESISTÊNCIA DE ESTATOR**

O teste para a determinação da resistência de estator foi efetuado aplicando um vetor fixo de tensão no motor e medindo a corrente resultante. A resistência de estator é calculada, então, pela relação entre a tensão aplicada e a corrente medida.

# APÊNDICE B

## DESCRIÇÃO DO DSP56001

### B.1 Características básicas de operação do DSP56001

O coração do processador consiste de 3 unidades de execução operando em paralelo: A unidade aritmética lógica (ALU), a unidade de geração de endereços (AGU) e o controlador de programa. O DSP56001 inclui periféricos no estilo MCU, memória de programa, memória de dados e porta de expansão de memória. Tem 512 palavras de RAM e 2 ROMs de dados de alta velocidade além de hardware especial de partida para permitir a carga de programas de usuário na RAM de programa. O modelo de programação de estilo MPU e seu conjunto de instruções permite uma geração simples de código compacto e eficiente. O DSP56001, graças a seu alto desempenho, é apropriado para aplicações em comunicações, controle de alta velocidade, processamento numérico, aplicações em computação e aplicações em áudio. Estas características podem ser resumidas da seguinte maneira:

**Velocidade:** 10.25 milhões de instruções por segundo (MIPS), pode executar uma FFT de 1024 pontos em 3.23 mseg.

**Precisão:** O comprimento dos barramentos de dados é de 24 bits.

**Paralelismo:** Cada unidade interna de execução (AGU, Controlador de programa, ALU de dados), memória e periféricos opera independentemente e em paralelo com as outras unidades por meio de um sofisticado barramento do sistema. O DSP 56001 pode executar até 4 instruções em um ciclo de instrução.

**Integração:** Possui 3 unidades de execução independentes, 6 memórias, 3 periféricos, um gerador de relógio e sete barramentos.

**Pipeline:** Tem um “pipeline” de 3 etapas e é invisível ao programador.

**Instruções:** Possui 62 mnemônicos de instruções de estilo MCU, isto faz a transição da programação em microprocessador a DSP56001 muito fácil. O uso das instruções REP e laço de “hardware” DO permitem programação estruturada.

## **B.2 Diagrama de blocos do DSP56001**

A Figura B.1 mostra os componentes principais do DSP56001. As partes principais são explicadas a continuação:

**Barramentos de dados:** Possui 4 "barramentos" de dados de 24 bits, barramento X (XDB), barramento Y (YDB), o barramento de dados de programa (PDB) e o barramento global de dados (GDB). Os barramentos de dados X e Y podem também ser tratados por certas instruções como um barramento de dados de 48 bits concatenando XDB e YDB.

**Barramentos de endereço:** Possui 3 "barramentos" de endereços, o barramento X (XAB) e o barramento Y (YAB) são usados para endereçar dados; o barramento P (PAB) é usado para endereçar memória de programa.

**Switch de barramento interno:** Realiza as transferências entre barramentos. Similar a uma matriz de chaves, pode conectar dois barramentos internos sem agregar atrasos no “pipeline”. Esta flexibilidade simplifica a programação.

**Unidade de manipulação de bit:** Esta unidade é fisicamente situada no “switch” de barramento interno porque o barramento de dados interno pode acessar cada espaço de memória. A unidade de manipulação de bit executa operações de manipulação de bit em endereços de memória, registros de endereço, registros de controle e registros de dados via os barramentos XDB, YDB e GDB.

**ALU de dados:** Executa todas as operações aritméticas e lógicas nos operandos de dados. Tem 4 registros de 24 bits, 2 registros acumuladores de 48 bits, 2 registros de extensão de acumulador de 8 bits, um deslocador de acumulador, 2 circuitos deslocador/limitador do barramento de dados e uma unidade multiplicadora e acumuladora (MAC) paralela de um ciclo de instrução. As operações da ALU de dados usam aritmética fracional em complemento de 2.

**Unidade de geração de endereço(AGU):** Esta unidade realiza o armazenamento de endereços e os cálculos necessários de operandos de dados para endereçamento indireto. Esta unidade opera em paralelo com outros recursos do chip para minimizar o consumo de tempo na geração do endereço. A AGU contém 8 registros de endereço (R0-R7), 8 registros de OFFSET (N0-N7) e 8 registros modificadores (M0-M7). Os registros Rn são de 16 bits e podem conter endereços ou dados.

**Memória de dados X:** A memória interna RAM X de dados de 24 bits, usa os 256 endereços mais baixos no espaço de memória X. A ROM de dados X usa os endereços 256-511 para armazenar 128 endereços de lei Mu e 128 endereços de lei A.

**Memória de dados Y:** A memória interna RAM Y de dados de 24 bits, usa os 256 endereços mais baixos no espaço de memória X. A ROM de dados Y usa os endereços 256-511 que tem sido programados com uma tabela de seno de 24 bits de 4 quadrantes.

**Memória de programa:** Consiste de 512 palavras de 24 bits. Os endereços são recebidos desde a lógica do programa. Os endereços dos vetores de interrupção dos recursos incluídos são localizados nos 64 endereços mais baixos da memória de programa. As memórias de dados X, Y e de programa podem ser expandidas até 65536 endereços.

**Controlador de programa:** Executa a pré-busca, decodificação de instrução, controle do laço DO de hardware e processamento de exceção. O controlador de programa contém uma memória de "stack" de 32 bits de 15 níveis e 6 registros endereçáveis diretamente: O contador de programa (PC), endereço de laço (LA), contador de laço (LC), registro de estado (SR), registro modo de operação (OMR) e ponteiro de stack (SP).

**Entrada/Saída:** A estrutura de E/S consiste de uma porta de expansão de 48 bits (PORT A) e 24 pinos adicionais de 24 bits. Este podem ser usados como pinos de propósito geral chamados PORTB (15 linhas de E/S) e PORTC (9 linhas de E/S) ou usados como periféricos on-chip baixo controle do software. O DSP56001 tem 3 periféricos on-chip: Uma interface de acesso direto a memória ou unidade para microprocessador host paralelo de 8 bits, um SCI (interface de comunicação serial) e um SSI(interface serial síncrona).

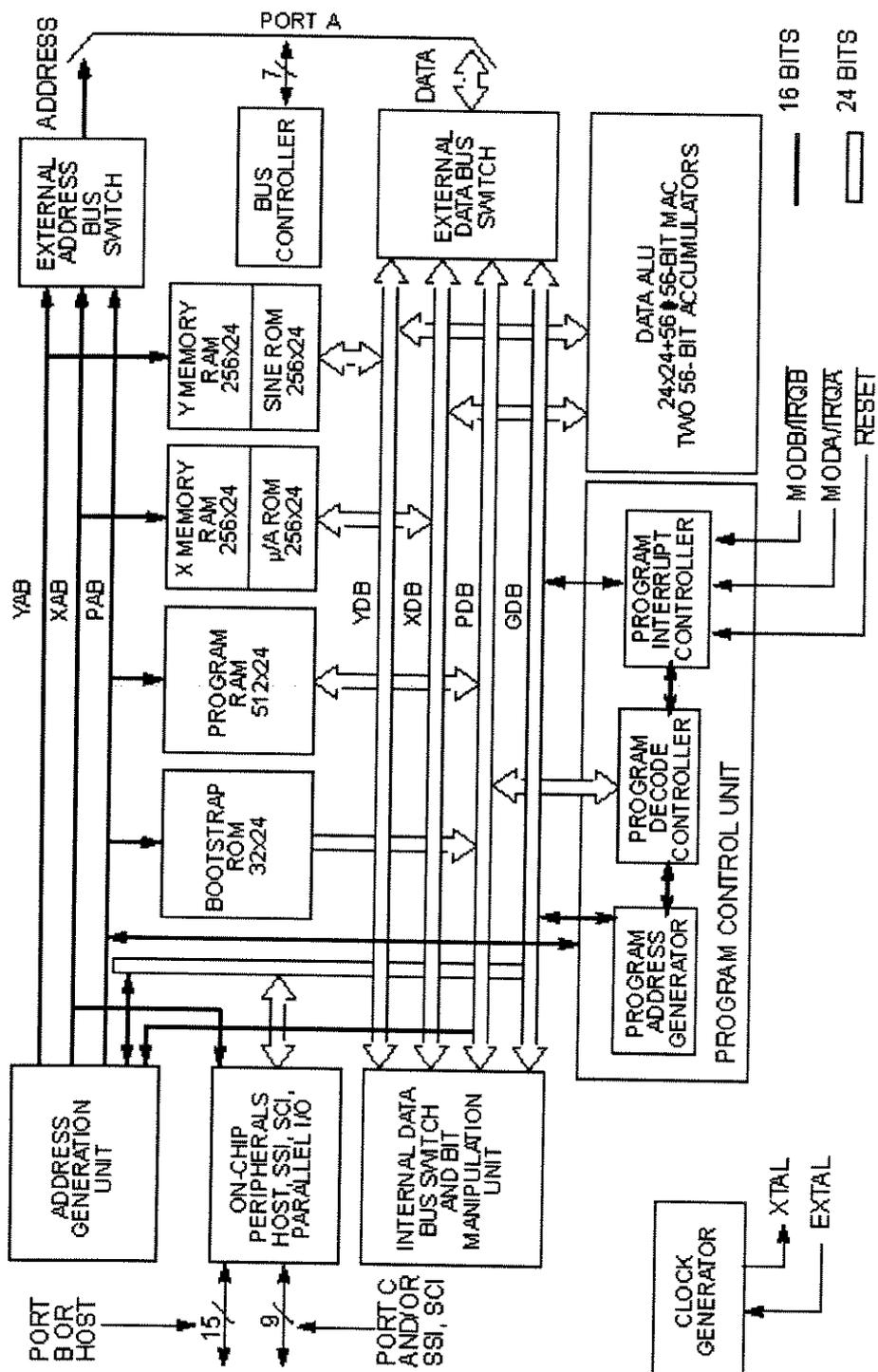


Figura B.1-Diagrama de blocos do DSP56001

A Figura B.2 mostra os grupos de sinais organizados por função.

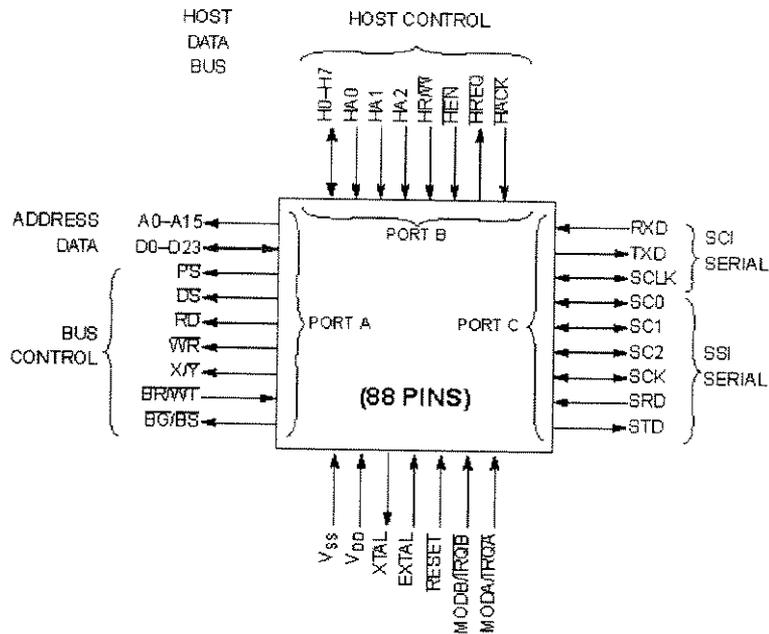
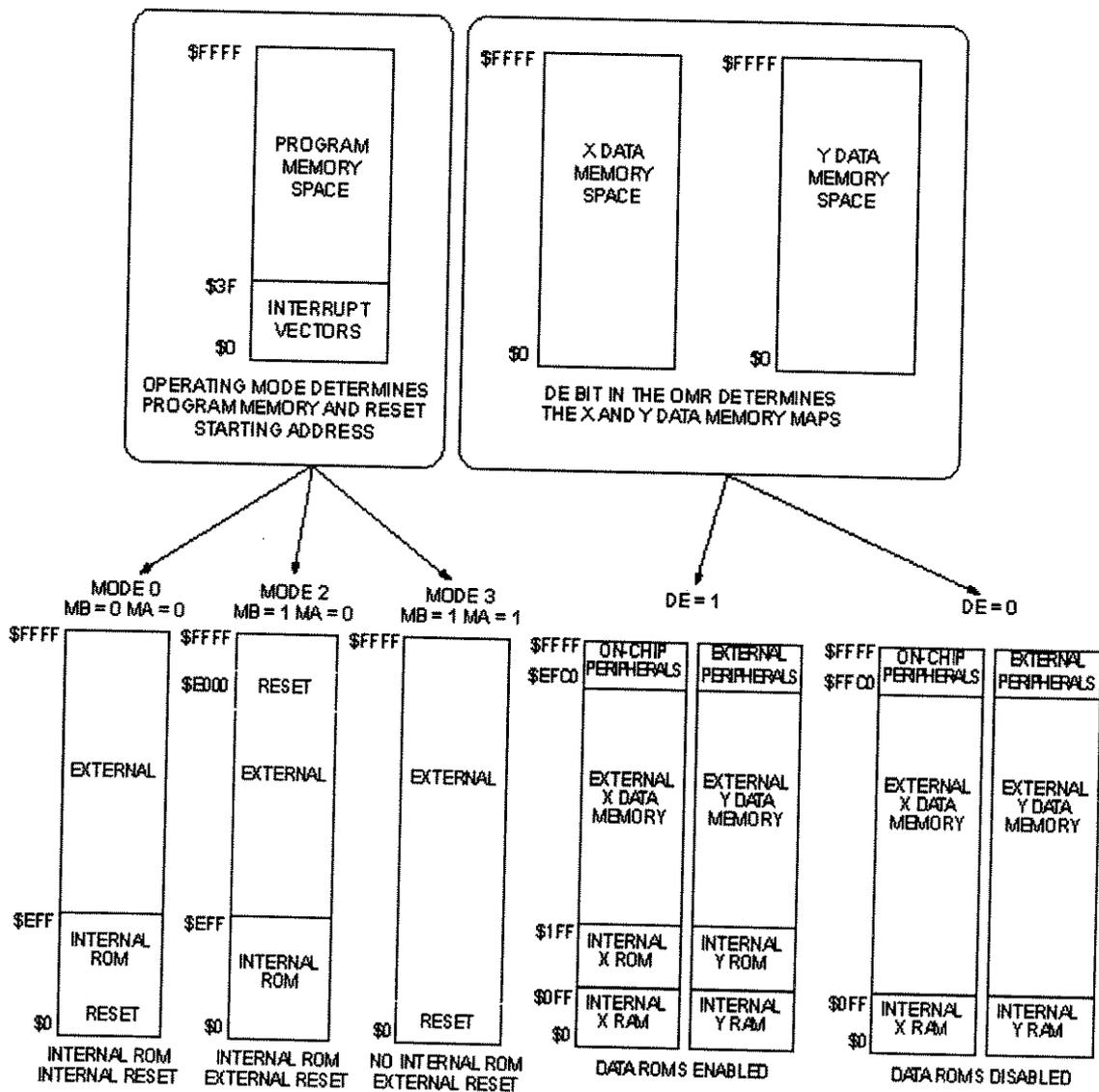


Figura B.2-Grupos de sinais do DSP56001

### B.3 Espaços de memória

A memória do DSP56001 pode ser dividida em diferentes formas para proporcionar operação paralela de alta velocidade e expansão de memória externa. As memórias de dados e programa são separadas e por sua vez a memória de dados está dividida em dois espaços de memória separados X e Y. As memórias de programas e dados podem ser expandidas externamente. Tem também memórias ROM internas que podem sobrepor-se com parte da memória RAM de programa. As memórias de dados estão divididas em 2 espaços independentes para trabalhar com as ALUs de 2 endereços que armazenam 2 operandos simultaneamente à ALU de dados.

Os 3 espaços independentes de memória do DSP56001, dados X, dados Y e programa, são mostrados na Figura 3. Os espaços de memória são configurados pelos bits de controle no registro modo de operação (OMR). Os bits de controle de modo de operação (MA e MB) no OMR controlam o mapa de memória de programa e selecionam o endereço do vetor de "reset". O bit de habilitação de ROM (DE) no OMR controla os mapas de memória de dados X e Y e habilita/desabilita as ROMs de dados X e Y.



**Figura B.3** Mapa de memória do DSP56001

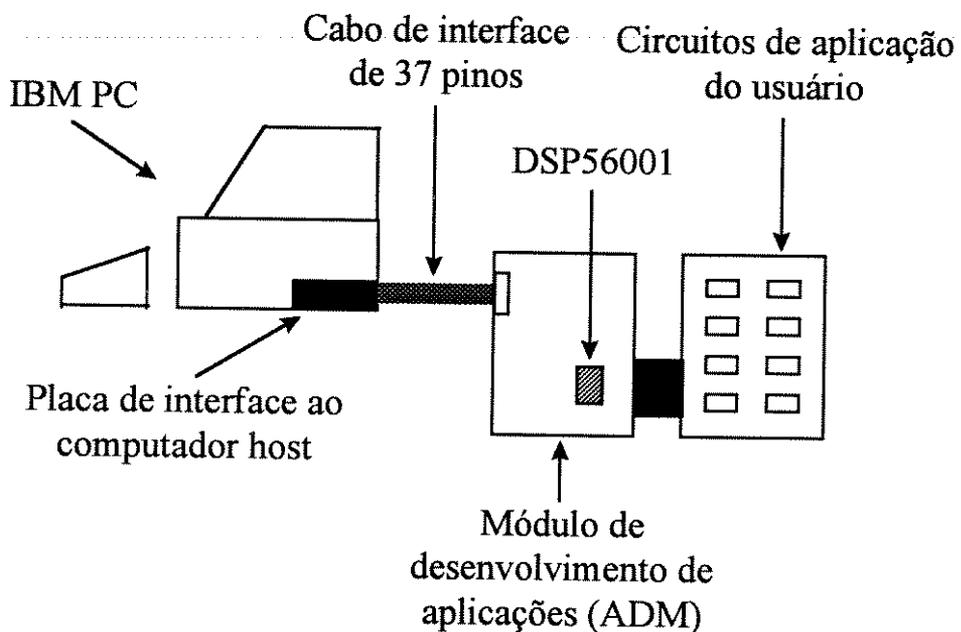


# APÊNDICE C

## DESCRIÇÃO DO SISTEMA DE DESENVOLVIMENTO

Foi empregado o Sistema de Desenvolvimento de Aplicações (ADS) de Motorola, que está composto dos seguintes três componentes, conforme mostrado na figura C.1:

- Módulo de desenvolvimento de aplicações (ADM), que contém um processador DSP56001 e circuitos de controle.
- Placa de interface HOST-BARRAMENTO, que suporta as comunicações entre o ADM e o computador "HOST".
- Programa de interface do usuário, interage com o usuário e controla até 8 ADMs.



**Figura C.1-Sistema de desenvolvimento**

## **CARACTERÍSTICAS DO SISTEMA ADS**

### **Hardware**

- Opera a 27 MHz
- Suporta múltiplas ADMs com endereçamento de ADM programável
- 8K palavras de RAM configurável para desenvolvimento de código do usuário, expandível a 32K palavras
- 2K palavras de EPROM monitor expandível a 4K palavras
- Suporte completo de múltiplos mapas de memória de dados
- Conectores separados para acessar portas seriais e paralelas
- Conector de 96 Pinos que fornece acesso a todos os pinos do DSP56001
- Não é requerida fonte externa quando está conectado ao "HOST computer"
- Operação "Stand-Alone" do ADM depois do desenvolvimento inicial
- Capacidade de emulação "in-circuit" quando é usado com o cabo DSP56EMUL-RCABL

### **Software**

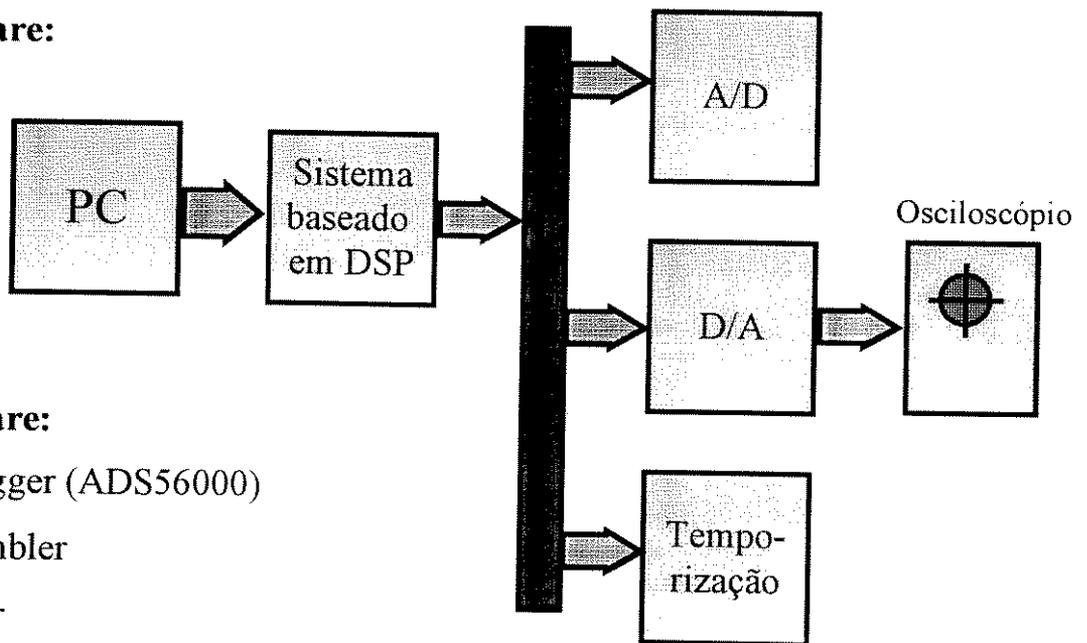
- Capacidade de execução passo a passo simples ou múltiplo nos programas objeto do DSP56001
- Pontos de rotura de software condicionais ou incondicionais
- Atualizador do programa usando assembler/disassembler de linha
- Registro da seqüência para posterior referência
- Registro de comandos para definição de macro
- Carga e armazenamento de arquivos a ou desde memória do ADM
- Display configurável de registros e conteúdo de memória
- Ajuda on-line para cada comando do ADS e registro do DSP56001

## Programadas de desenvolvimento

- Simulador DSP56000CLASA versão 3.2 (1990)
- Assembler DSP56000CLASA versão 3.1.12 (1990)
- Linker 3.1.14 DSP56000CLASA versão librarian 3.1.1 (1990)
- Compilador C DSP56KCCA versão G1.01 (1991)

A Figura C.2 mostra a configuração experimental do sistema implementado

### Hardware:



### Software:

- Debugger (ADS56000)
- Assembler
- Linker
- Simulador
- Compilador C

**Figura C.2-Diagrama de blocos do sistema implementado**

A Figura C.3 mostra o mapa de memória do sistema ADM, os programas foram armazenados a partir do endereço P:\$0040, os dados a partir de X:\$0000 e as constantes a partir de Y:\$0000.

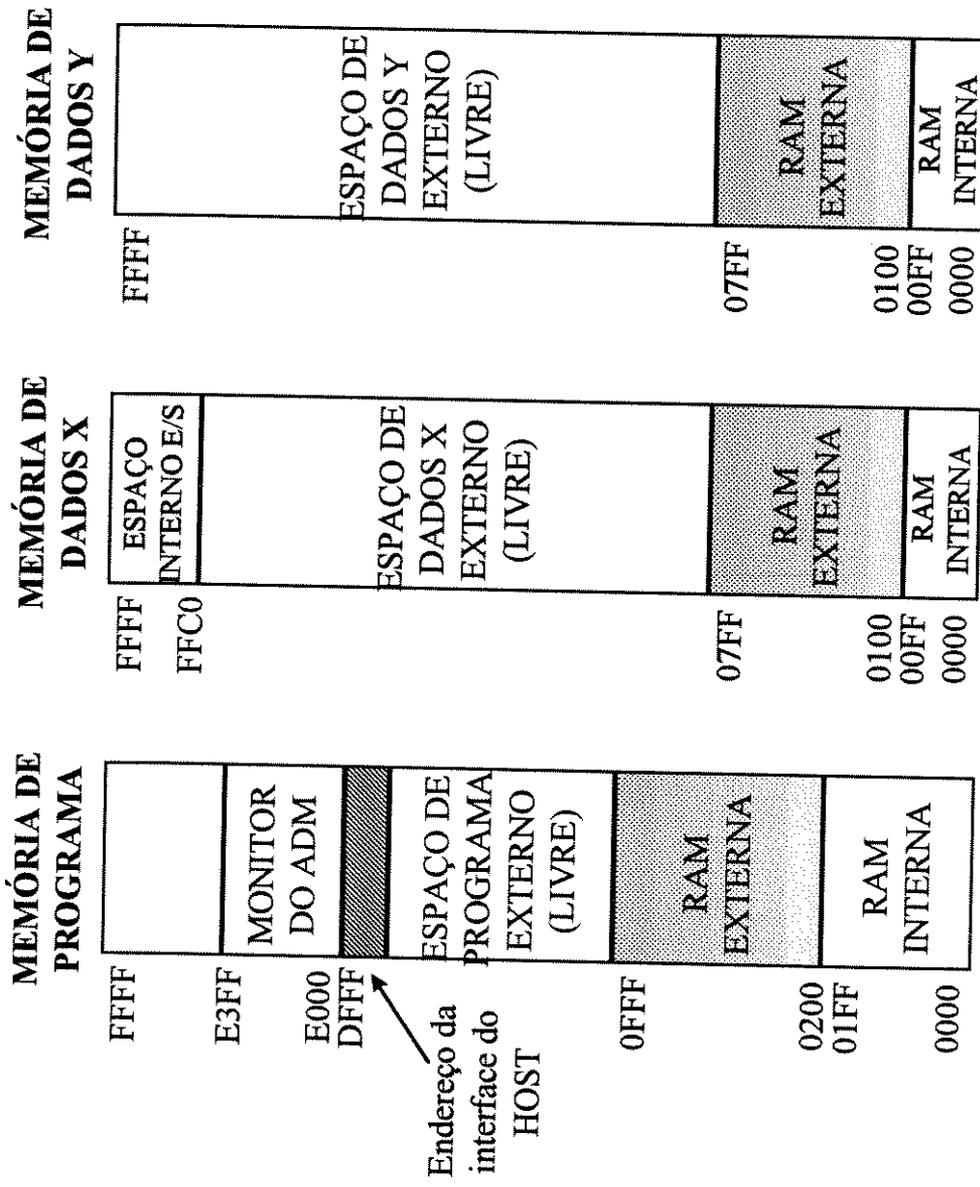


Figura C.3-Mapa de memória do ADM

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Lorenz, R.D.; Lipo, T.A. and Novotny, D., "Motion Control with Induction Motors", *Proceedings of IEEE-IAS*, Vol. 82, No. 8, pp. 1215-1240, August 1994.
- [2] Blaschke, F., "The Principle of Field Orientation as Applied to the New Transvector Closed Loop Control System for Rotating Machines", *Siemens Review*, Vol. 39, No. 5, pp. 217-220.
- [3] Plunkett, A.B., "Direct Flux and Torque Regulation in a PWM Inverter Induction Motor Drive", *IEEE Transactions On Industry Applications*, Vol. IA-13, No. 2, pp. 139-146, 1977.
- [4] Lipo, T.A., "Flux Sensing and Control of Static AC Drives by the Use of Flux Coils", *IEEE Transactions On Magnetics*, Vol. MAG-13, No. 5, pp. 1403-1408, 1977.
- [5] Lipo, T.A. and Chang, C., "A New Approach to Flux and Torque-Sensing in Induction Motors", *IEEE Transactions On Industry Applications*, Vol. IA-22, No. 4, pp. 731-737, July/August 1986.
- [6] Zinger, D.; Benzi, F. and Lipo, T.A., "A New Approach to Induction Motor Torque and Speed Control", *Record of Conference on Applied Motion Control*, pp. 25-29, 1986.
- [7] Moreira, J.C., "Direct Field Orientation Control Using the Third Harmonic Component of the Stator Voltage", *IEEE Transactions on Industry Applications*, Vol. IA-13, No. 2, pp. 139-146, March/April 1977.
- [8] Xu, X. and Novotny, D.W., "Implementation of Direct Stator Flux Orientation Control on a Versatile DSP Based System", *IEEE Transactions on Industry Applications*, Vol. 27, No. 4, pp. 694-700, July/August 1991.

- [9] Bose, M.K.; Simões, M.G.; Crecelius, D.R.; Rajashekara, K. and Martin, R., "Speed Sensorless Hybrid Vector Controlled Induction Motor Drive", *IEEE/IAS Ann. Meet. Conf. Rec.*, pp. 137-143, 1995.
- [10] Krause, P.C., "*Analysis of Electric Machinery*", McGraw Hill, 1986.
- [11] Leonhard, W., "*Control of Electrical Drives*", Springer Verlag, pp. 151-154, 1984.
- [12] Xu, X.; Doncker, R.D. and Novotny, D.W., "A Stator Flux Oriented Induction Machine Drive", *PESC'88 Conference Record*, pp. 870-876, April 1988.
- [13] Moham, N.; Robbins, W.P.; Underland, T.M.; Nilssen, R. and Mo, O., "Simulation of Power Electronic and Motion Control Systems an Overview", *Proceedings of the IEEE*, Vol. 82, No. 8, pp. 1287-1302, August 1994.
- [14] Pan, C-T. and Chang, T-Y., "A Microcomputer Based Vector Controlled Induction Motor Drive", *IEEE Transactions on Energy Conversion*, Vol. 8, No. 4, pp. 750-756, December 1993.
- [15] Rupperecht, G. and Leonhard, W., "Microprocessor Control of Induction Motor", *Proceedings of IEEE International Semiconductor and Power Converter Conference*, pp. 385-396, 1982.
- [16] Xu, X.; Doncker, R.D. and Novotny, D.W., "A Stator Flux Oriented Induction Machine Drive", *Conf. Rec. IEEE-PESC'88*, Kyoto, Japan, pp. 870-876, April 1988.
- [17] Ben-Brahim, L. and Kawamura, A., "A Fully Digitized Field-Oriented Controlled Induction Motor Drive Using only Current Sensors", *IEEE Transactions on Industrial Electronics*, Vol. 39, No. 3, pp. 241-249, June 1992.
- [18] Ohtani, T.; Takada, N. and Tanaka, K., "Vector Control of Induction Motor without Shaft Encoder", *IEEE Transactions on Industry Applications*, Vol. 28, pp. 157-164, January/February 1992.

- [19] Kerkman, R.J.; Seibel, B.J.; Rowan, T.M. and Schlegel, D.W., "A New Flux and Stator Resistance Identifier for AC Drive Systems", *IEEE Transactions on Industry Applications*, Vol. 32, No. 3, May/June 1996.
- [20] Khambadkone, A.M. and Holtz, J., "Vector-Controlled Induction Motor Drive with a Self Commissioning Scheme", *IEEE Transactions on Industrial Electronics*, Vol. 38, pp. 322-327, October 1991.
- [21] Verghese, G.C., "Observers for Flux Estimation in Induction Machines", *IEEE Transactions on Industrial Electronics*, Vol. 35, pp. 85-94, February 1988.
- [22] Hori, Y. and Umeno, T., "Flux Observer Based Field Orientation Controller for High Performance Torque Control", *Conf. Rec. on Int'l Pow. Elec. Conf.*, Tokyo, pp. 1219-1226, 1990.
- [23] Kubota, H.; Matsuse, K. and Nakano, T., "DSP-Based Speed Adaptive Flux Observer of Induction Motor", *IEEE/IAS Ann. Meet. Conf. Rec.*, pp. 380-384, 1991.
- [24] Habetle, T.G.; Profumo, F.; Griva, G.; Pastorelli, M. and Bettini, A., "Stator Resistance Tuning in a Stator Flux Field Oriented Drive Using an Instantaneous Hybrid Flux Estimator", in *European Power Electron. Conf.*, Brighton, England, pp. 101-107, 1993.
- [25] Okuyama, T.; Fujimoto, N. and Fujii, H., "A Simplified Vector Control System without Speed and Voltage Sensors-Effect of Setting Errors of Controls Parameters and their Compensation", *Elect. Eng.*, Japan, Vol. 110, No. 4, pp. 129-139, 1990.
- [26] Cabrera, L.A. and Elbuluk, M.E., "Tuning the Stator Resistance of Induction Motors Using Artificial Neural Network", *IEEE IAS Annual Meeting*, pp. 421-427, 1995.
- [27] Ho, E. and Sen, P., "Decoupling Control of Induction Motor Drives", *IEEE Transactions on Industrial Electronics*, Vol. IE-35, No. 2, pp. 253-262, May 1988.

- [28] Van Der Broeck, H.W. and Skudelny, H.C., "Analysis and Realization of a Pulsewidth Modulator Based on Voltage Space-Vector", *IEEE Transactions on Industry Applications*, Vol. IA-22, No. 1, pp. 814-821, 1988.
- [29] Kazmierkowski, M.P. and Sulkowski, W., "A Novel Vector Control Scheme for Transistor PWM Inverter-Fed Induction Motor Drive", *IEEE Transactions on Industrial Electronics*, Vol. 38, No. 1, pp. 41-47, February 1991.
- [30] Pollmann, A.J., "Software Pulsewidth Modulation for up Control of AC Drives", *IEEE Transactions on Industry Applications*, Vol. IA-22, No. 4, pp. 691-696, July/August 1986.
- [31] Plunkett, A.B., "A Current-Controlled PWM Transistor Inverter Drive", *Conf. Rec. IEEE/IAS Annual Meeting*, pp. 785-792, 1979.
- [32] Kazmierkowski, M.P., "Control Philosophies of PWM Inverter-Fed Induction Motors", *XI CBA*, S. Paulo, Brasil, September 1996.
- [33] DSP56000/DSP56001 Digital Signal Processor User's Manual.
- [34] Application Development System Reference Manual, Motorola Inc.