

Este exemplar corresponde a redação final da tese
defendida por Lino Rosell Valdenebro
e aprovada pela Comissão
Julgada em 20 / 04 / 2002
[Assinatura]
Orientador

UNICAMP

BIBLIOTECA CENTRAL

SECÃO CIRCULANTE

OBSERVADORES ADAPTATIVOS DE FLUXO E VELOCIDADE
PARA MOTORES DE INDUÇÃO – ESTUDO E IMPLEMENTAÇÃO

Autor: *Lino Rosell Valdenebro*

Banca Examinadora:

Prof. Dr. Edson Bim - Orientador – FEEC – UNICAMP

Prof. Dr. Benjamin Rodrigues de Menezes – CPDEE - UFMG

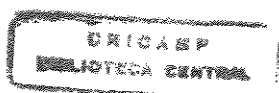
Prof. Dr. Ivan Eduardo Chabu – PEA – USP

Prof. Dr. Paulo Augusto Valente Ferreira – FEEC – UNICAMP

Prof. Dr. Akebo Yamakami – FEEC – UNICAMP

Tese 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 Doutor em Engenharia Elétrica.

abril/2001



N.º CHAMADA :
T/ UNICAMP
R7340
V. Ex.
TOMBO BC/ 45733
PROC. 16-392/01
C D
PREÇO R\$ 11,00
DATA 04/08/01
N.º CPD

CM0015854B-5

BR10-220231

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

R734o Rosell Valdenebro, Lino
Observadores adaptativos de fluxo e velocidade para
motores de indução – estudo e implementação / Lino
Rosell Valdenebro. --Campinas, SP: [s.n.], 2001.

Orientador: Edson Bim.

Tese (doutorado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

1. Motores elétricos de indução. 2. Sistemas difusos.
3. Algoritmos genéticos. 4. Sistemas de controle
ajustável. 5. Sistemas de controle digital. I. Bim, Edson
II. Universidade Estadual de Campinas. Faculdade de
Engenharia Elétrica e de Computação. III. Título.

*À minha família,
Em especial minha esposa Mariza,
meus avós Mirta e Agustin, minha mãe Mirtha,
minha irmã Leysi e a pequena Saylin.*

Agradecimentos

Ao Prof. Dr. Edson Bim pela orientação, oportunidade de realizar este trabalho, apoio, incentivos e confiança em mim depositada.

Aos professores Giovanni Griva e Francesco Profumo do Politecnico di Torino – Itália, pela experiência transmitida.

Ao amigo Jaime Reyes Hernández pela ajuda na implementação e testes do sistema de controle. Além disso, pela amizade e incentivos oferecidos.

Ao amigo Zanoni Dueire Lins pela amizade, importante colaboração com meu trabalho, incentivo e conselhos sempre oportunos.

Aos amigos José Alberto Torrico Altuna e Leonardo Araújo pelas sugestões e companheirismo.

Aos amigos Evelio, Rosa, Edgardo e Victor, pela amizade.

Ao amigo Prof. Raúl Diaz del Mazo e sua esposa e amiga Profa. Elsa León, ambos da Universidade de Oriente, pelo apoio e amizade.

À Ju e Dani pelo carinho.

À querida Mariza, inspiração de todas as horas...

À Faculdade de Engenharia Elétrica e de Computação, em especial ao Departamento de Sistemas de Controle e Energia, pela oportunidade dada à realização do curso de Doutorado.

À FAPESP pelo suporte financeiro.

SUMÁRIO

| | |
|--|------|
| Resumo | XI |
| Abstract | XIII |
| Índice de Figuras | XV |
| Índice de Tabelas | XIX |
| Nomenclatura | XXI |
| | |
| Capítulo 1 Introdução | 1 |
| | |
| Capítulo 2 Breve revisão de observadores de fluxo em máquinas de indução | 5 |
| 2.1 Introdução | 5 |
| 2.2 Estimação de fluxo com base no observador de Luenberger | 7 |
| 2.3 Observador de fluxo com base no filtro de Kalman | 9 |
| 2.4 Observador de fluxo com base em modos deslizantes | 11 |
| 2.5 Considerações finais | 13 |
| | |
| Capítulo 3 Observador adaptativo de fluxo de rotor utilizando filtro de Kalman | 15 |
| 3.1 Introdução | 15 |
| 3.2 Modelo matemático discretizado do motor de indução para aplicação do algoritmo EKF | 16 |
| 3.3 Implementação do algoritmo EKF discreto | 22 |
| 3.4 Resultados Experimentais | 26 |
| 3.4.1 Testes em regime permanente | 28 |
| 3.4.2 Testes em regime transitório | 35 |

| | |
|--|----|
| Capítulo 4 Observador adaptativo de Luenberger para estimação do fluxo do rotor | 41 |
| 4.1 Introdução | 41 |
| 4.2 Modelo matemático do motor de indução para implementação do observador de Luenberger | 43 |
| 4.3 Formulação matemática do observador de Luenberger no domínio contínuo do tempo | 44 |
| 4.3.1 Critérios de projeto | 47 |
| 4.4 Esquema adaptativo para estimação de velocidade | 50 |
| 4.5 Projeto do observador de Luenberger no domínio discreto do tempo | 53 |
| 4.6 Otimização do observador de Luenberger utilizando algoritmos genéticos | 55 |
| 4.6.1 Formulação do problema de otimização | 56 |
| 4.6.2 Breve revisão de algoritmos genéticos (GA) | 57 |
| 4.6.3 Solução ao problema de otimização utilizando algoritmos genéticos | 60 |
| 4.7 Resultados Experimentais | 61 |
| 4.7.1 Testes em regime permanente | 63 |
| 4.7.2 Testes em regime transitório | 70 |
| | |
| Capítulo 5 Observador adaptativo de fluxo de rotor em modos deslizantes | 77 |
| 5.1 Introdução | 77 |
| 5.2 Breve histórico de sistemas de estrutura variável | 79 |
| 5.3 Definições básicas dos SEV | 80 |
| 5.4 Observadores de estado em modos deslizantes | 81 |
| 5.5 Observador de fluxo de rotor em modos deslizantes | 84 |
| 5.5.1 Modelo matemático do observador | 84 |
| 5.5.2 Esquema adaptativo para estimação de velocidade | 87 |
| 5.5.3 Projeto do observador | 88 |
| 5.6 Redução de <i>chattering</i> utilizando Lógica Fuzzy | 91 |

| | |
|--|------------|
| 5.7 Resultados Experimentais | 95 |
| 5.7.1 Testes em regime permanente | 95 |
| 5.7.2 Testes em regime transitório | 103 |
| | |
| Capítulo 6 Conclusões | 109 |
| | |
| Apêndice A Dados do motor de indução | 113 |
| | |
| Apêndice B Fundamentos teóricos dos sistemas de estrutura variável em modos deslizantes | 115 |
| | |
| Apêndice C Descrição do sistema implementado | 127 |
| | |
| Apêndice D Programa do sistema de controle por orientação direta do fluxo de rotor | 149 |
| | |
| Referências Bibliográficas | 185 |

RESUMO

Neste trabalho é apresentado o estudo e a implementação digital de observadores adaptativos de fluxo e velocidade para o controle por orientação do fluxo do rotor em motores de indução. Os observadores com base no filtro de Kalman estendido, de Luenberger e em modos deslizantes são estudados. Uma nova abordagem é apresentada para obter a matriz de ganho do observador de Luenberger utilizando algoritmos genéticos. O algoritmo genético minimiza uma função de custo relacionada com a resposta dinâmica do erro entre os valores estimados e reais das variáveis de estado. Um algoritmo com base em lógica fuzzy para redução de *chattering* é utilizado para o observador de fluxo de rotor em modos deslizantes. O método de controle por orientação direta do fluxo de rotor utilizando os observadores estudados é implementado num sistema com base no Processador Digital de Sinais (DSP) TMS320F240 da Texas Instruments. O desempenho dinâmico do sistema de controle é avaliado através de resultados experimentais.

ABSTRACT

This work presents a study and the digital implementation of adaptive flux and velocity observers for the rotor flux field oriented based control of induction motors. The extended Kalman filter, Luenberger and the sliding mode based observers are studied. A new approach is presented to obtain the Luenberger observer gain matrix using genetic algorithms. The genetic algorithm minimizes a cost function related to the dynamic response of the error between the estimated state variables and their real values. A fuzzy logic based algorithm for chattering reduction is used for the sliding mode rotor flux observer. The direct rotor flux orientation control method using the studied observers is implemented on a Texas Instruments TMS320F240 Digital Signal Processor (DSP) based system. The dynamic performance of the control system is evaluated by experimental results.

ÍNDICE DE FIGURAS

| Figura | Descrição | Página |
|--------------------|--|--------|
| Figura 3.1 | Trajatória resultante dos pólos da matriz discreta A_d utilizando (1) aproximação de primeira ordem (2) aproximação de segunda ordem (3) função expm. (a) 0-10.000 rpm (b) 0-4.000 rpm (c) 0-1.800 rpm | 20 |
| Figura 3.2 | Diagrama de blocos do algoritmo EKF | 22 |
| Figura 3.3 | Diagrama de blocos do sistema de controle utilizando o algoritmo EKF | 27 |
| Figura 3.4 | Correntes reais e estimadas para $\omega_r^* = 70$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 29 |
| Figura 3.5 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 70$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 30 |
| Figura 3.6 | Correntes reais e estimadas para $\omega_r^* = 450$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 31 |
| Figura 3.7 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 450$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 32 |
| Figura 3.8 | Correntes reais e estimadas para $\omega_r^* = 1.000$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 33 |
| Figura 3.9 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 1.000$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 34 |
| Figura 3.10 | Reversão de velocidade de 1.000 rpm a -1.000 rpm. (a) $\omega_r^*, \omega_r, \hat{\omega}_r$ (b) ω_r^*, θ_r | 36 |
| Figura 3.11 | Teste de onda quadrada na referencia de velocidade de 20 a 100 rpm (a) $\omega_r^*, \omega_r, \hat{\omega}_r$ (b) instante diferente do teste | 37 |
| Figura 3.12 | Teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $\omega_r, \hat{\omega}_r$ (2.5 s/div) $\omega_r, \hat{\omega}_r$ (1 s/div) | 38 |
| Figura 3.13 | Correntes reais e estimadas para o teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 39 |

| | | |
|--------------------|--|----|
| Figura 3.14 | Teste de onda quadrada na referencia i_{sq}^* de 0.5 a -0.5 pu. | 40 |
| Figura 4.1 | Diagrama de blocos da estimação de velocidade com base em MRAS. | 52 |
| Figura 4.2 | Trajatórias dos pólos do motor (1) e do observador de Luenberger (2) utilizando aproximação de primeira ordem no cálculo das matrizes discretas: 0-2.000 rad/sec. | 54 |
| Figura 4.3 | Trajatórias dos pólos do motor (1) e do observador de Luenberger (2) utilizando aproximação de segunda ordem no calculo das matrizes discretas: 0-2.000 rad/sec. | 55 |
| Figura 4.4 | Operador de cruzamento. | 60 |
| Figura 4.5 | Operador de mutação. | 60 |
| Figura 4.6 | Diagrama de blocos do sistema de controle utilizando o observador de Luenberger. | 62 |
| Figura 4.7 | Correntes reais e estimadas para $\omega_r^* = 70$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 64 |
| Figura 4.8 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 70$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 65 |
| Figura 4.9 | Correntes reais e estimadas para $\omega_r^* = 900$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 66 |
| Figura 4.10 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 900$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 67 |
| Figura 4.11 | Correntes reais e estimadas para $\omega_r^* = 1.100$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 68 |
| Figura 4.12 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 1.100$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 69 |
| Figura 4.13 | Reversão de velocidade de -900 rpm a 900 rpm (a) $\omega_r^*, \omega_r, \hat{\omega}_r$ (b) ω_r^*, θ_r | 71 |
| Figura 4.14 | Teste de aceleração desaceleração de -900 rpm a 900 rpm (a) $\omega_r, \hat{\omega}_r$ (2.5 s/div) (b) $\omega_r, \hat{\omega}_r$ (1 s/div) | 72 |
| Figura 4.15 | Correntes reais e estimadas para o teste de aceleração desaceleração de 900 a -900 (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 73 |

| | | |
|--------------------|--|-----|
| Figura 4.16 | Teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $\omega_r, \hat{\omega}_r$ (2.5 s/div) (b) $\omega_r, \hat{\omega}_r$ (1 s/div) | 74 |
| Figura 4.17 | Correntes reais e estimadas para o teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 75 |
| Figura 4.18 | Teste de onda quadrada na referencia i_{sq}^* de 0.5 a -0.5 pu | 76 |
| Figura 5.1 | Trajatória dos pólos do motor para diferentes valores dos parâmetros q e γ . | 90 |
| Figura 5.2 | Funções de pertinência das variáveis de entrada. (a) erro (b) derivada do erro | 93 |
| Figura 5.3 | Superfície do regulador. | 94 |
| Figura 5.4 | Diagrama de blocos do sistema de controle utilizando o observador em modos deslizantes. | 96 |
| Figura 5.5 | Correntes reais e estimadas para $\omega_r^* = 100$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 97 |
| Figura 5.6 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 100$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 98 |
| Figura 5.7 | Correntes reais e estimadas para $\omega_r^* = 200$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 99 |
| Figura 5.8 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 200$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 100 |
| Figura 5.9 | Correntes reais e estimadas para $\omega_r^* = 900$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$ | 101 |
| Figura 5.10 | Componentes estimadas do fluxo de rotor para $\omega_r^* = 900$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$ | 102 |
| Figura 5.11 | Onda quadrada na referência de velocidade. (a) ω_r^* de 0 a 1.260 rpm (b) ω_r^* de -100 a 100 rpm | 105 |
| Figura 5.12 | Onda quadrada na referência de velocidade. (a) ω_r^* de -50 a 50 rpm (b) ω_r^* de -20 a 20 rpm | 106 |

| | | |
|--------------------|---|-----|
| Figura 5.13 | Onda quadrada na referência de velocidade de 200 a -200 rpm (a) sem controle de <i>chattering</i> (b) com controle de <i>chattering</i> . | 107 |
| Figura 5.14 | Teste de aceleração desaceleração de -900 a 900 rpm | 108 |
| Figura 5.15 | Dinâmica do sistema com aplicação de carga (0.45 pu) para $\omega_r=1.100$ rpm. | 108 |
| Figura B.1 | Planos de fase para o sistema definido por B.3 e B.6. | 117 |
| Figura B.2 | Planos de fase para o sistema definido por B.3 e B.9. | 119 |
| Figura B.3 | Plano de fase para o SEV em modos deslizantes. | 120 |
| Figura B.4 | Interpretação geométrica do princípio de Filippov. | 124 |
| Figura C.1 | Diagrama do TMS320C24x Evaluation Module (C24x EVM). | 128 |
| Figura C.2 | Arquitetura do DSP TMS320F240. | 129 |
| Figura C.3 | Comunicação Computador-C24x EVM. | 130 |
| Figura C.4 | Representação da variável ou parâmetro no formato 4.12f. | 131 |
| Figura C.5 | Circuito de acondicionamento dos sinais de corrente. | 132 |
| Figura C.6 | Representação dos sinais de corrente antes do escalamento. | 133 |
| Figura C.7 | Representação da variável ou parâmetro no formato 8.8f. | 135 |
| Figura C.8 | Representação dos vetores fixos de tensão para as diferentes combinações de estados das chaves no plano α,β . | 137 |
| Figura C.9 | Sinais de PWM e tempos de serviço para vetor de referência no setor 3. | 141 |
| Figura C.10 | Transformação (a,b,c) \rightarrow (α,β). (a) $i_{s\alpha}$, $i_{s\beta}$ (b) $i_{s\alpha}$ vs $i_{s\beta}$ | 142 |

ÍNDICE DE TABELAS

| Tabela | Descrição | Página |
|-------------------|--|---------------|
| Tabela 4.1 | Codificação binária. | 58 |
| Tabela 4.2 | Operador de reprodução. | 59 |
| Tabela 4.3 | Parâmetros de otimização. | 61 |
| Tabela 5.1 | Regras de inferência do regulador. | 94 |
| Tabela C.1 | Definição dos tempos para os sinais de PWM | 139 |
| Tabela C.2 | Registradores de comparação em função do setor e do tempo de serviço. | 140 |
| Tabela C.3 | Representação dos coeficientes das matrizes do modelo discreto em 1.15f. | 148 |

NOMENCLATURA

| Abreviaturas | Descrição |
|---------------------|-------------------------------|
| CA | Corrente alternada |
| CC | Corrente contínua |
| EKF | Filtro de Kalman estendido |
| SEV | Sistema de estrutura variável |
| GA | Algoritmo genético |

| Simbologia | Descrição |
|-------------------|-----------------------------------|
| v | Valor instantâneo da tensão (V) |
| V | Valor máximo da tensão (V) |
| i | Valor instantâneo da corrente (A) |
| I | Valor máximo da corrente (A) |
| λ | Fluxo magnético (wb) |
| ω | Velocidade angular (rad/s) |
| P | Número de pólos |
| L | Indutância (H) |
| R | Resistência (Ω) |
| X | Reatância (Ω) |
| σ | Constante de dispersão |

| | |
|-------------------|---|
| θ_e | Ângulo elétrico do vetor de fluxo (rad) |
| $s(\mathbf{x},t)$ | Superfície de chaveamento |
| $u(\mathbf{x},t)$ | Vetor de controle |
| T | Constante de tempo (s) |
| t | Tempo (s) |
| \mathbf{z} | Vetor sinal de chaveamento |
| \mathbf{I}_n | Matriz identidade de ordem n |

| Subscritos | Descrição |
|-------------------|---|
| abc | Relativo às fases do motor |
| α,β | Relativo ao sistema de referência estacionário |
| d,q | Relativo ao sistema de referência síncrono |
| r | Relativo ao rotor |
| s | Relativo ao estator |
| m | Relativo ao ramo de magnetização |
| max | Referente ao valor máximo da função relacionada |
| min | Referente ao valor mínimo da função relacionada |

| Sobrescritos | Descrição |
|---------------------|---|
| * | Símbolo indicativo de valores de referência |
| ^ | Símbolo indicativo de quantidades estimadas |
| . | Símbolo indicativo do operador derivada |

Capítulo 1

Introdução

O surgimento da teoria de controle por orientação de campo [1], teve sem dúvida, extraordinária repercussão no controle de acionamentos elétricos, que utilizam como elemento de conversão eletromecânica de energia o motor de corrente alternada. O princípio da orientação de campo consiste em conhecer a posição espacial do vetor de fluxo escolhido (de estator, de rotor ou de entreferro) e, a partir desta informação, alinhar o eixo direto do sistema de referência síncrono com o vetor espacial do fluxo em questão. Com este procedimento obtém-se um modelo para o motor de indução em que as variáveis de fluxo e torque estão desacopladas ou fracamente acopladas.

O sucesso da orientação depende da determinação exata do vetor de fluxo, isto é, a amplitude e posição espacial. Além disso, quando o objetivo é maximizar a eficiência ajustando o fluxo da máquina, o conhecimento desta variável é imprescindível para a operação satisfatória do algoritmo de controle. Essa informação pode ser obtida através de medições diretas na máquina (bobinas exploratórias) ou calculada a partir dos valores de corrente e tensão (estimador de fluxo). Um outro método consiste em estimar a posição espacial deste vetor, utilizando a informação de velocidade, as constantes do motor e sinais de referência gerados pelo controle.

A utilização de bobinas exploratórias para implementação do método direto de orientação tem custo elevado e é impraticável na maior parte das aplicações. Quando este for o caso, a utilização de um estimador de fluxo torna-se necessária. Os estimadores de fluxo podem ser implementados em malha aberta ou fechada, sendo que a diferença entre eles está na existência do termo de correção associado ao erro de estimação, que é utilizado para ajustar a resposta dinâmica do estimador. Geralmente, o estimador em malha fechada recebe

a denominação de observador. Dois observadores são, geralmente, utilizados na estimação do vetor de fluxo: o observador com base no filtro de Kalman e o observador de Luenberger.

O observador de estado baseado no filtro de Kalman é considerado um estimador de natureza estocástica, dado que o algoritmo leva em consideração o conhecimento estatístico dos sinais de ruído de estado do sistema e de medição. A implementação é feita através de um algoritmo recursivo de elevada complexidade computacional, por causa do grande número de operações matriciais envolvidas, o que dificulta a implementação digital. Além disso, as características dos sinais de ruído necessárias para a estimação são difíceis de se obter do ponto de vista prático, tornando difícil a sintonia do observador.

Por sua vez, o observador de Luenberger é baseado somente nas equações do modelo matemático da planta e o custo computacional não é elevado, quando comparado com o estimador com base no filtro de Kalman. A dinâmica da estimação de estado utilizando o observador de Luenberger pode ser controlada através da seleção adequada da matriz de ganho do observador. Esta seleção deve representar uma solução de compromisso entre uma boa resposta dinâmica e baixa sensibilidade às perturbações e ruídos do sistema.

Outro aspecto importante no controle de motores elétricos é o projeto e a implementação de reguladores robustos com o objetivo de diminuir os efeitos negativos de perturbações externas, e das incertezas paramétricas do modelo. Neste sentido, os avanços obtidos na área de controle não linear têm sido aproveitados convenientemente na área de acionamentos elétricos. Entre estas técnicas, o controle com base em modos deslizantes (*Sliding Mode Control (SMC)*) tem recebido especial atenção, dada sua característica de alta robustez ante variações de parâmetros, assim como da baixa sensibilidade a ruídos no sistema. O problema encontrado na estimação de fluxo em motores de indução é um exemplo típico que pode ser solucionado através da aplicação destas técnicas.

Quando a estimação de estado é feita com base em SMC, o termo de correção associado ao erro de estimação é não linear. O algoritmo deve forçar a trajetória do sistema a seguir a superfície de comutação previamente projetada. Para isto, a matriz de realimentação de estado do observador deve ser selecionada de forma a garantir as condições de existência do modo deslizante. Na condição ideal, o projeto conduz a oscilações com frequência infinita

na vizinhança na superfície de comutação. Nos sistemas reais, esta situação não é possível. Um dos motivos é a existência de atrasos finitos de tempo relacionados com o cálculo dos algoritmos de controle em tempo real. Outro motivo não menos importante, são as limitações físicas do mecanismo de atuação encarregado de executar a lei de controle. No regime estacionário, estas limitações levam às oscilações de alta frequência na vizinhança do ponto de equilíbrio, sendo este fenômeno conhecido como *chattering*. A presença de *chattering* pode servir como fonte para excitar dinâmicas de alta frequência não modeladas, além de desviar o sistema da condição desejada de projeto.

Além da determinação do vetor de fluxo, outra variável de grande importância no controle do motor de indução é a velocidade mecânica. Esta é utilizada no sistema de controle por orientação para gerar a componente de corrente de estator associada à produção de conjugado eletromagnético, a partir do erro entre a velocidade de referência e a real do motor. Além disso, os observadores de fluxo utilizam as equações do modelo matemático do motor, as quais por sua vez têm como parâmetro a velocidade. Em alguns casos não é possível instalar um sensor de velocidade, ou porque o ambiente de operação não permite ou por limitações econômicas do projeto em questão. Nestes casos é necessária a existência de métodos de estimação adequados para a velocidade, de forma a oferecer esta informação ao algoritmo de controle. Uma das soluções é o uso de algoritmos de estimação com base em Modelo de Referência (MRAS) aproveitando o observador de fluxo já projetado.

Neste trabalho são apresentados um estudo e implementação digital de observadores adaptativos de fluxo de rotor e velocidade para o motor de indução. O desempenho de cada um dos observadores é validado a partir dos resultados experimentais obtidos num sistema de controle por orientação direta do fluxo do rotor. O hardware de controle utilizado tem como base o Processador Digital de Sinais (DSP) TMS320F240 da Texas Instruments.

No capítulo 2 é feita uma breve revisão bibliográfica dos principais trabalhos na área de observadores adaptativos de fluxo e velocidade, orientada principalmente para os seguintes tópicos: critérios de projeto, métodos de adaptação de parâmetros e algoritmos de estimação de velocidade.

Os fundamentos teóricos do observador de fluxo com base no filtro de Kalman são apresentados no capítulo 3. Os detalhes relacionados com a discretização do modelo e a implementação do algoritmo de estimação de fluxo e velocidade são discutidos nas seções deste capítulo. O desempenho do observador é verificado através de resultados experimentais.

A estimação do fluxo e velocidade utilizando o observador de Luenberger é apresentada no capítulo 4. Um estudo dos diferentes critérios de projeto, assim como também uma análise da estabilidade do modelo discreto são apresentados e na seqüência, um novo projeto do observador é proposto. O novo projeto tem como base a minimização de uma função de custo relacionada com a dinâmica do erro de estimação. Para resolver este problema de otimização foi utilizado um algoritmo genético e como resultado, a matriz ótima de retração de estado do observador é obtida. Resultados experimentais são apresentados para validar o método proposto.

O capítulo 5 é dedicado ao estudo dos observadores de fluxo e de velocidade em modos deslizantes. É realizado um breve estudo dos fundamentos teóricos desta técnica de controle, sendo apresentada a aplicação desta no problema de estimação de fluxo. Uma nova abordagem para a redução de *chattering* com base em Lógica Fuzzy é estudada neste capítulo. O desempenho do observador é avaliado através de resultados experimentais.

No capítulo 6 são apresentadas as conclusões sobre os resultados obtidos, como também os tópicos que poderão ser objetos de estudos futuros.

Capítulo 2

Breve revisão de observadores de fluxo em máquinas de indução

2.1 Introdução

A literatura especializada na área de estimação de fluxo e velocidade para o motor de indução tem apresentado importantes contribuições nos últimos anos. Excelentes revisões destes temas podem ser encontradas em [2,3,4]. Entre as diversas abordagens propostas, os estimadores de fluxo e velocidade em malha aberta com base em modelo de referência (MRAS) e aqueles que utilizam observadores têm sido os mais utilizados.

Dados os objetivos de nossa pesquisa, neste capítulo é feita uma breve descrição dos principais trabalhos que utilizam observadores na estimação de fluxo e de velocidade do motor de indução. Os estimadores em malha aberta são utilizados freqüentemente, dada a sua simplicidade e facilidade de implementação. Em contrapartida, eles apresentam alguns problemas principalmente na faixa de baixas velocidades. Isso ocorre por problemas associados à integração em malha aberta, os quais podem levar à instabilidade do sistema. Uma análise da estabilidade desse tipo de estimador pode ser encontrada no trabalho de Giménez et al. [5] e algumas referências de sua aplicação aparecem em [4,5,6,7,8].

Os primeiros trabalhos na aplicação da teoria de observadores aplicados na estimação de fluxo em máquinas de indução podem ser encontrados nas publicações de Bellini et al. [9,10]. Paralelamente a estes trabalhos, os estudos realizados por Verghese e Sanders [11,12], tiveram grande repercussão no desenvolvimento de observadores de fluxo aplicados

às máquinas de indução. Em primeiro lugar, foi apresentada uma metodologia de análise e de projeto do observador de fluxo, que poderia ser estendida a outros problemas de estimação e controle em máquinas elétricas. Além disso, através de uma linguagem matemática bem compreensível, foi feita uma análise crítica de diferentes esquemas de estimação de fluxo. Foram apresentadas propostas para evitar a utilização de termos diferenciais nos observadores de ordem reduzida iniciais, assim como uma análise dos efeitos de incerteza nos parâmetros do motor e erros de medição na dinâmica do observador.

Em decorrência do trabalho de Verghese e Sanders, o número de publicações relacionadas a observadores de fluxo do motor de indução cresceu de maneira considerável. Os principais pontos abordados nestes estudos podem ser resumidos nos seguintes tópicos:

- Métodos de projeto do observador de fluxo. A preocupação é a obtenção da matriz de realimentação de estado do observador a partir de um dado critério de projeto. Essa seleção deve representar uma solução de compromisso entre uma boa resposta dinâmica e baixa sensibilidade às perturbações e ruídos no sistema.
- Robustez do observador ante incertezas nos parâmetros. Com este objetivo duas abordagens são consideradas. A primeira contempla o efeito negativo das perturbações dentro do critério de projeto, tentando minimizar um índice de desempenho adequado. A segunda abordagem preocupa-se, inicialmente, em elaborar algoritmos de identificação de parâmetros, para depois atualizar as matrizes do observador.
- Estimação de velocidade. Os estudos são direcionados ao projeto de algoritmos de estimação de velocidade para operar em sincronismo com o observador de fluxo. Neste sentido, a convergência do algoritmo de estimação de velocidade é provada através do método de Lyapunov. A estimação é feita utilizando a técnica de modelo de referência (MRAS), na qual o modelo de referência são as equações do motor e o modelo ajustável é o próprio observador.
- Implementação do observador de fluxo. Nestes estudos, especial atenção é dada aos métodos de discretização do modelo do motor e à estabilidade da estimação. Não menos

importantes são os esforços destinados a minimizar o tempo de execução do algoritmo discreto e à seleção do processador digital utilizado na implementação.

2.2 Estimação de fluxo com base no observador de Luenberger

O observador de Luenberger tem sido utilizado freqüentemente na estimação de fluxo em máquinas de indução. O principal motivo disto é o baixo custo computacional de sua implementação e a estabilidade deste observador. Além disso, a sintonia deste observador é facilitada pela sua própria natureza determinística, a qual permite a utilização de técnicas clássicas de alocação de pólos. Alguns dos principais trabalhos que utilizam este tipo de observador na estimação de fluxo e de velocidade do motor de indução são comentados a seguir.

No trabalho de Kubota et al. [14] é proposto a utilização de um observador de Luenberger na estimação de fluxo de rotor, como também é apresentado um esquema adaptativo de identificação de parâmetros que opera em sincronismo com o observador. O algoritmo de identificação estima os valores da resistência do estator e da constante de tempo do rotor, os quais são utilizados para atualizar as matrizes do observador. Além disso, a obtenção da matriz de realimentação de estado utiliza o critério de proporcionalidade entre os pólos do motor e do observador para cada ponto de velocidade. O método proposto é validado através de resultados de simulação num sistema de controle por orientação direta do fluxo do rotor. Exemplos da aplicação da técnica de projeto proposta por estes autores podem ser encontrados nos trabalhos de Yang et al. [15] e Sousa et al. [16].

Resultados da implementação, utilizando DSP, do observador de fluxo proposto em [14] são apresentados por Kubota et al. em [17]. Neste trabalho, além do observador e os métodos de identificação de parâmetros, é implementado um esquema adaptativo para estimação de velocidade. O algoritmo de estimação é obtido com base no método de Lyapunov. Os problemas associados à estimação simultânea da velocidade mecânica e da resistência do rotor levaram os autores a propor um novo método de estimação, o qual é

apresentado em [18]. O objetivo é separar o erro de estimação de velocidade do erro na resistência de rotor, a partir dos valores em estado estacionário das variáveis de estator. O método propõe a imposição de uma componente alternada na corrente de campo i_{sd} , sendo a frequência desta componente diferente da fundamental. O método de estimação é verificado a partir de resultados de simulação e de experimentos.

Uma nova metodologia para a obtenção do modelo matemático discreto do motor de indução é apresentada no trabalho de Bottura et al. [19]. O modelo discreto resultante é obtido a partir da discretização parcial da equação de estado do sistema. O desempenho de um observador de fluxo de rotor de ordem reduzida, que utiliza o modelo discreto proposto, é analisado. A partir dos resultados de simulação obtidos os autores mostram a melhora da precisão e estabilidade do observador, com base no método proposto, quando comparado com aqueles que utilizam o método de Euler.

O Observador de Luenberger Tipo Fuzzy (FLO) proposto no trabalho de Profumo et al. [20] resolve de maneira efetiva os problemas de instabilidade associados à implementação no tempo discreto, permitindo a operação em uma faixa maior de velocidade do que aquelas permitidas na implementação tradicional do Observador de Luenberger. A implementação do FLO utiliza um conjunto de matrizes discretas do observador calculadas para diferentes pontos de velocidade. O cálculo destas matrizes é realizado de maneira numérica sem aproximação. Para uma dada velocidade mecânica, o cálculo das equações do observador é feito através da interpolação entre os dois conjuntos de matrizes discretas adjacentes no ponto de velocidade considerado. O desempenho satisfatório deste observador é verificado pelos autores através de resultados experimentais obtidos, de um sistema com Controle Direto de Torque (DTC), operando com uma velocidade máxima de 12.000 rpm. Uma análise da estabilidade do observador discreto de Luenberger, quando utilizados métodos de discretização de primeira, segunda, quarta ordem e do observador FLO é apresentada pelos autores em [21].

Um observador adaptativo de fluxo de estator e de rotor é apresentado no trabalho de Maes e Melkebeek [22]. A obtenção das equações do observador é feita a partir do modelo

do motor no sistema de referência que gira com a velocidade do rotor. O mecanismo de estimação de velocidade é similar àquele adotado no trabalho de Kubota et al. [17]. Com o objetivo de melhorar a estabilidade e a velocidade de convergência da estimação, o ângulo de rotação, utilizado na transformação entre o sistema estacionário e girante, é atualizado com o termo de correção utilizado na identificação da velocidade. Um novo critério para a obtenção da matriz de ganho do observador é proposto, no qual os pólos do observador são deslocados à esquerda em relação aos pólos do motor. Os autores comentam as vantagens desse método, nas faixas de baixas e altas velocidades, quando comparado com o critério proporcional. Resultados experimentais são obtidos a partir da operação do observador num sistema de controle DTC, sendo verificado o bom desempenho do método proposto.

O observador de fluxo apresentado por Maes e Melkebeek em [22] é melhorado por estes autores através da incorporação da equação eletromecânica do motor nas equações do observador. Os resultados deste estudo são apresentados em [23]. Considerando constante o conjugado de carga, os autores propõem um método para a identificação desta variável, que é a variável de entrada nas equações de estimação da velocidade. Através de resultados de simulação e experimentais os autores mostram o melhor desempenho dinâmico do sistema de controle, tanto no transitório como em regime, quando comparado com os obtidos em [22]. O observador adaptativo de fluxo resultante é similar àquele proposto por Hoffman e Sanders em [24].

2.3 Observador de fluxo com base no filtro de Kalman

Quando o objetivo é diminuir os erros de estimação de fluxo e de velocidade provenientes de ruídos no sistema, o observador com base em filtro de Kalman é indicado. O filtro de Kalman é um algoritmo recursivo de uma elevada complexidade computacional quando comparado com outros métodos de estimação, dado o grande número de operações matriciais envolvidas, o qual dificulta a implementação em tempo real. Além disso, as equações do filtro têm como base as estatísticas dos sinais de ruído de estado e de medição,

as quais do ponto de vista prático são difíceis de se obter. Esses problemas são abordados convenientemente na literatura como é comentado a seguir.

Henneberger et al. [25] apresentam um estimador de velocidade com base no filtro de Kalman estendido (EKF). O estimador proposto utiliza o modelo do motor tendo como variáveis de estado somente as correntes de estator e a corrente de magnetização, sendo esta última admitida constante.

Resultados da implementação de um observador para a estimação simultânea do fluxo de rotor e da velocidade do motor de indução utilizando o algoritmo EKF, são apresentados no trabalho de Kim et al. [26], que utiliza o modelo de quinta ordem para o motor ao invés do modelo de quarta ordem adotado por Henneberger et al. em [25]. Como variáveis de estado foram consideradas as componentes da corrente de estator e do fluxo do rotor, além da velocidade mecânica do motor. O observador proposto foi aplicado em um sistema por orientação direta do fluxo do rotor, mostrando o desempenho do observador na faixa de baixas e altas velocidades. Por outra parte, na implementação do algoritmo de estimação é utilizada somente a informação referente às correntes de estator e a tensão DC na entrada do inversor.

A utilização do filtro de Kalman para a estimação do fluxo do rotor é reportada no trabalho de Raumer et al. [27]. A informação de fluxo é utilizada na implementação de um controle não linear de fluxo e de conjugado eletromagnético com base em técnicas de linearização entrada-saída. Neste trabalho, aspectos práticos da implementação em tempo real da lei não-linear de controle são discutidos. Além disto, são abordados problemas relacionados com a observação de fluxo e limitações da corrente de estator. Como principal conclusão do trabalho, os autores colocam a potencialidade dos Processadores Digitais de Sinais (DSPs) na implementação dos algoritmos não-lineares de controle, além de mostrar o melhora no desempenho dinâmico do algoritmo de controle proposto quando comparado com o método de orientação pelo campo convencional.

No trabalho de Hilairet et al. [29], a principal preocupação é diminuir a complexidade computacional do algoritmo do filtro de Kalman na estimação do fluxo do rotor. Primeiramente, a velocidade do motor é considerada conhecida e o modelo de quarta ordem

do motor de indução é utilizado na estimação. Aproveitando as características de simetria deste modelo, os autores concluem como o tempo de cálculo das operações matriciais envolvidas no algoritmo pode ser reduzido de maneira considerável. Além disso, é apresentado um método para a sintonia do filtro. O método obtém os valores da matriz de covariância do ruído de estado a partir da minimização do erro medio quadrático do vetor de fluxo, enquanto a matriz de covariância de ruído de medição é considerada igual à matriz identidade. A seguir é considerada a estimação simultânea de fluxo e velocidade com base no filtro de Kalman estendido. Neste caso, os autores utilizam um modelo de sexta ordem, através da adição de uma variável de estado artificial ao modelo de quinta ordem original. Isso é feito com o objetivo de preservar as características de simetria nas matrizes de estado do sistema, facilitando a implementação do algoritmo. Novamente, os valores da matriz de covariância de ruído de estado do sistema são obtidos a partir de um processo de otimização, sendo a função de custo valor meio da velocidade estimada. O desempenho do observador é avaliado através de resultados de simulação.

Uma outra abordagem para minimizar o tempo de cálculo do algoritmo de EKF, preservando as características de robustez da estimação ante ruídos no sistema é a combinação deste com um observador adaptativo como proposto no trabalho de Lee e Chen [28].

2.4 Observador de fluxo com base em modos deslizantes

A utilização da técnica de controle com base em modos deslizantes (*Sliding Mode Control (SMC)*) [35,36] tem recebido especial atenção, dada sua característica de alta robustez ante variações de parâmetros, assim como de baixa sensibilidade a ruídos no sistema. O problema de estimação de fluxo e velocidade do motor de indução não tem sido uma exceção na aplicação destas técnicas.

Os primeiros trabalhos a aplicarem esta técnica na estimação de fluxo de rotor em máquinas de indução podem ser encontrados nas publicações de Sangwongwanich et al.

[30,31,32]. Em [30] os autores apresentam uma análise detalhada do projeto do observador utilizando o critério de rejeição a perturbações e da velocidade de convergência do erro de estimação. O método de projeto utiliza a minimização da norma H_∞ da função de transferência entre o erro de estimação e a variável de perturbação, considerando como restrição o valor máximo da norma H_2 da função de transferência em malha fechada. Os resultados da otimização são apresentados considerando diferentes variáveis de perturbação. Entre os tipos de perturbações consideradas, encontram-se as variações nas resistências de estator e de rotor, assim como também na velocidade mecânica do motor. O método de projeto proposto pelos autores foi validado através de simulação computacional. Os resultados da implementação do observador de fluxo de rotor proposto utilizando DSP são apresentados pelos autores em [31], num sistema de controle por orientação de fluxo. Em [32] foi incorporado o mecanismo de estimação de velocidade ao observador implementado em [31]. Os resultados experimentais apresentados nesse trabalho mostram a robustez da estimação de fluxo e de velocidade ante variações nas resistências de estator e de rotor, assim como também do conjugado de carga.

A utilização de um observador de fluxo de rotor com base em modos deslizantes para um sistema de controle por orientação direta é reportada por Luis et al. [33], que utilizam o critério de projeto proposto por Sangwongwanich et al. [30]. A implementação do observador no domínio discreto do tempo é feita utilizando o método descrito em [19]. Os resultados experimentais evidenciam o comportamento satisfatório do sistema de controle na região de baixas velocidades, assim como durante o transitório. Além disso, os autores apresentam a diminuição do custo computacional da implementação do algoritmo discreto como uma das suas vantagens.

Benchaib et al. [34] fizeram um estudo comparativo entre diferentes métodos de controle do motor de indução: 1) controle por orientação pelo campo 2) controle não-linear utilizando a técnica de linearização entrada-saída e 3) controle não-linear baseado em modos deslizantes. Na estimação das componentes do vetor fluxo do rotor foi utilizado um observador baseado em SMC em cada um dos métodos. A comparação entre os diferentes métodos foi feita analisando a operação do sistema de controle ante variações nos parâmetros

do motor, como também para diferentes valores de velocidade (baixas e altas velocidades). A partir dos resultados experimentais obtidos, os autores concluem que o controle não linear com base em modos deslizantes possui uma menor sensibilidade ante as variações de parâmetros, assim como um comportamento dinâmico satisfatório para altas e baixas velocidades. Para minimizar os efeitos negativos de *chattering* inerentes ao controle baseado em SMC, a função não-linear de ganho do observador foi substituída por uma função linear com saturação.

2.5 Considerações finais

A maior parte dos trabalhos que utilizam o filtro de Kalman na estimação de fluxo e de velocidade preocupam-se principalmente em tentar diminuir o tempo de cálculo do algoritmo de estimação. Embora algumas regras gerais possam ser utilizadas para o ajuste dos elementos das matrizes de covariâncias do ruído de estado e de medição, a falta de conhecimento das características estatísticas destes sinais, dificulta a sintonia do observador. Em muitos casos, a preocupação pela redução no tempo de cálculo no algoritmo de Kalman é dada pela não disponibilidade de um hardware de controle adequado para a implementação. Atualmente, os problemas da complexidade computacional do algoritmo EKF têm sido minimizados pelo surgimento de processadores digitais de sinais com elevada capacidade de processamento, muitos dos quais com funções dedicadas ao controle digital de motores.

Já no caso do observador de Luenberger, os critérios de projeto estudados tentam posicionar os pólos do observador de forma a conseguir uma dinâmica mais rápida que a do motor, sem comprometer a robustez do observador diante dos ruídos no sistema. Exemplos destas técnicas de projeto comentadas na revisão são o critério proporcional e aquele que utiliza o deslocamento à esquerda em relação aos pólos do motor. Além disso, na maior parte dos trabalhos não é utilizado um índice de desempenho específico no projeto do observador. Dado que os projetos quase sempre são feitos no domínio contínuo do tempo, as condições de projeto não são mais válidas quando é feita a passagem ao domínio discreto, que

dependendo do método de discretização utilizado e da velocidade de operação do motor, podem levar a problemas de instabilidade no processo de estimação. Este aspecto é abordado no nosso trabalho, onde para o projeto do observador é proposta a minimização de uma função de custo associada à dinâmica do erro de estimação das variáveis de estado. Isso é feito a partir das equações do observador no domínio discreto do tempo.

Quando o observador de fluxo com base em modos deslizantes é utilizado, o projeto é facilitado pela redução na ordem do sistema original quando este encontra-se na superfície de modo deslizante. Assim, por exemplo, nos observadores de ordem total apresentados na seção 2.4, o projeto é realizado a partir do sistema de segunda ordem, tendo como variáveis de estado o erro nas componentes do vetor fluxo. O erro nas componentes da corrente de estator é considerado nulo, admitindo que o sistema encontra-se na condição de modo deslizante. O ganho do observador é selecionado suficientemente grande com o objetivo de garantir a existência do modo deslizante. Por outra parte, ganhos elevados levam ao aumento da amplitude das oscilações na vizinhança na superfície de comutação, afetando a estimação. Este fenômeno, denominado na literatura como *chattering*, tem sido abordado em outras aplicações de SMC. Os efeitos negativos ocasionados pela presença de *chattering* nos observadores de fluxo e de velocidade com base em modos deslizantes têm sido pouco estudados nos trabalhos revisados. Este aspecto é tratado em nosso trabalho, sendo proposto um algoritmo adaptativo para o ajuste do ganho do observador, operando em sincronismo com o observador de fluxo.

Capítulo 3

Observador Adaptativo de Fluxo utilizando Filtro de Kalman

3.1 Introdução

Um dos algoritmos utilizados com maior frequência para a estimação do estado de sistemas lineares e não-lineares baseia-se no filtro de Kalman (KF). Na sua forma básica, o filtro de Kalman pode ser aplicado somente em sistemas lineares. No caso de sistemas não-lineares, o algoritmo de estimação utilizado é conhecido como filtro de Kalman estendido (EKF), que permite a estimação simultânea do estado e parâmetros do sistema. O filtro de Kalman estendido pode ser aplicado na estimação de fluxo de rotor ou de estator, assim como também da velocidade mecânica do motor de indução. Neste caso, o modelo matemático do motor é aumentado por considerar a velocidade como variável de estado, além das componentes de fluxo de rotor e da corrente de estator.

As componentes de fluxo de rotor obtidas a partir do observador podem ser utilizadas pelo algoritmo de controle para fins determinados. No caso do controle por orientação pelo campo, estas componentes são empregadas para calcular a posição espacial do vetor fluxo escolhido (estator, rotor ou entreferro) e, a partir desta, realizar as transformações necessárias pelo algoritmo para garantir a orientação. Por outra parte, a informação de velocidade resultante na saída do estimador é utilizada na malha de velocidade do sistema de controle com o objetivo de gerar a componente adequada de corrente de conjugado, eliminando desta forma a necessidade de utilizar o sensor de velocidade.

Para a implementação do observador de fluxo baseado no filtro de Kalman estendido é necessário o conhecimento das características estatísticas dos sinais de ruído de estado do sistema e de medição, as quais, na prática, são difíceis de se obter. Normalmente, são considerados ruídos de estado e de medida do tipo Gaussiano, não correlacionados com média zero e covariâncias conhecidas. As características de ruído podem mudar durante a estimação, dependendo da operação do método de controle. Neste caso, estes sinais podem ser considerados como quase – estacionários, tendo magnitude constante ao longo de um intervalo de tempo suficientemente grande, sendo o algoritmo de estimação responsável pela sua adaptação diante das variações dos sinais de ruído.

O filtro de Kalman estendido é um algoritmo recursivo de elevada complexidade computacional quando comparado com outros métodos de estimação existentes, porque exige um grande número de operações matriciais que devem ser feitas no intervalo de amostragem. Por este motivo, na implementação deste algoritmo de estimação é imprescindível a utilização de processadores de elevada capacidade e velocidade de processamento. Neste contexto, os processadores digitais de sinais ("DSP") apresentam-se como a alternativa natural na implementação do algoritmo EKF.

Neste capítulo são apresentados os fundamentos teóricos dos observadores de fluxo de rotor e de velocidade, utilizando o algoritmo de filtro de Kalman estendido. Primeiramente, é apresentado o modelo matemático do motor de indução no espaço de estados para aplicação do algoritmo EKF e na seqüência é feita a discretização do modelo. A partir do modelo discreto são descritas as etapas da implementação do algoritmo EKF. Resultados experimentais são apresentados a partir da implementação do observador num sistema de controle por orientação direta do fluxo de rotor.

3.2 Modelo matemático discretizado do motor de indução para aplicação do algoritmo EKF

Embora a estimação da velocidade mecânica e das componentes do vetor fluxo de rotor utilizando o algoritmo do filtro de Kalman estendido possa ser realizada utilizando o

modelo matemático do motor de indução tanto no sistema de referência síncrono $d-q$, como no sistema de referência estacionário $\alpha-\beta$, este último é preferido. Quando é utilizado o modelo do motor no sistema síncrono, transformações adicionais são necessárias. Por exemplo, as tensões do motor devem ser transformadas do sistema estacionário ao síncrono, o qual implica a presença de termos associados ao seno e cosseno do ângulo do vetor de fluxo nas matrizes de entrada e saída do sistema. Este fato incrementa o número de não linearidades do sistema, além de aumentar o tempo de cálculo do algoritmo e diminuir a precisão nas variáveis estimadas. Por isso é recomendável a utilização do modelo do motor de indução no sistema estacionário de referência para a implementação do algoritmo EKF.

O modelo matemático do motor de indução, descrito no sistema de referência estacionário, tendo como variáveis de estado as componentes do fluxo de rotor e das correntes de estator e a velocidade do motor, pode ser expresso na forma de equação de estado como segue [4]:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \end{aligned} \tag{3.1}$$

nas quais:

$$\mathbf{A} = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r T_r} & \frac{L_m}{\sigma L_s L_r} \omega_r & 0 \\ 0 & -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & -\frac{L_m}{\sigma L_s L_r} \omega_r & \frac{L_m}{\sigma L_s L_r T_r} & 0 \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -\omega_r & 0 \\ 0 & \frac{L_m}{T_r} & \omega_r & -\frac{1}{T_r} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.2}$$

sendo o vetor de estado $\mathbf{x} = [i_{s\alpha} \ i_{s\beta} \ \lambda_{r\alpha} \ \lambda_{r\beta} \ \omega_r]^T$ formado pelas componentes dos vetores corrente de estator e fluxo de rotor no sistema de referência estacionário, além da velocidade

mecânica do motor. Os vetores de entrada $\mathbf{u} = [v_{sa} \ v_{sb}]^T$ e de saída $\mathbf{y} = [i_{sa} \ i_{sb}]^T$ são definidos pelas componentes dos vetores tensão e corrente de estator no sistema de referência estacionário.

Como pode ser observado, a partir das equações (3.1) e (3.2) a incorporação da velocidade como variável de estado conduz a um modelo de quinta ordem não linear, sendo que agora a matriz do sistema é função do estado, ou seja, $\mathbf{A} = \mathbf{A}(\mathbf{x})$. Por outro lado, a aproximação $\dot{\omega}_r = 0$ feita no modelo, não é válida do ponto de vista prático, mas constitui uma boa aproximação para taxas de amostragens elevadas. Além disso, o termo de correção do algoritmo EKF leva em consideração o ruído de estado do sistema, no qual encontram-se as imprecisões no cálculo, diminuindo desta maneira o erro de estimação resultante.

A implementação digital do algoritmo EKF requer a discretização do modelo contínuo do motor apresentado anteriormente. A partir das equações (3.1) e (3.2), utilizando o método de Euler, o modelo discreto pode ser escrito como [4]:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_d \mathbf{x}(k) \end{aligned} \quad (3.3)$$

nas quais \mathbf{A}_d , \mathbf{B}_d e \mathbf{C}_d são as matrizes discretas de estado, entrada e saída, respectivamente; $\mathbf{x}(k)$ e $\mathbf{y}(k)$ representam o estado e saída do sistema no k -ésimo instante de amostragem. A matriz discreta da saída coincide com a matriz de saída no tempo contínuo, isto é, $\mathbf{C}_d = \mathbf{C}$. As matrizes discretas de estado e entrada são calculadas utilizando a aproximação em série de Taylor da exponencial matricial como mostrado a seguir:

$$\mathbf{A}_d = \exp[\mathbf{A}T] = \mathbf{I}_n + \mathbf{A}T + \frac{(\mathbf{A}T)^2}{2} + \dots \quad (3.4)$$

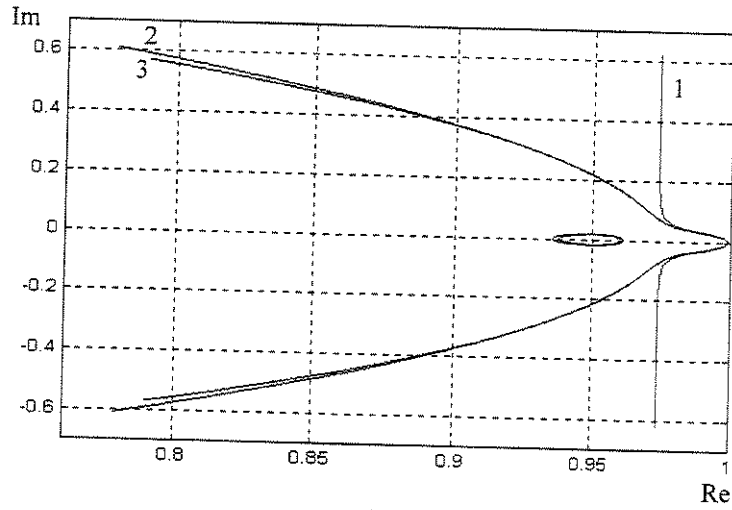
$$\mathbf{B}_d = \mathbf{A}^{-1}(\exp[\mathbf{A}T] - \mathbf{I}_n)\mathbf{B} = \mathbf{B}T + \frac{\mathbf{A}\mathbf{B}T^2}{2} + \dots \quad (3.5)$$

na qual T é o período de amostragem.

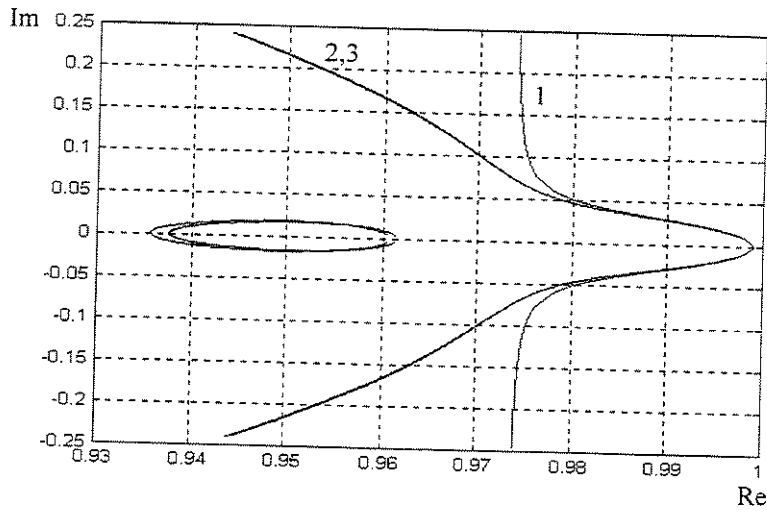
Do ponto de vista prático, para o cálculo das matrizes discretas são utilizados os termos de primeira ou segunda ordem da série de Taylor. De maneira geral, a taxa de amostragem deve ser elevada de maneira a não comprometer a estabilidade do sistema discreto resultante. A aproximação de segunda ordem, embora garanta uma melhor precisão no cálculo, exige um elevado esforço computacional na implementação, que somado à complexidade inerente do algoritmo EKF, impõe uma limitação na seleção do período de amostragem. A aproximação de primeira ordem diminui o tempo de cálculo, mas certa atenção deve ser dada aos problemas de estabilidade do sistema, como será discutido a seguir.

Dado que a velocidade do motor aparece como parâmetro dentro da matriz do sistema, a precisão associada à aproximação adotada é função da faixa de operação desta variável no sistema de controle. A Figura 3.1 mostra a trajetória resultante dos pólos da matriz discreta A_d , para diferentes intervalos da velocidade, quando utilizadas aproximações de primeira e segunda ordem, assim como aquela obtida com a função "expm" do Matlab. O período de amostragem utilizado no cálculo da matriz A_d foi de 300 μ s. Da figura fica evidente que a medida que a velocidade aumenta, o erro associado à utilização da aproximação de primeira ordem também aumenta, passando a ser crítico para velocidades elevadas. Quando utilizada a aproximação de segunda ordem é possível aumentar a faixa de operação de velocidade, mantendo o erro dentro de limites adequados.

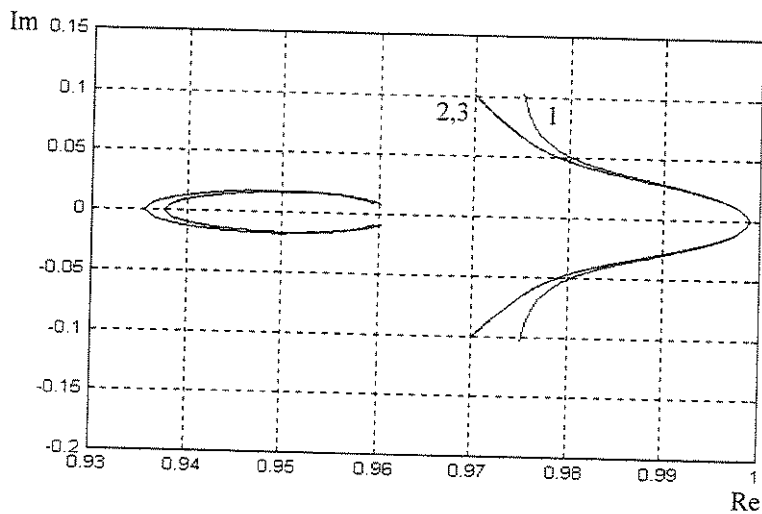
Da análise anterior podemos concluir que a escolha do tipo de aproximação utilizado no cálculo das matrizes discretas deve representar uma solução de compromisso entre a complexidade de cálculo e os problemas de estabilidade do sistema discreto resultante.



(a)



(b)



(c)

Figura 3.1- Trajetória resultante dos pólos da matriz discreta A_d utilizando (1) aproximação de primeira ordem (2) aproximação de segunda ordem (3) função expm.
 (a) 0-10.000 rpm (b) 0-4.000 rpm (c) 0-1.800 rpm

As matrizes discretas da equação (3.3) para o caso de se utilizar aproximação de primeira ordem podem ser escritas como [4]:

$$\mathbf{A}_d = \begin{bmatrix} 1 - T \left(\frac{1}{\sigma T_s} + \frac{1 - \sigma}{\sigma T_r} \right) & 0 & \frac{TL_m}{\sigma L_s L_r T_r} & \frac{TL_m}{\sigma L_s L_r} \omega_r & 0 \\ 0 & 1 - T \left(\frac{1}{\sigma T_s} + \frac{1 - \sigma}{\sigma T_r} \right) & -\frac{TL_m}{\sigma L_s L_r} \omega_r & \frac{TL_m}{\sigma L_s L_r T_r} & 0 \\ \frac{TL_m}{T_r} & 0 & 1 - \frac{T}{T_r} & -T \omega_r & 0 \\ 0 & \frac{TL_m}{T_r} & T \omega_r & 1 - \frac{T}{T_r} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B}_d = \begin{bmatrix} \frac{T}{\sigma L_s} & 0 \\ 0 & \frac{T}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{C}_d = \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (3.6)$$

Consideremos ruídos de estado e de medição não correlacionados e com média zero, representados pelos vetores $\mathbf{v}(k)$ e $\mathbf{w}(k)$ respectivamente, com matrizes de covariância \mathbf{Q} e \mathbf{R} , respectivamente. Considerando estes sinais, o sistema discreto é escrito como [4]:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d \mathbf{u}(k) + \mathbf{v}(k) \\ \mathbf{y}(k) &= \mathbf{C}_d \mathbf{x}(k) + \mathbf{w}(k) \end{aligned} \quad (3.7)$$

As matrizes de covariância \mathbf{Q} e \mathbf{R} são matrizes quadradas de dimensões 5 e 2, respectivamente. Dada a suposição de que os sinais de ruído são não-correlacionados, estas matrizes são consideradas diagonais. Além disso, os elementos de \mathbf{Q} e \mathbf{R} associados às componentes de fluxo ou corrente de estator no sistema estacionário de referência são iguais. Sendo assim, o número total de elementos das matrizes de covariância que é necessário conhecer para a implementação do algoritmo EKF, reduz-se a quatro, diminuindo desta maneira as exigências de memória do sistema.

3.3 Implementação do algoritmo EKF discreto

Na estrutura do filtro de Kalman estendido podem ser distinguidas duas etapas. A primeira etapa é chamada etapa de predição, onde a primeira estimativa do estado do sistema no instante $k + 1$ é feita a partir do modelo do motor e da informação referente ao estado no instante anterior k . Além disso, nessa etapa de predição, é feita uma estimativa da matriz de covariância do erro de estimação \mathbf{P} no instante $k + 1$. Isto é feito utilizando a informação referente à própria matriz \mathbf{P} no instante k , além da matriz de covariância de estado \mathbf{Q} e as equações do modelo.

Na segunda etapa do algoritmo EKF, chamada de etapa de filtragem, ao estado do sistema obtido na etapa de predição é somado o termo de correção relacionado com o erro entre o estado medido na saída do sistema e o próprio estado estimado pelo algoritmo. As variáveis de estado medidas são, neste caso, as componentes das correntes de estator no sistema estacionário de referência, $i_{s\alpha}$ e $i_{s\beta}$. Sendo assim, a estimação de estado utilizando o algoritmo EKF pode ser representada pela seguinte equação:

$$\hat{\mathbf{x}} = \mathbf{A}(\hat{\mathbf{x}})\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}(i_s - \hat{i}_s) \quad (3.8)$$

onde \mathbf{K} é matriz de ganho do filtro, a qual é calculada de forma a minimizar o erro de estimação. A Figura 3.2 mostra um diagrama de blocos do algoritmo EKF.

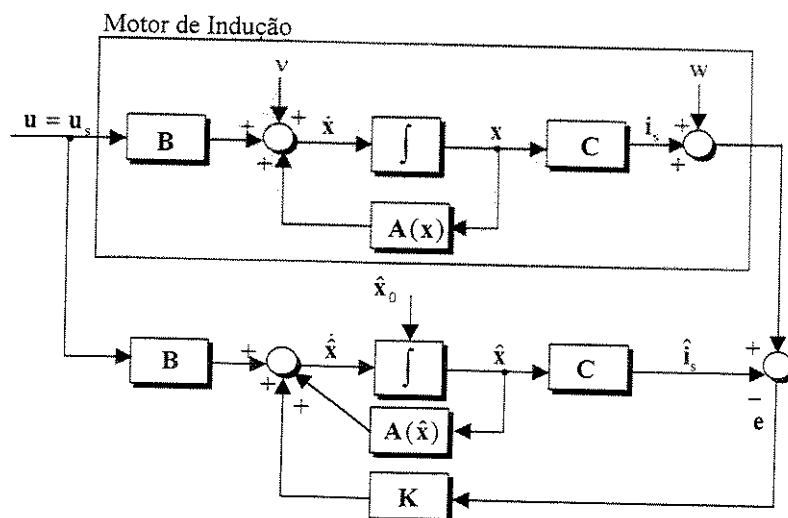


Figura 3.2 - Diagrama de blocos do algoritmo EKF

A execução detalhada do algoritmo EKF pode ser explicada através dos seguintes passos [4]. O passo 1 e 2 correspondem a etapa de predição, enquanto os passos de 3 até 5 à etapa de filtragem.

a) Etapa de predição

1. Predição de estado

A predição do vetor de estado no instante $k+1$ é feita utilizando o modelo do motor e a informação do estado no instante k como segue:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{A}_d \hat{\mathbf{x}}(k|k) + \mathbf{B}_d \mathbf{u}(k) \quad (3.9)$$

2. Estimação da matriz de covariância do erro

Neste passo a matriz de covariância do erro no instante $k+1$ é estimada utilizando as equações do modelo e a matriz de covariância de estado da seguinte maneira:

$$\hat{\mathbf{P}}(k+1|k) = \mathbf{f}(k+1|k) \hat{\mathbf{P}}(k|k) \mathbf{f}^T(k+1|k) + \mathbf{Q} \quad (3.10)$$

onde $\mathbf{f}(k+1|k)$ é uma matriz gradiente definida como:

$$\mathbf{f}(k+1|k) = \frac{\partial}{\partial \mathbf{x}} (\mathbf{A}_d \mathbf{x} + \mathbf{B}_d \mathbf{u})_{\mathbf{x}=\hat{\mathbf{x}}(k|k)} \quad (3.11)$$

Substituindo (3.6) em (3.11) e derivando em relação ao estado obtemos:

$$\mathbf{f}(k+1) = \begin{bmatrix} 1 - T \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) & 0 & \frac{TL_m}{\sigma L_s L_r T_r} & \frac{TL_m}{\sigma L_s L_r} \omega_r & \frac{TL_m}{\sigma L_s L_r} \lambda_{r\beta} \\ 0 & 1 - T \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) & -\frac{TL_m}{\sigma L_s L_r} \omega_r & \frac{TL_m}{\sigma L_s L_r T_r} & -\frac{TL_m}{\sigma L_s L_r} \lambda_{r\alpha} \\ \frac{TL_m}{T_r} & 0 & 1 - \frac{T}{T_r} & -T \omega_r & -T \lambda_{r\beta} \\ 0 & \frac{TL_m}{T_r} & T \omega_r & 1 - \frac{T}{T_r} & T \lambda_{r\alpha} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

b) Etapa de filtragem

3. Cálculo da matriz de ganho do filtro de Kalman

Neste ponto estamos em condições de calcular a matriz de ganho do filtro que é dada pela seguinte equação matricial:

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{h}^T(k+1)\left[\mathbf{h}(k+1)\mathbf{P}(k+1|k)\mathbf{h}^T(k+1) + \mathbf{R}\right]^{-1} \quad (3.13)$$

No caso do motor de indução a matriz de ganho do filtro possui cinco linhas e duas colunas, sendo $\mathbf{h}(k+1)$ definido como:

$$\mathbf{h}(k+1) = \frac{\partial}{\partial \mathbf{x}} (\mathbf{C}_d \mathbf{x})_{\mathbf{x}=\hat{\mathbf{x}}(k+1|k)} \quad (3.14)$$

Substituindo (3.6) em (3.14) e derivando em relação ao estado obtemos:

$$\mathbf{h}(k+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (3.15)$$

4. Atualização da matriz de covariância do erro

Neste passo é feita a atualização da matriz de covariância do erro como segue:

$$\mathbf{P}(k+1|k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k) \quad (3.16)$$

5. Estimação de estado

A estimação do vetor de estado no instante $k+1$ é feita somando o termo de correção ao estado obtido na etapa de predição como mostrado a seguir:

$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\left[\mathbf{y}(k+1) - \hat{\mathbf{y}}(k+1)\right] \quad (3.17)$$

onde:

$$\hat{\mathbf{y}}(k+1) = \mathbf{C}_d \hat{\mathbf{x}}(k+1|k) = \begin{bmatrix} \hat{i}_{s\alpha}(k+1|k) & \hat{i}_{s\beta}(k+1|k) \end{bmatrix}^T \quad (3.18)$$

A partir da nova estimação do estado o algoritmo continua de maneira iterativa do passo 1 até o 5. Para isto é necessário a seguinte atualização nas variáveis:

$$k = k + 1, \mathbf{x}(k) = \mathbf{x}(k - 1), \mathbf{P}(k) = \mathbf{P}(k - 1) \quad (3.19)$$

Os valores estimados das componentes de fluxo de rotor no sistema de referência estacionário podem ser utilizados para a determinação do seno e cosseno do ângulo elétrico do vetor fluxo, dados por:

$$\text{sen}(\theta_e) = \frac{\hat{\lambda}_{r\beta}}{\sqrt{\hat{\lambda}_{r\alpha}^2 + \hat{\lambda}_{r\beta}^2}}, \quad \text{cos}(\theta_e) = \frac{\hat{\lambda}_{r\alpha}}{\sqrt{\hat{\lambda}_{r\alpha}^2 + \hat{\lambda}_{r\beta}^2}} \quad (3.20)$$

Estes valores servem como entradas às rotinas de transformação entre os sistemas de referências síncrono e estacionário do sistema de controle. Por outra parte, a velocidade mecânica estimada $\hat{\omega}_r$ é utilizada na malha de velocidade do controle, eliminando a necessidade do sensor de velocidade.

A partir da descrição dos passos do algoritmo EKF fica evidente a influência das matrizes de covariância \mathbf{Q} e \mathbf{R} no cálculo da matriz de ganho do filtro, determinando o desempenho do algoritmo de estimação tanto no transitório como no estado de regime. Por exemplo, um aumento nos elementos da matriz de covariância \mathbf{Q} representa um maior grau de incerteza em relação ao estado sistema. Este aumento produz um incremento na matriz de ganho do filtro, aumentando a velocidade de convergência da estimação. Além disso, um aumento nos elementos da matriz de covariância \mathbf{R} significa que as medições das correntes estão sujeitas a uma maior influência de ruído e portanto devem ser pesadas em um menor grau no algoritmo de estimação. Isto implica na diminuição da matriz de ganho do filtro e, como consequência, da velocidade de reposta do filtro. Valores muito elevados da matriz \mathbf{Q} ou muito pequenos da matriz \mathbf{R} podem levar à instabilidade no processo de estimação.

Uma outra característica do algoritmo EKF, que é evidenciada nos passos da implementação descritos nesta seção (equações 3.9-3.18), é a complexidade computacional do método. Este fato deve ser levado em consideração na implementação, dado que impõe uma limitação na seleção nos tempos de amostragem e de execução das rotinas de controle.

3.4 Resultados Experimentais

O algoritmo do filtro de Kalman estendido foi testado experimentalmente num sistema de controle por orientação direta do fluxo de rotor. Os parâmetros do motor utilizado nos testes aparecem no Apêndice A deste trabalho. O sistema de controle por orientação, assim como o observador de fluxo de rotor e de velocidade baseados no algoritmo EKF foram implementados utilizando o processador digital de sinais DSP TMS320F240 da Texas Instruments. A descrição detalhada da implementação do sistema de controle aparece no Apêndice C. A Figura 3.3 mostra um diagrama de blocos do sistema implementado.

A discretização do modelo do motor para a implementação do algoritmo EKF foi feita utilizando a aproximação de primeira ordem. O algoritmo de controle, incluindo as rotinas de entrada – saída, foi feito utilizando a linguagem *assembler* do DSP. A execução do algoritmo de controle está sincronizada com período do sinal de PWM, sendo executada a cada 300 μ s (3.33 kHz). O tempo de execução do algoritmo de controle, incluindo o observador, foi de 190 μ s, sendo de 150 μ s o tempo aproximado de execução do algoritmo EKF. Durante o desenvolvimento do programa de controle um encoder incremental de 1200 pulsos/rev foi utilizado com objetivo de comparar a velocidade medida com o valor estimado a partir do observador. Na obtenção das curvas experimentais foi utilizado um osciloscópio digital da Tektronix conectado aos canais selecionados do conversor D/A da placa EVM do DSP.

A apresentação dos resultados é dividida em duas partes fundamentais. Na primeira são mostrados os resultados de regime permanente. A segunda parte mostra resultados do funcionamento do sistema de controle no regime transitório, sendo aplicadas ao motor referências variáveis de velocidade mecânica e de corrente de conjugado, avaliando-se o desempenho do processo de estimação. As matrizes de covariância \mathbf{Q} e \mathbf{R} , assim como também a matriz inicial de covariância \mathbf{P}_0 , utilizadas na implementação foram as seguintes:

$$\mathbf{Q} = \begin{bmatrix} 0.015259 & 0 & 0 & 0 & 0 \\ 0 & 0.015259 & 0 & 0 & 0 \\ 0 & 0 & 0.004577 & 0 & 0 \\ 0 & 0 & 0 & 0.004577 & 0 \\ 0 & 0 & 0 & 0 & 0.007629 \end{bmatrix}, \mathbf{R} = 0.30518 \cdot \mathbf{I}_2, \mathbf{P}_0 = 0.004882 \cdot \mathbf{I}_5$$

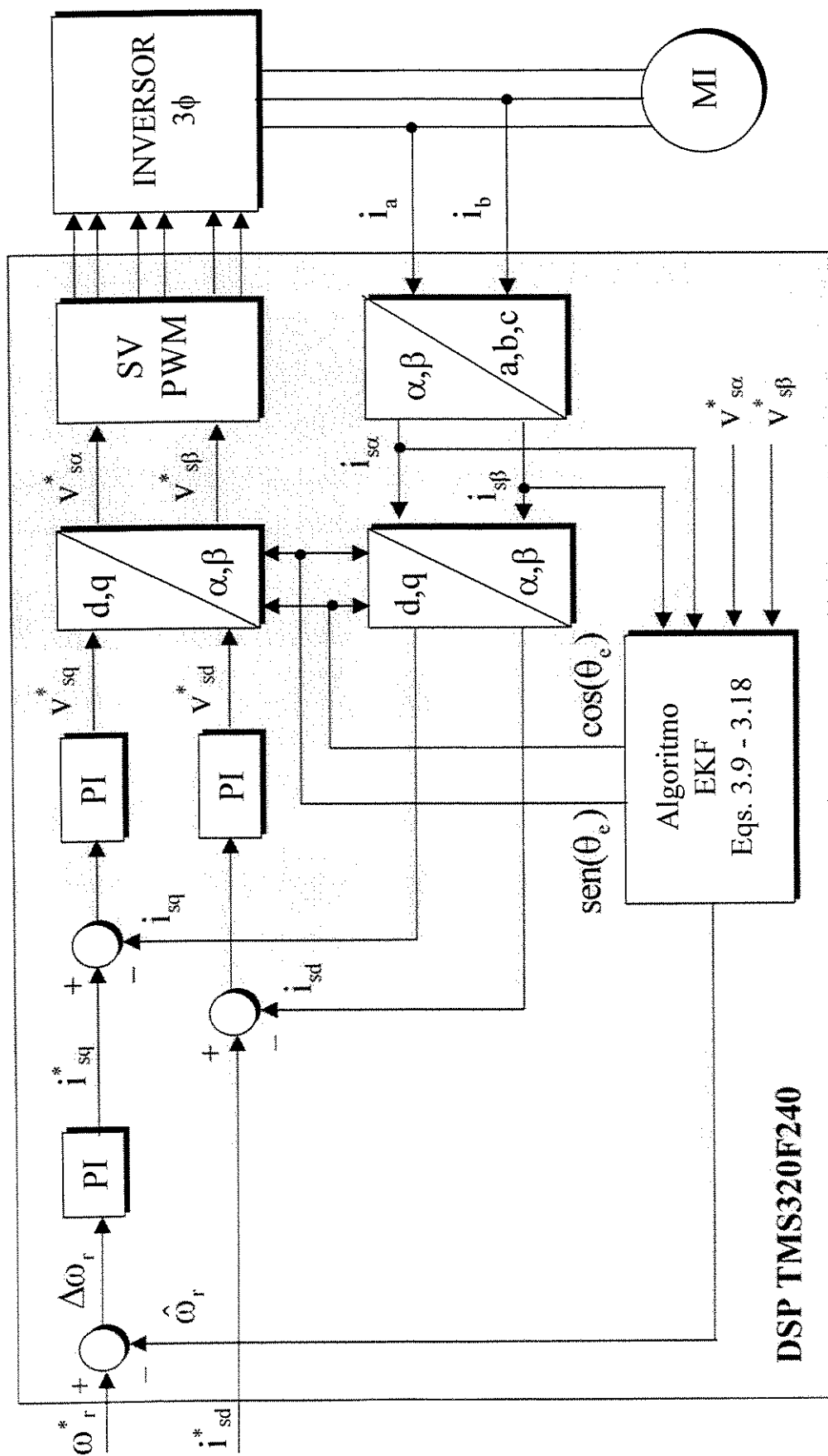
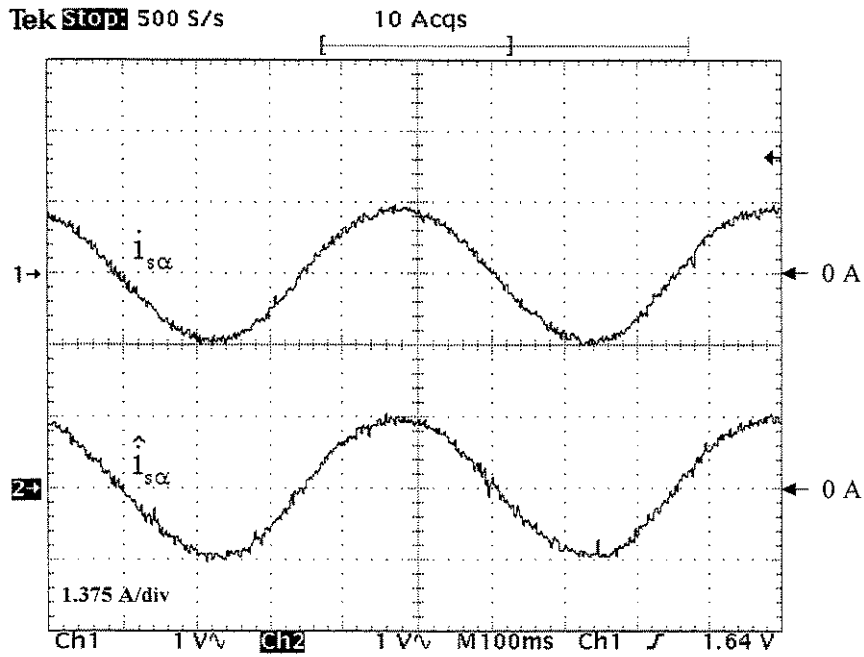


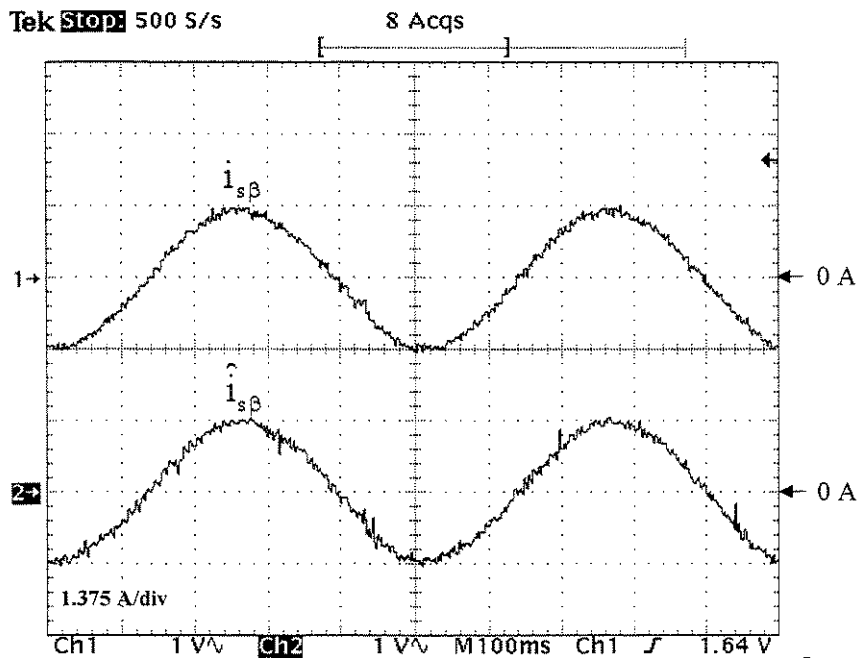
Figura 3.3 - Diagrama de blocos do sistema de controle utilizando o algoritmo EKF

3.4.1 Testes em regime permanente

A Figura 3.4 mostra as curvas experimentais das componentes reais da corrente de estator, assim como os valores estimados a partir do observador para uma velocidade de operação de 70 rpm. A seguir na Figura 3.5 são apresentados os resultados da estimação das componentes do vetor fluxo de rotor para a mesma velocidade de operação. Este teste foi feito para outras velocidades de operação. As Figuras 3.6 e 3.7 mostram o resultado do teste para uma velocidade de 450 rpm e as Figuras 3.8 e 3.9, para 1.000 rpm. Os resultados apresentados para regime permanente mostram o bom funcionamento do observador para diferentes velocidades, sendo verificado como as componentes estimadas da corrente de estator seguem os valores reais resultantes da medição. No caso da estimação de fluxo é verificada a correspondência entre os valores estimados e o valor de referência imposto pelo controle, assim como a defasagem de $\pi/2$ radianos elétricos entre as duas componentes.

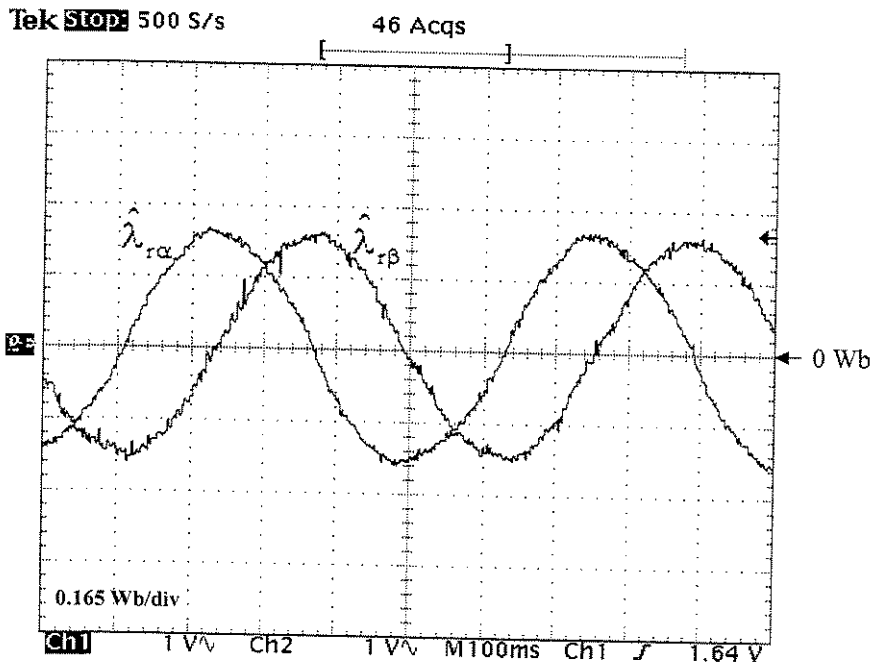


(a)

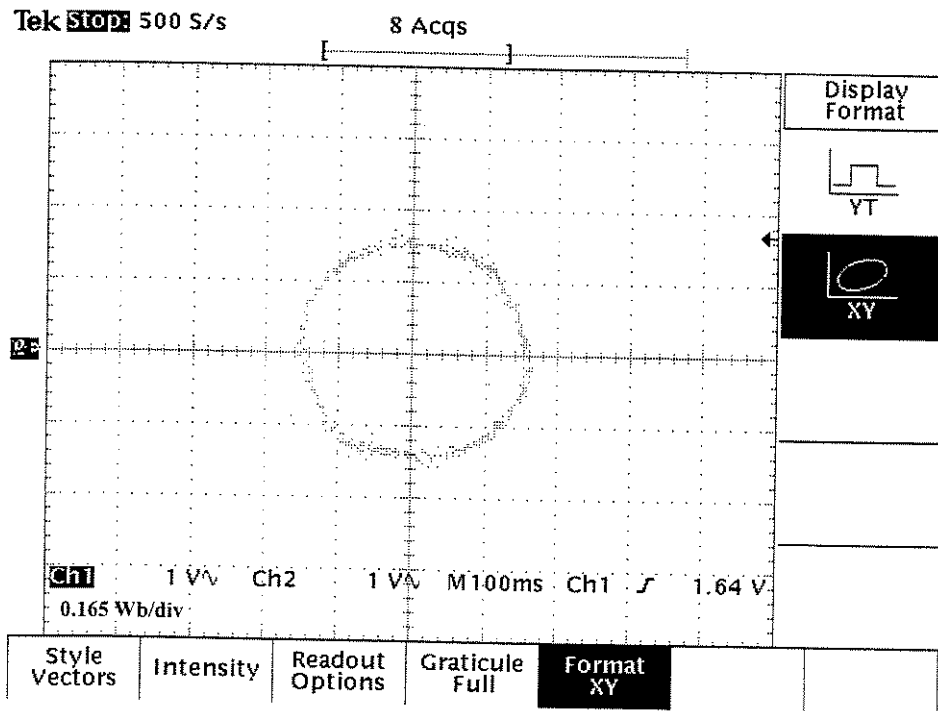


(b)

Figura 3.4 - Correntes reais e estimadas para $\omega_r^* = 70$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$

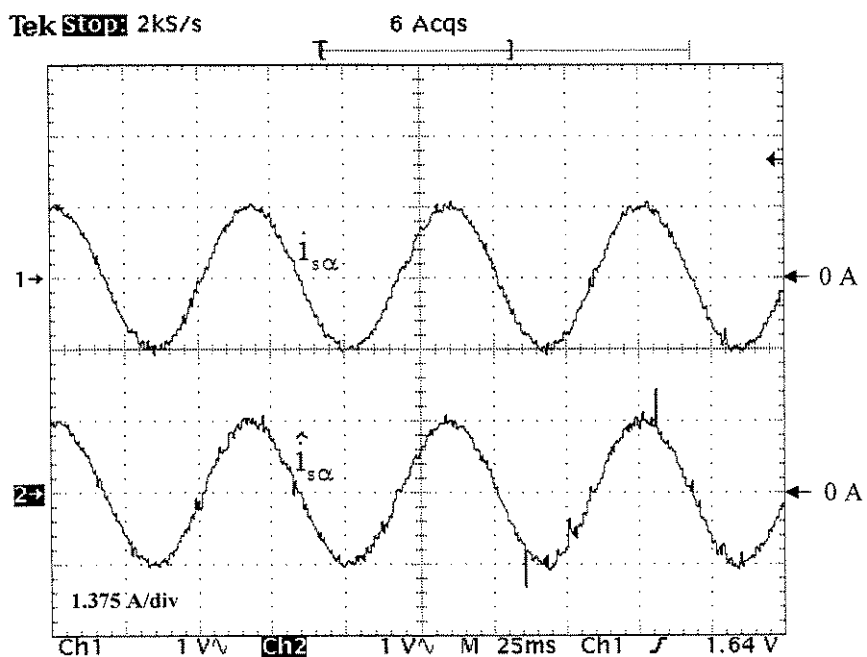


(a)

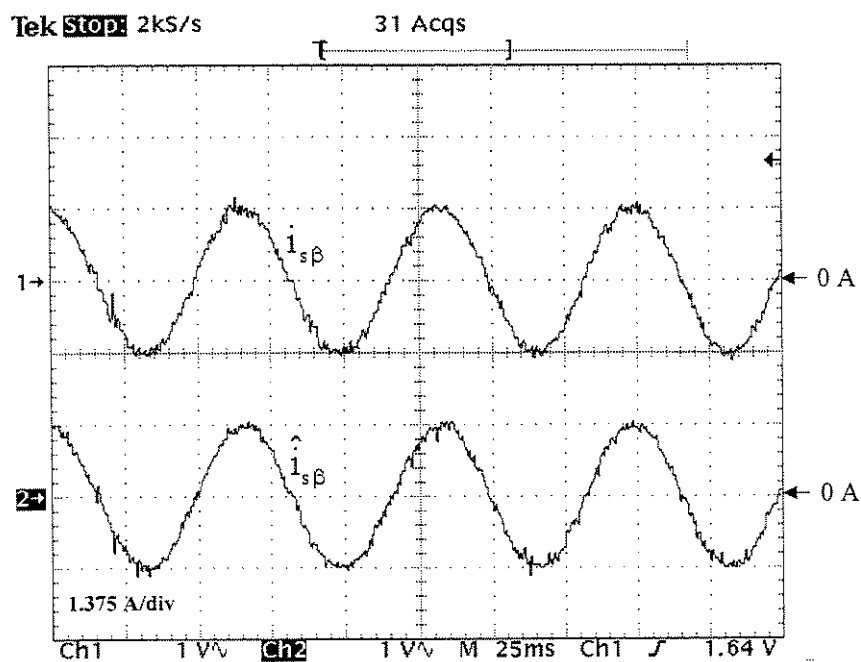


(b)

Figura 3.5 - Componentes estimadas do fluxo de rotor para $\omega_r^* = 70$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$ (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

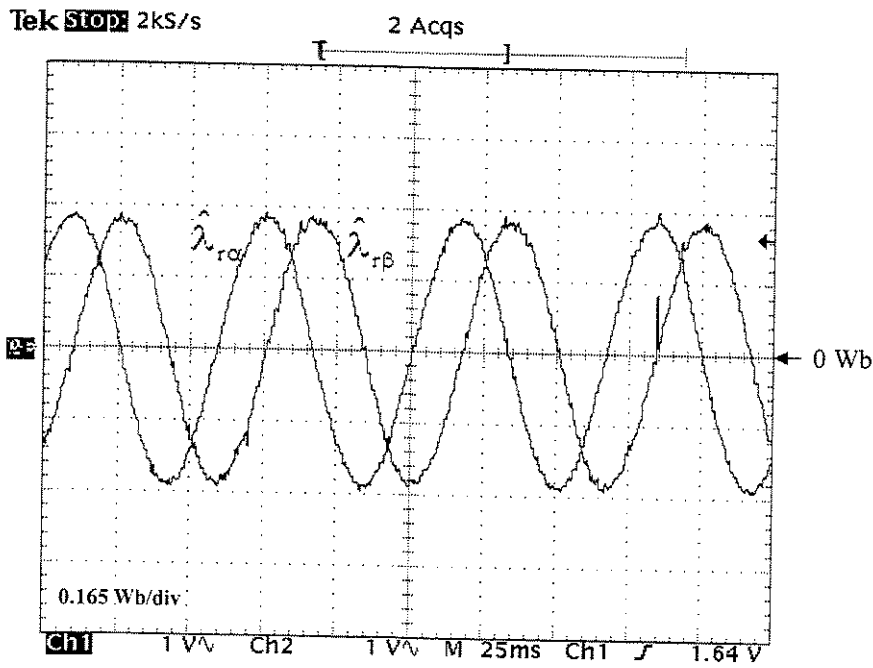


(a)

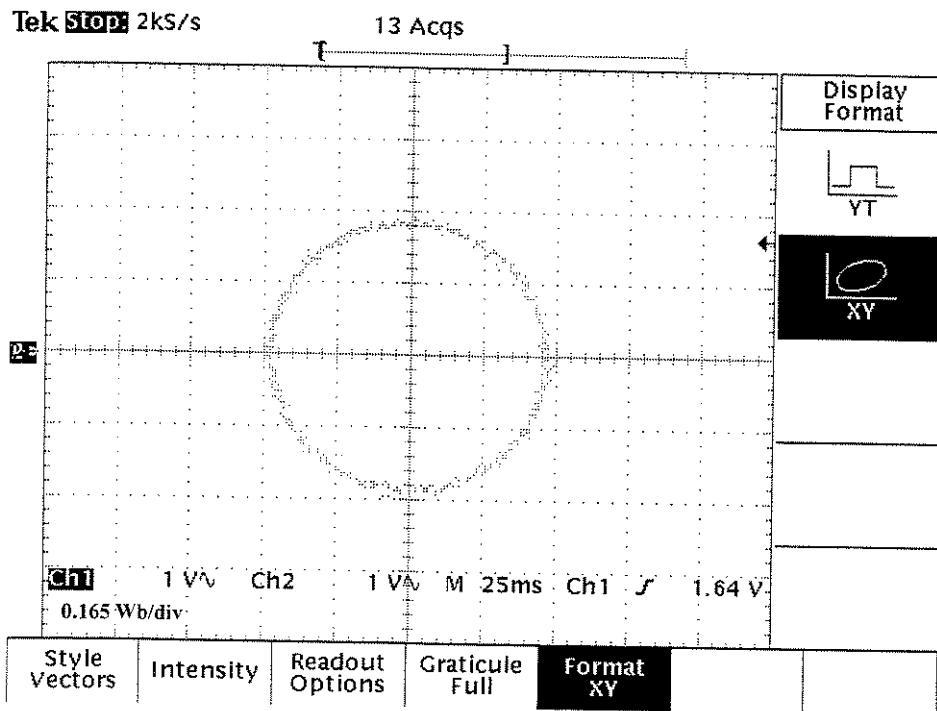


(b)

Figura 3.6 - Correntes reais e estimadas para $\omega_r^* = 450$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$

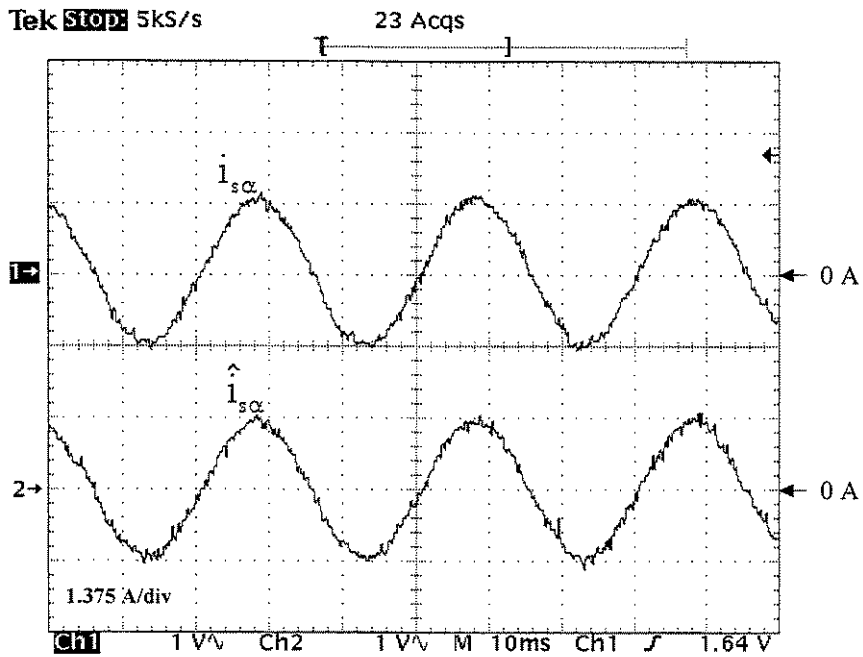


(a)

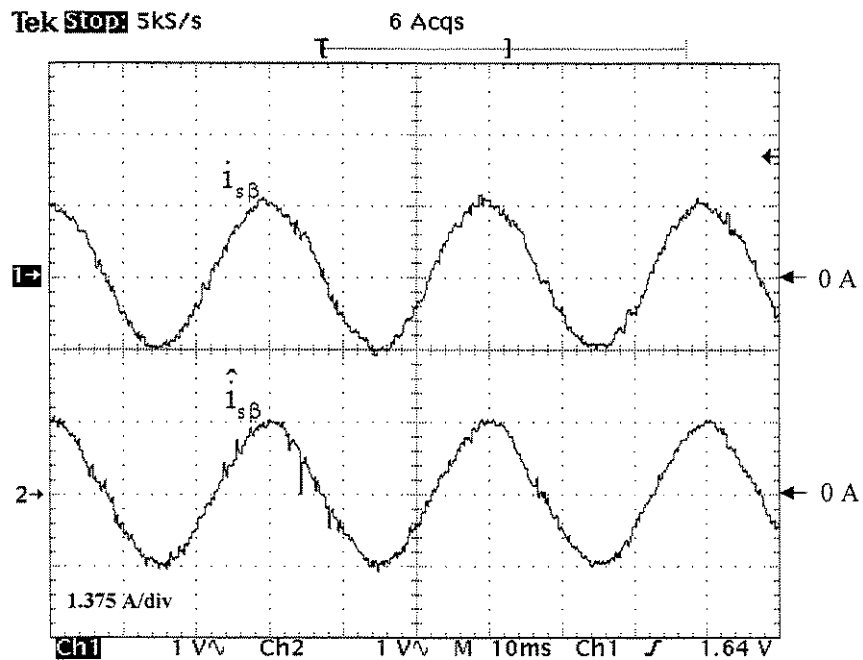


(b)

Figura 3.7 - Componentes estimadas do fluxo de rotor para $\omega_r^* = 450$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$
 (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

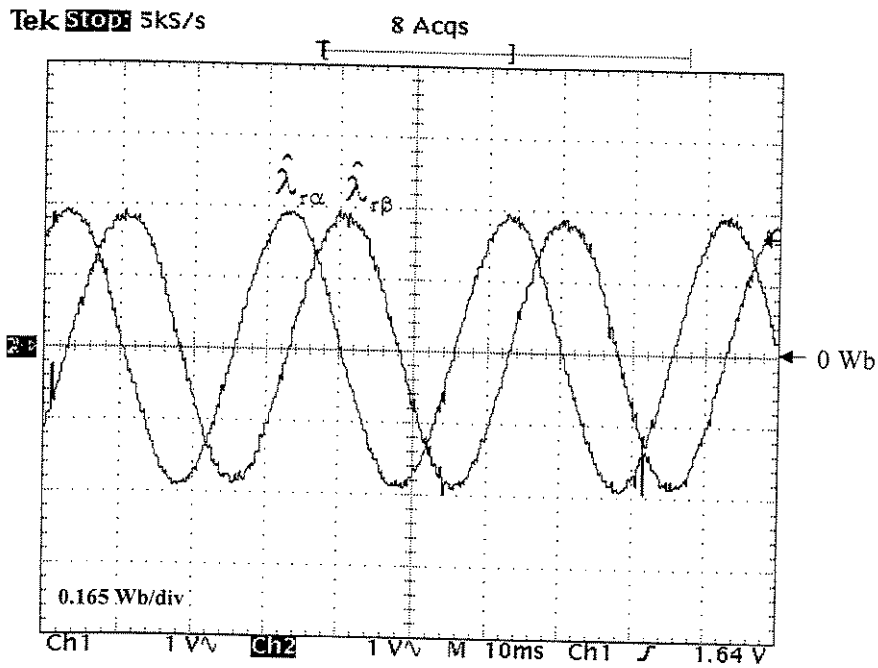


(a)

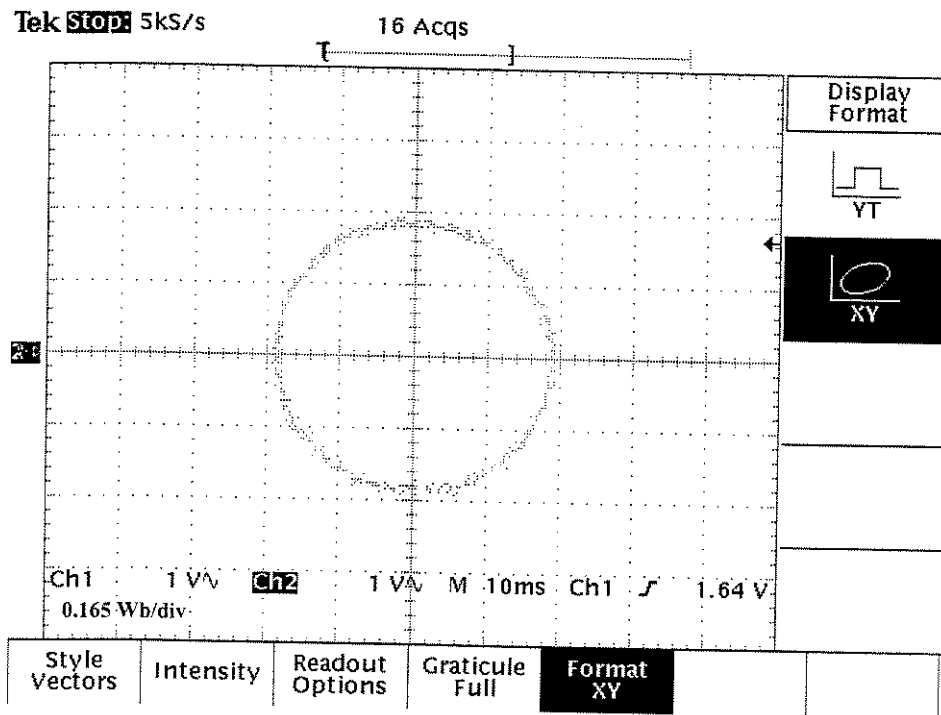


(b)

Figura 3.8 - Correntes reais e estimadas para $\omega_r^* = 1.000 \text{ rpm}$ (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$



(a)



(b)

Figura 3.9 - Componentes estimadas do fluxo de rotor para $\omega_r^* = 1.000$ rpm (a) $\hat{\lambda}_{r\alpha}$, $\hat{\lambda}_{r\beta}$
 (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

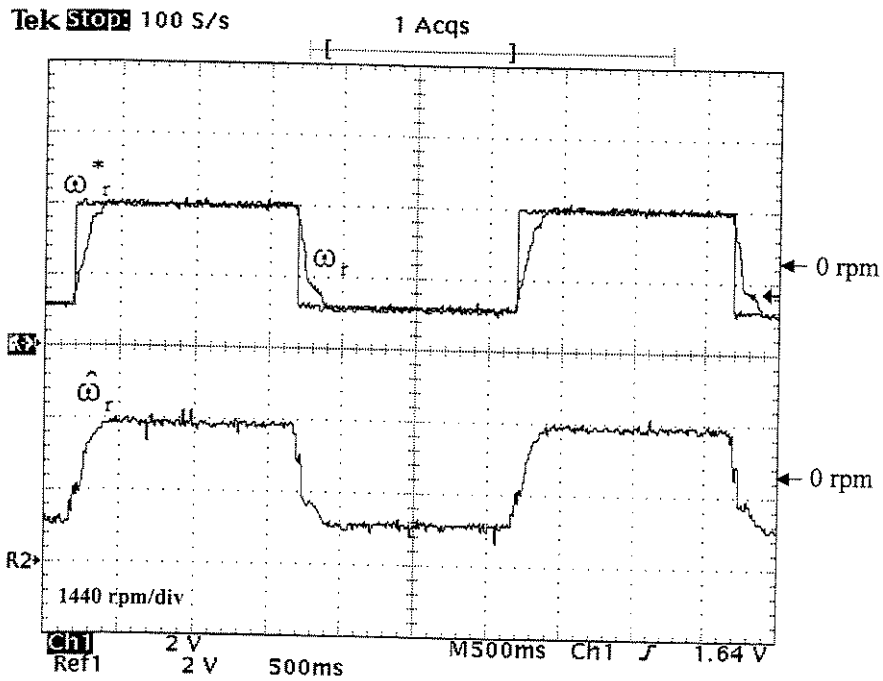
3.4.2 Testes em regime transitório

Dois testes foram feitos para avaliar o desempenho do sistema no regime transitório: 1) referência variável de velocidade 2) referência variável na componente de conjugado da corrente de estator.

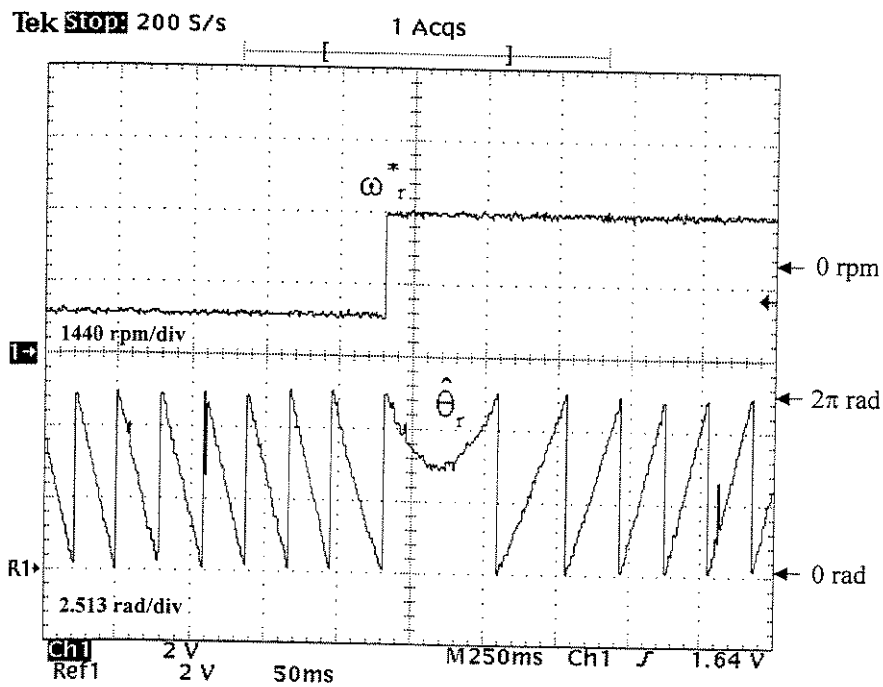
A Figura 3.10(a) mostra a velocidade mecânica medida e a estimada quando é aplicada uma referência variável de velocidade do tipo onda quadrada de 1000 a -1000 rpm. Na Figura 3.10(b) é mostrado o sinal de referência de velocidade e a variação do ângulo elétrico do vetor fluxo de rotor durante o transitório. Das figuras verifica-se como a velocidade estimada pelo observador segue com exatidão o valor medido. O mesmo teste foi repetido com a referência de velocidade mudando de 20 a 100 rpm. As curvas experimentais referentes a este teste aparecem nas Figuras 3.11(a) e 3.11(b), mostrando como a velocidade estimada ainda segue com uma boa aproximação a velocidade medida. O erro de regime é de aproximadamente 6 rpm, ou seja 0.3% da velocidade nominal o que é um erro aceitável em relação à velocidade medida.

A seguir foi feito o teste de aceleração – desaceleração com inversão de velocidade. Para o motor foi acelerado de uma velocidade de -450 rpm até 450 rpm e após 3 segundos é feita a desaceleração até -450 rpm, repetindo-se o ciclo de operação. As Figuras 3.12(a) e 3.12(b) mostram as curvas de velocidade medida e estimada e as Figuras 3.13(a) e 3.13(b) as componentes reais e estimadas da corrente de estator durante o transitório. A partir do teste verifica-se como a velocidade estimada pelo observador segue com exatidão o valor medido. Este fato é também verificado para o caso das componentes da corrente de estator.

Por último foi aplicado um sinal do tipo onda quadrada na componente de conjugado da corrente de estator i_{sq}^* . A Figura 3.14 apresenta as curvas experimentais da velocidade com a referência de corrente mudando de 0.5 até -0.5 pu com um período de 300 ms.

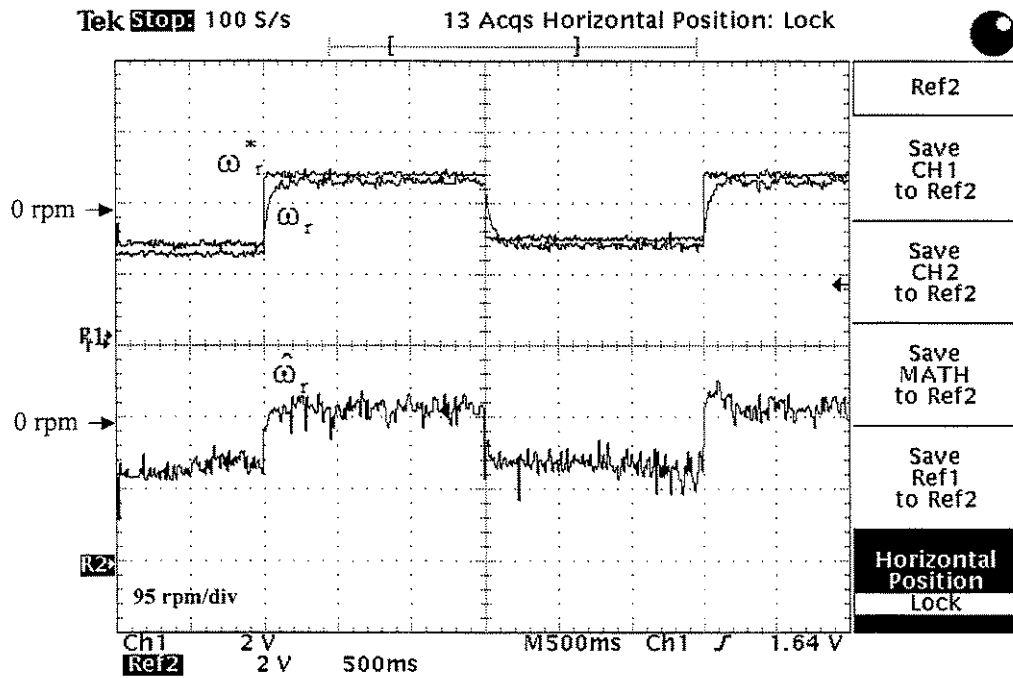


(a)

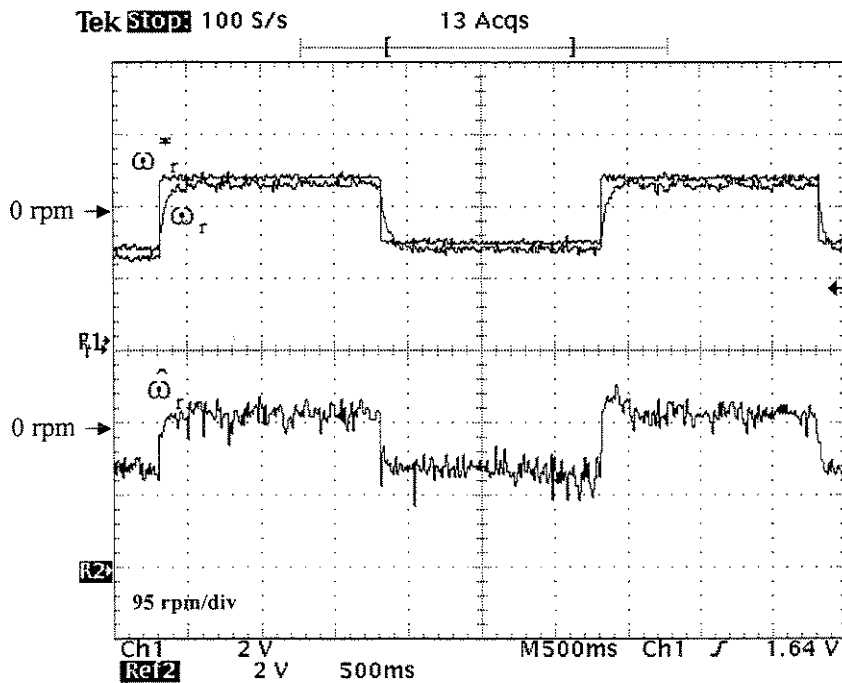


(b)

Figura 3.10 - Reversão de velocidade de -1.000 rpm a 1.000 rpm. (a) ω_r^* , ω_r , $\hat{\omega}_r$ (b) ω_r^* , θ_r

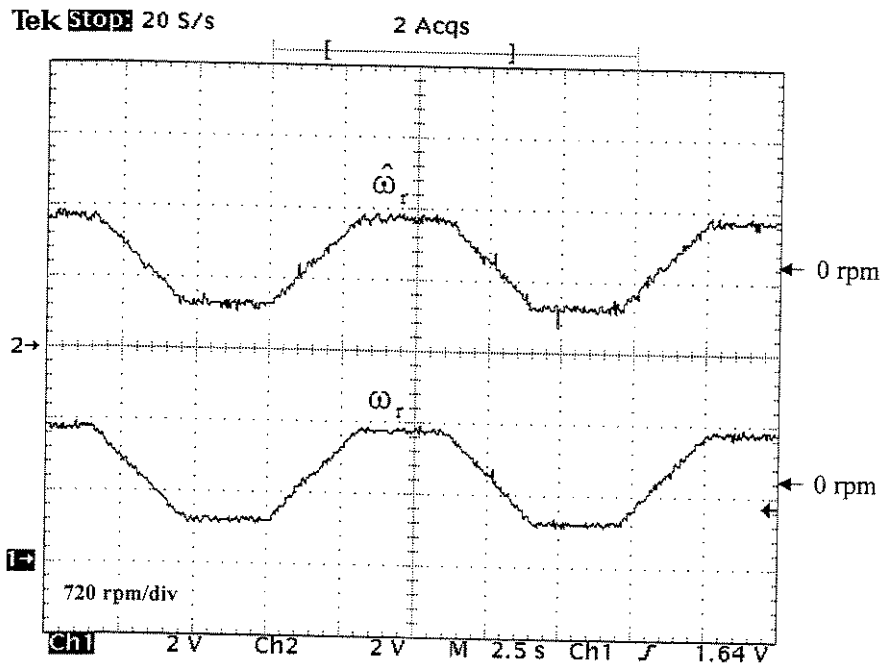


(a)

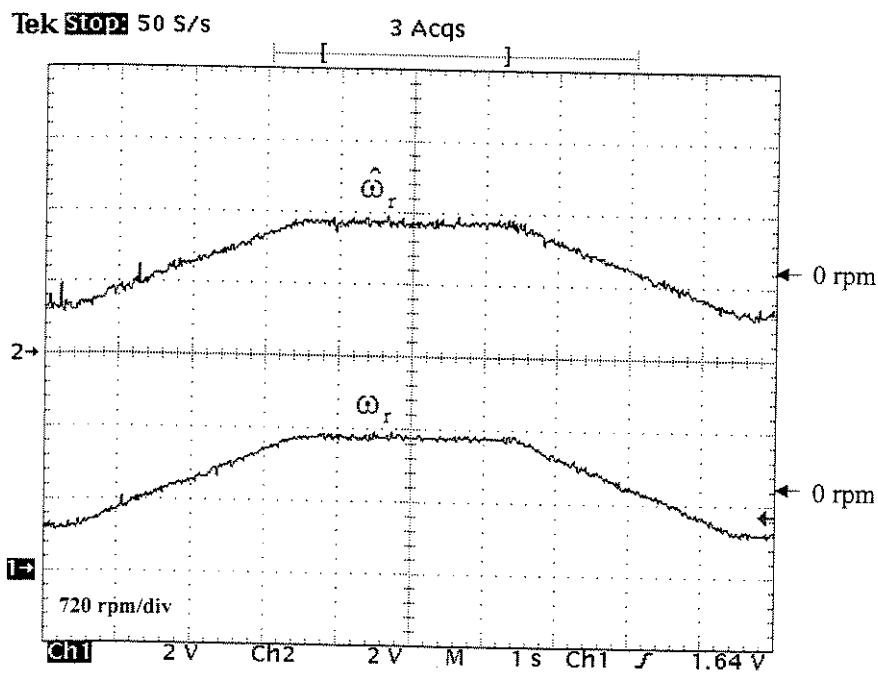


(b)

Figura 3.11 - Teste de onda quadrada na referência de velocidade de 20 a 100 rpm (a) $\omega_r^*, \omega_r, \hat{\omega}_r$
 (b) instante diferente do teste

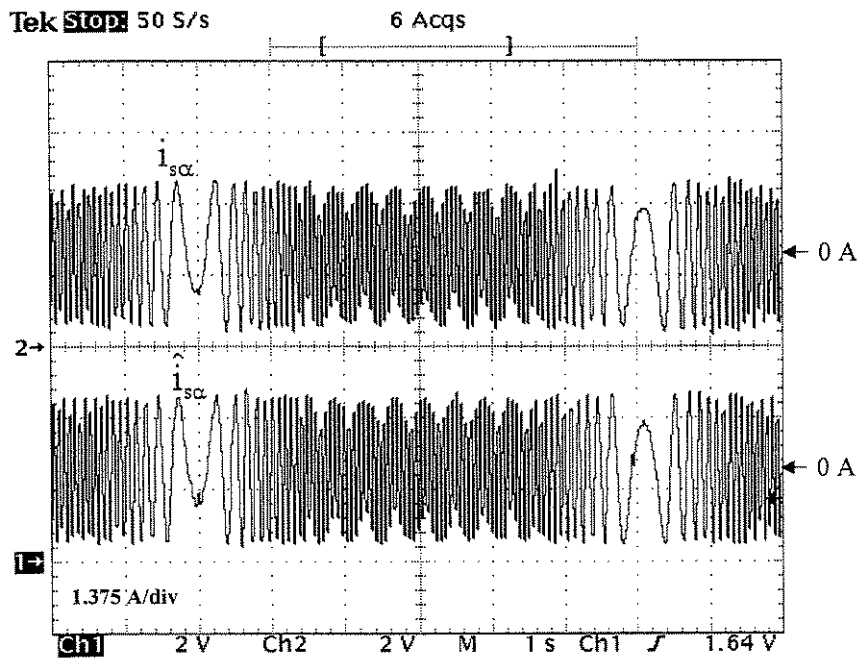


(a)

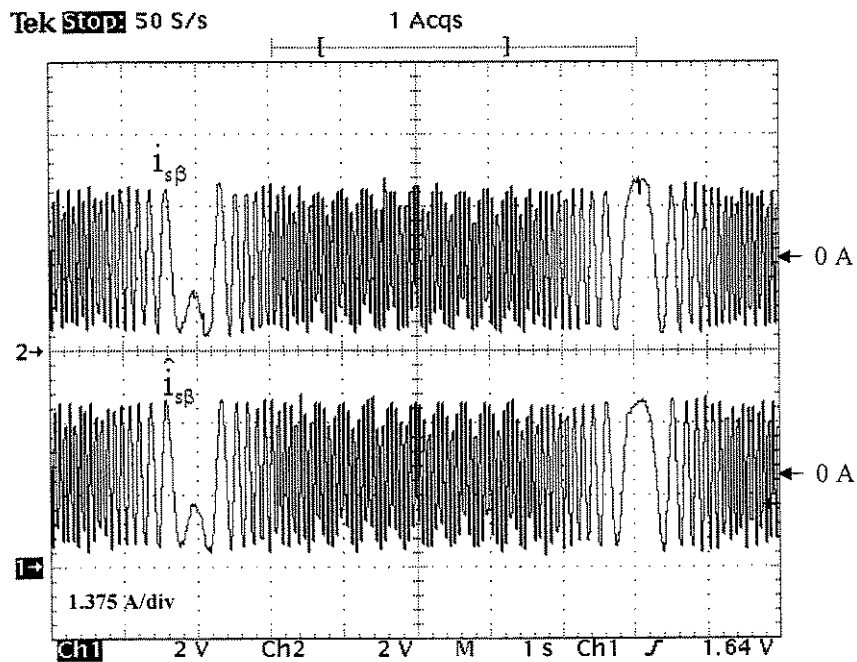


(b)

Figura 3.12 - Teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $\omega_r, \hat{\omega}_r$ (2.5 s/div)
 (b) $\omega_r, \hat{\omega}_r$ (1 s/div)



(a)



(b)

Figura 3.13 - Correntes reais e estimadas para o teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $i_{s\alpha}$, $\hat{i}_{s\alpha}$ (b) $i_{s\beta}$, $\hat{i}_{s\beta}$

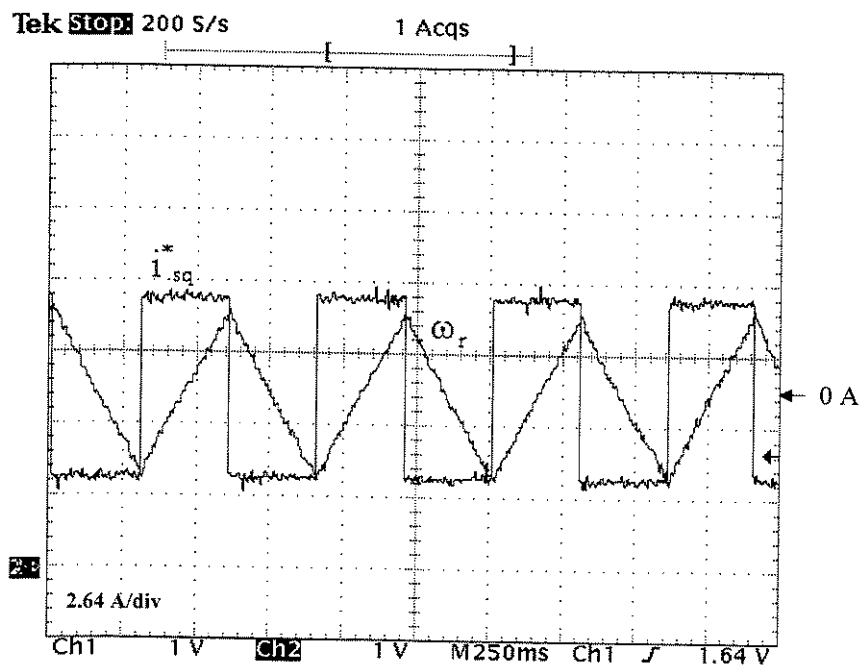


Figura 3.14 - Teste de onda quadrada na referência i_{sq}^* de 0.5 a -0.5 pu.

Capítulo 4

Observador adaptativo de Luenberger para estimação do fluxo de rotor

4.1 Introdução

No capítulo anterior foi estudado o algoritmo do filtro de Kalman estendido para a estimação do fluxo de rotor e da velocidade do motor de indução. Como foi comentado, o algoritmo EKF exige o conhecimento das características estatísticas dos sinais de ruído de estado do sistema e de medição que, do ponto de vista prático, são difíceis de se obter. Normalmente, são considerados sinais de ruído de estado e de medida do tipo Gaussiano, não-correlacionados com média zero. Como as características estatísticas destes sinais podem mudar durante a estimação, o algoritmo tem a função de realizar a atualização das matrizes de covariâncias associadas.

Por outra parte, como foi explicado, o algoritmo EKF possui uma elevada complexidade computacional devido ao grande número de operações matriciais necessárias na sua implementação, exigindo a utilização de processadores digitais de elevada capacidade e velocidade de processamento.

Diante dos problemas de difícil sintonia e de tempo de execução do observador de fluxo com base no algoritmo EKF, uma solução é a utilização do observador de Luenberger, que baseia-se somente no modelo matemático da planta em questão, não sendo modeladas as características estocásticas dos sinais de ruído, sendo, por esta razão, considerado um observador do tipo determinístico. De maneira semelhante ao algoritmo EKF, este

observador pode ser aplicado na estimação do fluxo do rotor e da velocidade do motor de indução.

Neste capítulo, são apresentados os fundamentos teóricos essenciais do observador de Luenberger. Primeiramente, é apresentado o modelo dinâmico do motor no sistema de referência $\alpha\beta$ utilizando equações de estado. A partir do modelo matemático são derivadas as equações do Observador de Luenberger no domínio contínuo do tempo. Para o projeto do observador são apresentados dois esquemas: no primeiro esquema a matriz de realimentação de estado do observador é selecionada de maneira que os pólos resultantes do observador sejam proporcionais aos correspondentes pólos do motor em cada ponto de velocidade. A segunda abordagem é baseada no critério de amplificação e rotação dos pólos do motor em relação àqueles do motor. A seguir são apresentadas as equações do esquema adaptativo para a estimação da velocidade baseado no princípio de Modelo de Referência (MRAS). Uma análise da discretização do modelo do observador é apresentada, visando o estudo do comportamento da trajetória dos pólos no domínio discreto do tempo.

Por último, uma nova abordagem para o projeto do Observador de Luenberger é apresentada. O novo projeto tem como base a minimização de uma função de custo relacionada com o tempo de convergência das variáveis de estado estimadas a partir do observador em relação ao valor real do estado. Para resolver este problema de otimização foi utilizado um algoritmo genético, devido às vantagens deste método de otimização em comparação aos métodos convencionais. Como resultado, a matriz ótima de realimentação de estado é obtida para cada ponto de velocidade considerado. Para validar o método de projeto, resultados experimentais são apresentados a partir da implementação do observador num sistema de controle por orientação direta do fluxo de rotor.

4.2 Modelo matemático do motor de indução para implementação do observador de Luenberger

O observador de Luenberger para a estimação do fluxo de rotor descrito neste capítulo, utiliza o modelo matemático do motor de indução no sistema de referência estacionário definido pelas coordenadas α e β [4]:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\mathbf{x}\end{aligned}\tag{4.1}$$

onde:

$$\mathbf{A} = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r T_r} & \frac{L_m}{\sigma L_s L_r} \omega_r \\ 0 & -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & -\frac{L_m}{\sigma L_s L_r} \omega_r & \frac{L_m}{\sigma L_s L_r T_r} \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -\omega_r \\ 0 & \frac{L_m}{T_r} & \omega_r & -\frac{1}{T_r} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\tag{4.2}$$

sendo $\mathbf{x} = [i_{s\alpha} \ i_{s\beta} \ \lambda_{r\alpha} \ \lambda_{r\beta}]^T$, $\mathbf{u} = [v_{s\alpha} \ v_{s\beta}]^T$ e $\mathbf{y} = [i_{s\alpha} \ i_{s\beta}]^T$ os vetores de estado, entrada e de saída respectivamente.

Como pode ser observado, neste caso a velocidade não é considerada variável de estado, ao contrário do modelo utilizado na implementação do algoritmo EKF. Sendo assim, as equações (4.1) e (4.2) representam um modelo de quarta ordem. Quando o objetivo é eliminar o sensor de velocidade, algoritmos de estimação desta variável devem ser desenvolvidos e a matriz de estado do sistema atualizada.

Aproveitando a simetria presente nas matrizes do sistema e com o objetivo de simplificar a notação matemática, o modelo matemático representado pela equação (4.1),

pode ser escrito na forma complexa. Para isso adotamos as seguintes definições para o vetor de estado e as matrizes do sistema:

$$\mathbf{x} = \begin{bmatrix} \bar{i}_s \\ \bar{\lambda}_r \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & \frac{L_m}{\sigma L_s L_r} \left(\frac{1}{T_r} - j\omega_r\right) \\ \frac{L_m}{T_r} & -\left(\frac{1}{T_r} - j\omega_r\right) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{1}{\sigma L_s} \\ 0 \end{bmatrix}, \quad \mathbf{C} = [1 \ 0] \quad (4.3)$$

4.3 Formulação matemática do observador de Luenberger no domínio contínuo do tempo

A partir das expressões na forma real do modelo do motor (equações 4.1 e 4.2) obtêm-se as seguintes equações para o Observador de Luenberger [4]:

$$\begin{aligned} \dot{\hat{\mathbf{x}}} &= \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}(\mathbf{y} - \hat{\mathbf{y}}) \\ \hat{\mathbf{y}} &= \mathbf{C}\hat{\mathbf{x}} \end{aligned} \quad (4.4)$$

onde:

$$\mathbf{K} = \begin{bmatrix} k_{11} & -k_{12} \\ k_{12} & k_{11} \\ k_{21} & -k_{22} \\ k_{22} & k_{21} \end{bmatrix} \quad (4.5)$$

é a matriz de ganho do observador. Os valores k_{ij} da matriz \mathbf{K} são calculados de acordo com o critério de projeto adotado.

As equações do Observador de Luenberger podem ser escritas utilizando a forma complexa. Neste caso a matriz de ganho do observador é dada por:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix} = \begin{bmatrix} k_{11} + jk_{12} \\ k_{21} + jk_{22} \end{bmatrix} \quad (4.6)$$

A partir da equação de estado para o motor e a correspondente ao observador é possível obter as equações dinâmicas do erro entre as variáveis de estados reais e as estimadas:

$$\dot{\mathbf{e}} = \dot{\mathbf{x}} - \dot{\hat{\mathbf{x}}} = \mathbf{A}\mathbf{x} - \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} - \mathbf{B}\mathbf{u} - \mathbf{K}\mathbf{C}(\mathbf{x} - \hat{\mathbf{x}}) \quad (4.7)$$

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{K}\mathbf{C})\mathbf{e} = \mathbf{A}_{\text{obs}}\mathbf{e} \quad (4.8)$$

Da equação anterior, observa-se que o comportamento dinâmico do erro depende dos autovalores da matriz $(\mathbf{A} - \mathbf{K}\mathbf{C})$, comumente chamados de pólos do observador. Se a matriz $(\mathbf{A} - \mathbf{K}\mathbf{C})$ for uma matriz estável, o vetor de erro irá convergir para zero, independentemente do valor do erro inicial $\mathbf{e}(0)$. Isto quer dizer que, o vetor de estado estimado $\hat{\mathbf{x}}$ converge para o estado real \mathbf{x} para quaisquer condição inicial $\hat{\mathbf{x}}(0)$ e $\mathbf{x}(0)$. Se os autovalores de $(\mathbf{A} - \mathbf{K}\mathbf{C})$ forem selecionados de maneira adequada, será possível obter uma rápida convergência para zero do vetor de erro, independentemente da condição inicial $\mathbf{e}(0)$.

Uma vez definida a posição desejada dos pólos do observador, é possível determinar sua matriz de ganho \mathbf{K} . Normalmente, os pólos do observador são escolhidos de maneira que este responda mais rápido do que o sistema de malha fechada. Por outro lado, é necessário considerar que se for aumentada a velocidade de resposta dinâmica do erro, os efeitos negativos devidos às perturbações e ruídos no sistema aumentarão. Sendo assim, para determinar a matriz de ganho do observador é preciso examinar diversas matrizes \mathbf{K} baseadas em vários polinômios característicos desejados. Para cada uma destas matrizes, devem ser feitos testes de simulação e, então, escolher a matriz \mathbf{K} que fornece o melhor desempenho global na dinâmica do erro. Por conseguinte, a melhor matriz de ganho do observador deve representar uma solução de compromisso entre uma boa resposta dinâmica e baixa sensibilidade a perturbações e ruídos.

O polinômio característico do observador pode ser obtido em função dos elementos da matriz \mathbf{K} da seguinte maneira:

$$\bar{p}_{\text{obs}}(s) = \det(s\mathbf{I} - \mathbf{A}_{\text{obs}}) = \det(s\mathbf{I} - \mathbf{A} + \mathbf{K}\mathbf{C}) \quad (4.9)$$

Para simplificar o tratamento matemático, utilizamos a forma complexa da matriz do sistema que é expressa da seguinte maneira:

$$\mathbf{A} = \begin{bmatrix} a_1 & -\beta(a_2 + j\omega_r) \\ a_3 & a_2 + j\omega_r \end{bmatrix} \quad (4.10)$$

onde:

$$a_1 = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right), \quad a_2 = -\frac{1}{T_r}, \quad a_3 = \frac{L_m}{T_r}, \quad \beta = \frac{L_m}{\sigma L_s L_r} \quad (4.11)$$

Substituindo a equação (4.10) em (4.9) obtemos:

$$\bar{p}_{\text{obs}}(s) = \det \begin{bmatrix} s - a_1 + \mathbf{K}_1 & \beta(a_2 + j\omega_r) \\ -a_3 + \mathbf{K}_2 & s - (a_2 + j\omega_r) \end{bmatrix} \quad (4.12)$$

$$\bar{p}_{\text{obs}}(s) = s^2 + s(-a_1 + \mathbf{K}_1 - a_2 - j\omega_r) + [\beta(a_3 - \mathbf{K}_2)(a_2 + j\omega_r) + (a_1 - \mathbf{K}_1)(a_2 + j\omega_r)] \quad (4.13)$$

A partir da equação anterior, verifica-se que a posição dos pólos do observador depende dos elementos da matriz de ganho \mathbf{K} . Assim, para uma determinada posição dos pólos, é possível calcular a correspondente matriz \mathbf{K} . Para isto, igualam-se os coeficientes de expoentes iguais do polinômio característico do observador com os do polinômio característico desejado. O polinômio característico desejado pode ser escrito como:

$$\bar{p}_{\text{obs}}^*(s) = (s + \bar{p}_{o1})(s + \bar{p}_{o2}) = s^2 + s(\bar{p}_{o1} + \bar{p}_{o2}) + \bar{p}_{o1}\bar{p}_{o2} \quad (4.14)$$

onde \bar{p}_{o1} e \bar{p}_{o2} são as soluções complexas da equação de segundo grau.

A matriz de ganho \mathbf{K} do observador de estado pode ser obtida, também diretamente, através da fórmula de Ackerman, como segue [57]:

$$\mathbf{K} = \phi(\mathbf{A}) \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \cdot \\ \cdot \\ \mathbf{CA}^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 1 \end{bmatrix} \quad (4.15)$$

onde:

$$\phi(\mathbf{A}) = \mathbf{A}^n + \alpha_1 \mathbf{A}^{n-1} + \dots + \alpha_{n-1} \mathbf{A} + \alpha_n \mathbf{I} \quad (4.16)$$

e α_i são os coeficientes do polinômio característicos desejado.

É possível demonstrar que para o caso do sistema expresso na forma real, o polinômio característico resultante, de quarta ordem, pode ser expresso como:

$$p_{\text{obs}}(s) = \bar{p}_{\text{obs}}(s) \bar{p}_{\text{obs}}^*(s) = (s + \bar{p}_{o1})(s + \bar{p}_{o2})(s + \bar{p}_{o1}^*)(s + \bar{p}_{o2}^*) \quad (4.17)$$

onde \bar{p}_{o1}^* e \bar{p}_{o2}^* são os complexos conjugados das raízes \bar{p}_{o1} e \bar{p}_{o2} respectivamente.

A partir da representação do sistema na forma complexa é possível estudar a dinâmica do sistema na forma real, o qual simplifica em grande medida o tratamento matemático no projeto do observador.

4.3.1 Critérios de projeto

A seguir, são discutidas duas abordagens para o projeto do Observador de Luenberger. A primeira abordagem é baseada no critério de proporcionalidade entre os pólos do motor e os pólos do observador. A segunda utiliza amplificação e rotação dos pólos do motor para definir a posição dos pólos do observador.

Projeto do Observador de Luenberger baseado no critério de proporcionalidade

A expressão do polinômio característico do modelo do motor é obtido considerando $\mathbf{K}=0$ na expressão (4.9), sendo assim:

$$\bar{p}_{\text{mot}}(s) = \bar{p}_{\text{obs}}(s, \mathbf{K} = 0) = s^2 + s(-a_1 - a_2 - j\omega_r) + [\beta a_3(a_2 + j\omega_r) + a_1(a_2 + j\omega_r)] \quad (4.18)$$

$$\bar{p}_{\text{mot}}(s) = s^2 + s(\bar{p}_{m1} + \bar{p}_{m2}) + \bar{p}_{m1}\bar{p}_{m2} \quad (4.19)$$

Uma abordagem normalmente utilizada para o projeto do observador baseia-se no critério de proporcionalidade entre os pólos deste e os pólos do motor [14]. Para isso é necessário impor a seguinte condição:

$$\bar{p}_{o1,2} = \alpha \bar{p}_{m1,2} \quad (4.20)$$

A condição anterior leva ao seguinte polinômio característico para o observador:

$$\bar{p}_{\text{obs}}(s) = s^2 + \alpha s(\bar{p}_{m1} + \bar{p}_{m2}) + \alpha^2 \bar{p}_{m1}\bar{p}_{m2} \quad (4.21)$$

onde α é o fator de proporcionalidade escolhido.

Igualando os coeficientes dos termos de mesmo expoente nas duas representações do polinômio característico do observador, obtemos o seguinte sistema de equações:

$$\begin{cases} -a_1 + \mathbf{K}_1 - a_2 - j\omega_r = \alpha(-a_1 - a_2 - j\omega_r) \\ \beta(a_3 - \mathbf{K}_2)(a_2 + j\omega_r) + (a_1 - \mathbf{K}_1)(a_2 + j\omega) = \alpha^2[\beta a_3(a_2 + j\omega_r) + a_1(a_2 + j\omega_r)] \end{cases} \quad (4.22)$$

A partir da solução deste sistema de equações em termos de \mathbf{K}_1 e \mathbf{K}_2 , é possível verificar a dependência dos elementos da matriz de ganho do observador com os parâmetros do motor, com o fator de proporcionalidade α e com a velocidade mecânica do motor:

$$\mathbf{K}_1 = k_{11} + jk_{12} = (1 - \alpha)(a_1 + a_2 + j\omega_r) \quad (4.23)$$

$$\mathbf{K}_2 = k_{21} + jk_{22} = (1 - \alpha^2)(a_3 + \beta^{-1}a_1) - \beta^{-1}(1 - \alpha)(a_1 + a_2 + j\omega_r) \quad (4.24)$$

A partir das equações (4.23) e (4.24) são obtidos os seguintes valores para os coeficientes da matriz de ganho do observador.

$$\begin{cases} k_{11} = (1 - \alpha)(a_1 + a_2) \\ k_{12} = (1 - \alpha)\omega_r \\ k_{21} = (1 - \alpha^2)(a_3 + \beta^{-1}a_1) - \beta^{-1}(1 - \alpha)(a_1 + a_2) \\ k_{22} = -\beta^{-1}(1 - \alpha)\omega_r \end{cases} \quad (4.25)$$

Da solução do sistema de equações observa-se que dos quatro elementos da matriz de ganho do observador, dois deles são proporcionais à velocidade do motor e os outros dois são constantes.

Projeto do Observador de Luenberger baseado no critério de amplificação e rotação

Ao invés de uma pura proporcionalidade, pode-se desejar que entre os pólos do observador e os pólos do motor exista uma relação de ganho e ângulo para cada ponto de velocidade (amplificação e rotação), o qual conduz a uma resposta dinâmica mais rápida e ao mesmo tempo amortecida. Esta condição pode ser imposta através da seguinte equação:

$$\bar{p}_{o1,2} = \alpha e^{j\theta} \bar{p}_{m1,2} \quad (4.26)$$

que conduz ao seguinte polinômio característico:

$$\bar{p}_{obs}(s) = s^2 + \alpha e^{j\theta} s(\bar{p}_{m1} + \bar{p}_{m2}) + \alpha^2 e^{j2\theta} \bar{p}_{m1} \bar{p}_{m2} \quad (4.27)$$

Definindo a relação entre os pólos do observador e do motor, pode-se obter os elementos da matriz de ganho e seguindo o mesmo procedimento do caso proporcional, chega-se à seguinte solução na forma complexa:

$$\mathbf{K}_1 = k_{11} + jk_{12} = (1 - \alpha \cos \theta - j\alpha \sin \theta)(a_1 + a_2 + j\omega_r) \quad (4.28)$$

$$\begin{aligned} \mathbf{K}_2 = k_{21} + jk_{22} = & [1 - \alpha^2 (\cos \theta + j \sin \theta)^2] (a_3 + \beta^{-1} a_1) - \\ & - \beta^{-1} (1 - \alpha \cos \theta - j\alpha \sin \theta) (a_1 + a_2 + j\omega_r) \end{aligned} \quad (4.29)$$

Explicitando as partes real e imaginária da solução, obtemos:

$$\begin{cases} k_{11} = (1 - \alpha \cos\theta)(a_1 + a_2) + \omega_r \alpha \sin\theta \\ k_{12} = (1 - \alpha \cos\theta)\omega_r - \alpha(a_1 + a_2) \sin\theta \\ k_{21} = (1 - \alpha^2 \cos^2\theta + \alpha^2 \sin^2\theta)(a_3 + \beta^{-1}a_1) - \beta^{-1}[(1 - \alpha \cos\theta)(a_1 + a_2) + \alpha\omega_r \sin\theta] \\ k_{22} = -\beta^{-1}[(1 - \alpha \cos\theta)\omega_r - \alpha(a_1 + a_2) \sin\theta] - \alpha^2(a_3 + \beta^{-1}a_1) \sin(2\theta) \end{cases} \quad (4.30)$$

Como pode ser observado, quando o critério de amplificação e rotação é utilizado, os quatro coeficientes da matriz de ganho do observador dependem da velocidade mecânica do motor.

4.4 Esquema adaptativo para estimação de velocidade

O projeto do observador de Luenberger foi estudado até este momento considerando que a informação referente à velocidade mecânica do motor é disponível para o cálculo das equações do observador. Quando a velocidade do motor não está disponível, é necessário obter um esquema adaptativo para estimar a velocidade e com isso, atualizar as matrizes do observador. O esquema adaptativo de estimação de velocidade pode ser obtido utilizando o princípio de Modelo de Referência (MRAS).

A partir das equações dinâmicas de estado do sistema motor - observador, obtemos a seguinte equação para o erro considerando a velocidade como parâmetro a estimar:

$$\dot{\mathbf{e}} = (\mathbf{A} - \mathbf{KC})\mathbf{e} - \Delta\mathbf{A}\hat{\mathbf{x}} \quad (4.31)$$

na qual :

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}, \quad \Delta\mathbf{A} = \hat{\mathbf{A}} - \mathbf{A} = \begin{bmatrix} \mathbf{O}_2 & -\frac{L_m}{\sigma L_s L_r}(\hat{\omega}_r - \omega_r)\mathbf{J} \\ \mathbf{O}_2 & (\hat{\omega}_r - \omega_r)\mathbf{J} \end{bmatrix}, \quad \mathbf{O}_2 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (4.32)$$

Para obter o esquema adaptativo de estimação de velocidade, consideramos a seguinte função de Lyapunov:

$$V = \mathbf{e}^T \mathbf{e} + (\hat{\omega}_r - \omega_r)^2 / \lambda \quad (4.33)$$

onde λ é uma constante positiva.

A derivada da função de Lyapunov em relação ao tempo pode ser expressa como:

$$\dot{V} = \mathbf{e}\mathbf{e}^T + \mathbf{e}^T\dot{\mathbf{e}} + 2\hat{\omega}_r \frac{(\hat{\omega}_r - \omega_r)}{\lambda} \quad (4.34)$$

Substituindo (4.32) na equação (4.34) obtemos:

$$\dot{V} = \mathbf{e}^T \{(\mathbf{A} - \mathbf{K}\mathbf{C})^T + (\mathbf{A} - \mathbf{K}\mathbf{C})\} \mathbf{e} - (\hat{\mathbf{x}}^T \Delta \mathbf{A}^T \mathbf{e} + \mathbf{e}^T \Delta \mathbf{A} \hat{\mathbf{x}}) + 2\hat{\omega}_r \frac{(\hat{\omega}_r - \omega_r)}{\lambda} \quad (4.35)$$

Desenvolvendo o segundo termo da equação anterior obtemos:

$$\dot{V} = \mathbf{e}^T \{(\mathbf{A} - \mathbf{K}\mathbf{C})^T + (\mathbf{A} - \mathbf{K}\mathbf{C})\} \mathbf{e} - 2 \frac{L_m}{\sigma L_s L_r} (\hat{\omega}_r - \omega_r) (e_{is\alpha} \hat{\lambda}_{r\beta} - e_{is\beta} \hat{\lambda}_{r\alpha}) + 2 \frac{(\hat{\omega}_r - \omega_r) \hat{\omega}_r}{\lambda} \quad (4.36)$$

onde: $e_{is\alpha} = i_{s\alpha} - \hat{i}_{s\alpha}$, $e_{is\beta} = i_{s\beta} - \hat{i}_{s\beta}$

Igualando o segundo e terceiro da equação (4.36) é possível obter a equação que representa o esquema adaptativo para a estimação da velocidade:

$$\hat{\omega}_r = \frac{L_m}{\sigma L_s L_r} \lambda (e_{is\alpha} \hat{\lambda}_{r\beta} - e_{is\beta} \hat{\lambda}_{r\alpha}) \quad (4.37)$$

Se a matriz de ganho \mathbf{K} do observador é selecionada de maneira que o primeiro termo da derivada da função de Lyapunov seja negativo - definido, o esquema adaptativo do observador será estável. Com o objetivo de melhorar o desempenho dinâmico do estimador de velocidade, ao termo integral resultante da equação anterior é somado um termo proporcional, ou seja:

$$\hat{\omega}_r = K_p (e_{is\alpha} \hat{\lambda}_{r\beta} - e_{is\beta} \hat{\lambda}_{r\alpha}) + K_I \int (e_{is\alpha} \hat{\lambda}_{r\beta} - e_{is\beta} \hat{\lambda}_{r\alpha}) dt \quad (4.38)$$

onde: K_p e K_I são as constantes proporcional e integral do esquema adaptativo.

A Figura 4.1 mostra um diagrama de blocos da estimação de velocidade com base no Modelo de Referência.

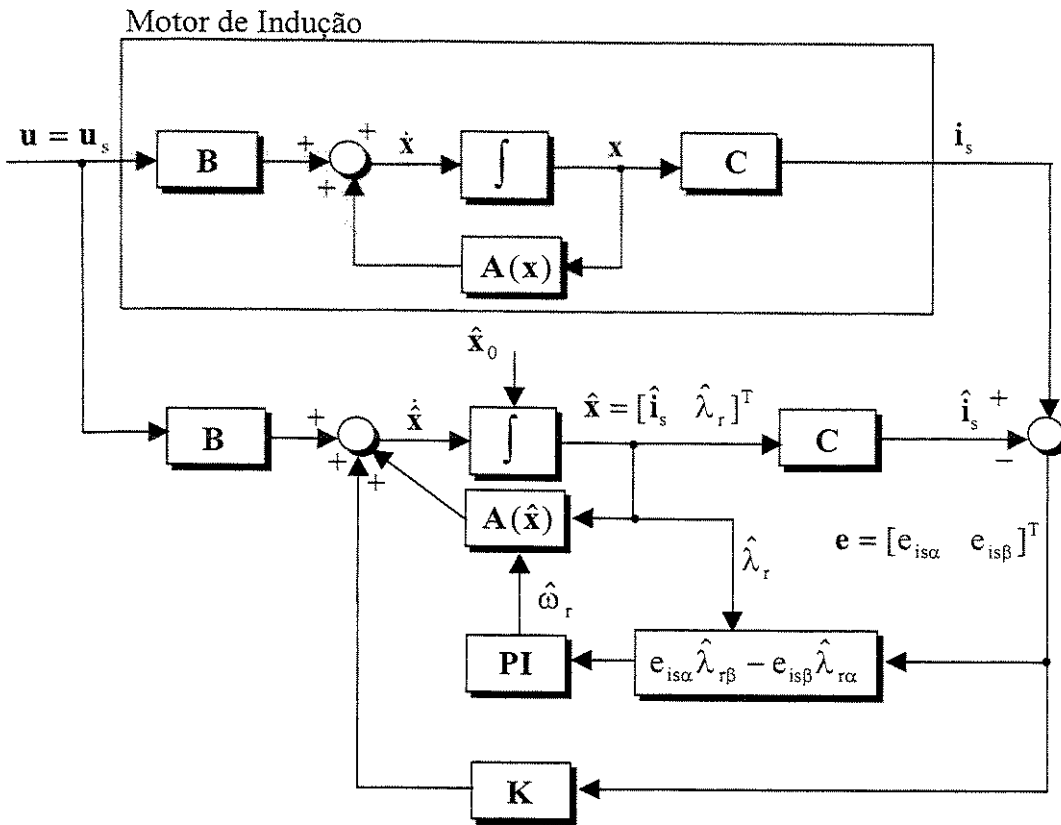


Figura 4.1 – Diagrama de blocos da estimação de velocidade com base em MRAS

4.5 Projeto do observador de Luenberger no domínio discreto do tempo

O observador de Luenberger no domínio discreto do tempo pode ser representado através da seguinte equação [21]:

$$\hat{\mathbf{x}}(k+1) = (\mathbf{A}_d - \mathbf{K}_d \mathbf{C})\hat{\mathbf{x}} + \mathbf{B}_d \mathbf{u}(k) + \mathbf{K}_d \mathbf{C}\mathbf{x}(k) \quad (4.39)$$

na qual $\hat{\mathbf{x}}(k)$ é o vetor de estado estimado pelo observador, $\mathbf{x}(k)$ o vetor de estado real e $\mathbf{u}(k)$ o vetor de entrada do sistema no instante k de amostragem. Para o cálculo das matrizes discretas é utilizada a expansão em Série de Taylor como mostrado no capítulo anterior.

De maneira geral, quando é utilizada a aproximação baseada na Série de Taylor, são utilizados somente os termos de primeira e de segunda ordem. Aproximações de ordem superior ($n > 2$) requerem um grande esforço computacional e exigem processadores digitais de elevada capacidade de processamento. Por outro lado, quando são utilizadas aproximações de ordem inferior pode acontecer que os pólos da matriz do observador ($\mathbf{A} - \mathbf{K}\mathbf{C}$) abandonem a trajetória desejada na medida que a velocidade do motor aumenta, podendo alcançar a área de instabilidade.

Quando é utilizada a expansão até primeira ordem da Série de Taylor para o cálculo das matrizes discretas do Observador de Luenberger, obtém-se a seguinte expressão [21]:

$$\mathbf{x}(k+1) = [\mathbf{I} + (\mathbf{A}(\omega_r) - \mathbf{K}(\omega_r)\mathbf{C})T]\mathbf{x}(k) + \mathbf{B}T\mathbf{u}(k) + \mathbf{K}(\omega_r)\mathbf{C}\mathbf{y}(k) \quad (4.40)$$

onde T é o período de amostragem.

A Figura 4.2 mostra a trajetória dos pólos do observador e do motor quando é utilizada uma aproximação de primeira ordem. O cálculo da matriz \mathbf{K} do observador foi baseado no critério de proporcionalidade. Pode-se observar que, no caso de se utilizar uma aproximação de primeira ordem, a proporcionalidade entre os pólos do motor e do observador é mantida somente para baixas velocidades. À medida que a velocidade aumenta perde-se a proporcionalidade e os pólos do observador alcançam a área de instabilidade.

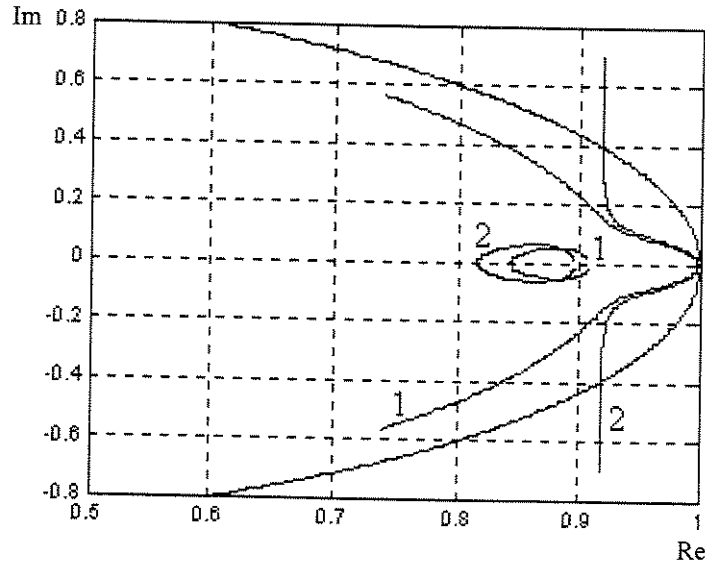


Figura 4.2 - Trajetórias dos pólos do motor (1) e do Observador de Luenberger (2) utilizando aproximação de primeira ordem no cálculo das matrizes discretas: 0-2.000 rad/sec.

Quando é utilizada a expansão até segunda ordem da Série de Taylor para o cálculo das matrizes discretas do observador de Luenberger, obtemos a seguinte equação [21]:

$$\begin{aligned} \mathbf{x}(k+1) = & (\mathbf{I} + \mathbf{A}'(\omega_r)T + \mathbf{A}'(\omega_r)^2 \frac{T^2}{2})\mathbf{x}(k) + (\mathbf{B}T + \mathbf{A}'(\omega_r)\mathbf{B} \frac{T^2}{2})\mathbf{u}(k) + \\ & + (\mathbf{K}(\omega_r)\mathbf{C}T + \mathbf{A}'(\omega_r)\mathbf{K}(\omega_r)\mathbf{C} \frac{T^2}{2})\mathbf{y}(k) \end{aligned} \quad (4.41)$$

A Figura 4.3 mostra a trajetória dos pólos do observador e do motor quando as matrizes discretas do observador são calculadas utilizando uma aproximação de segunda ordem da Série de Taylor. A partir desta figura, podemos concluir que utilizando uma aproximação de segunda ordem a condição de proporcionalidade permanece válida para uma banda maior da velocidade e somente para velocidades muito elevadas os pólos do observador abandonam a trajetória desejada aproximando-se rapidamente da área de instabilidade.

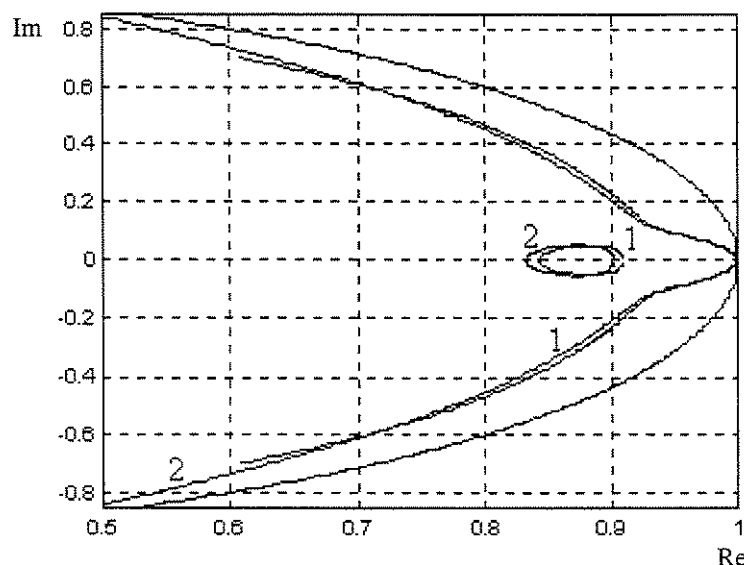


Figura 4.3 - Trajetórias dos pólos do motor (1) e do Observador de Luenberger (2) utilizando aproximação de segunda ordem no cálculo das matrizes discretas: 0-2.000 rad/sec.

4.6 Otimização do observador de Luenberger utilizando algoritmos genéticos

Como foi comentado, a abordagem convencional utilizada para especificar a posição dos pólos do observador é baseada no critério de proporcionalidade. Uma outra abordagem estudada para o posicionamento dos pólos utiliza o critério de amplificação e rotação dos pólos do observador em relação àqueles do motor. Esta estratégia permite utilizar aproximações de primeira ordem numa maior faixa de velocidade, mantendo os pólos dentro da área de estabilidade. As duas abordagens estudadas podem ser consideradas aceitáveis do ponto de vista de resposta dinâmica do erro, ou seja, na rapidez de convergência das variáveis de estado estimadas aos valores das variáveis reais. Por outro lado, em nenhum dos casos é analisado se a posição selecionada para os pólos do observador baseado no critério de projeto é a ótima para o ponto de velocidade considerado.

Recentemente tem aumentado a aplicação de Algoritmos Genéticos (GA) na área de otimização de sistemas. GA são algoritmos de busca baseados no mecanismos da seleção e da genética natural [38]. Quando utilizados como método de otimização somente é necessário como informação o valor da função objetivo avaliada no conjunto de parâmetros

do problema, dispensando informações de derivadas, as quais dependendo do problema estudado são difíceis de serem obtidas. Estas, sem dúvida, são as principais vantagens dos algoritmos genéticos em relação aos métodos tradicionais de otimização. Esta propriedade tem sido aproveitada por poucos pesquisadores na área de controle de máquinas elétricas. Pillay et al [39] tem aplicado GA para obter os parâmetros do motor de indução a partir dos dados de placa do motor assim como, também, no problema de determinação de eficiência [40]. Em outro trabalho, Souza et al [41] utilizam GA para a determinação dos parâmetros do motor de indução a partir dos valores de corrente e potência obtidos nos testes de laboratório. Em [42] é reportada a utilização de GA num sistema adaptativo de controle vetorial para o motor de indução baseado na identificação da constante de tempo do rotor.

Esta seção apresenta uma nova abordagem para o problema de seleção da trajetória ótima dos pólos do observador de Luenberger. O método proposto utiliza os resultados da otimização para obter a matriz de retração de estado do observador em cada ponto de velocidade. O método de otimização, baseado em algoritmos genéticos minimiza a função de custo relacionada com a dinâmica do erro de estimação das variáveis de estado. Como resultado, a matriz ótima de realimentação de estado é obtida, a qual determinará a posição ótima dos pólos em cada ponto de velocidade considerado.

4.6.1 Formulação do problema de otimização

O procedimento para selecionar a posição dos pólos para o observador de Luenberger pode ser escrito como um problema de otimização utilizando uma função de custo apropriada. Esta função de custo pode ser construída a partir das equações dinâmicas de erro do Observador Discreto de Luenberger:

$$\mathbf{e}(k+1) = \mathbf{x}(k+1) - \hat{\mathbf{x}}(k+1) = (\mathbf{A}_d - \mathbf{K}_d \mathbf{C})\mathbf{e}(k) \quad (4.42)$$

Com o objetivo de construir a função de custo CF define-se, primeiramente, a norma do vetor erro no instante k de amostragem, como:

$$\|\mathbf{e}(k)\| = \sqrt{\mathbf{e}(k)^T \mathbf{e}(k)} \quad (4.43)$$

Definimos agora a função de custo tentando englobar características da integral do erro, e o termo relacionado com a convergência da estimação:

$$CF = h \cdot n + \sum_{k=1}^n \|\mathbf{e}(k)\| \quad (4.44)$$

onde n é o valor discreto do tempo necessário pela norma do erro para alcançar o valor de parada especificado e h é uma constante selecionada com o objetivo de ponderar o peso relativo dos dois termos que aparecem na função de custo (CF).

Os valores nominais, correspondentes a cada uma das variáveis de estado do sistema, são utilizados como valores iniciais do erro para realizar o cálculo da função de custo. Uma vez definida a nossa função de custo, o problema de otimização pode ser formulado da seguinte maneira:

$$\begin{aligned} \min. \quad & CF \\ \text{s.a: } & k_{ji} < k_i < k_{ui} \quad i = 1,2,3,4 \end{aligned}$$

Os limites superior k_{ui} e inferior k_{ji} dos coeficientes da matriz de ganho do observador podem ser definidos seguindo critérios práticos de projeto e devem mudar em função da faixa de velocidade. Assim, por exemplo, para baixas velocidades ganhos altos não devem ser utilizados, para evitar os efeitos negativos do ruído no processo de estimação.

4.6.2 Breve revisão de algoritmos genéticos (GA)

GA são algoritmos de busca baseados nos mecanismos da seleção e a genética natural [38]. Quando utilizados como método de otimização somente é necessária como informação o valor da função objetivo avaliada no conjunto de parâmetros do problema, dispensando informações de derivadas, as quais, a depender do problema são difíceis de serem obtidas. Isto representa uma grande vantagem em relação aos métodos tradicionais de otimização.

Para a execução de um algoritmo genético, os parâmetros do problema de otimização devem ser codificados utilizando cadeias definidas sob um alfabeto finito. Como exemplo, consideremos o seguinte problema de otimização:

$$\max. \quad y = f(x), \quad x \in [0,31]$$

Para a codificação dos parâmetros do problema, geralmente é utilizada a codificação binária. Assim, em nosso exemplo, o parâmetro x é codificado utilizando cadeias binárias de comprimento 5, como mostrado a seguir:

Tabela 4.1- Codificação binária.

| Parâmetro x | Cadeia Binária |
|---------------|----------------|
| 0 | 00000 |
| 1 | 00001 |
| . | |
| . | |
| 30 | 11110 |
| 31 | 11111 |

Uma vez codificados os parâmetros do problema, é selecionada de maneira aleatória, uma população inicial de indivíduos, sobre os quais são aplicados sucessivamente os operadores genéticos: reprodução, cruzamento e mutação. Como resultado da aplicação destes operadores é gerada uma nova população. A nova população gerada desta maneira, pode ser considerada uma população mais “forte” quando comparada com a população inicial, no sentido que os indivíduos desta nova população contribuem em maior grau para a solução global do problema. Ou seja, o valor médio da função objetivo avaliada na nova população é maior em relação ao valor médio da população inicial.

Reprodução é um processo no qual os indivíduos são selecionados de acordo com o seu valor na função objetivo (“fitness”). Isto significa que aqueles indivíduos com maior valor na função objetivo terão uma maior probabilidade de contribuir com uma ou mais cópias na próxima geração. A regra frequentemente utilizada para efetuar a reprodução é a chamada "roulette wheel selection". Nesta regra, o número de cópias esperadas para cada um dos indivíduos na próxima geração pode ser expressa como:

$$nc_j = n \frac{f(x_j)}{\sum_{i=1}^n f(x_i)} \quad (4.45)$$

onde n é o comprimento da população e $f(x_i)$ é o valor da função objetivo avaliada para o elemento i .

Uma vez terminada a reprodução, os indivíduos resultantes são colocados na chamada “Mating Pool”, onde esperam serem selecionados para a aplicação dos restantes operadores genéticos. A Tabela 4.2 mostra um exemplo do operador de reprodução para uma população de 4 elementos.

Tabela 4.2- Operador de reprodução.

| i | Parâmetro x | Cadeia | $f(x)$ | n_c | Mating Pool |
|-----|---------------|--------|--------|-------|-------------|
| 1 | 13 | 01101 | 0 | 0 | 11011 |
| 2 | 27 | 11011 | 50 | 2 | 11011 |
| 3 | 30 | 11110 | 25 | 1 | 11110 |
| 4 | 23 | 10111 | 25 | 1 | 10111 |

Cruzamento é um operador executado em duas etapas. Primeiramente são selecionados de maneira aleatória um par de indivíduos a partir daqueles obtidos do resultado da reprodução. A seguir, no caso de cruzamento simples, uma posição aleatória é selecionada e, a informação binária contida nos indivíduos selecionados é trocada a partir desta posição até o final da cadeia. Esta operação é repetida sobre a população resultante da reprodução. Um exemplo do operador de cruzamento é mostrado na figura 4.4, onde as cadeias 1-2 e 3-4 foram selecionadas aleatoriamente para efetuar o cruzamento. Como resultado da aplicação deste operador obtêm-se os novos indivíduos $1'$, $2'$, $3'$ e $4'$.

O último operador genético a ser aplicado é o operador de mutação. Para isto, é selecionado um número de cadeias igual ao produto do número de indivíduos na população pela probabilidade de mutação. A partir daí, para cada cadeia é escolhida aleatoriamente uma posição de bit e então, a informação binária correspondente a esta posição é mudada de 0 para 1 ou vice-versa. Usualmente, a probabilidade de mutação é bem reduzida. O operador de mutação evita a perda de informação genética útil e é considerado um mecanismo secundário na operação do algoritmo genético. Um exemplo do operador de mutação é mostrado na figura 4.5. O processo de otimização pode terminar quando o valor médio da

função de custo na última geração é maior que um certo valor $1/\epsilon$, então o indivíduo com maior valor da função de custo é selecionado como a solução do problema de otimização.

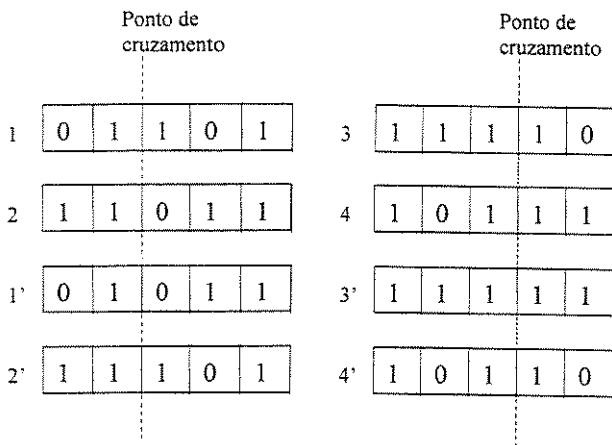


Figura 4.4 - Operador de cruzamento.

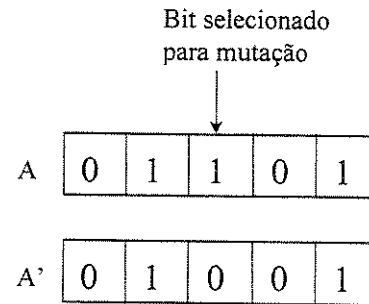


Figura 4.5 - Operador de mutação.

4.6.3 Solução ao problema de otimização utilizando algoritmos genéticos

O objetivo do algoritmo genético é minimizar a função de custo CF . Para uma correta operação do algoritmo é necessário transformar o problema de minimização em um problema de maximização equivalente. Isto pode ser feito como segue:

$$\begin{aligned} \max. \text{ Fitness} &= 1/(CF + c) \\ \text{s.a: } k_{li} &< k_i < k_{ui} \quad i = 1,2,3,4 \end{aligned}$$

onde k_{li} , k_{ui} definem os intervalos do espaço de busca e c é uma constante utilizada para garantir a adequada operação dos operadores genéticos.

Cada parâmetro do problema de otimização, os quais em nosso caso são os coeficientes k_1 , k_2 , k_3 e k_4 , da matriz de retração de estado \mathbf{K} , é codificado como uma cadeia binária de comprimento l_i , sendo então concatenados para formar o indivíduo da população ou seja, cada indivíduo dentro da população possui comprimento igual a: $\sum_{i=1}^4 l_i$. A precisão desta codificação binária pode ser expressa como: $(k_{ui} - k_{li}) / (2^{l_i} - 1)$. Cada indivíduo da

população foi representado por uma cadeia binária de 44 bits, a qual pode ser dividida em quatro sub-cadeias de comprimentos 10, 10, 12 e 12, correspondentes aos parâmetros k_1 , k_2 , k_3 e k_4 respectivamente.

Uma vez definida a codificação dos parâmetros do problema de otimização, os operadores genéticos de reprodução, cruzamento e mutação podem ser aplicados sob a seqüência de gerações. O processo de otimização continua até que o valor médio da função de custo na última geração seja maior que um certo valor $1/\epsilon$. O indivíduo com maior valor da função de custo é selecionado como a solução do processo de otimização. Na tabela 4.3 são mostrados os parâmetros de otimização utilizados para a execução do algoritmo genético.

Tabela 4.3 – Parâmetros de otimização

| Parâmetro | Valor |
|---|-------|
| Pmutation (probabilidade de mutação) | 0.01 |
| Pcross (probabilidade de cruzamento) | 0.8 |
| Popsiz (total da população) | 40 |

4.7 Resultados Experimentais

O método proposto para o projeto do observador de Luenberger foi testado experimentalmente num sistema de controle por orientação direta do fluxo de rotor. Os parâmetros do motor utilizados nos testes aparecem no Apêndice A deste trabalho. O sistema de controle por orientação, o observador de fluxo de rotor e o sistema de estimação de velocidade baseado no modelo de referência, foram implementados no processador digital de sinais DSP TMS320F240 da Texas Instruments. A Figura 4.6 mostra um diagrama de blocos do sistema implementado.

O modelo matemático do motor foi discretizado utilizando a aproximação de primeira ordem. O programa de controle e as rotinas de entrada-saída foram desenvolvidas utilizando a linguagem *assembler* do DSP. A execução do algoritmo de controle foi sincronizada com

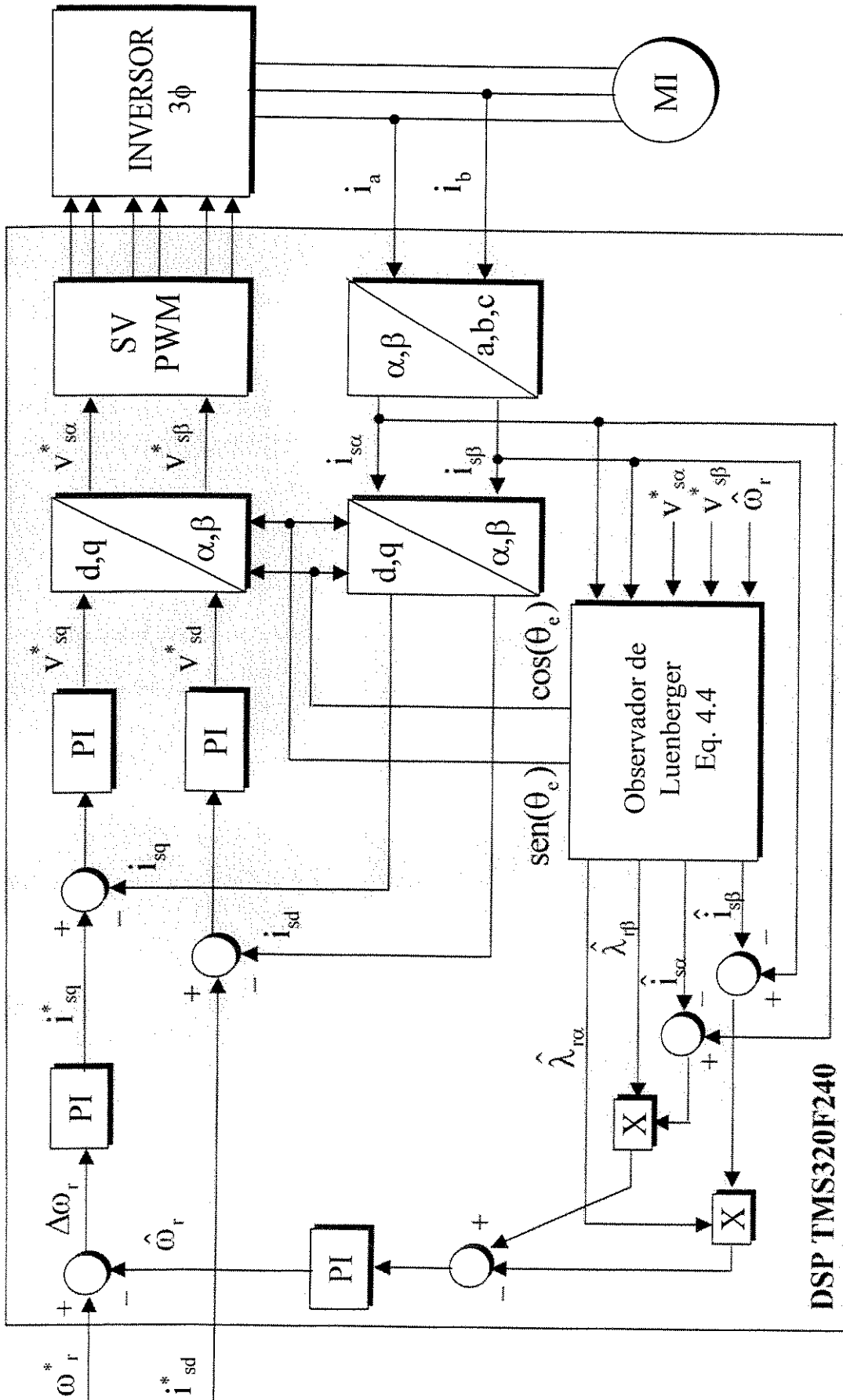


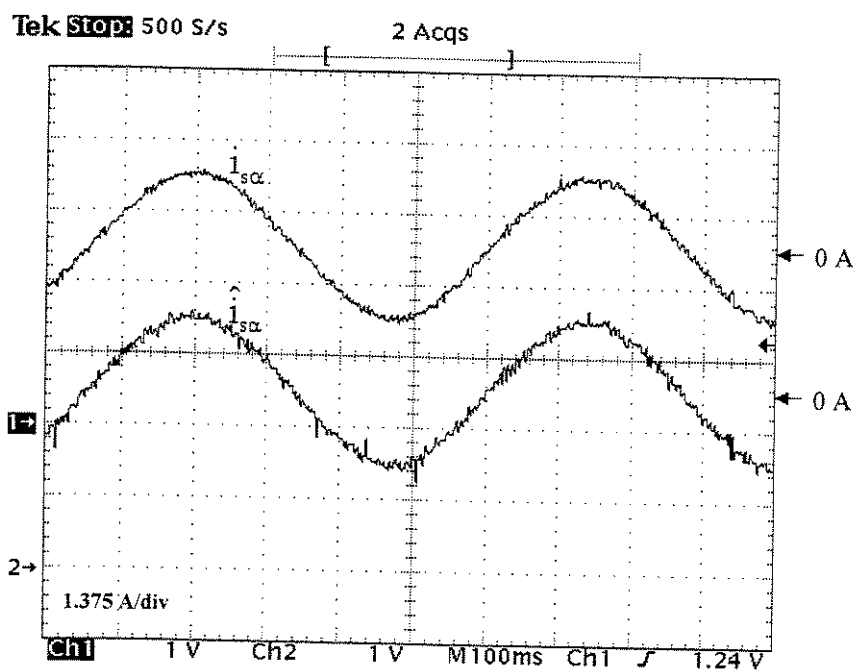
Figura 4.6 - Diagrama de blocos do sistema de controle utilizando o observador de Luenberger

o período do sinal de PWM, sendo executada a cada $200 \mu\text{s}$ (5 kHz). O tempo de execução do algoritmo de controle, incluindo a aquisição dos sinais e o observador foi de aproximadamente $60 \mu\text{s}$, sendo de $20 \mu\text{s}$ o tempo aproximado de execução do observador de Luenberger. É importante observar a grande diminuição no tempo de execução do algoritmo do observador quando comparado com o algoritmo EKF. Durante o desenvolvimento do algoritmo de controle, um encoder incremental de 1200 pulsos/rev foi utilizado com objetivo de comparar a velocidade medida com o valor estimado a partir do observador. Na obtenção das curvas experimentais foi utilizado um osciloscópio digital TDS 360 da Tektronix conectado aos canais selecionados do conversor digital–analógico da placa EVM do DSP.

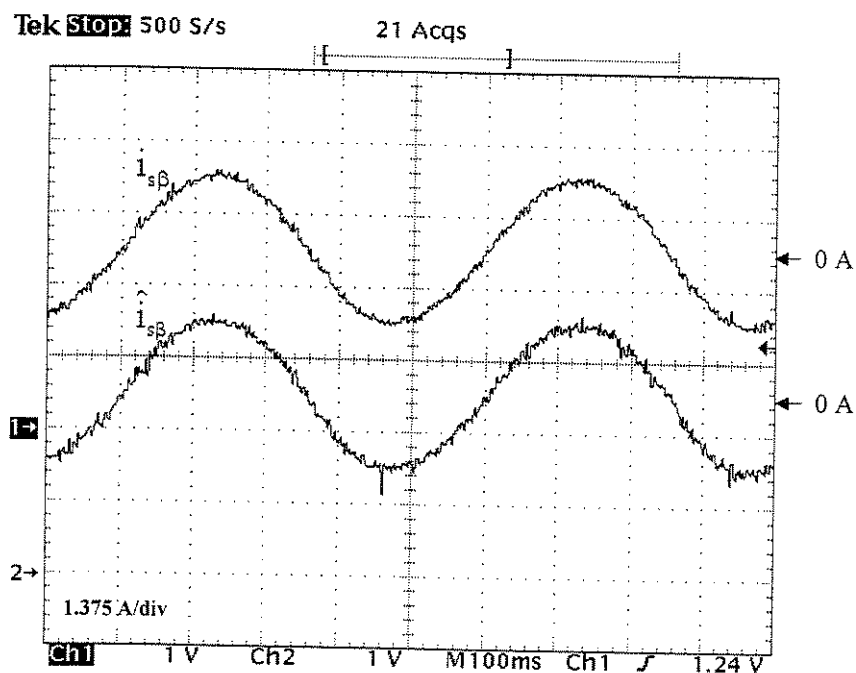
Como no caso do capítulo anterior, a apresentação dos resultados experimentais é dividida em duas partes. Primeiramente são mostrados resultados referentes à estimação das componentes de fluxo de rotor e correntes de estator em estado de regime permanente para diferentes velocidades de operação do motor. A segunda parte apresenta resultados do funcionamento do sistema durante o transitório, sendo aplicadas ao motor referências variáveis de velocidade mecânica e da componente da corrente de conjugado.

4.7.1 Testes em regime permanente

A Figura 4.7 mostra as curvas experimentais das componentes reais da corrente de estator, assim como os valores estimados a partir do observador para uma velocidade de operação de 70 rpm. A Figura 4.8 apresenta resultados da estimação das componentes do vetor fluxo de rotor para a mesma velocidade de operação. De maneira similar ao que foi feito para o caso do algoritmo EKF, foram obtidos resultados em regime permanente do observador de Luenberger para outras velocidades de operação. As Figuras 4.9 e 4.10 mostram o resultado do teste para uma velocidade de 900 rpm e as Figuras 4.11 e 4.12 para 1.100 rpm. A correspondência entre os valores estimados das componentes da corrente de estator e os valores reais fica evidente a partir dos resultados. Por outra parte as componentes estimadas do fluxo de rotor coincidem com o valor de referência especificado no controle, além de ser observada a defasagem de $\pi/2$ radianos elétricos entre estas.

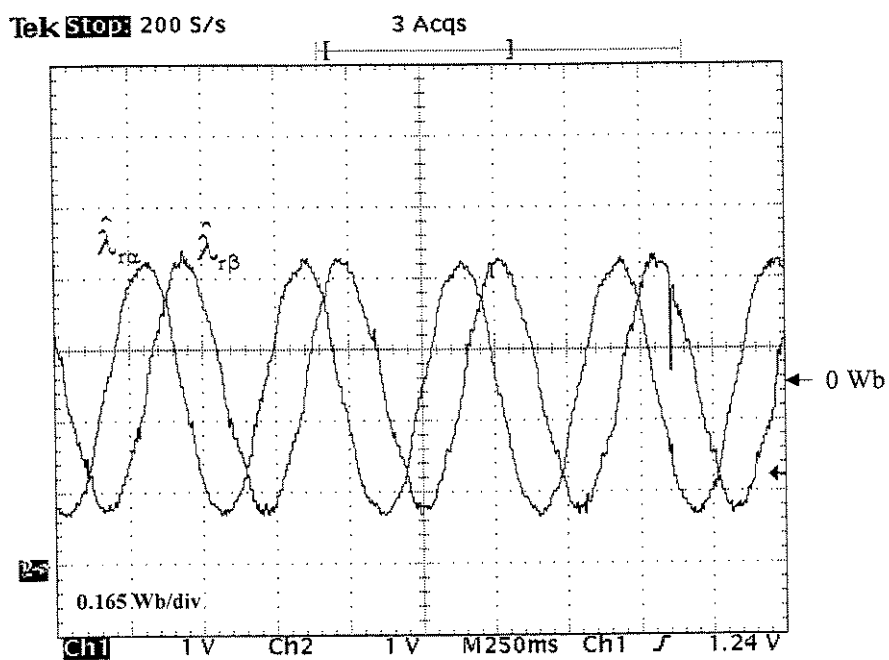


(a)

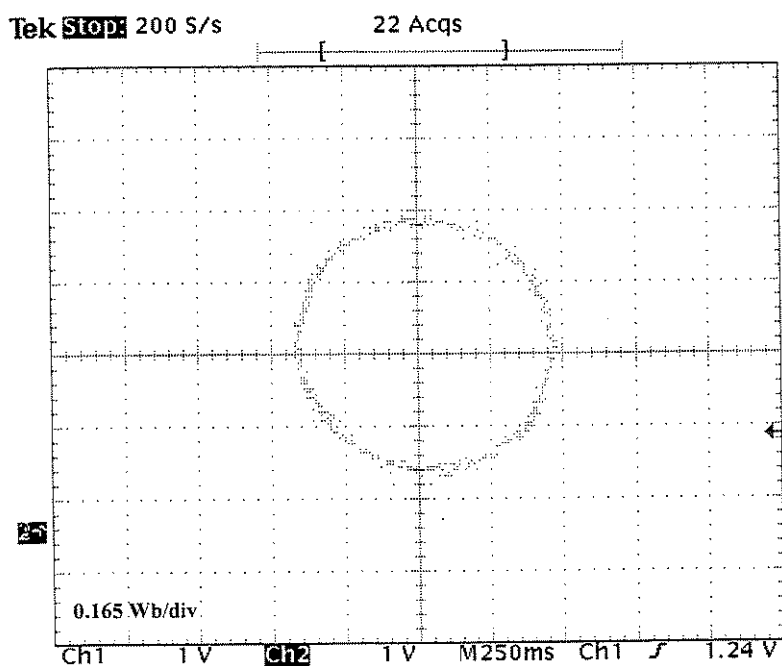


(b)

Figura 4.7 - Correntes reais e estimadas para $\omega_r^* = 70$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$



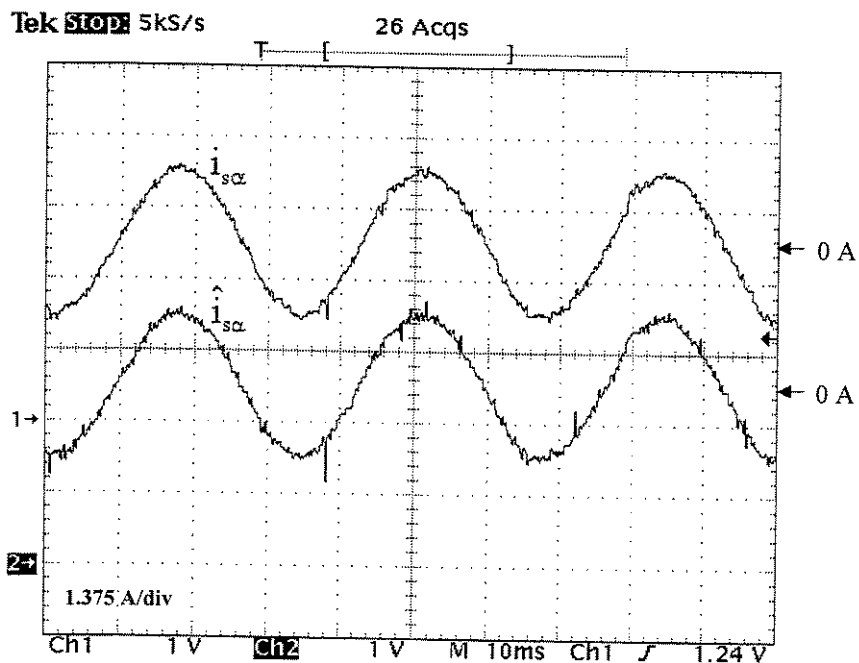
(a)



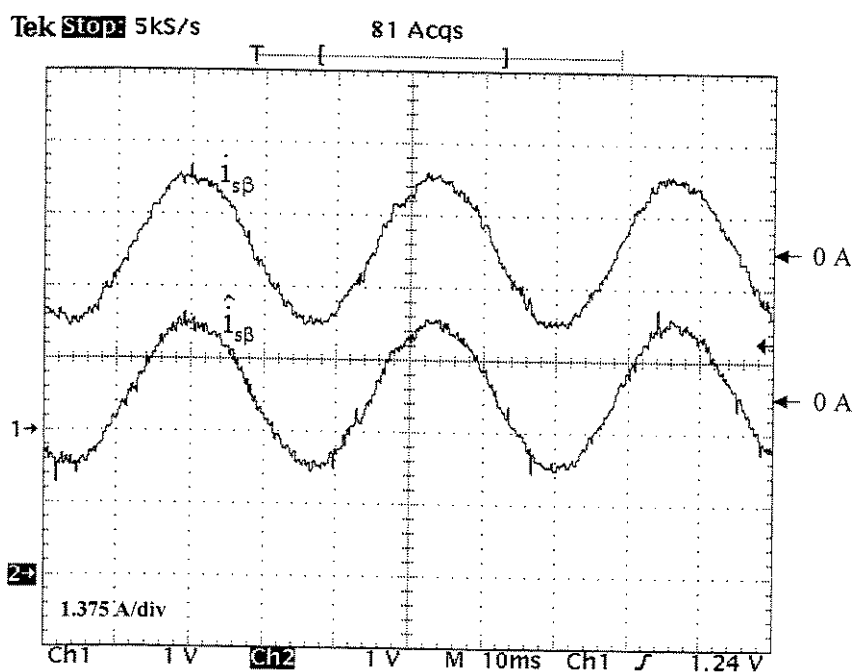
(b)

Figura 4.8 - Componentes estimadas do fluxo de rotor para $\omega_r^* = 70$ rpm (a) $\hat{\lambda}_{r\alpha}$, $\hat{\lambda}_{r\beta}$

(b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

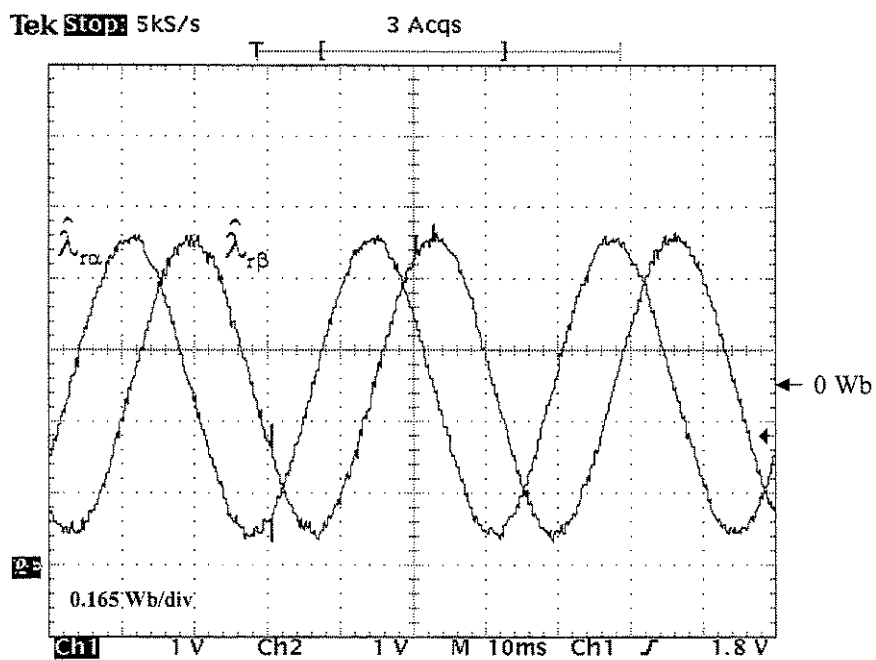


(a)

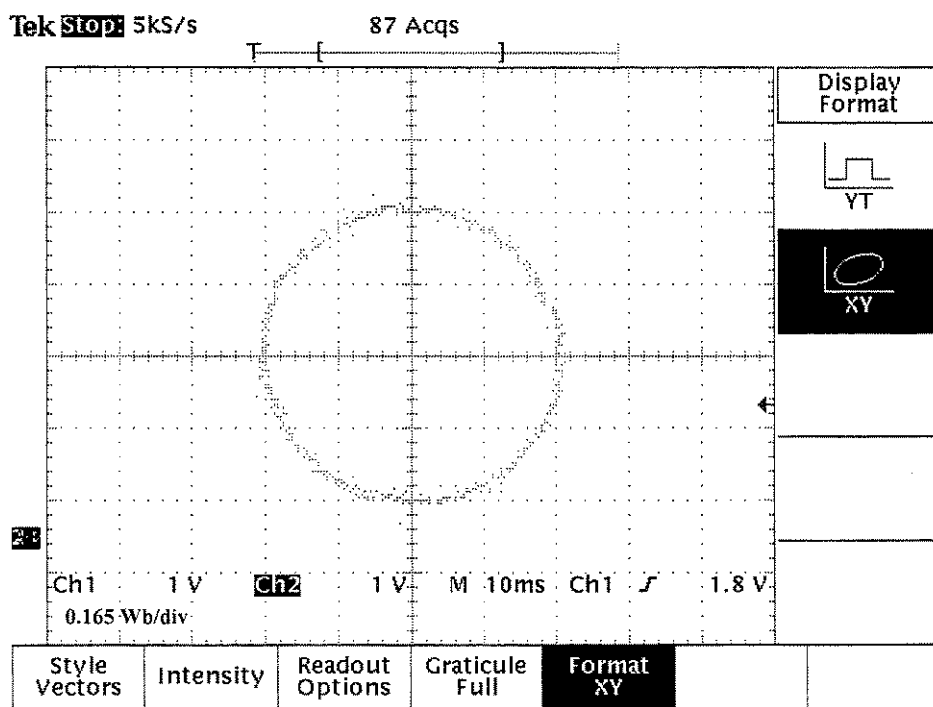


(b)

Figura 4.9 - Correntes reais e estimadas para $\omega_r^* = 900$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$

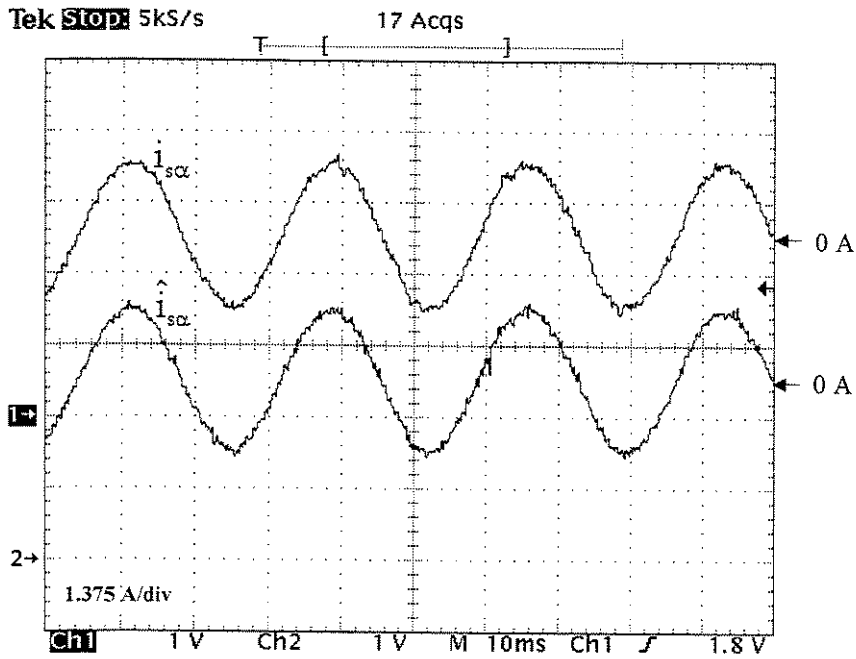


(a)

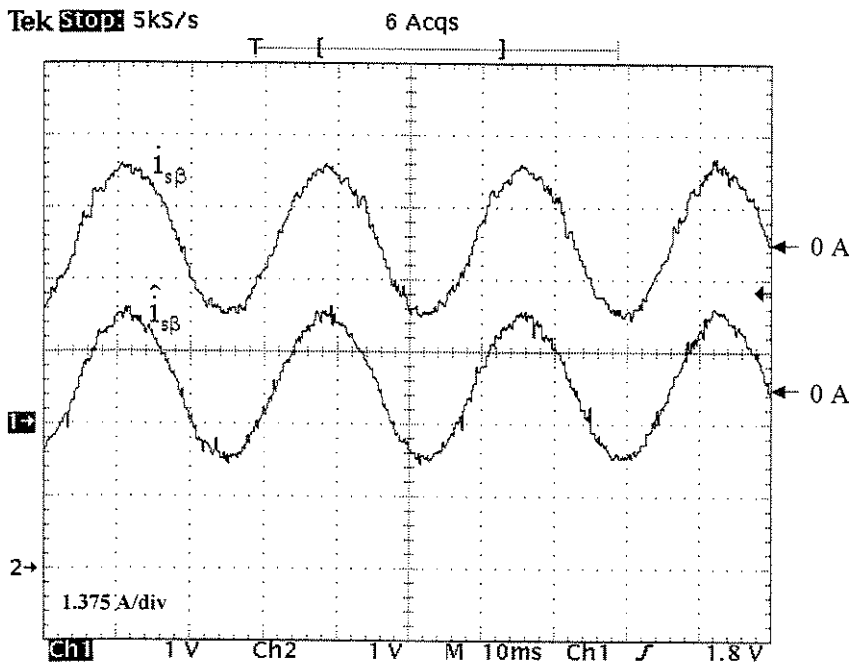


(b)

Figura 4.10 - Componentes estimadas do fluxo de rotor para $\omega_r^* = 900$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$
 (b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

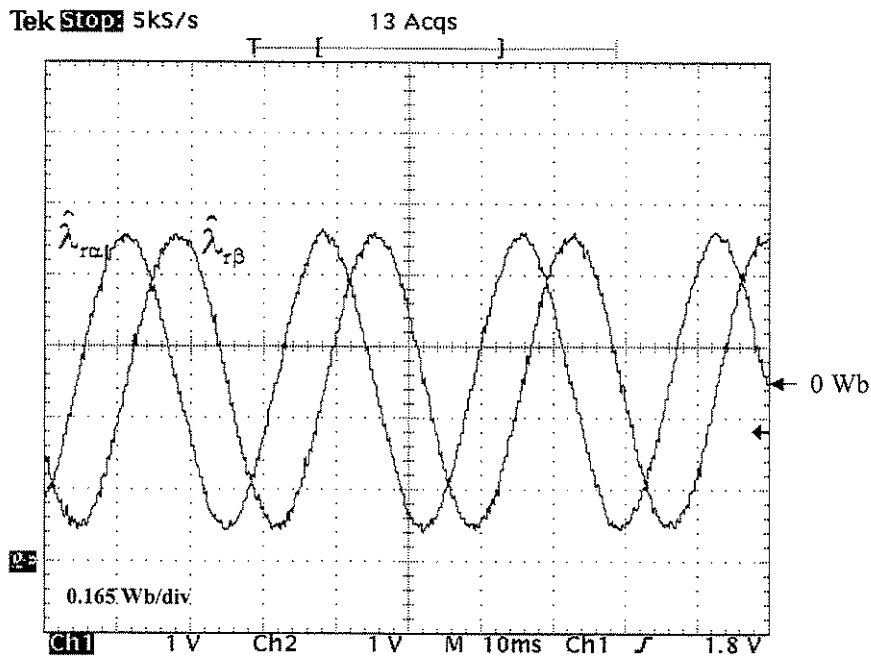


(a)

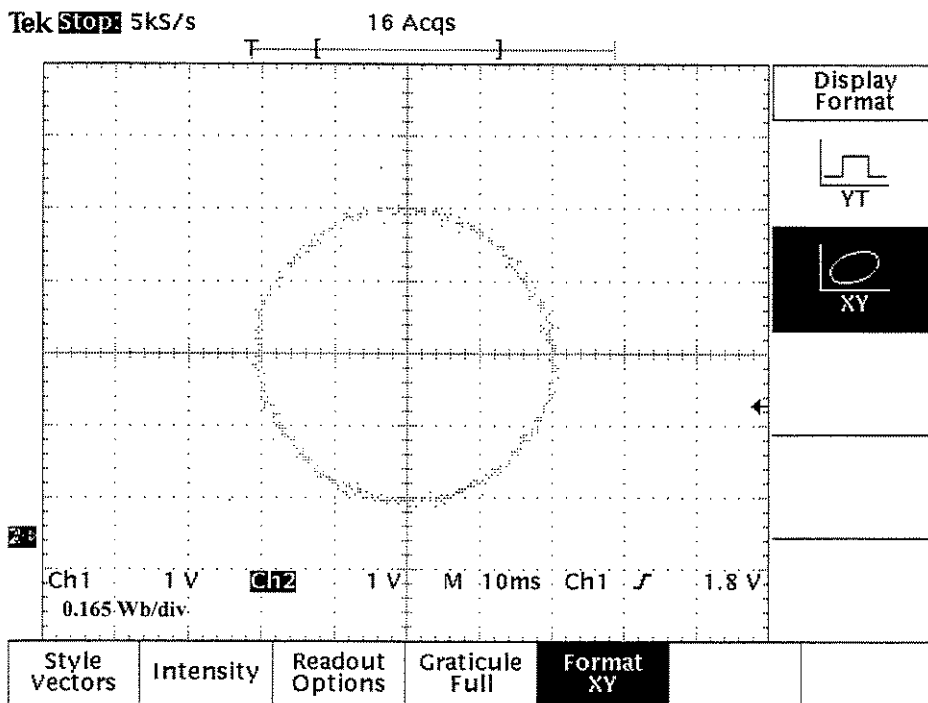


(b)

Figura 4.11 - Correntes reais e estimadas para $\omega_r^* = 1.100$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$



(a)



(b)

Figura 4.12 - Componentes estimadas do fluxo de rotor para $\omega_r^* = 1.100$ rpm (a) $\hat{\lambda}_{r\alpha}, \hat{\lambda}_{r\beta}$

(b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

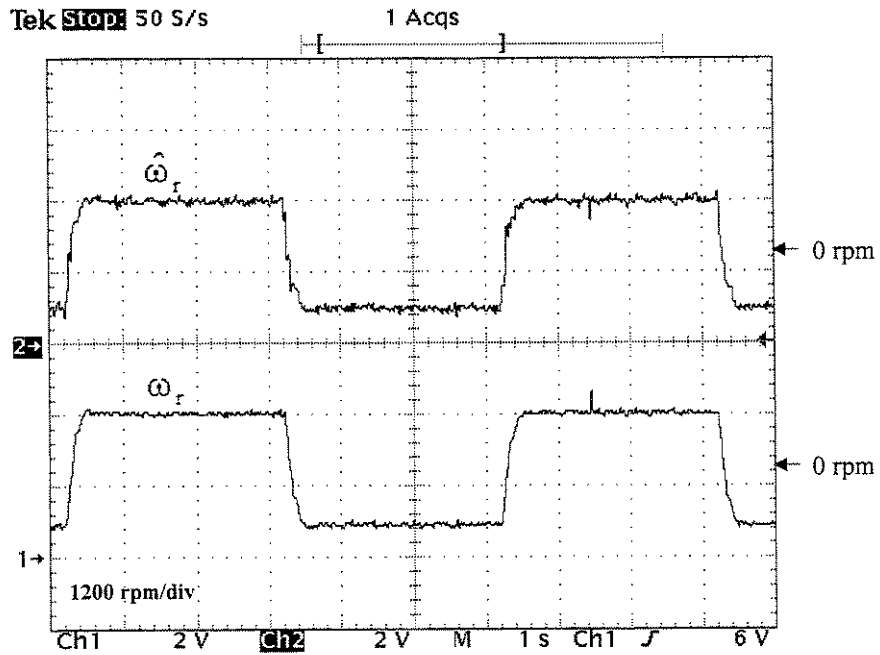
4.7.2 Testes em regime transitório

Para testar o desempenho do sistema de controle durante o transitório, referências variáveis de velocidade e da componente de conjugado da corrente de estator foram aplicadas.

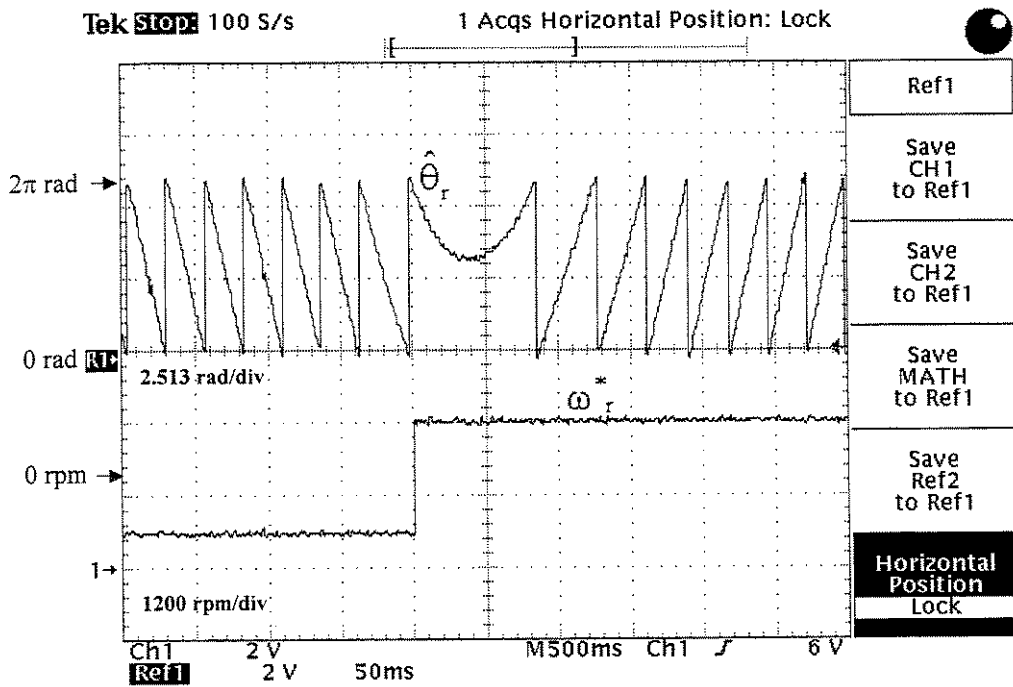
A Figura 4.13(a) mostra a velocidade mecânica medida e a estimada quando é aplicada uma referência variável de velocidade do tipo onda quadrada de 900 a -900 rpm. Na Figura 4.13(b) é mostrado o sinal de referência de velocidade e a variação do ângulo elétrico do vetor fluxo de rotor durante o transitório. A partir dos resultados, verifica-se como a velocidade estimada pelo observador segue com boa aproximação o valor medido.

Os testes de aceleração – desaceleração com inversão de velocidade foram feitos para duas faixas de velocidade: -900 até 900 rpm e de -450 até 450 rpm. No primeiro caso, o tempo de espera entre a aceleração e desaceleração foi de 3 seg e no segundo de 2.5 seg. As Figuras 4.14 até 4.17 mostram as curvas de velocidade medida e estimada, assim como aquelas referentes às componentes reais e estimadas da corrente de estator durante o transitório para os dois testes. A partir das figuras verifica-se como a velocidade estimada pelo observador segue com precisão o valor medido. Este fato é também verificado para o caso das componentes da corrente de estator, verificando-se o excelente desempenho do observador durante o transitório.

O sinal de onda quadrada na componente de conjugado da corrente de estator i_{sq}^* foi aplicado e a Figura 4.18 apresenta as curvas experimentais da velocidade com a referência de corrente indo de 0.5 até -0.5 pu com um período de 300 ms.

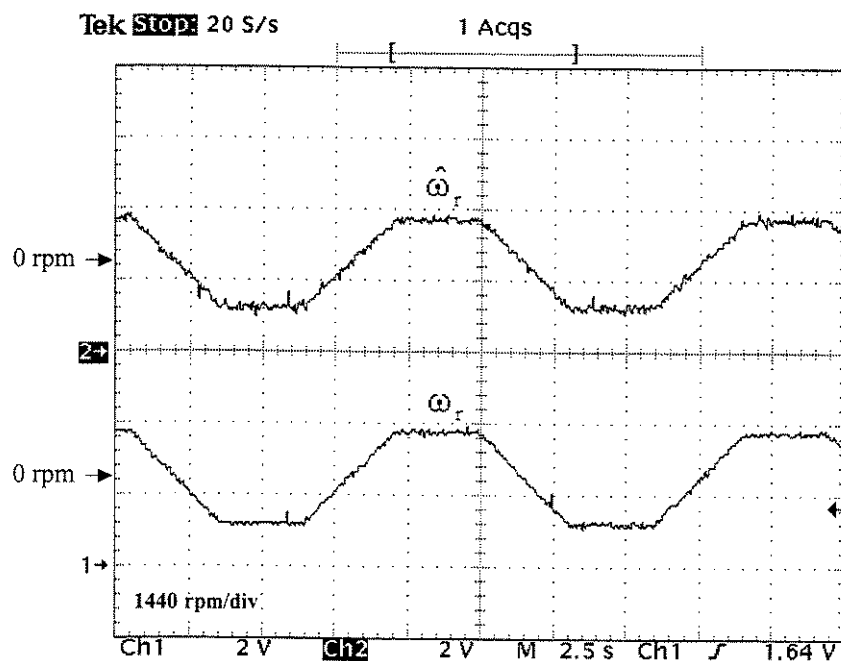


(a)

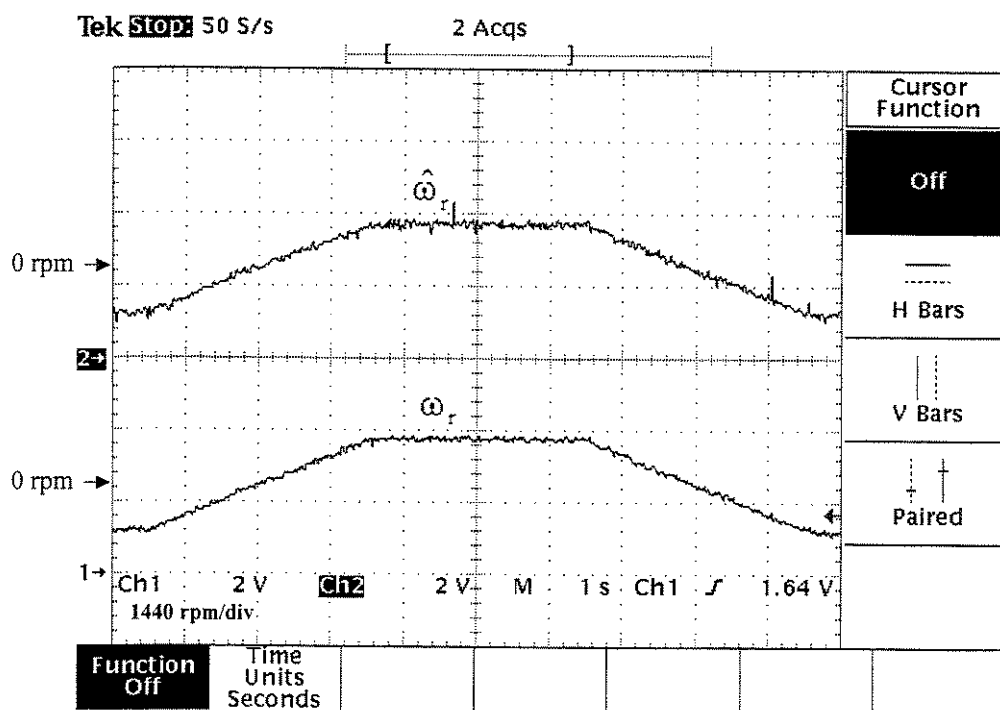


(b)

Figura 4.13 - Reversão de velocidade de -900 rpm a 900 rpm (a) $\omega_r^*, \omega_r, \hat{\omega}_r$ (b) ω_r^*, θ_r

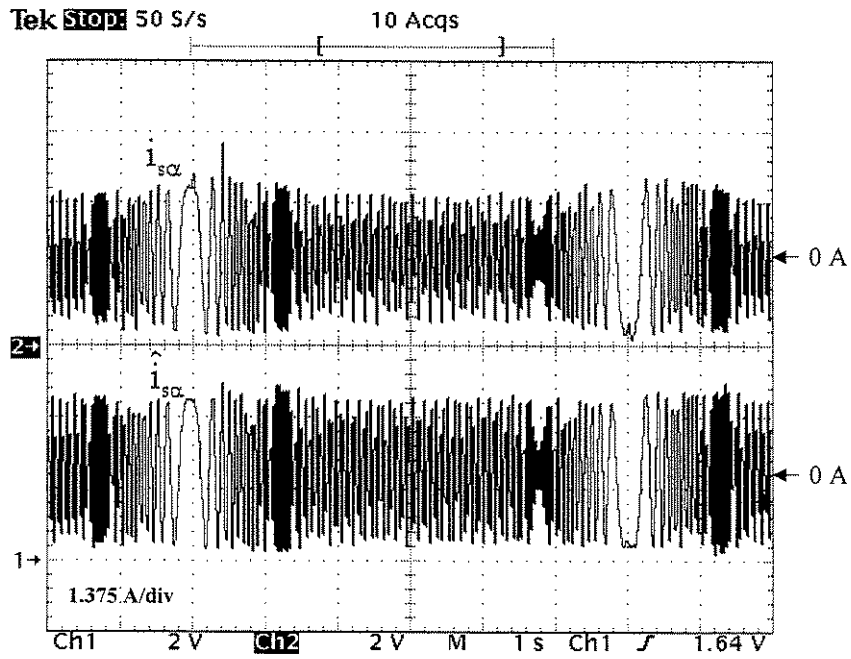


(a)

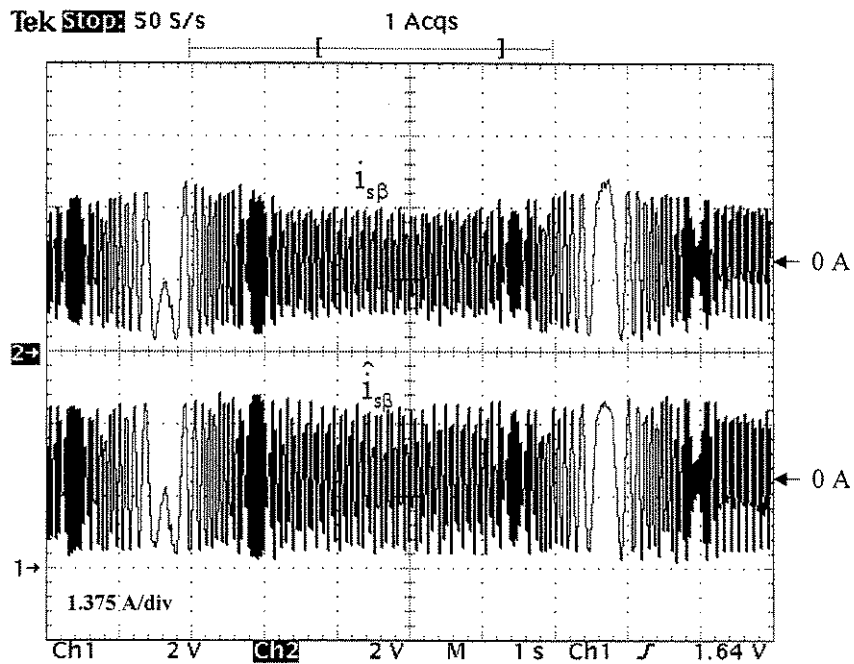


(b)

Figura 4.14 - Teste de aceleração desaceleração de -900 rpm a 900 rpm (a) $\omega_r, \hat{\omega}_r$ (2.5 s/div)
 (b) $\omega_r, \hat{\omega}_r$ (1 s/div)

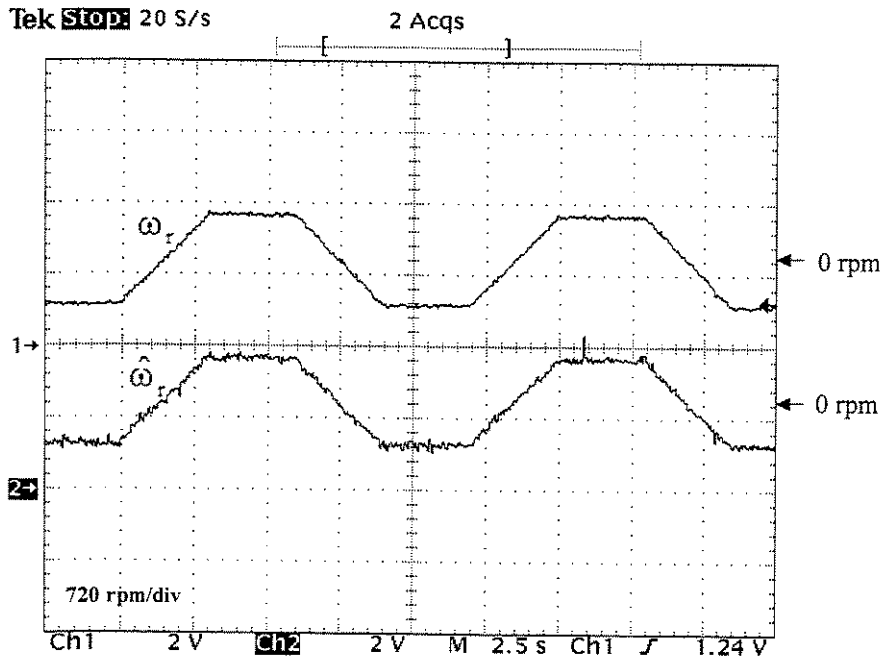


(a)

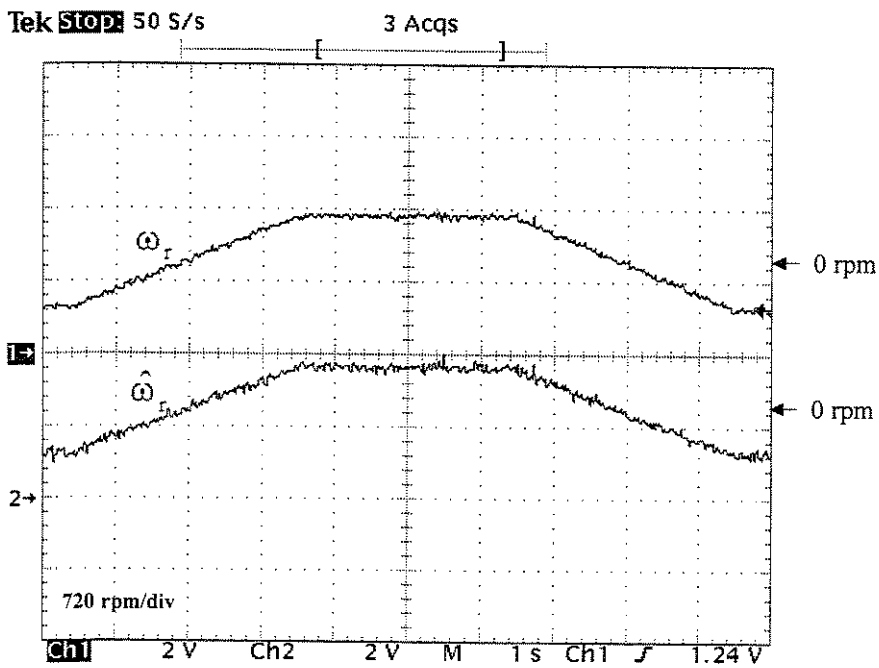


(b)

Figura 4.15 - Correntes reais e estimadas para no teste de aceleração desaceleração de -900 a 900 rpm (a) $i_{s\alpha}$, $\hat{i}_{s\alpha}$ (b) $i_{s\beta}$, $\hat{i}_{s\beta}$

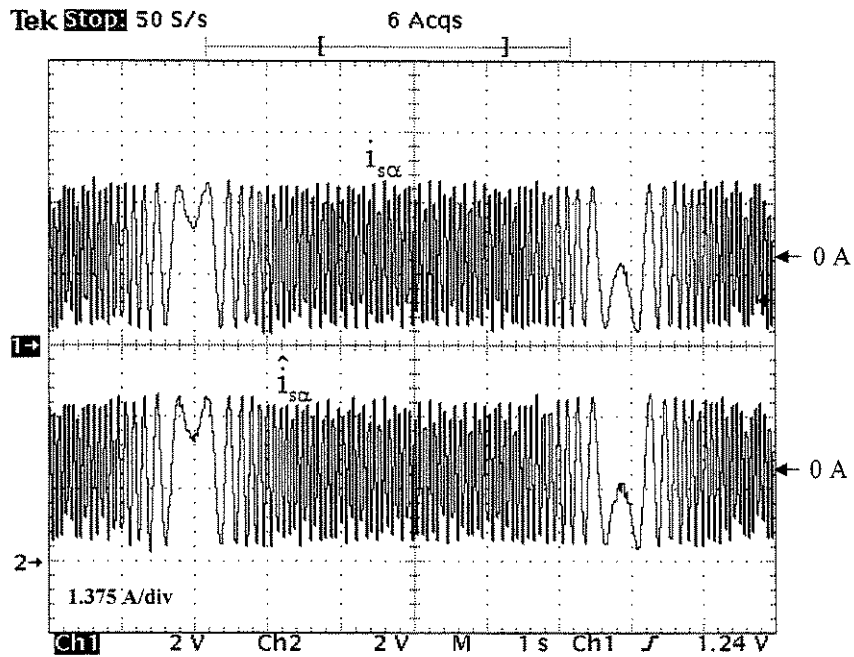


(a)

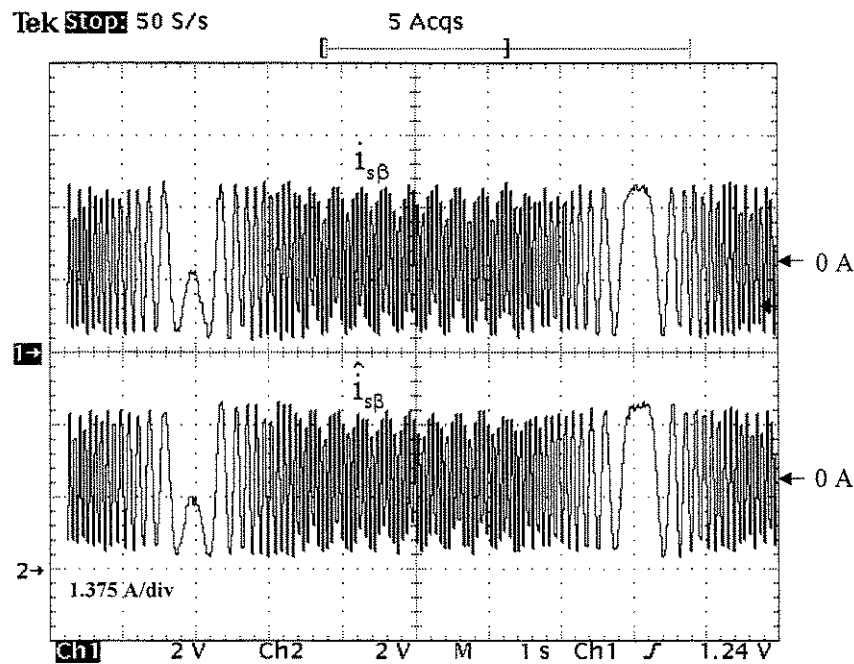


(b)

Figura 4.16 - Teste de aceleração desaceleração de -450 rpm a 450 rpm (a) $\omega_r, \hat{\omega}_r$ (2.5 s/div)
 (b) $\omega_r, \hat{\omega}_r$ (1 s/div)



(a)



(b)

Figura 4.17 - Correntes reais e estimadas para o teste de aceleração desaceleração de -450 rpm a 450 rpm (a) i_{sc} , \hat{i}_{sc} (b) $i_{s\beta}$, $\hat{i}_{s\beta}$

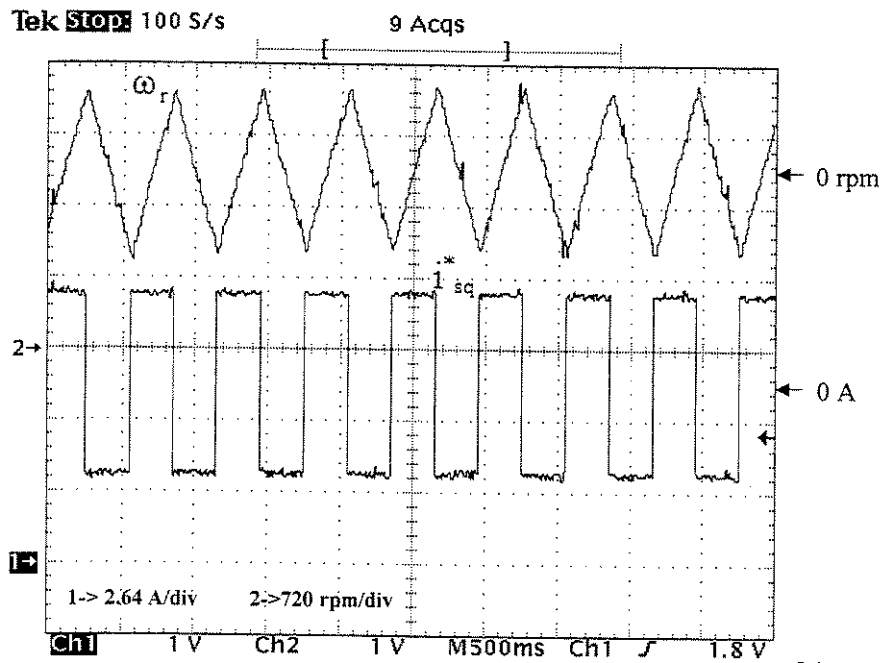


Figura 4.18 - Teste de onda quadrada na referência i_{sq}^* de 0.5 a -0.5 pu

Capítulo 5

Observador adaptativo de fluxo de rotor em modos deslizantes

5.1 Introdução

Nos capítulos 3 e 4 foram estudados e implementados os observadores de Kalman e de Luenberger, para estimação do fluxo de rotor em máquinas de indução. O termo de correção utilizado em cada um destes observadores foi uma função contínua do erro entre as componentes reais e estimadas da corrente de estator. No caso do observador de fluxo com base no filtro de Kalman, foram salientados os problemas de sintonia do observador, assim como a complexidade computacional do algoritmo de estimação. Visando a solução destes problemas, foi estudado e implementado o observador de Luenberger a partir do qual, dois critérios de projeto foram analisados e apresentada uma proposta para a obtenção da matriz ótima de retroação de estado do observador com base na minimização de uma função de custo associada à dinâmica do erro de estimação.

As características inerentes da técnica dos modos deslizantes quanto à baixa sensibilidade à variação paramétrica e a sua alta robustez aos distúrbios de medição, [35,37,45] têm sido a principal motivação na aplicação desta técnica no projeto de sistemas de controle. Inicialmente, esta técnica foi utilizada somente no controle de sistemas e a partir dos trabalhos de Misawa e Hedrick (1979), estendida sua aplicação à observação de estados. Recentemente, os observadores com base em modos deslizantes vêm sendo aplicados na estimação de fluxo, velocidade e posição em máquinas CA [32,33,47]. Diferentemente da

utilização de modos deslizantes no controle de sistemas, na observação de estados não existem limitações físicas em relação aos ganhos de chaveamento. A principal restrição na aplicação desta técnica está na presença de oscilações inerentes aos sistemas de estrutura variável. Este fenômeno, conhecido como *chattering*, pode levar a excitar dinâmicas de alta frequência não modeladas, limitando a utilização das variáveis estimadas no sistema de controle. Novas técnicas para redução de *chattering* e a aplicação destas aos problemas de controle e observação de estados têm sido atualmente estudadas [48,49,50].

Neste capítulo são apresentados os fundamentos teóricos do observador de fluxo de rotor em modos deslizantes. Inicialmente, são abordados os princípios dos sistemas de estrutura variável em modos deslizantes. A partir do modelo matemático do motor são obtidas as equações do observador de fluxo de rotor e os critérios utilizados para o projeto deste observador são apresentados, assim como também o algoritmo adaptativo de estimação de velocidade. Em seguida, é proposto um algoritmo para redução de *chattering* utilizando Lógica Fuzzy, no qual os ganhos do observador são calculados em função da dinâmica do erro de estimação das componentes da corrente de estator. No final, resultados experimentais são apresentados para avaliar o desempenho do observador no sistema de controle por orientação direta do fluxo de rotor.

5.2 Breve histórico de sistemas de estrutura variável

Os conceitos iniciais de sistemas de estrutura variável (SEV) foram introduzidos na década de 50 pelos pesquisadores russos S. V. Emelyanov e A. I. Fedotova na análise de sistemas lineares de segunda ordem com realimentação da saída e controle descontínuo.

Numa primeira etapa de desenvolvimento dos sistemas de estrutura variável, que abrange o período de 1957 a 1970 [45], foi estudada somente a aplicação desta técnica em sistemas lineares com entrada de controle escalar. Estes sistemas eram modelados utilizando equações diferenciais lineares ou através da sua forma canônica controlável. Além disso, para a compensação das perturbações externas eram utilizadas realimentações das variáveis de saída e de suas derivadas. Durante este período foram feitos diversos estudos das condições de existência e estabilidade dos modos deslizantes, dos sistemas com parâmetros variantes no tempo, dos efeitos das perturbações externas e incertezas paramétricas, assim como também dos sistemas com variáveis de estado não mensuráveis.

No período de 1970 a 1980, a teoria dos SEV foi estendida para uma classe mais geral de sistemas lineares. Porém, os avanços teóricos obtidos durante este período não foram aproveitados em muitas aplicações práticas. O principal motivo foi a maior aceitação das bem estabelecidas técnicas lineares de controle existentes, além de que as propriedades de robustez dos SEV não foram reconhecidas totalmente [45].

A partir de 1980 as pesquisas teóricas e as aplicações práticas dos SEV tiveram um desenvolvimento acelerado, graças à existência de uma teoria bem fundamentada, que podia ser aplicada no projeto de sistemas mais complexos, ao reconhecimento das importantes propriedades de robustez diante das perturbações e incertezas paramétricas dos SEV, assim como aos avanços nas áreas de circuitos eletrônicos microprocessados e técnicas de controle de *chattering*.

Atualmente, novos estudos e aplicações dos SEV às mais diversas áreas da engenharia vêm sendo realizados. Neste sentido a aplicação da teoria dos SEV em sistemas não-lineares, discretos, sistemas com atrasos e estocásticos são temas de intensas pesquisas. Além disso, são reportados estudos da utilização desta técnica no projeto de sistemas de controle ótimo e

adaptativos, assim como também a sua aplicação ao problema de observação de estado. Não menos importantes são as pesquisas direcionadas à eliminação ou redução de *chattering* nos SEV, onde novas abordagens vêm sendo estudadas.

5.3 Definições básicas dos SEV

A descrição matemática de um sistema de estrutura variável proposta por Utkin [36] considera um sistema dinâmico genérico expresso através da seguinte equação diferencial:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t, \mathbf{u}), \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{u} \in \mathbb{R}^m \quad (5.1)$$

A estrutura do controle é mudada utilizando um vetor de controle \mathbf{u} descontínuo sobre uma superfície de chaveamento $\mathbf{s}(\mathbf{x})$, definida sobre o espaço de estados do sistema, ou seja:

$$\mathbf{u}_i(\mathbf{x}, t) = \begin{cases} \mathbf{u}_i^+(\mathbf{x}, t), & \text{se } s_i(\mathbf{x}, t) > 0 \\ \mathbf{u}_i^-(\mathbf{x}, t), & \text{se } s_i(\mathbf{x}, t) < 0 \end{cases} \quad i = 1, \dots, m \quad (5.2)$$

$$\mathbf{s}(\mathbf{x})^T = (s_1(\mathbf{x}), \dots, s_m(\mathbf{x})) \quad (5.3)$$

onde \mathbf{u}_i^+ , \mathbf{u}_i^- e s_i são funções contínuas do estado \mathbf{x} .

O sistema de estrutura variável definido pelas equações (5.1)-(5.3) pode ser visto como uma composição de 2^m subsistemas contínuos e a sua estrutura muda em m superfícies do espaço de estado. O projeto do sistema consiste na seleção das componentes do vetor de controle \mathbf{u}_i^+ , \mathbf{u}_i^- e das superfícies $s_i = 0$.

O controle em modos deslizantes para o sistema de estrutura variável consiste na definição da superfície de chaveamento $\mathbf{s}(\mathbf{x})=0$ de acordo com a dinâmica desejada para o sistema e da ação de controle de estrutura variável $\mathbf{u}(\mathbf{x}, t)$, de maneira que qualquer estado \mathbf{x} fora da superfície de chaveamento seja direcionado para esta, garantindo a condição de

convergência. Uma vez que o sistema encontra-se na superfície $s(\mathbf{x})=0$, o modo deslizante será estabelecido e o sistema apresentará o comportamento dinâmico para o qual foi projetado.

5.4 Observadores de estado em modos deslizantes

O problema de controle consiste em levar o sistema a partir de um estado genérico \mathbf{x} a um estado desejado \mathbf{x}_d na presença de incertezas no modelo e perturbações externas. Uma solução natural deste problema, utilizando modos deslizantes, consiste em definir a superfície de chaveamento em função do erro atual do vetor de estado, ou seja:

$$\mathbf{s}(\mathbf{e}(\mathbf{x}), t) = \mathbf{0} \quad (5.4)$$

onde $\mathbf{e}(\mathbf{x}) = \mathbf{x} - \mathbf{x}_d = \mathbf{0}$.

A partir disso, a escolha da lei de controle deve ser tal que garanta as condições de convergência para os modos deslizantes. Ou seja, o controle deverá fazer com que qualquer trajetória do sistema em torno da superfície de chaveamento aponte na direção desta.

No projeto de observadores de estado em modos deslizantes, a escolha de uma superfície de chaveamento do tipo expresso pela equação 5.4 não é adequada, dado que não estão disponíveis todos os estados do sistema. Neste caso, a solução consiste em definir a superfície de chaveamento em função do erro de observação das variáveis de estado mensuráveis no sistema. A lei de controle descontínua está dada pelo termo de correção do observador, o qual direcionará a dinâmica do erro de observação destas variáveis para a superfície de chaveamento, conseqüentemente o erro do sistema observado tenderá a zero [46,47].

Para ilustrar a consideração anterior, consideremos o seguinte sistema linear invariante no tempo dado por [46]:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu}, \quad \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{u} \in \mathbb{R}^m \quad (5.5)$$

$$\mathbf{y} = \mathbf{Cx}, \quad \mathbf{y} \in \mathbb{R}^l, \quad 1 \leq l < n \quad (5.6)$$

Supondo que $\text{rank } \mathbf{C} = l$, o vetor de saída \mathbf{y} pode ser escrito como segue:

$$\mathbf{y} = \mathbf{C}_1 \mathbf{x}_1 + \mathbf{C}_2 \mathbf{x}_2, \quad \mathbf{x}^T = (\mathbf{x}_1^T, \mathbf{x}_2^T), \quad \mathbf{x}_2 \in \mathbb{R}^l, \quad \det \mathbf{C}_2 \neq 0 \quad (5.7)$$

A dinâmica do sistema pode ser expressa em função das variáveis de estado \mathbf{x}_1 e \mathbf{y} através das seguintes equações diferenciais:

$$\dot{\mathbf{x}}_1 = \bar{\mathbf{A}}_{11} \mathbf{x}_1 + \bar{\mathbf{A}}_{12} \mathbf{y} + \bar{\mathbf{B}}_1 \mathbf{u} \quad (5.8)$$

$$\dot{\mathbf{y}} = \bar{\mathbf{A}}_{21} \mathbf{x}_1 + \bar{\mathbf{A}}_{22} \mathbf{y} + \bar{\mathbf{B}}_2 \mathbf{u} \quad (5.9)$$

onde:

$$\mathbf{TAT}^{-1} = \begin{bmatrix} \bar{\mathbf{A}}_{11} & \bar{\mathbf{A}}_{12} \\ \bar{\mathbf{A}}_{21} & \bar{\mathbf{A}}_{22} \end{bmatrix}, \quad \mathbf{TB} = \begin{bmatrix} \bar{\mathbf{B}}_1 \\ \bar{\mathbf{B}}_2 \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{0} \\ \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \quad (5.10)$$

Para o projeto do observador de estado em modos deslizantes é selecionada como superfície de chaveamento o erro de observação das variáveis mensuráveis do sistema, ou seja,

$$\mathbf{y} - \hat{\mathbf{y}} = \mathbf{0} \quad (5.11)$$

A partir disto, as equações do observador de estado em modos deslizantes são escritas como segue:

$$\dot{\hat{\mathbf{x}}}_1 = \mathbf{A}_{11} \hat{\mathbf{x}}_1 + \mathbf{A}_{12} \hat{\mathbf{y}} + \mathbf{B}_1 \mathbf{u} - \mathbf{Lv} \quad (5.12)$$

$$\dot{\hat{\mathbf{y}}} = \mathbf{A}_{21} \hat{\mathbf{x}}_1 + \mathbf{A}_{22} \hat{\mathbf{y}} + \mathbf{B}_2 \mathbf{u} + \mathbf{v} \quad (5.13)$$

onde $\hat{\mathbf{x}}_1$ e $\hat{\mathbf{y}}$ são os valores estimados de \mathbf{x}_1 e \mathbf{y} a partir do observador.

O termo não-linear de correção \mathbf{v} tem como função forçar a trajetória do sistema na direção da superfície de chaveamento de maneira a estabelecer o regime deslizante ao longo desta:

$$\mathbf{v} = \mathbf{K}\text{sign}(\mathbf{e}_2) \quad (5.14)$$

onde:

$$\mathbf{e}_2 = \mathbf{y} - \hat{\mathbf{y}}, \text{sign}(\mathbf{e}_2) = (\text{sign}(y_1 - \hat{y}_1), \dots, \text{sign}(y_l - \hat{y}_l)) \quad (5.15)$$

A partir das equações (5.8), (5.9), (5.12) e (5.13) obtemos as equações dinâmicas do erro de observação:

$$\dot{\mathbf{e}}_1 = \dot{\mathbf{x}}_1 - \dot{\hat{\mathbf{x}}}_1 = \mathbf{A}_{11}\mathbf{e}_1 + \mathbf{A}_{12}\mathbf{e}_2 + \mathbf{L}\mathbf{v} \quad (5.16)$$

$$\dot{\mathbf{e}}_2 = \dot{\mathbf{y}} - \dot{\hat{\mathbf{y}}} = \mathbf{A}_{21}\mathbf{e}_1 + \mathbf{A}_{22}\mathbf{e}_2 - \mathbf{v} \quad (5.17)$$

A dinâmica do observador no regime de modos deslizantes pode ser descrita utilizando a lei de controle equivalente (equação B.27). Para isto consideramos válida a condição:

$$\dot{\mathbf{e}}_2 = \mathbf{e}_2 = \mathbf{0} \quad (5.18)$$

Substituindo (5.18) em (5.17) obtemos:

$$\mathbf{v} = \mathbf{v}_{\text{eq}} = \mathbf{A}_{21}\mathbf{e}_1 \quad (5.19)$$

Substituindo o resultado de (5.19) em (5.17) resulta:

$$\dot{\mathbf{e}}_1 = (\mathbf{A}_{11} + \mathbf{L}\mathbf{A}_{21})\mathbf{e}_1 \quad (5.20)$$

Se o par $\{\mathbf{A}_{11}, \mathbf{A}_{21}\}$ é observável, os autovalores de $(\mathbf{A}_{11} + \mathbf{L}\mathbf{A}_{21})$ podem ser arbitrariamente alocados e então o erro \mathbf{e}_1 pode ser controlado de acordo com a dinâmica

desejada na estimação de x_1 . O vetor x_2 , o qual forma parte do estado x , pode ser calculado a partir da equação (5.7) como segue:

$$x_2 = C_2^{-1}(y - C_1 x_1) \quad (5.21)$$

5.5 Observador de fluxo de rotor em modos deslizantes

Os fundamentos teóricos dos observadores de estado em modos deslizantes são utilizados nesta seção no projeto do observador de fluxo de rotor para o motor de indução. Inicialmente é apresentado o modelo matemático do observador e a seguir é obtido o mecanismo adaptativo de estimação de velocidade, assim como as equações de projeto para a matriz de ganho do observador.

5.5.1 Modelo matemático do observador

O observador de fluxo de rotor em modos deslizantes baseia-se no mesmo modelo do motor de indução utilizado para o observador de Luenberger (equações 4.1 e 4.2), repetido a seguir por comodidade:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \quad (5.22)$$

onde:

$$A = \begin{bmatrix} -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & 0 & \frac{L_m}{\sigma L_s L_r T_r} & \frac{L_m}{\sigma L_s L_r} \omega_r \\ 0 & -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right) & -\frac{L_m}{\sigma L_s L_r} \omega_r & \frac{L_m}{\sigma L_s L_r T_r} \\ \frac{L_m}{T_r} & 0 & -\frac{1}{T_r} & -\omega_r \\ 0 & \frac{L_m}{T_r} & \omega_r & -\frac{1}{T_r} \end{bmatrix} \quad B = \begin{bmatrix} \frac{1}{\sigma L_s} & 0 \\ 0 & \frac{1}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} B_1 \\ \mathbf{0} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (5.23)$$

sendo: $\mathbf{x} = [i_{s\alpha} \ i_{s\beta} \ \lambda_{r\alpha} \ \lambda_{r\beta}]^T$, $\mathbf{u} = [v_{s\alpha} \ v_{s\beta}]^T$ e $\mathbf{y} = [i_{s\alpha} \ i_{s\beta}]^T$ os vetores de estado, de entrada e de saída, respectivamente.

Com o objetivo de simplificar a notação matemática define-se:

$$\mathbf{A}_{11} = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)\mathbf{I}, \quad \mathbf{A}_{12} = \frac{L_m}{\sigma L_s L_r T_r}\mathbf{I} - \frac{L_m \omega_r}{\sigma L_s L_r}\mathbf{J}, \quad \mathbf{A}_{21} = \frac{L_m}{T_r}\mathbf{I}, \quad \mathbf{A}_{22} = -\frac{1}{T_r}\mathbf{I} + \omega_r \mathbf{J} \quad (5.24)$$

$$\text{sendo: } \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ e } \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

A partir do modelo matemático do motor, a equação do observador de fluxo de rotor com base em modos deslizantes pode ser expressa como [30,51]:

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K} \text{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) \quad (5.25)$$

onde $\hat{\mathbf{x}} = [\hat{i}_{s\alpha} \ \hat{i}_{s\beta} \ \hat{\lambda}_{r\alpha} \ \hat{\lambda}_{r\beta}]^T$ é o vetor de estado, composto pelos valores estimados pelo observador das componentes da corrente de estator e do vetor de fluxo de rotor. A superfície de chaveamento é o erro entre as componentes estimadas da corrente de estator e aquelas medidas.

A matriz \mathbf{K} pode ser escrita de maneira geral como segue:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ -\mathbf{L}\mathbf{K}_1 \end{bmatrix}, \quad \mathbf{K}_1 = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{bmatrix} \quad (5.26)$$

A matriz \mathbf{L} é selecionada de maneira a garantir a estabilidade do sistema, como também uma reposta dinâmica satisfatória em relação ao erro entre as componentes reais e estimadas do vetor de fluxo de rotor.

A partir do modelo do motor e do observador e considerando as variações dos parâmetros na matriz \mathbf{A} do sistema, a equação dinâmica do erro entre as variáveis de estado reais e as estimadas a partir do observador é escrita como:

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \Delta\mathbf{A}\hat{\mathbf{x}} + \mathbf{K} \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) \quad (5.27)$$

onde:

$$\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x} = [\mathbf{e}_i \quad \mathbf{e}_\lambda]^T, \quad \mathbf{e}_i = \hat{\mathbf{i}}_s - \mathbf{i}_s, \quad \mathbf{e}_\lambda = \hat{\lambda}_r - \lambda_r, \quad \Delta\mathbf{A} = \begin{bmatrix} \Delta\mathbf{A}_{11} & \Delta\mathbf{A}_{12} \\ \Delta\mathbf{A}_{21} & \Delta\mathbf{A}_{22} \end{bmatrix} \quad (5.28)$$

Desenvolvendo a equação (5.27), obtemos as equações dinâmicas do erro para os vetores corrente de estator e fluxo de rotor separadamente:

$$\dot{\mathbf{e}}_i = \mathbf{A}_{11}\mathbf{e}_i + \mathbf{A}_{12}\mathbf{e}_\lambda + \Delta\mathbf{A}_{11}\hat{\mathbf{i}}_s + \Delta\mathbf{A}_{12}\hat{\lambda}_r + \mathbf{K}_1 \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) \quad (5.29)$$

$$\dot{\mathbf{e}}_\lambda = \mathbf{A}_{21}\mathbf{e}_i + \mathbf{A}_{22}\mathbf{e}_\lambda + \Delta\mathbf{A}_{21}\hat{\mathbf{i}}_s + \Delta\mathbf{A}_{22}\hat{\lambda}_r - \mathbf{L}\mathbf{K}_1 \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) \quad (5.30)$$

A dinâmica do sistema no regime de modos deslizantes pode ser descrita a partir da aplicação da lei de controle equivalente. Para isto consideramos válida a seguinte condição:

$$\mathbf{e}_i = \dot{\mathbf{e}}_i = \mathbf{0} \quad (5.31)$$

Substituindo (5.31) em (5.29) e (5.30) resulta:

$$\mathbf{0} = \mathbf{A}_{12}\mathbf{e}_\lambda + \Delta\mathbf{A}_{11}\hat{\mathbf{i}}_s + \Delta\mathbf{A}_{12}\hat{\lambda}_r + \mathbf{K}_1 \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) \quad (5.32)$$

$$\dot{\mathbf{e}}_\lambda = \mathbf{A}_{22}\mathbf{e}_\lambda + \Delta\mathbf{A}_{21}\hat{\mathbf{i}}_s + \Delta\mathbf{A}_{22}\hat{\lambda}_r - \mathbf{L}\mathbf{K}_1 \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) \quad (5.33)$$

Fazendo $\mathbf{v} = -\mathbf{K}_1 \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s)$, a partir da equação (5.32) obtemos:

$$\mathbf{v} = \mathbf{v}_{\text{eq}} = \mathbf{A}_{12}\mathbf{e}_\lambda + \Delta\mathbf{A}_{11}\hat{\mathbf{i}}_s + \Delta\mathbf{A}_{12}\hat{\lambda}_r \quad (5.34)$$

Substituindo o resultado de (5.34) em (5.33) obtemos a equação diferencial do modo deslizante:

$$\dot{\mathbf{e}}_\lambda = (\mathbf{A}_{22} + \mathbf{L}\mathbf{A}_{12})\mathbf{e}_\lambda + (\Delta\mathbf{A}_{21} + \mathbf{L}\Delta\mathbf{A}_{11})\hat{\mathbf{i}}_s + (\Delta\mathbf{A}_{22} + \mathbf{L}\Delta\mathbf{A}_{12})\hat{\lambda}_r \quad (5.35)$$

Se a velocidade do motor é uma variável medida no sistema e não são consideradas variações de outros parâmetros, a equação (5.35) resulta em:

$$\dot{\mathbf{e}}_\lambda = (\mathbf{A}_{22} + \mathbf{L}\mathbf{A}_{12})\mathbf{e}_\lambda \quad (5.36)$$

A equação anterior representa a dinâmica do erro entre as componentes estimadas e reais do vetor fluxo de rotor em modos deslizantes.

5.5.2 Esquema adaptativo para estimação de velocidade

Se a velocidade do motor é considerada como parâmetro variável, os elementos da matriz $\Delta\mathbf{A}$ podem ser escritos como:

$$\Delta\mathbf{A}_{11} = \mathbf{0}, \quad \Delta\mathbf{A}_{12} = -\frac{\Delta\omega_r}{\varepsilon}\mathbf{J}, \quad \Delta\mathbf{A}_{21} = \mathbf{0}, \quad \Delta\mathbf{A}_{22} = \Delta\omega_r\mathbf{J} \quad (5.37)$$

$$\text{onde: } \varepsilon = \frac{\sigma L_s L_r}{L_m} \quad \text{e} \quad \Delta\omega_r = \hat{\omega}_r - \omega_r$$

Para obter o esquema adaptativo de estimação de velocidade, considera-se a seguinte função de Lyapunov [51]:

$$\mathbf{V} = \mathbf{e}_\lambda^T \mathbf{e}_\lambda + W \quad (5.38)$$

na qual a função W deve ser determinada com o objetivo de garantir a convergência do algoritmo de estimação.

A derivada da função de Lyapunov em relação ao tempo pode ser expressa como:

$$\dot{\mathbf{V}} = \mathbf{v}^T \mathbf{\Lambda}^T \mathbf{A}_{12}^{-1} \mathbf{v} + \mathbf{v}^T \mathbf{\Lambda}^T \mathbf{A}_{12}^{-1} \frac{\Delta\omega_r}{\varepsilon} \mathbf{J} \hat{\boldsymbol{\lambda}}_r + \dot{W} \quad (5.39)$$

$$\text{onde: } \mathbf{\Lambda} = \mathbf{L} - \varepsilon \mathbf{I}$$

Para que a derivada da função de Lyapunov seja negativa definida, impõem-se as seguintes condições:

$$\mathbf{v}^T \mathbf{\Lambda}^T \mathbf{A}_{12}^{-1} \mathbf{v} < 0 \quad (5.40)$$

$$\mathbf{v}^T \mathbf{\Lambda}^T \mathbf{A}_{12}^{-1} \frac{\Delta\omega_r}{\varepsilon} \mathbf{J} \hat{\boldsymbol{\lambda}}_r + \dot{W} = 0 \quad (5.41)$$

Para satisfazer a condição da equação (5.40) fazemos:

$$\mathbf{\Lambda}^T = -\gamma \mathbf{A}_{12}, \quad \gamma > 0 \quad (5.42)$$

Substituindo este resultado na equação (5.41), obtêm-se:

$$\dot{W} = \gamma \mathbf{v}^T \frac{\Delta\omega_r}{\varepsilon} \mathbf{J} \hat{\boldsymbol{\lambda}}_r \quad (5.43)$$

A partir da equação anterior selecionamos a função W como sendo:

$$W = \frac{\Delta\omega_r^2}{2\mu\varepsilon}, \quad \mu > 0 \quad (5.44)$$

Derivando a equação (5.44) em relação ao tempo e igualando o resultado com (5.43), obtemos a equação que descreve o esquema adaptativo para a estimação da velocidade:

$$\dot{\hat{\omega}} = \mu\gamma [k_1 \operatorname{sgn}(\hat{i}_{s\alpha} - i_{s\alpha}) \hat{\lambda}_{r\beta} - k_2 \operatorname{sgn}(\hat{i}_{s\beta} - i_{s\beta}) \hat{\lambda}_{r\alpha}] \quad (5.45)$$

5.5.3 Projeto do observador

Para uma maior liberdade no projeto do observador, a condição expressa pela equação (5.42) pode ser formulada como segue [51]:

$$\mathbf{\Lambda} \leq -\gamma \mathbf{A}_{12}, \quad \gamma > 0 \quad (5.46)$$

Consideremos a matriz \mathbf{L} definida como [51]:

$$\mathbf{L} = -\alpha\mathbf{I} - \gamma\mathbf{J} = \begin{bmatrix} -\alpha & \gamma \\ -\gamma & -\alpha \end{bmatrix} \quad (5.47)$$

Substituindo as definições de Λ e \mathbf{A}_{12} em (5.46) e resolvendo a desigualdade em relação aos elementos x e y da matriz \mathbf{L} obtêm-se:

$$x = (q-1)\varepsilon + \frac{\gamma}{T_r\varepsilon}, \quad q \geq 0 \quad \text{e} \quad y = \frac{\gamma\omega_r}{\varepsilon} \quad (5.48)$$

onde q é um parâmetro de projeto.

Substituindo as definições de \mathbf{A}_{22} , \mathbf{L} e \mathbf{A}_{12} em (5.36) a equação dinâmica do erro de estimação no regime deslizante pode ser escrita como:

$$\dot{\mathbf{e}}_\lambda = (\mathbf{A}_{22} + \mathbf{L}\mathbf{A}_{12})\mathbf{e}_\lambda = (-\alpha\mathbf{I} + \beta\mathbf{J})\mathbf{e}_\lambda \quad (5.49)$$

na qual:

$$-\alpha = -\frac{1}{T_r\varepsilon}x - \frac{\omega_r}{\varepsilon}y - \frac{1}{T_r}, \quad \beta = \frac{\omega_r}{\varepsilon}x - \frac{1}{T_r\varepsilon}y + \omega_r \quad (5.50)$$

A dinâmica do erro de estimação, dada pela equação (5.49), é determinada pelos autovalores da matriz $\mathbf{A}_{22} + \mathbf{L}\mathbf{A}_{12}$:

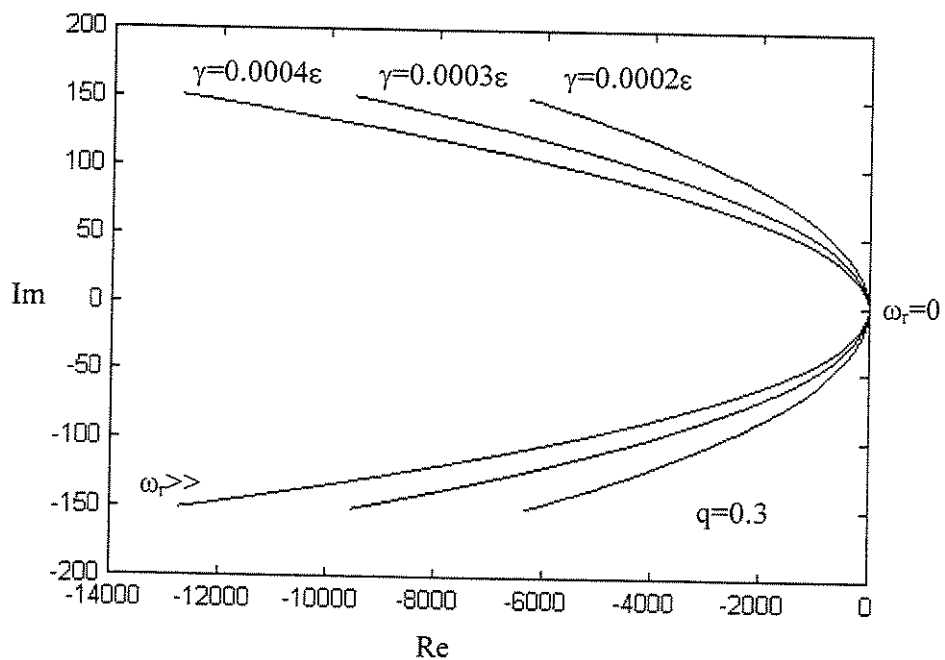
$$\lambda_{1,2} = -\alpha \pm j\beta \quad (5.51)$$

Substituindo (5.48) em (5.50) e o resultado desta substituição em (5.51) obtêm-se:

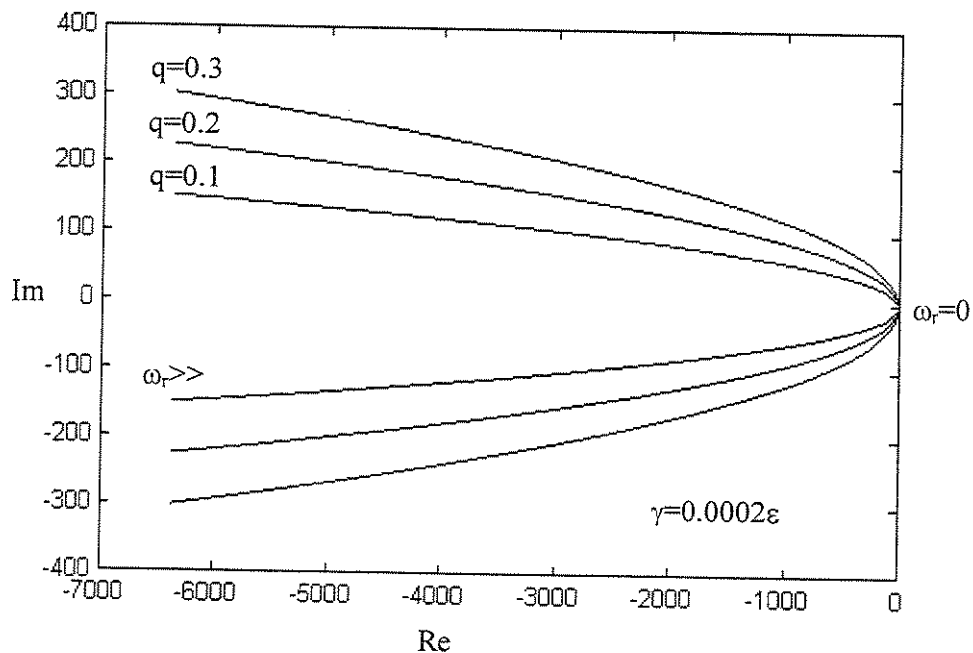
$$\lambda_{1,2} = \left(-\frac{q}{T_r} - \gamma \frac{1}{T_r^2\varepsilon^2} - \gamma \frac{\omega_r^2}{\varepsilon^2} \right) \pm jq\omega_r \quad (5.52)$$

Visto que $q > 0$ e $\gamma > 0$, os autovalores $\lambda_{1,2}$ têm parte real negativa, garantindo a estabilidade do observador. Os parâmetros de projeto q e γ devem ser selecionados adequadamente com o objetivo de obter uma resposta dinâmica satisfatória na estimação do

fluxo. Por sua vez, o parâmetro μ influencia na dinâmica do algoritmo adaptativo de estimação da velocidade. As trajetórias dos pólos do observador em função da velocidade do motor para diferentes valores dos parâmetros de projeto q e γ são mostradas na Figura 5.1.



(a)



(b)

Figura 5.1 – Trajetória dos pólos do motor para diferentes valores dos parâmetros q e γ .

5.6 Redução de *chattering* utilizando Lógica Fuzzy

No projeto e análise dos sistemas de controle com base em modos deslizantes considera-se que o controle pode mudar instantaneamente de um valor a outro. Porém, nos sistemas reais isto não é possível, devido principalmente aos atrasos de tempo necessários para o cálculo da lei de controle, o qual é característico da implementação digital, além das limitações dos dispositivos atuadores. Por causa disso, o sistema nem sempre conseguirá manter a trajetória na superfície de chaveamento, dando origem a um chaveamento de alta frequência em torno desta superfície, sendo este fenômeno conhecido como *chattering*. Dado que é inevitável a presença de *chattering* em sistemas reais, muitos esforços têm sido direcionados com o objetivo de eliminar ou diminuir os efeitos negativos deste.

A estratégia comumente utilizada para a redução de *chattering* introduz uma camada limite (Φ) em torno das superfícies de chaveamento. Para isto a função $sgn(\cdot)$ é substituída pela função $sat(\cdot)$, definida como segue [50]:

$$sat(\mathbf{s}(t), \Phi) = \begin{cases} \text{sign}(\mathbf{s}(t)) & \text{se } |\mathbf{s}(t)| > \Phi \\ \mathbf{s}(t) / \Phi & \text{se } |\mathbf{s}(t)| \leq \Phi \end{cases} \quad (5.53)$$

Com esta estratégia, o controle deixa de ser descontínuo dentro da camada limite e passa a ser um controle contínuo de ganho elevado. Para preservar as características de robustez do controle com base em modos deslizantes, esta camada deve ser bem estreita, de maneira que a trajetória do sistema permaneça o mais próxima da superfície deslizante. Porém, atrasos de tempo inerentes à execução da lei de controle podem exigir camadas de largura maior. Alguns autores propõem a utilização de algoritmos adaptativos para ajustar a largura da camada limite, considerando as incertezas na dinâmica do sistema [50].

As características adaptativas dos algoritmos com base em Lógica *Fuzzy* têm sido aproveitadas convenientemente pelos pesquisadores da área, com o objetivo de aumentar a robustez do controle em modos deslizantes. A utilização conjunta destas técnicas no projeto de sistemas de controle têm sido objeto de muitas pesquisas nos últimos anos [52,53]. Com

base no trabalho de De Neyer e Gorez [53], um algoritmo adaptativo utilizando Lógica Fuzzy para a redução de *chattering* no observador de fluxo de rotor em modos deslizantes é elaborado e apresentado nesta seção.

O algoritmo substitui o termo não-linear de controle presente no modelo do observador por um mapeamento *fuzzy* em função da dinâmica do erro entre os valores reais e estimados das componentes da corrente de estator. Para a elaboração deste algoritmo, define-se a função não linear de chaveamento como sendo:

$$\mathbf{w} = -\mathbf{v} = \mathbf{K}_1 \operatorname{sgn}(\hat{\mathbf{i}}_s - \mathbf{i}_s) = \begin{bmatrix} k_1 \operatorname{sgn}(\xi_1) \\ k_2 \operatorname{sgn}(\xi_2) \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (5.54)$$

onde $\xi_1 = \hat{i}_{s\alpha} - i_{s\alpha}$ e $\xi_2 = \hat{i}_{s\beta} - i_{s\beta}$.

Como variáveis de entrada, o algoritmo recebe a diferença ou erro entre o valor estimado e medido do componente da corrente de estator em questão, assim como também a derivada desta variação. O algoritmo fornece na saída a componente do vetor \mathbf{w} (w_1 ou w_2), em função do conjunto de variáveis de entrada utilizado ($[\xi_1, \dot{\xi}_1]$ ou $[\xi_2, \dot{\xi}_2]$). Para cada uma das variáveis de entrada foram empregados 5 conjuntos *fuzzy*, resultando no total de 25 regras de inferência. A Figura 5.2 mostra os gráficos das funções de pertinência utilizadas para as variáveis de entrada do sistema. Para a variável de saída do algoritmo foram empregados *fuzzy singletons* nas posições -1 , 0 e 1 respectivamente. A Figura 5.3 mostra a superfície do regulador.

Os valores do universo de discurso das variáveis de entrada e de saída estão normalizados no intervalo $[-1, +1]$. Os termos usados para os conjuntos *fuzzy* das variáveis de entrada e de saída tem os seguintes significados lingüísticos:

- Variáveis de Entrada

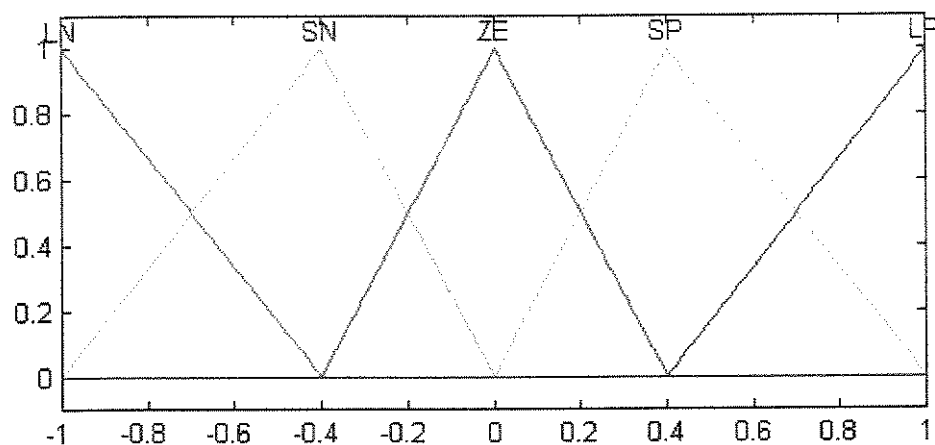
LN: Grande Negativo, SN: Pequeno Negativo, ZE: Zero, SP: Pequeno Positivo,

LP: Grande Positivo

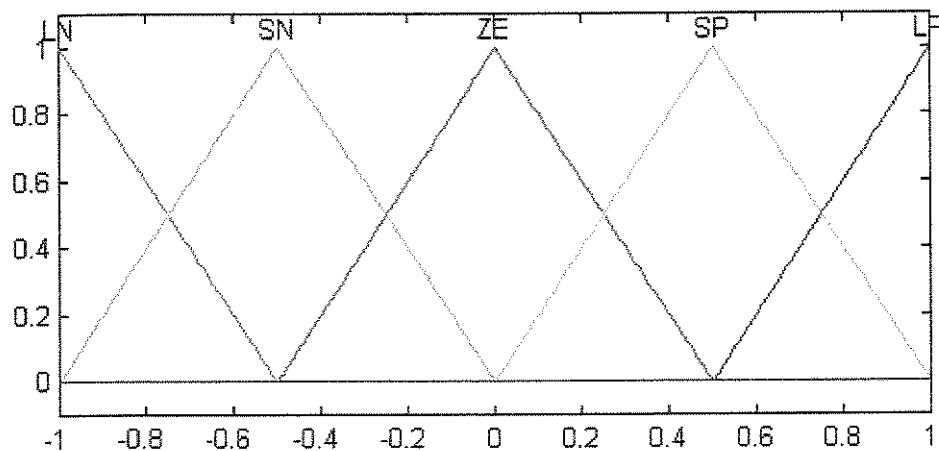
- Variável de Saída

N: Negativo, Z: Zero, P: Positivo

As regras de inferência são mostradas na tabela 5.1. O método “MAX-MIN” foi utilizado no processo de inferência e a “defuzzificação” foi feita com base no método de Centro do Máximo (CoM). O desempenho do algoritmo *fuzzy* para controle de *chattering* foi simulado antes de ser implementado no DSP utilizando o software *fuzzyTech* da *Inform Software*, o qual permite gerar um código assembler compatível com a família de DSPs TMS320C2x da Texas Instruments.



(a) Variável: erro



(b) Variável: derivada do erro (derro)

Figura 5.2 - Funções de pertinência das variáveis de entrada. (a) erro (b) derivada do erro.

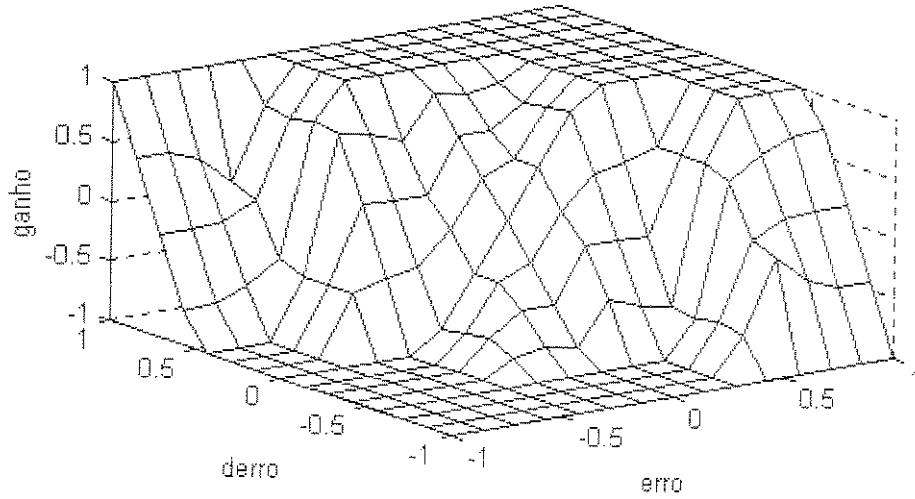


Figura 5.3- Superfície do regulador.

Tabela 5.1 – Regras de Inferência do regulador

| derro \ erro | LN | SN | ZE | SP | LP |
|--------------|----|----|----|----|----|
| LP | P | P | P | P | P |
| SP | N | P | P | P | P |
| ZE | N | N | Z | P | P |
| SN | N | N | N | N | P |
| LN | N | N | N | N | N |

5.7 Resultados Experimentais

A Figura 5.4 mostra um diagrama de blocos da implementação do sistema de controle por orientação direta do fluxo de rotor utilizando o observador em modos deslizantes com base no processador digital de sinais DSP TMS320F240 da Texas Instruments.

O tempo de execução do algoritmo de controle, da aquisição de sinais e do observador de fluxo sem o algoritmo de redução de chattering foi de aproximadamente 55 μs , sendo deste total, aproximadamente 15 μs para o cálculo das equações do observador. A incorporação do algoritmo *fuzzy* para o ajuste dos ganhos do observador resultou no aumento do tempo total de execução para 180 μs . As funções de aquisição, estimação e controle são sincronizadas com o período de 300 μs do sinal PWM. O valor da velocidade estimada, utilizado na malha de velocidade do sistema de controle, é atualizado a cada 3 ms.

A apresentação dos resultados experimentais é feita de forma similar ao que foi feito para os observadores de Kalman e o de Luenberger, ou seja, em testes de regime permanente e de transitório. Na condição de regime permanente são apresentados resultados referentes à estimação a partir do observador das componentes no sistema estacionário da corrente de estator e do vetor fluxo de rotor. Na seqüência, são mostrados os resultados do funcionamento do sistema durante o transitório no qual foram aplicadas ao motor referências variáveis na velocidade mecânica.

5.7.1 Testes em regime permanente

Os resultados experimentais das componentes reais e estimadas da corrente de estator no sistema estacionário de referência para uma velocidade de operação constante de 100 rpm, são apresentados na Figura 5.5, sendo os resultados de estimação das componentes do vetor de fluxo para esta mesma velocidade apresentados na Figura 5.6. Resultados similares foram obtidos para as velocidades de 200 rpm (Figuras 5.7 e 5.8) e 900 rpm (Figuras 5.9 e 5.10). Os resultados de regime apresentados mostraram o bom desempenho do observador para altas e baixas velocidades de operação, sendo verificada a correta estimação das componentes da corrente de estator e do vetor fluxo de rotor.

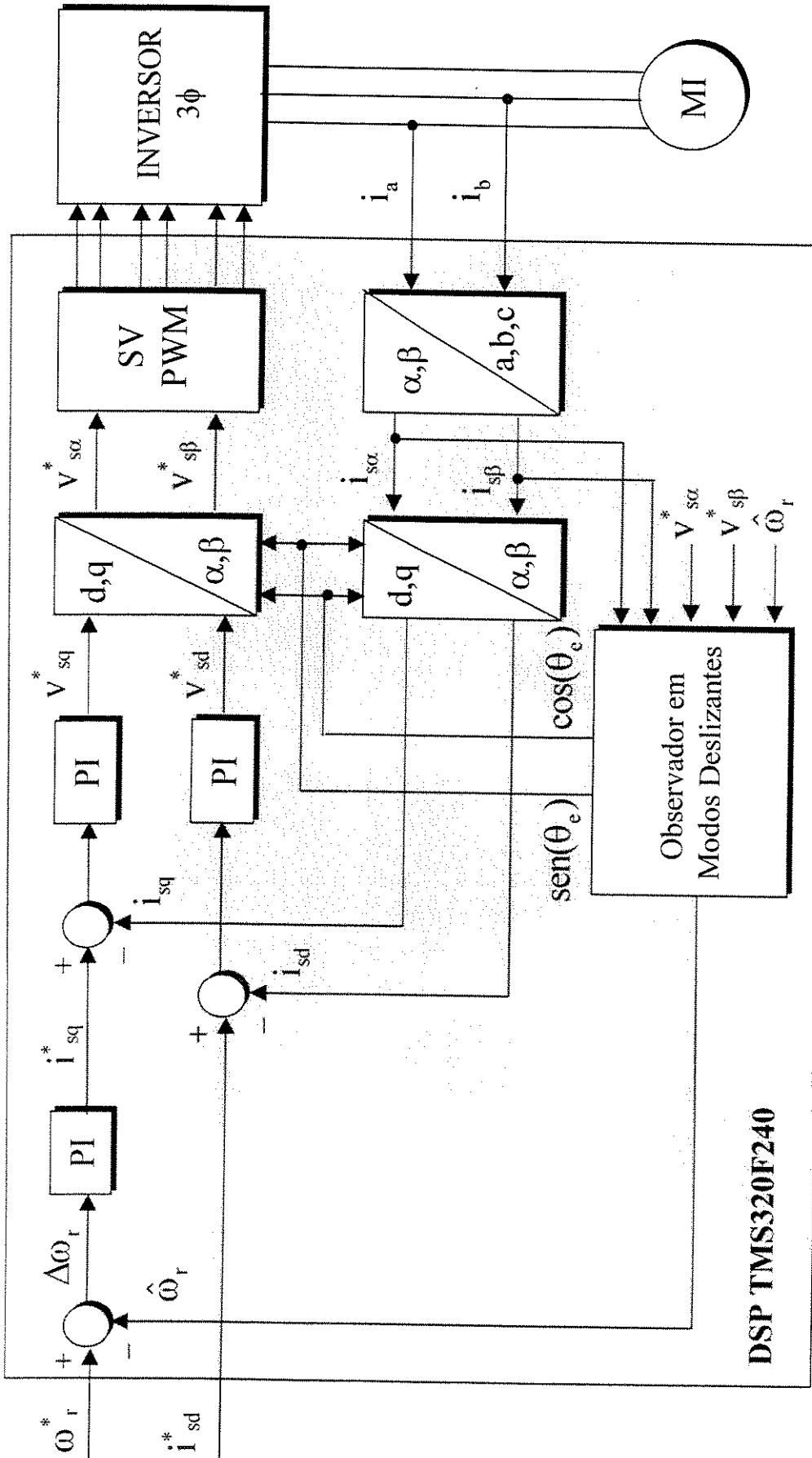
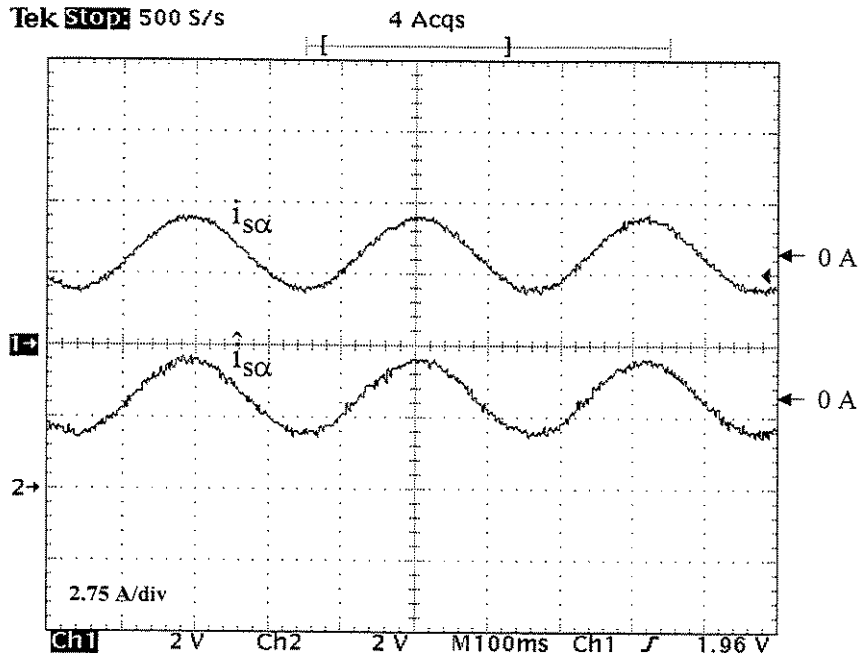
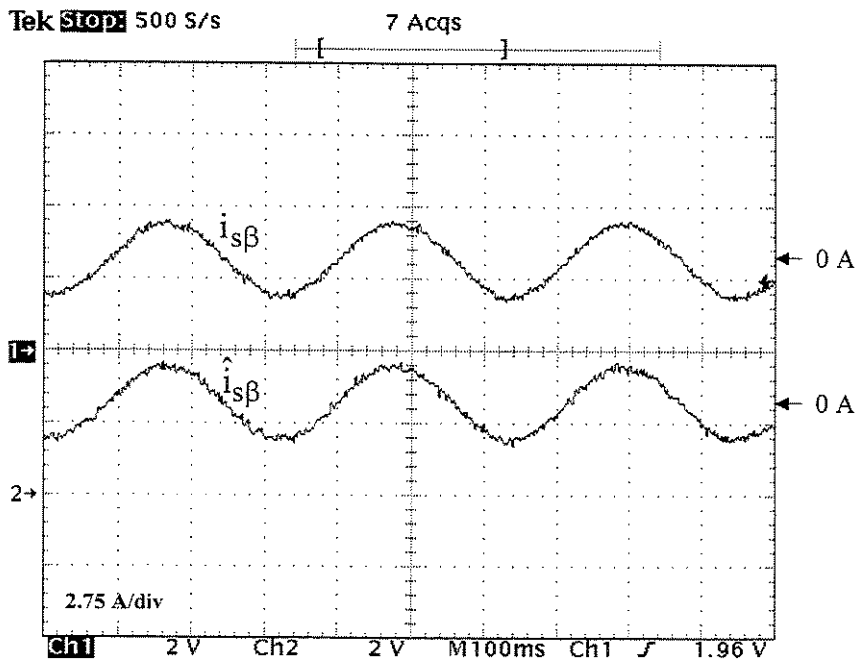


Figura 5.4 - Diagrama de blocos do sistema de controle utilizando o observador em modos deslizantes

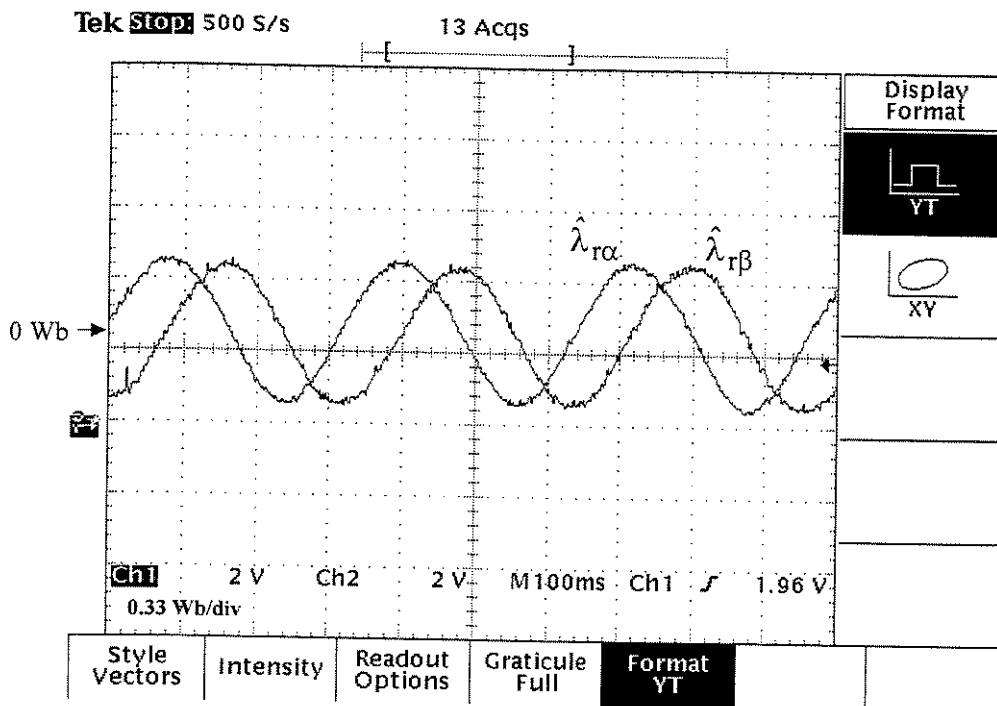


(a)

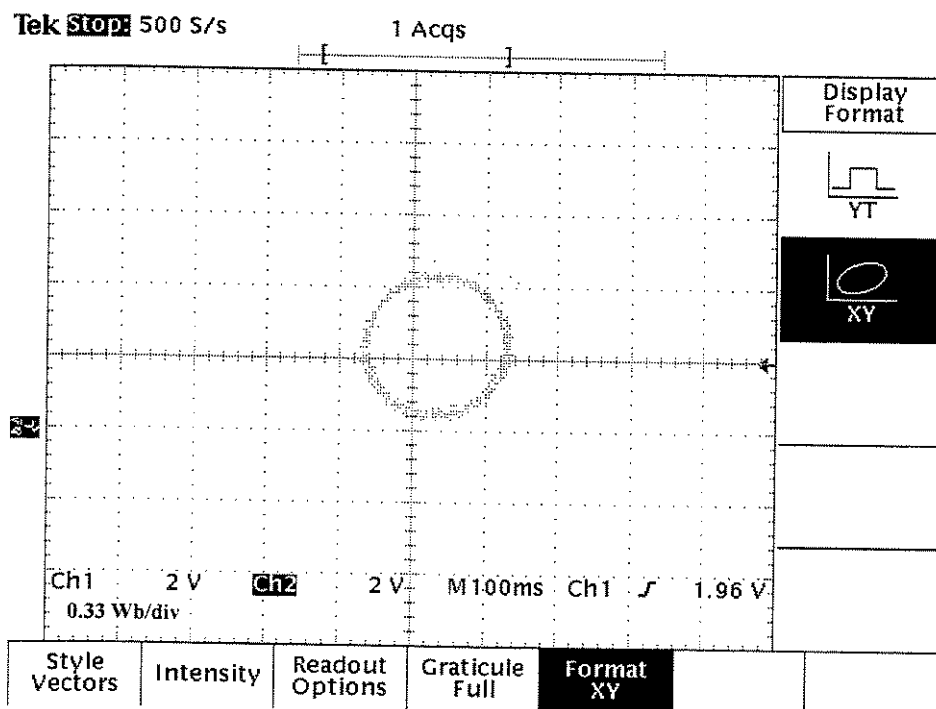


(b)

Figura 5.5- Correntes reais e estimadas para $\omega_r^* = 100$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$



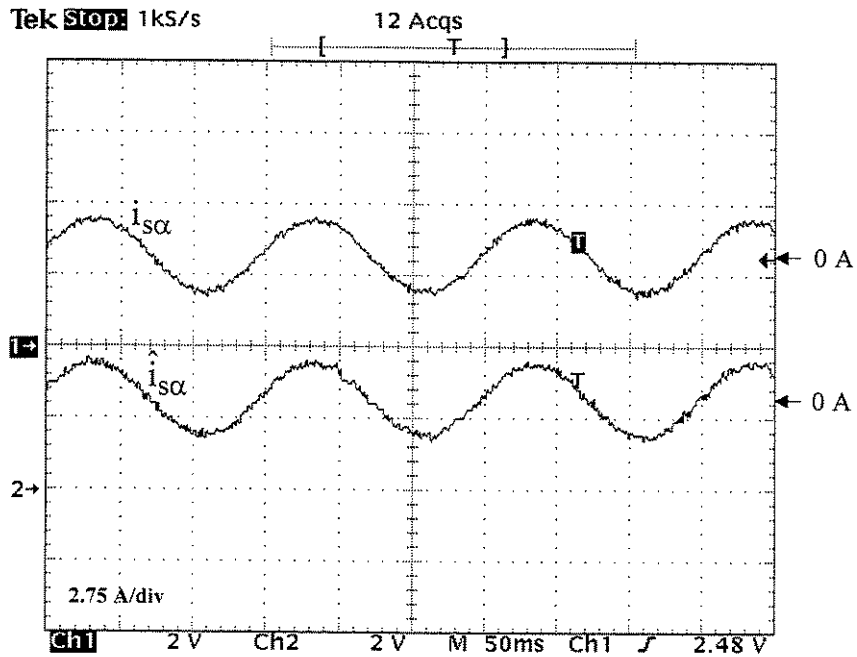
(a)



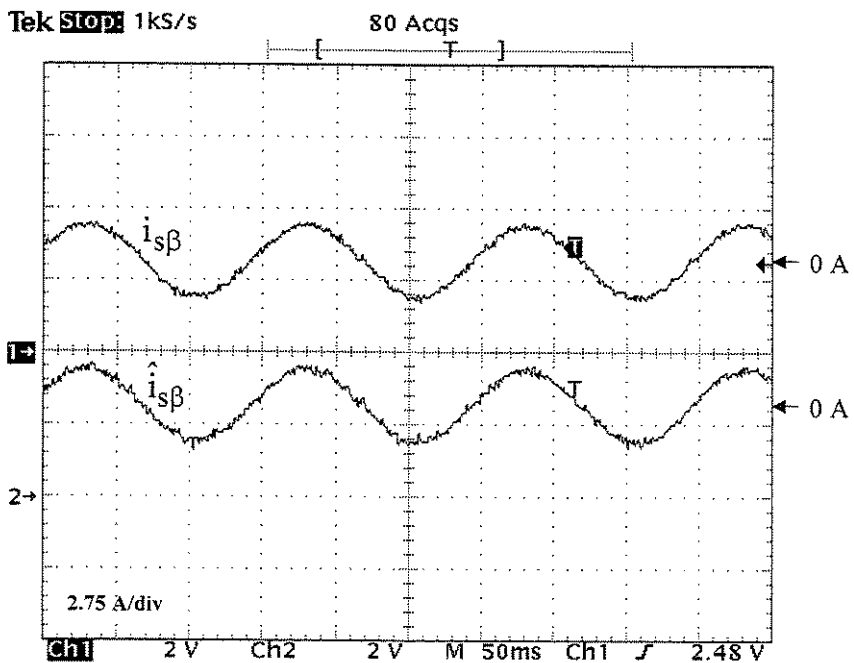
(b)

Figura 5.6- Componentes estimadas do fluxo de rotor para $\omega_r^* = 100$ rpm (a) $\hat{\lambda}_{r\alpha}$, $\hat{\lambda}_{r\beta}$

(b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

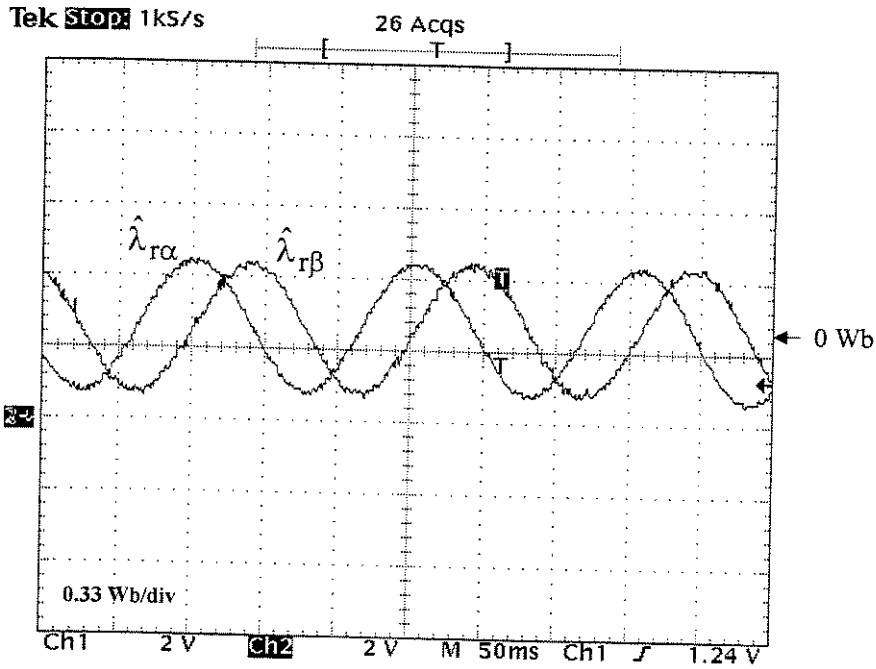


(a)

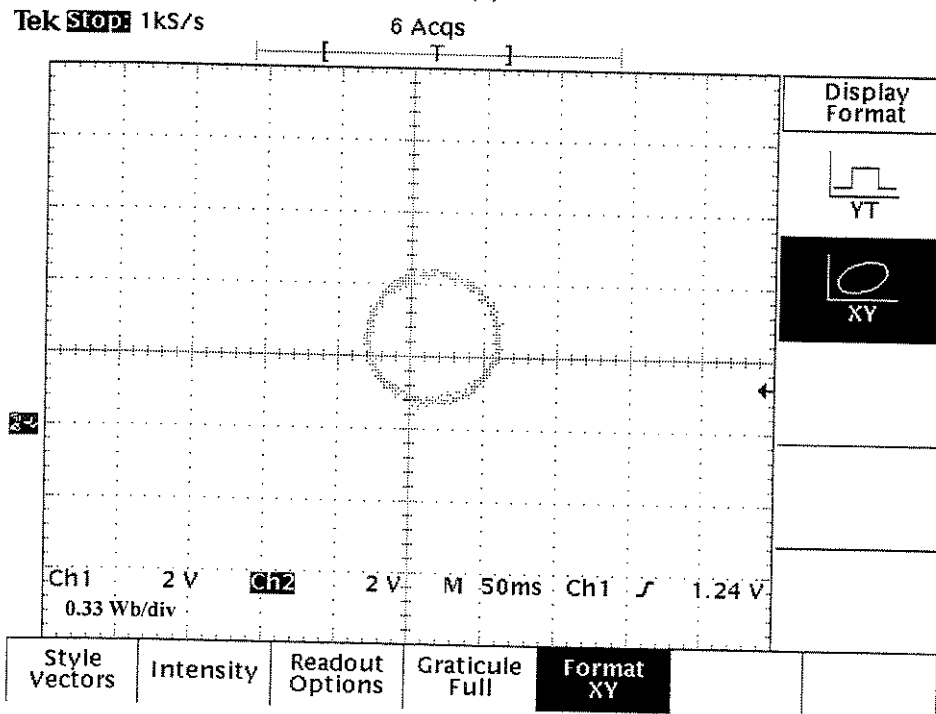


(b)

Figura 5.7- Correntes reais e estimadas para $\omega_r^* = 200$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$



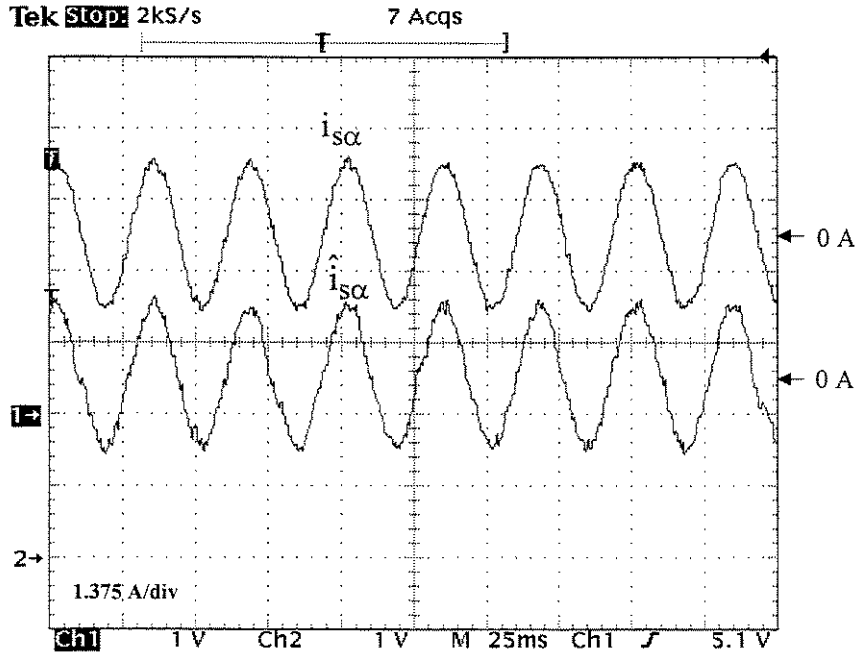
(a)



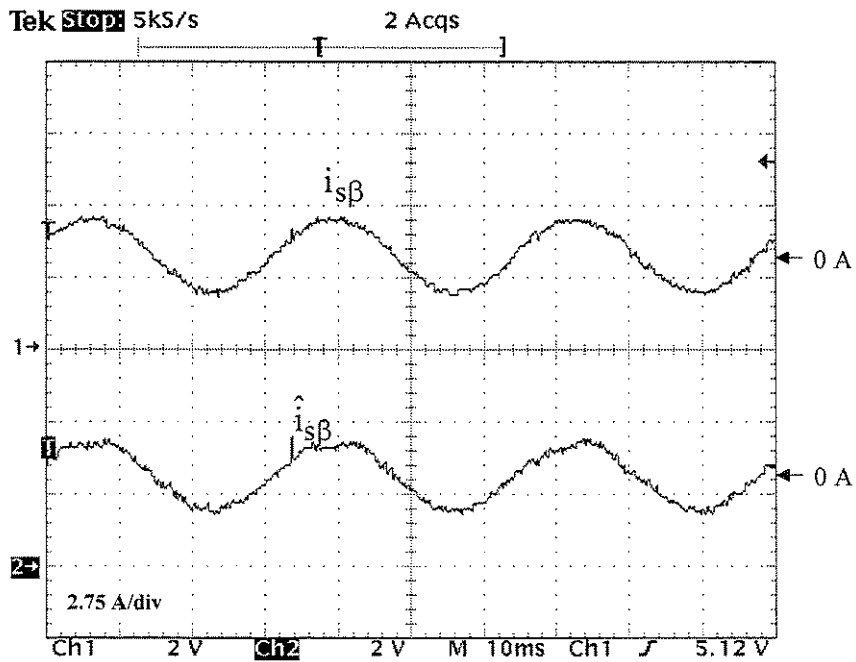
(b)

Figura 5.8- Componentes estimadas do fluxo de rotor para $\omega_r^* = 200$ rpm (a) $\hat{\lambda}_{r\alpha}$, $\hat{\lambda}_{r\beta}$

(b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

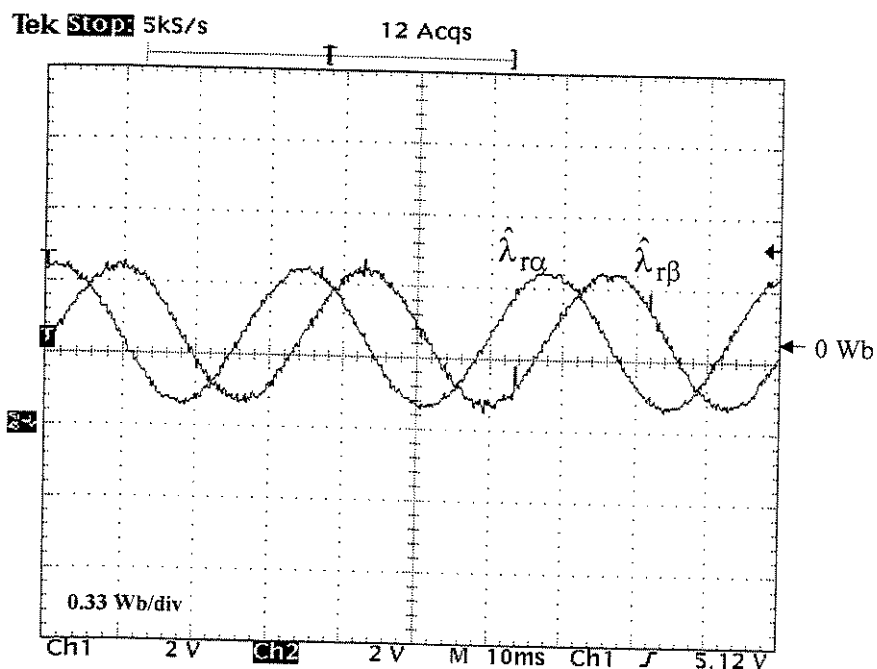


(a)

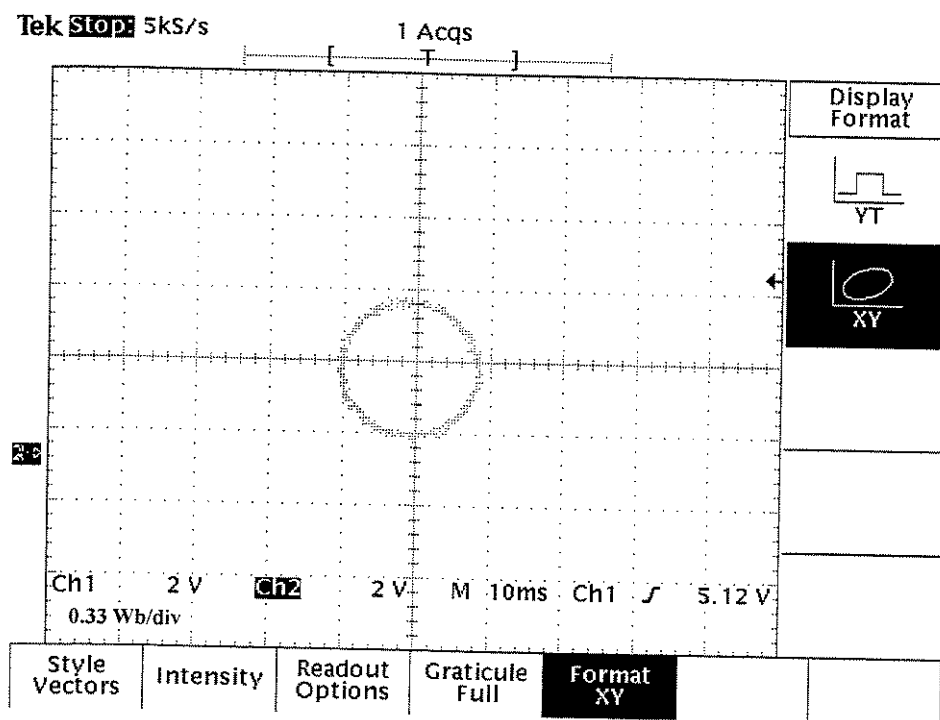


(b)

Figura 5.9- Correntes reais e estimadas para $\omega_r^* = 900$ rpm (a) $i_{s\alpha}, \hat{i}_{s\alpha}$ (b) $i_{s\beta}, \hat{i}_{s\beta}$



(a)



(b)

Figura 5.10- Componentes estimadas do fluxo de rotor para $\omega_r^* = 900$ rpm (a) $\hat{\lambda}_{r\alpha}$, $\hat{\lambda}_{r\beta}$
(b) $\hat{\lambda}_{r\alpha}$ vs $\hat{\lambda}_{r\beta}$

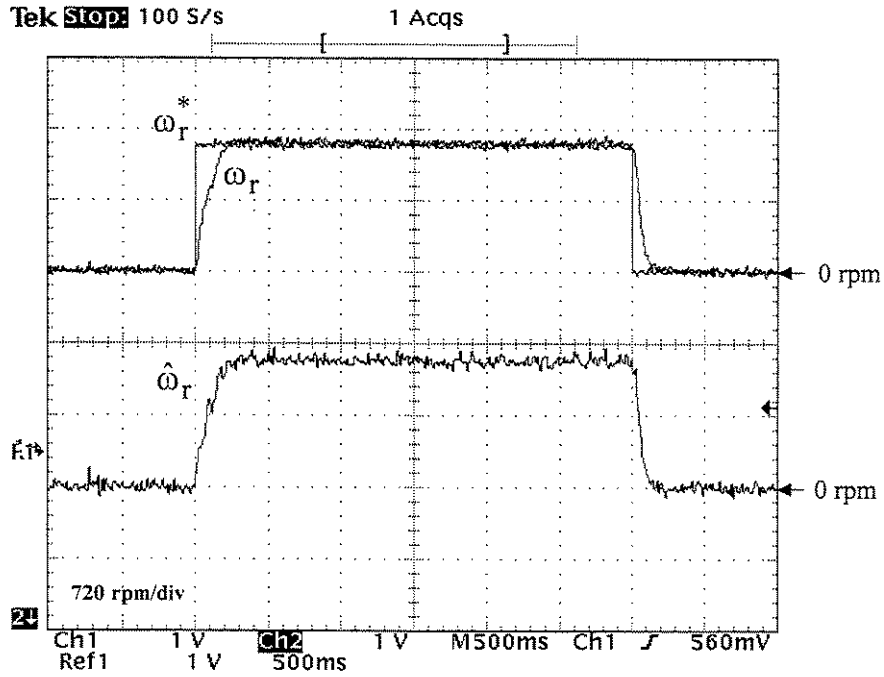
5.7.2 Testes em regime transitório

Para avaliar o desempenho do sistema de controle durante o regime transitório foram aplicadas referências variáveis da velocidade do motor. Na Figura 5.11(a) são apresentadas as curvas experimentais da velocidade mecânica medida e a estimada quando da aplicação de velocidade de referência variável do tipo onda quadrada de 0 a 1260 rpm. Neste caso verifica-se que a velocidade estimada segue com uma boa aproximação o valor medido. O teste anterior foi repetido para uma variação de velocidade de -100 a 100 rpm e as curvas da velocidade real e de referência são mostradas na Figura 5.11(b). O erro de regime verificado neste caso foi de aproximadamente 8 rpm, o qual pode ser considerado aceitável em relação à velocidade medida. Nas Figuras 5.12(a) e 5.12(b) são apresentados os resultados dos testes de reversão de velocidade de -50 a 50 rpm e de -20 a 20 rpm, respectivamente. Nesta figura, pode ser observado o aumento do erro de regulação a medida que ocorre a diminuição da velocidade de referência, chegando a ser maior que 40% do valor de regime no caso da reversão de -20 a 20 rpm.

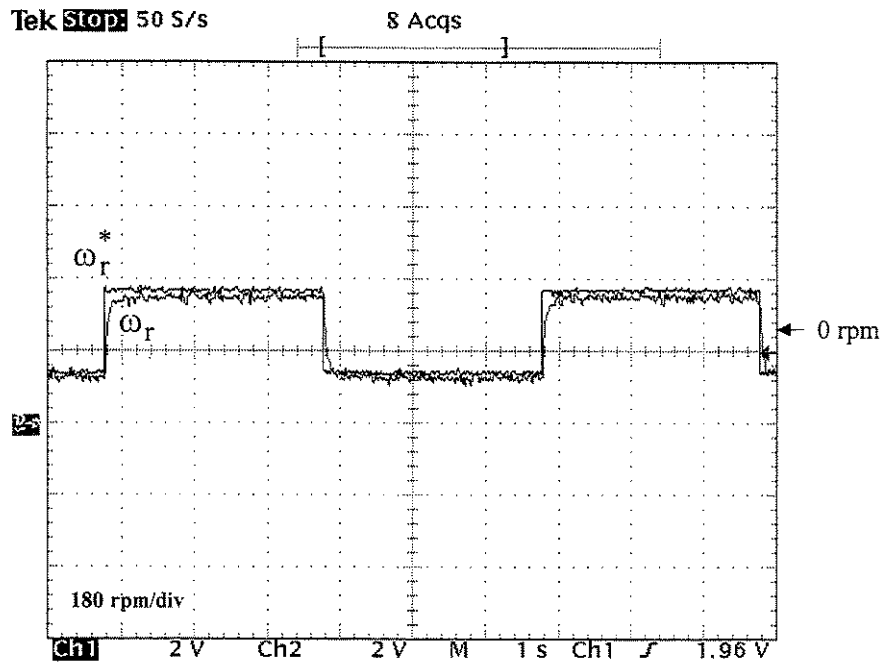
O efeito de *chattering* na dinâmica do sistema de controle é mostrado na Figura 5.13 para uma reversão de velocidade de -200 a 200 rpm. Na Figura 5.13(a), na qual o algoritmo de redução de *chattering* é desativado, a resposta do sistema apresenta oscilações notáveis em torno do valor de regime da velocidade. As curvas da Figura 5.13(b), onde o algoritmo de redução de *chattering* está em operação, mostram uma diminuição considerável nas oscilações no valor de regime da velocidade do motor obtendo-se uma resposta dinâmica satisfatória.

A seguir foi realizado o teste de aceleração – desaceleração do motor com inversão de velocidade. Neste teste, o motor é acelerado de -900 rpm até 900 rpm e após 3 segundos na condição de regime é feita a desaceleração até -900 rpm, repetindo-se o ciclo de operação. As respostas obtidas para a velocidade real e a estimada do motor são mostradas na Figura 5.14. Como pode ser observado nesta figura, o valor da velocidade estimada segue com boa aproximação, o valor medido.

Na Figura 5.15 são mostradas as curvas experimentais da velocidade e das componentes da corrente de estator no sistema síncrono com aplicação de carga no eixo do motor para a velocidade de 1.100 rpm. A partir da condição de regime é aplicado um conjugado de carga de aproximadamente 0.45 pu durante 3 segundos e a partir deste instante é retirada a carga. Como pode ser observado na figura, a componente i_{sq} da corrente de estator aumenta, devido à nova demanda de carga e a componente i_{sd} permanece em valor aproximadamente igual ao da referência imposto pelo controle (0.4 pu). Assim, é obtido um bom desacoplamento entre o controle de fluxo e do conjugado eletromagnético para o sistema de controle implementado.



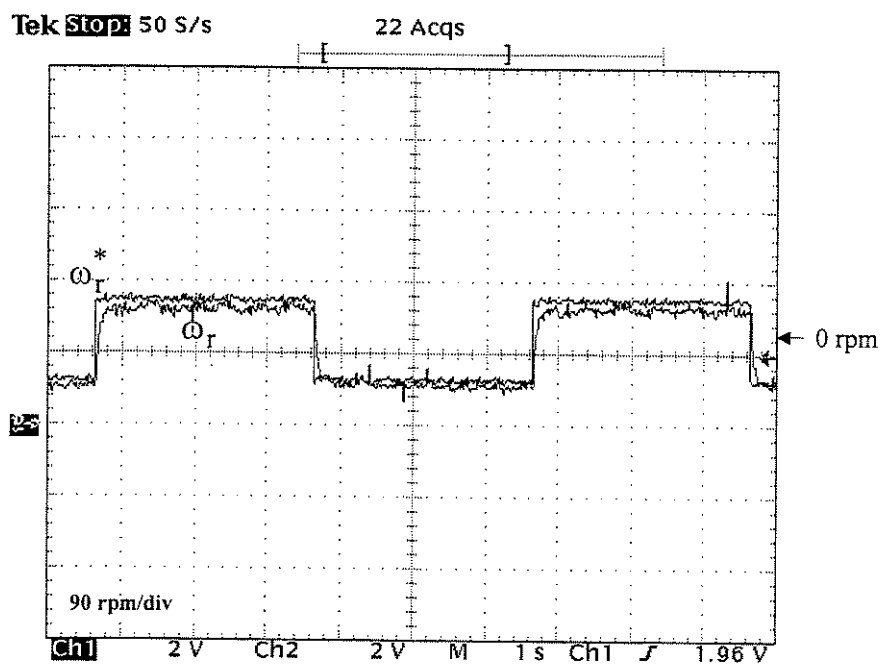
(a)



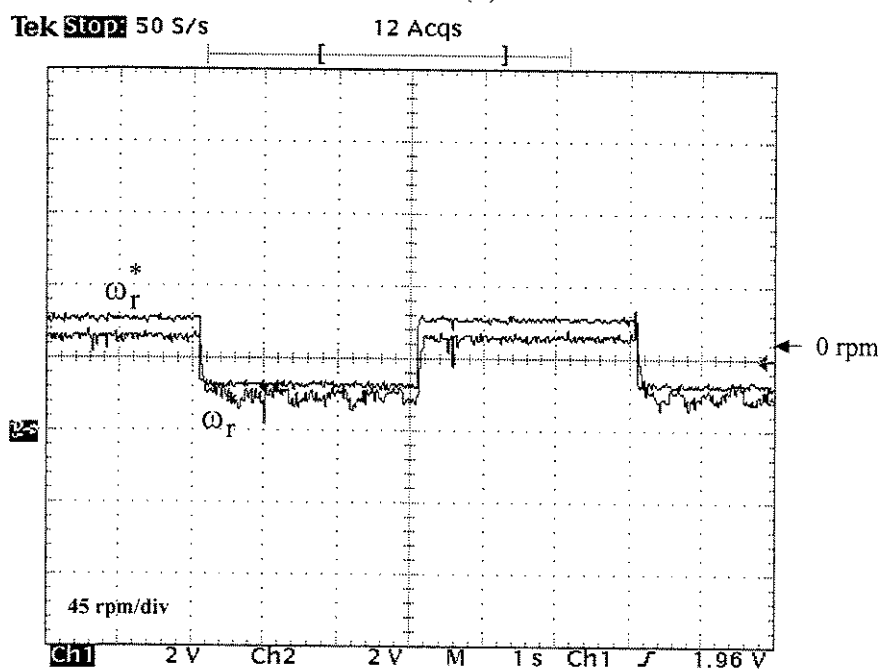
(b)

Figura 5.11- Onda quadrada na referência de velocidade. (a) ω_r^* de 0 a 1.260 rpm

(b) ω_r^* de -100 a 100 rpm



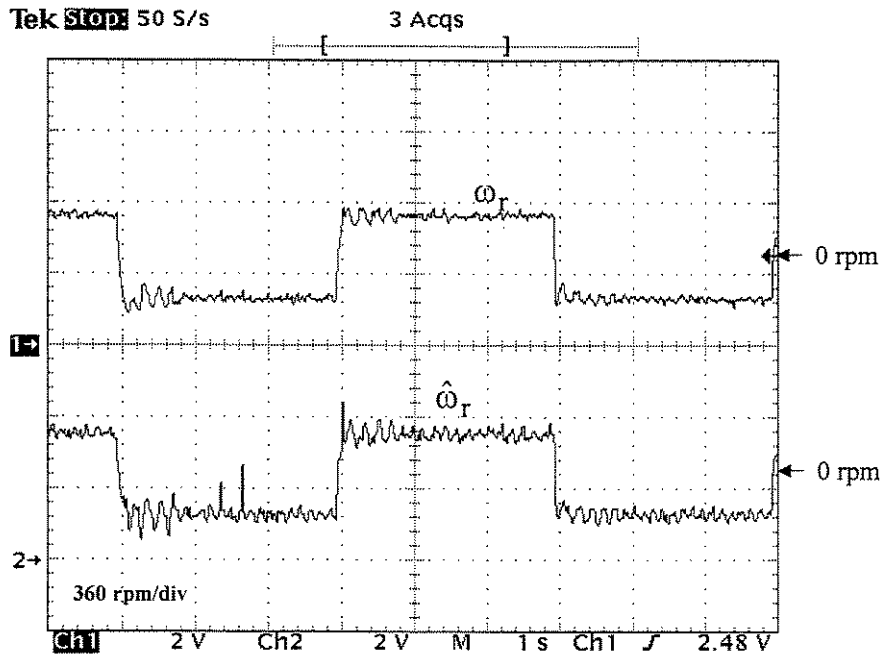
(b)



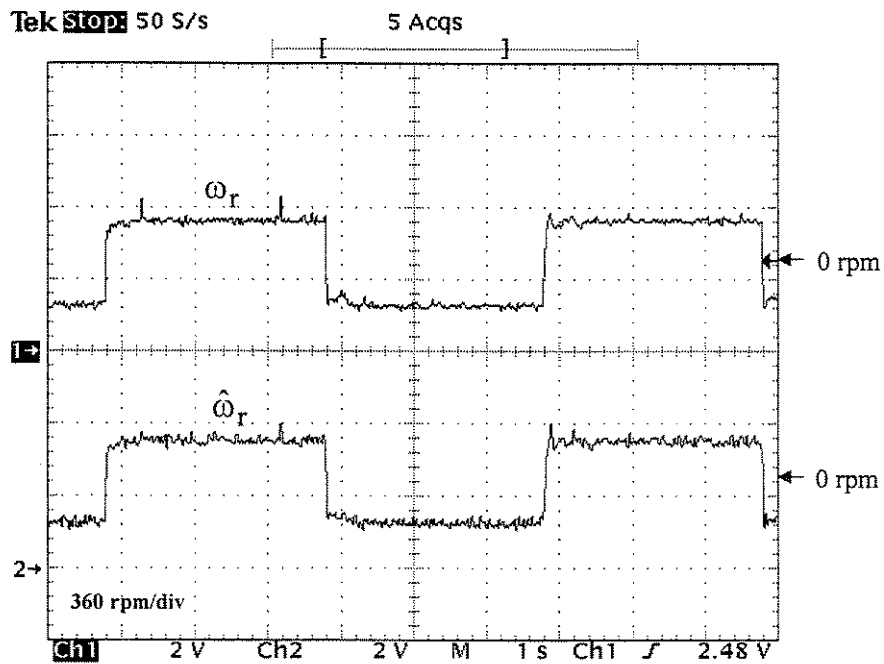
(b)

Figura 5.12- Onda quadrada na referência de velocidade. (a) ω_r^* de -50 a 50 rpm

(b) ω_r^* de -20 a 20 rpm



(a)



(b)

Figura 5.13- Onda quadrada na referência de velocidade de 200 a -200 rpm (a) sem controle de *chattering* (b) com controle de *chattering*.

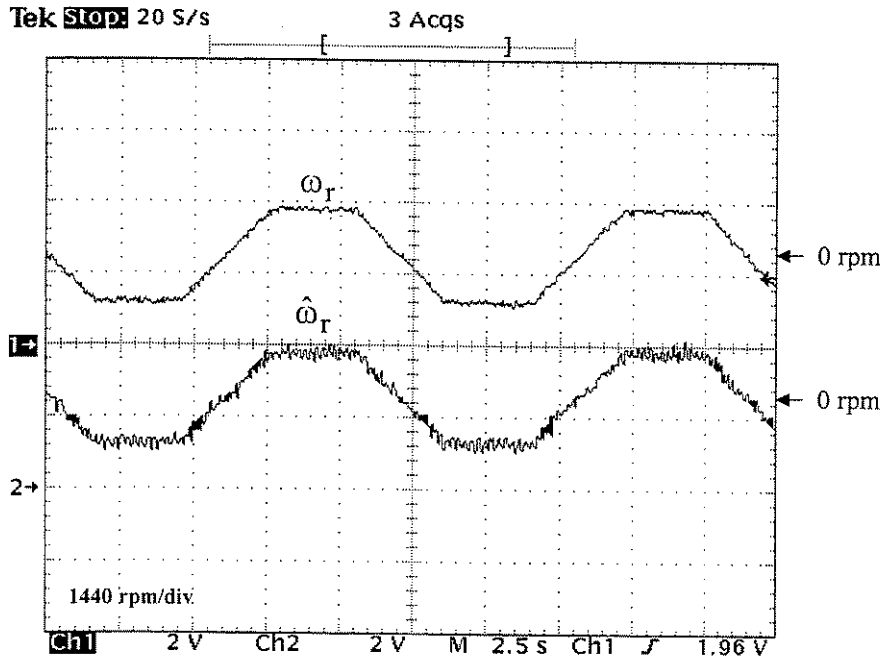


Figura 5.14- Teste de aceleração desaceleração de -900 a 900 rpm.

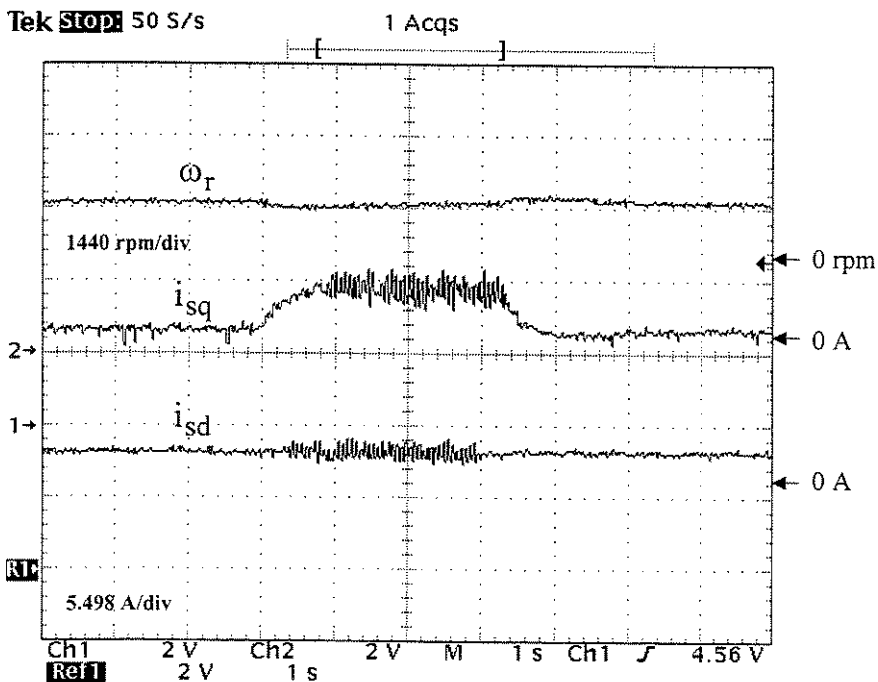


Figura 5.15- Dinâmica do sistema com aplicação de carga (0.45 pu) para $\omega_r = 1.100$ rpm.

Capítulo 6

Conclusões

Neste trabalho foi feito um estudo e implementação de diferentes observadores para a estimação do fluxo de rotor e da velocidade mecânica em motores de indução. O desempenho dinâmico de cada um dos observadores foi avaliado através dos resultados experimentais, obtidos no sistema de controle por orientação direta do fluxo de rotor. O *software* de controle e os algoritmos de estimação foram implementados utilizando o Processador Digital de Sinais (DSP) TMS320F240 da *Texas Instruments*.

O observador com base no algoritmo do filtro de Kalman estendido mostrou bons resultados em toda a faixa de variação da velocidade. Porém, o tempo de execução do algoritmo de estimação foi aproximadamente 150 μs e, este tempo é relativamente elevado quando comparado com o tempo dos outros dois observadores estudados neste trabalho: 20 μs para o Luenberger e 15 μs para o de modos deslizantes. Isto é dado pela complexidade computacional do algoritmo do filtro de Kalman, o qual impõe restrições na seleção dos tempos de execução do algoritmo de controle.

A utilização do observador de Luenberger na estimação do fluxo de rotor e da velocidade do motor mostrou ser uma solução satisfatória diante dos problemas de sintonia e tempo de execução do algoritmo com base no filtro de Kalman. A proposta de projeto do observador com base na minimização da função de custo relacionada ao erro de estimação permitiu obter a matriz ótima de retroação de estado para cada ponto de velocidade considerado. A utilização de algoritmos genéticos na solução do problema de otimização permitiu explorar as vantagens deste método de otimização, além de mostrar a sua aplicação num dos principais problemas de controle em acionamentos elétricos. Os resultados experimentais mostraram um bom desempenho do sistema de controle operando com o

observador tanto em regime permanente como durante o transitório. O tempo de execução da rotina deste observador foi de aproximadamente $20 \mu\text{s}$, tempo este consideravelmente menor que o necessário para a execução do algoritmo do filtro de Kalman, o que permitiu, desta forma, a utilização de uma frequência maior para a execução da rotina de controle.

No que diz respeito ao observador de fluxo em modos deslizantes, foi dada especial atenção ao problema de *chattering* inerente a esta técnica. Como solução deste problema foi aplicado um algoritmo com base em Lógica Fuzzy para o cálculo do termo de correção do observador. Os resultados experimentais obtidos evidenciaram uma boa resposta dinâmica nas estimações de fluxo e de velocidade. O tempo de execução do algoritmo do observador é de aproximadamente $15 \mu\text{s}$. Com a incorporação do algoritmo de controle de *chattering*, este tempo aumentou em $125 \mu\text{s}$, ainda menor que o tempo de execução da rotina do observador com base no filtro de Kalman ($150 \mu\text{s}$).

Os resultados experimentais obtidos mostraram um aumento no erro de regulação de velocidade para baixas velocidades. Este aumento no erro é devido principalmente às variações dos parâmetros do motor utilizados nas equações do observador, em particular das resistências de estator e do rotor, as quais mudam com a temperatura de trabalho da máquina. No controle por orientação direta de fluxo, a velocidade mecânica do motor foi estimada a partir do observador e, para baixas velocidades, foi verificado o aumento no erro de regulação. Uma das causas possíveis pode estar nas variações do valor da resistência do estator, as quais produzem erro na regulação de conjugado de referência e de estimação de velocidade, principalmente para baixas velocidades. Além disso, variações da resistência do rotor tem uma influência uniforme no erro de estimação, em toda a faixa de variação da velocidade. Isto sugere a utilização de algoritmos de identificação de parâmetros para atualizar as equações do modelo e, assim, conseguir um melhor desempenho dinâmico do sistema de controle.

A utilização do DSP TMS320F240 acelerou bastante o processo de implementação do sistema de controle. Embora este processador seja de ponto fixo, o qual requer uma maior preocupação do programador no escalamento dos sinais do sistema, a incorporação dentro da sua arquitetura de módulos dedicados ao controle digital de motores facilita de maneira

considerável a implementação do software de controle. A unidade de PWM do DSP, utilizada no algoritmo de modulação por vetores espaciais, evitou a necessidade de utilizar um número grande de interrupções para gerar os sinais de PWW do inversor, diminuindo assim a complexidade da implementação digital. Além disso, quando a informação de velocidade é requerida pelo sistema, a unidade QEP do DSP realiza esta operação de maneira independente a partir dos sinais provenientes do encoder liberando o processador para a execução das outras tarefas do sistema.

A partir do estudo realizado e dos resultados experimentais obtidos, são propostas as seguintes sugestões para trabalhos futuros:

- Com o objetivo de diminuir o erro de regulação e estimação para baixas velocidades, devido às variações das resistências de rotor e de estator, sugere-se o estudo e a implementação de algoritmos de identificação destes parâmetros para atualizar as equações do observador de fluxo utilizado.
- Avaliar a utilização de outras funções de custo para a otimização dos ganhos do observador de Luenberger, sendo importante a incorporação das incertezas nos parâmetros do motor dentro do problema de otimização.
- Realizar uma análise matemática rigorosa em relação à robustez e estabilidade da aplicação dos algoritmos adaptativos com base em Lógica Fuzzy para o controle de *chattering* no observador de fluxo em modos deslizantes. Neste sentido, um estudo comparativo entre esta nova abordagem e os métodos de controle de *chattering* já propostos poderá ser realizado.
- Estender a aplicação dos observadores de fluxo e de velocidade ao problema de controle não-linear do motor de indução, com o objetivo de obter o desacoplamento entre as variáveis de fluxo e conjugado eletromagnético.
- Aplicação simultânea da técnica de modos deslizantes no controle e observação de estado do motor de indução, com objetivo de aumentar a robustez e desempenho dinâmico do método de controle por orientação pelo campo para o motor de indução.

- Apesar das facilidades de implementação oferecidas pelo DSP utilizado, as características de ponto fixo do processador, assim como não disponibilidade de um compilador C otimizado oferecem algumas limitações ao desenvolvimento do *software*. Este problema torna-se maior na medida que aumenta o número e a complexidade dos algoritmos a serem implementados. Para futuras implementações, sugere-se a utilização de sistemas de desenvolvimento que ofereçam maiores facilidades para o desenvolvimento do programa de aplicação, sem comprometer as vantagens de utilizar recursos em *hardware* dedicados ao controle.

APÊNDICE A

DADOS DO MOTOR DE INDUÇÃO

Condições Nominais

Tensão Nominal: $V_{\text{nom}} = 220/380 \text{ V}$

Corrente Nominal: $I_{\text{nom}} = 8.40/4.86 \text{ A}$

Frequência Nominal: $F_{\text{nom}} = 60 \text{ Hz}$

Potência Nominal: $PE_{\text{nom}} = 3 \text{ HP}$

Velocidade Nominal: $W_{\text{nom}} = 1760 \text{ rpm}$

Condições Nominais

Número de pólos: $P = 4$

Resistência de estator: $R_s = 2.229 \Omega$

Resistência de rotor: $R_r = 1.522 \Omega$

Indutância de magnetização: $L_m = 0.23848 \text{ H}$

Indutância de dispersão de estator: $L_{ls} = 0.00632 \text{ H}$

Indutância de dispersão de rotor: $L_{lr} = 0.01123 \text{ H}$

APÊNDICE B

Fundamentos teóricos dos sistemas de estrutura variável em modos deslizantes

B.1 Sistemas de estrutura variável (SEV)

O conceito de *sistemas de estrutura variável* pode ser explicado através da comparação com o regulador linear de estado para um sistema dinâmico com entrada simples [37], representado pela seguinte equação de estado:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}u \\ u &= \mathbf{k}^T \mathbf{x}\end{aligned}\tag{B.1}$$

No projeto do regulador linear de estado o controle é uma combinação linear do estado do sistema. Os parâmetros do vetor de realimentação de estado \mathbf{k} são selecionados a partir de um determinado critério de projeto, como exemplo utilizando técnicas de alocação de pólos ou através da minimização de um índice quadrático de desempenho. Uma vez definidos os valores deste vetor, o controle passa a ter uma *estrutura fixa*.

Nos sistemas de estrutura variável o controle muda de um elemento a outro do conjunto de possíveis funções contínuas do estado, onde cada elemento deste conjunto definirá uma nova estrutura. Sendo assim, o projeto do sistema de controle deve selecionar os parâmetros de cada uma das estruturas e decidir a lógica para efetuar a mudança entre estas estruturas.

Para uma melhor compreensão do conceito de estrutura, consideremos um sistema linear de segunda ordem sem excitação representado através da seguinte equação diferencial [37]:

$$\ddot{x} + a_2 \dot{x} + a_1 x = 0 \quad (\text{B.2})$$

ou de maneira equivalente pela seguinte equação de estado:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (\text{B.3})$$

A dinâmica do sistema depende das raízes da equação característica:

$$\lambda^2 + a_2 \lambda + a_1 = 0 \quad (\text{B.4})$$

Para a condição em que a solução da equação característica é puramente imaginária tem-se:

$$\text{Re}(\lambda_1) = \text{Re}(\lambda_2) = a_2 = 0 < a_1 \quad (\text{B.5})$$

A partir de uma dada condição inicial x_0 , a trajetória do sistema no plano de fase é representada por uma família de elipses como mostradas nas Figuras B.1(a) e B.1(b). Se $a_1 < 1$, as trajetórias são estendidas ao longo do eixo das abcissas (Figura B.1(a)) e para o caso $a_1 > 1$, as trajetórias são estendidas ao longo do eixo das ordenadas (Figura B.1(b)).

A partir deste exemplo, vale ressaltar algumas características inerentes à dinâmica do sistema. Fica evidente a mudança na trajetória de fase em função do valor do parâmetro a_1 ou seja, para valores do parâmetro $a_1 < 1$ o sistema apresenta um modo de movimento ou dinâmica diferente daquele apresentado para valores $a_1 > 1$. Assim, a variação da *estrutura* do sistema resultou em modos diferentes de movimento. Uma outra característica evidenciada neste exemplo é o fato de nenhuma das estruturas ser estável assintoticamente.

Supondo que a estrutura do sistema é mudada durante o transitório do sistema de acordo com a seguinte lei:

$$a_1 = \begin{cases} \alpha_1 & (\alpha_1 > 1), & \text{se } x_2 x_1 \geq 0 \\ \alpha_2 & (\alpha_2 < 1), & \text{se } x_2 x_1 < 0 \end{cases} \quad (\text{B.6})$$

A lógica de mudança da estrutura imposta pela equação (B.6) divide o plano de fase em quatro regiões separadas pelos eixos de coordenadas como mostrado na Figura B.1(c). Considerando que no instante $t_0 = 0$ o ponto representativo definido pelo vetor de estado $[x_1 \ x_2]^T$ encontra-se na região I, o movimento seguirá uma trajetória elíptica estendida ao longo do eixo das ordenadas ($a_1 = \alpha_1$). No instante de tempo $t_1 > 0$, o ponto representativo alcança a região IV onde a estrutura do sistema muda forçando o movimento através da trajetória elíptica estendida ao longo do eixo das abscissas ($a_1 = \alpha_2$). O movimento continua até a trajetória alcançar a região III, onde novamente acontece a mudança na estrutura do sistema e assim sucessivamente. Dada as diferenças de excentricidade das duas trajetórias elípticas o movimento do sistema é caracterizado por oscilações periódicas amortecidas. A Figura B.1(d) mostra a nova trajetória de fase. A partir da figura nota-se que o sistema de estrutura variável resultante apresenta estabilidade assintótica. Ou seja, a partir da combinação de duas estruturas instáveis é possível gerar um sistema assintoticamente estável.

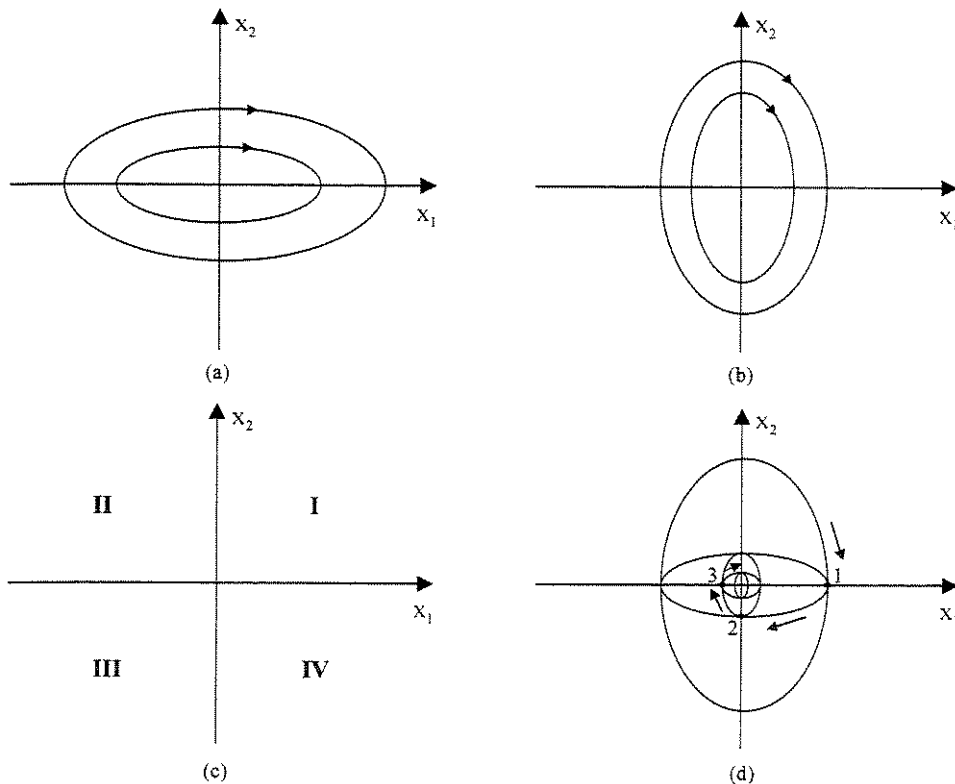


Figura B.1 – Planos de fase para o sistema definido por B.3 e B.6.

Para uma melhor visualização das diferenças de estrutura do sistema, valores positivos e negativos do parâmetro a_1 no exemplo anterior são considerados. Para valores positivos de a_1 , o sistema é caracterizado por trajetórias elípticas estendidas ao longo do eixo das abcissas ou das ordenadas dependendo do valor de a_1 , como foi analisado anteriormente. Para valores negativos de a_1 , as raízes da equação característica são reais com sinal diferente, isto é:

$$\text{Im}(\lambda_1) = \text{Im}(\lambda_2) = a_2 = 0 > a_1, \quad \lambda_1 < 0 < \lambda_2 \quad (\text{B.7})$$

A trajetória do sistema neste caso é formada por uma família de hipérboles com duas assíntotas como mostrado na Figura B.2 (a), sendo as equações das assíntotas dadas por:

$$\sigma_1 = x_2 + \lambda_1 x_1 = 0, \quad \sigma_2 = x_2 + \lambda_2 x_1 = 0 \quad (\text{B.8})$$

onde : $\lambda_{1,2} = \pm\sqrt{a_1}$ são as raízes da equação característica.

Novamente, as duas estruturas são instáveis, sendo o único movimento que converge para a origem aquele ao longo da assíntota $x_2 + \sqrt{a_1}x_1 = 0$. Supondo que a estrutura é mudada durante o transitório de acordo com a seguinte lei:

$$a_1 = \begin{cases} \alpha, & \text{se } x_1 s > 0 \\ -\alpha, & \text{se } x_1 s \leq 0 \end{cases}, \quad \alpha > 0 \quad (\text{B.9})$$

onde: $s = x_2 + \sqrt{\alpha}x_1$

A partir da equação (B.9) o plano de fase é dividido em quatro regiões delimitadas pelo eixo das ordenadas e a assíntota $x_2 + \sqrt{\alpha}x_1 = 0$ como mostrado na Figura B.2(c). Supondo que no instante $t_0 = 0$ o ponto representativo definido pelo vetor de estado $[x_1 \ x_2]^T$ encontra-se na região I, o movimento seguirá uma trajetória elíptica com valor do parâmetro $a_1 = \alpha$. No instante de tempo $t_1 > 0$ o ponto representativo alcança a assíntota $x_2 + \sqrt{\alpha}x_1 = 0$ entrando na região IV onde a estrutura do sistema muda forçando o movimento através de

uma trajetória hiperbólica com valor do parâmetro $a_1 = -\alpha$. Dado que a posição do estado no instante t_1 coincide com a assíntota da família de hipérbolas, a trajetória do sistema seguirá esta linha aproximando-se assintoticamente para a origem. Uma análise similar pode ser feita para diferentes posições iniciais no plano de fase, verificando-se a estabilidade assintótica do sistema de estrutura variável resultante.

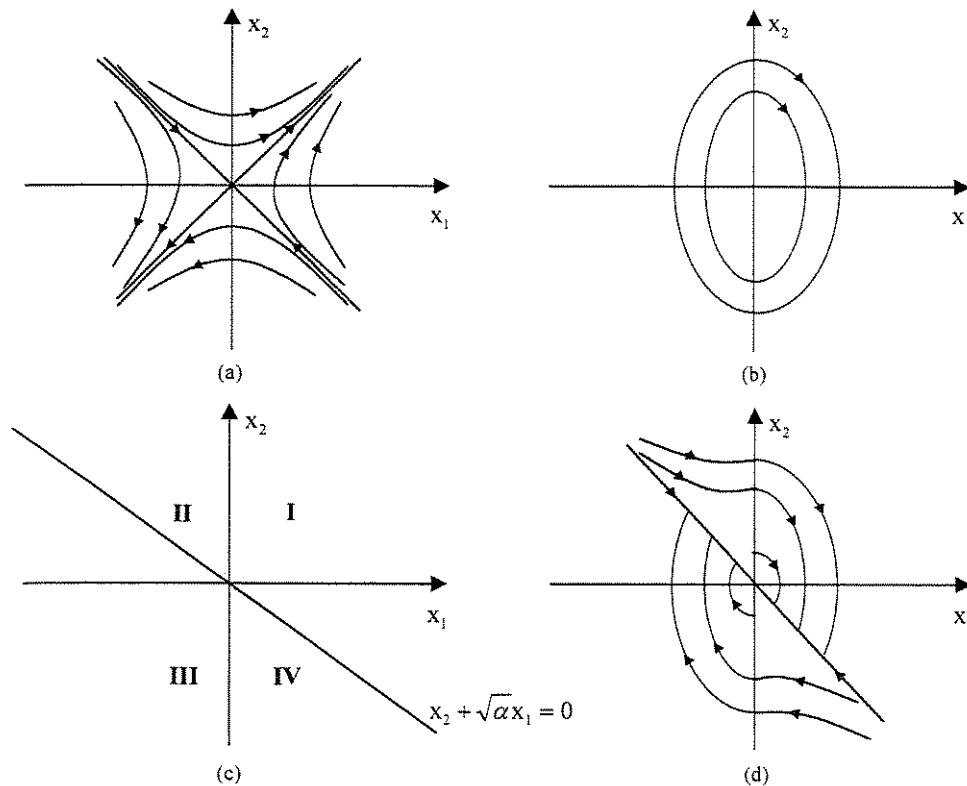


Figura B.2 - Planos de fase para o sistema definido por B.3 e B.9.

B.2 Sistemas de estrutura variável em modos deslizantes.

Nos dois casos analisados o vetor de estado em qualquer instante de tempo encontra-se sobre uma das estruturas inerentes do sistema (elíptica ou hiperbólica). A partir da combinação destas duas estruturas instáveis foi possível obter um sistema assintoticamente estável. De uma maior importância nos sistemas de estrutura variável é a possibilidade de obter trajetórias não presentes em nenhuma das estruturas inerentes ao sistema. Estas trajetórias descrevem um novo tipo de movimento chamado de modos deslizantes [35].

Para mostrar como este tipo de movimento acontece, considera-se que no exemplo anterior a estrutura do sistema é mudada seguindo a lei:

$$a_1 = \begin{cases} \alpha, & \text{se } x_1(x_2 + cx_1) > 0 \\ -\alpha, & \text{se } x_1(x_2 + cx_1) \leq 0 \end{cases}, \quad \alpha > 0, 0 < c < \sqrt{\alpha} \quad (\text{B.10})$$

Nesse caso, a trajetória do sistema é dirigida no sentido de alcançar a linha $\sigma = x_2 + cx_1 = 0$, para qualquer condição inicial no plano de fase. Uma vez que o ponto representativo encontra-se sobre a reta, a trajetória de fase seguirá ao longo desta aproximando-se assintoticamente para a origem como mostrado na Figura B.3. Dado que as trajetórias elípticas e hiperbólicas possuem direções opostas na vizinhança de $\sigma = 0$, qualquer que seja o desvio do ponto a partir de uma dada posição sobre a linha, o mesmo será corrigido pela variação da estrutura imposta pela equação (B.10). O movimento ao longo de $\sigma = 0$, que não faz parte de nenhuma das estruturas do sistema, define um novo regime de movimento chamado de regime de modos deslizantes.

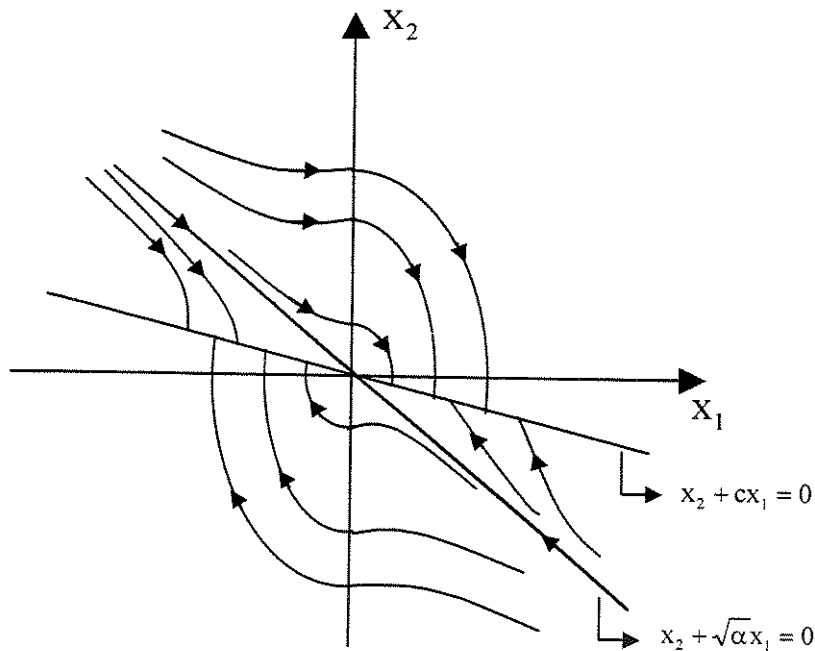


Figura B.3 – Plano de fase para o SEV em modos deslizantes.

A dinâmica do sistema no regime de modos deslizantes é determinada pela equação:

$$\dot{x}_2 + cx_1 = 0 \quad (B.11)$$

É importante ressaltar que esta dinâmica depende unicamente do valor do parâmetro c . A independência do regime de modos deslizantes em relação aos parâmetros da planta e perturbações é de grande importância no projeto de controladores para sistemas com parâmetros variantes no tempo, assim como nos problemas de rejeição de perturbações.

Para mostrar matematicamente a dinâmica do sistema no regime de modos deslizantes, será considerado um sistema dinâmico linear de entrada simples representado na forma canônica controlável pelo seguinte sistema de equações [35]:

$$\begin{cases} \dot{x}_i = x_{i+1}, & i = 1, \dots, n-1 \\ \dot{x}_n = -\sum_{i=1}^n a_i x_i + f(t) + bu \end{cases} \quad (B.12)$$

onde: a_i são parâmetros constantes ou variantes no tempo, u é o controle e $f(t)$ a função de perturbação.

O controle u no sistema (B.12) é uma função do estado com descontinuidade em alguma superfície $s = 0$:

$$u(\mathbf{x}, t) = \begin{cases} u^+(\mathbf{x}, t), & \text{se } s(\mathbf{x}, t) > 0 \\ u^-(\mathbf{x}, t), & \text{se } s(\mathbf{x}, t) < 0 \end{cases} \quad (B.13)$$

$$s = \sum_{i=1}^n c_i x_i, \quad c_i = \text{const}, \quad c_n = 1 \quad (B.14)$$

Se na vizinhança de $s = 0$ as trajetórias de fase possuem direções opostas o regime de modos deslizantes será estabelecido. Resolvendo a equação $s = 0$ para o estado x_n tem-se:

$$x_n = -\sum_{i=1}^{n-1} c_i x_i \quad (B.15)$$

Substituindo o resultado da equação (B.15) no sistema original obtêm-se o sistema de equações que caracteriza o movimento no regime de modos deslizantes:

$$\begin{cases} \dot{x}_i = x_{i+1}, & i = 1, \dots, n-2 \\ \dot{x}_{n-1} = -\sum_{i=1}^{n-1} c_i x_i \end{cases} \quad (\text{B.16})$$

A equação anterior mostra claramente a independência da dinâmica do sistema respeito aos parâmetros da planta a_i e perturbações representadas por $f(t)$, dependendo unicamente dos coeficientes c_i .

B.3 Formulação matemática do regime de modos deslizantes.

Nesta seção é feita uma breve descrição das condições suficientes para a existência do regime de modos deslizantes e as principais abordagens utilizadas para descrever a dinâmica do sistema neste regime.

Seja um sistema dinâmico descrito pela seguinte equação diferencial:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t, \mathbf{u}) \quad (\text{B.17})$$

com descontinuidade em alguma superfície $s(\mathbf{x}) = 0$.

O regime de modos deslizantes na superfície $s(\mathbf{x}) = 0$ será estabelecido se a distância entre o estado atual do sistema e esta superfície, e a velocidade de variação desta distância $\dot{s}(\mathbf{x})$ possuírem sinais opostos [45]. Matematicamente isso pode se descrito por :

$$\begin{aligned} \dot{s}(\mathbf{x}) &> 0, \text{ se } s(\mathbf{x}) < 0 \\ \dot{s}(\mathbf{x}) &< 0, \text{ se } s(\mathbf{x}) > 0 \quad i = 1, \dots, m \end{aligned} \quad (\text{B.18})$$

ou de maneira equivalente pela desigualdade:

$$s(\mathbf{x})\dot{s}(\mathbf{x}) < 0 \quad (\text{B.19})$$

Uma condição suficiente de existência similar à expressa pela equação (B.19), mas de natureza local foi proposta por V. I. Utkin [46]:

$$\lim_{s \rightarrow 0^+} \dot{s}(\mathbf{x}) < 0 \quad \text{e} \quad \lim_{s \rightarrow 0^-} \dot{s}(\mathbf{x}) > 0 \quad (\text{B.20})$$

A dinâmica do sistema expresso pela equação (B.17) na superfície de descontinuidade não pode ser descrita de maneira adequada através da teoria clássica de equações diferenciais. Para resolver este problema, várias abordagens têm sido propostas, nas quais o objetivo é o de transformar o sistema original a uma forma que permita a aplicação das técnicas clássicas de análise. A seguir são descritos, de forma sintética, dois destes métodos: 1) Princípio de Filippov; 2) Lei de controle equivalente.

Método 1: Princípio de Filippov [43]

A. F. Filippov estabeleceu uma rigorosa definição matemática do movimento do sistema no regime de modos deslizantes. Considerando que a dinâmica do sistema é representada pela seguinte equação diferencial:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (\text{B.21})$$

com descontinuidade em alguma superfície $s(\mathbf{x}) = 0$.

Dado que o sistema é de estrutura variável, a dinâmica do mesmo pode ser descrita por:

$$\mathbf{f}(\mathbf{x}) = \begin{cases} \mathbf{f}^+(\mathbf{x}, \mathbf{u}), & \text{se } s(\mathbf{x}) > 0 \\ \mathbf{f}^-(\mathbf{x}, \mathbf{u}), & \text{se } s(\mathbf{x}) < 0 \end{cases} \quad (\text{B.22})$$

O movimento do sistema em $s(\mathbf{x}) = 0$ é descrito por A. F. Filippov como uma ponderação das duas estruturas representadas na equação (B.22):

$$\dot{\mathbf{x}} = \mathbf{f}_0(\mathbf{x}) = \mu \mathbf{f}_0^+ + (1 - \mu) \mathbf{f}_0^-, \quad 0 \leq \mu \leq 1 \quad (\text{B.23})$$

onde:

$$\mathbf{f}_0^+ = \lim_{s \rightarrow 0} \mathbf{f}^+(\mathbf{x}, \mathbf{u}), \quad \mathbf{f}_0^- = \lim_{s \rightarrow 0} \mathbf{f}^-(\mathbf{x}, \mathbf{u}) \quad (\text{B.24})$$

A constante μ depende do estado do sistema sendo selecionada de maneira que a função $f_0(\mathbf{x})$ seja tangente à superfície $s(\mathbf{x}) = 0$. Esta constante pode ser interpretada como a parte do período durante a qual é válida a condição $s(\mathbf{x}) > 0$ e $1-\mu$ representa a parte do período na qual $s(\mathbf{x}) < 0$. A Figura B.4 mostra a interpretação geométrica do princípio de Filippov.

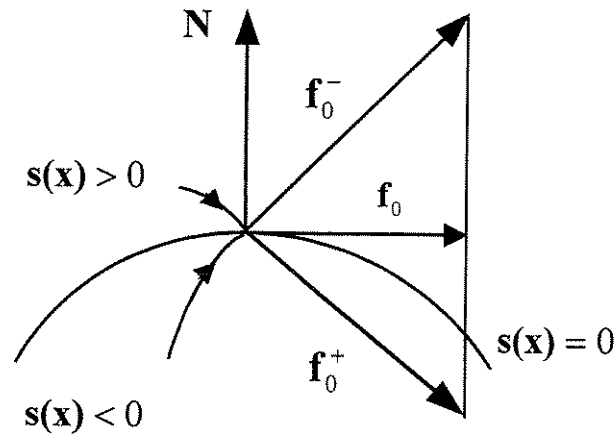


Figura B.4 - Interpretação geométrica do princípio de Filippov.

Método 2: Lei de controle equivalente [44]

Outra maneira de descrever o movimento do sistema no regime de modos deslizantes é através da lei de controle equivalente. Consideremos um sistema dinâmico representado pela seguinte equação diferencial:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}) + \mathbf{B}(\mathbf{x})\mathbf{u} \quad (\text{B.25})$$

com descontinuidade em alguma superfície $s(\mathbf{x}) = 0$.

A lei de controle equivalente é obtida considerando $\dot{s}(\mathbf{x}) = 0$ como condição necessária para a trajetória do sistema permanecer na superfície $s(\mathbf{x}) = 0$, ou seja:

$$\dot{s}(\mathbf{x}) = \frac{\partial s}{\partial \mathbf{x}} \mathbf{A}(\mathbf{x}) + \frac{\partial s}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x})\mathbf{u} = 0 \quad (\text{B.26})$$

Resolvendo a equação (B.26) em relação ao vetor de controle \mathbf{u} obtemos:

$$\mathbf{u} = \mathbf{u}_{\text{eq}}(\mathbf{x}) = -\left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x})\right)^{-1} \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{A}(\mathbf{x}) \quad (\text{B.27})$$

Substituindo (B.27) em (B.25), obtemos a equação que representa a dinâmica do sistema no regime de modos deslizantes:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}) - \left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x})\right)^{-1} \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{A}(\mathbf{x}) \quad (\text{B.28})$$

na qual a existência da inversa da matriz $\left(\frac{\partial \mathbf{s}}{\partial \mathbf{x}} \mathbf{B}(\mathbf{x})\right)$ é uma condição necessária.

Descrição funcional do sistema implementado

C.1 Introdução

Embora o desenvolvimento das técnicas de controle por orientação pelo campo data do final da década de 60 e início da década de 70 (Hasse 1969, Blaschke 1972), estas só passaram a ser largamente implementadas nos últimos anos, dados os crescentes avanços na área da eletrônica digital, e o conseqüente surgimento de microprocessadores com uma alta velocidade de processamento e de características funcionais específicas, facilitando a implementação dos mais diversos algoritmos de controle.

O controle em acionamentos elétricos sempre foi visto com especial interesse pelos fabricantes de microprocessadores, dada a complexidade de cálculo e as especificidades funcionais dos algoritmos de controle a serem implementados. No final da década de 80, o surgimento de microcontroladores dedicados ao controle de máquinas elétricas deram um grande impulso à implementação de controladores vetoriais em acionamentos elétricos, em particular aos métodos de orientação pelo campo. No início dos anos 90, os processadores digitais de sinais (DSP) passaram a ser predominantes no controle digital de máquinas elétricas, pelas características de alta velocidade de processamento, flexibilidade operacional e custo declinante destes processadores.

Neste apêndice é feita uma descrição funcional da implementação do sistema de controle por orientação do fluxo de rotor utilizando o DSP TMS320F240 da *Texas Instruments*.

C.2 Características Gerais do DSP TMS320F240

Para o desenvolvimento do sistema de controle foi escolhida como plataforma de hardware o *TMS320C24x Evaluation Module* (C24x EVM) da Texas Instruments™. O C24x EVM é um sistema de desenvolvimento baseado no Processador Digital de Sinais DSP TMS320F240 projetado e otimizado para aplicações de controle de motores. A utilização desta plataforma de hardware permite aproveitar a capacidade de processamento da família de DSPs C24x e a integração de periféricos dedicados ao controle, reduzindo os custos totais de implementação e aumentando o desempenho total do sistema. A Figura C.1 mostra um diagrama do C24x EVM.

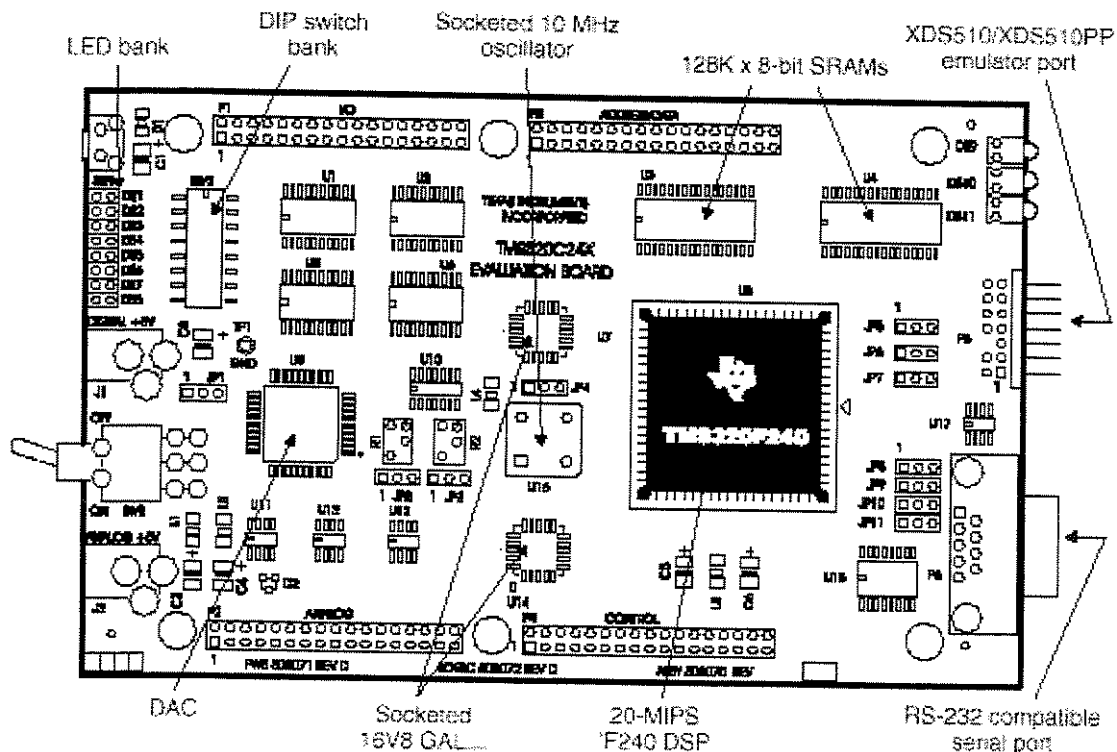


Figura C.1 – Diagrama do TMS320C24x Evaluation Module (C24x EVM).

O C24x EVM utiliza o DSP TMS320F240 o qual tem como características principais as seguintes:

- Tempo do ciclo de instrução: 50 ns (20 MIPS)
- 16K words de memória Flash.
- 544 words de memória de acesso duplo (DARAM).
- Timers de propósito geral com 6 modos de operação.
- Dois conversores A/D de 10 bits e tempo máximo de conversão de 6.6 μ s.
- Interface com capacidade para captura de pulsos provenientes de encoder (QEP).
- 9 canais independentes de “pulse-with modulation” (PWM)
- Módulo de comunicação serial (SCI).

A Figura C.2 mostra um diagrama de blocos da arquitetura do DSP TMS320F240. Além das características associadas ao processador, o C24x EVM apresenta um conversor digital analógico de 12 bits com 4 canais de conversão, sendo isso de grande utilidade nas etapas de desenvolvimento e depuração do programa. Um total de 128K de memória externa é suportada pelo C24x EVM, sendo esta particionada da seguinte maneira:

- 64K words de memória externa de programa
- 32K words de memória externa de dados locais
- 32K words de memória externa de dados globais

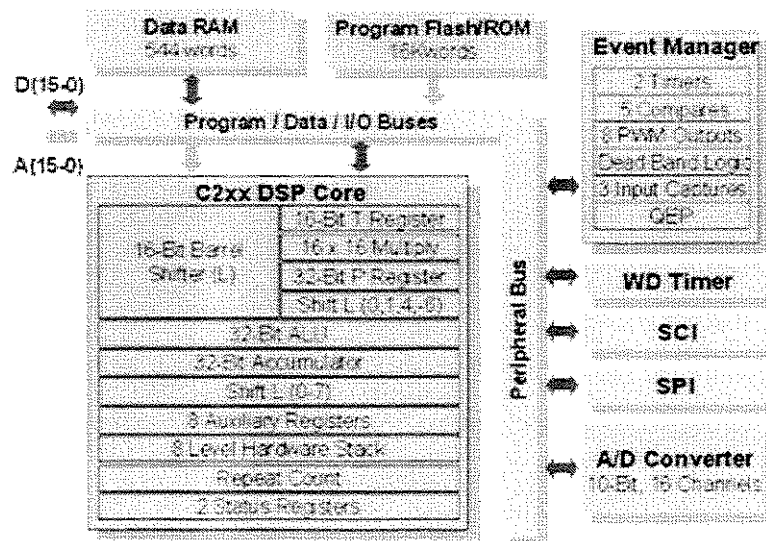


Figura C.2 – Arquitetura do DSP TMS320F240

O desenvolvimento dos programas de controle é feito no computador utilizando as ferramentas de software disponíveis (assembler, compilador C, depurador). A comunicação entre o computador e a placa C24x EVM através do emulador XDS10PP como mostrado na Figura C.3.

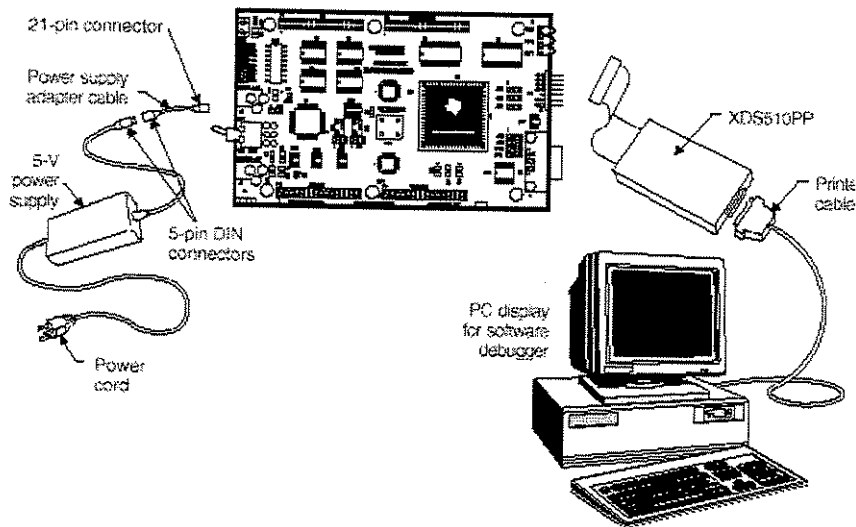


Figura C.3 – Comunicação Computador-C24x EVM.

C.3 Modelo PU e considerações numéricas da implementação

Dado que o TMS320F240 é um DSP de ponto fixo, para a implementação do controle é mais conveniente a utilização do modelo por unidade (pu). Os valores base do modelo em pu são obtidos a partir dos valores de pico nominais da corrente e da tensão de estator, assim como da frequência elétrica do motor:

$$I_b = \sqrt{2} \cdot I_n = \sqrt{2} \cdot 4.86 = 6.873 \text{ A} \quad (\text{C.1})$$

$$V_b = \sqrt{2} \cdot V_n = \sqrt{2} \cdot 220 = 311.127 \text{ V} \quad (\text{C.2})$$

$$\omega_b = 2 \cdot \pi \cdot f_n = 2 \cdot \pi \cdot 60 = 377 \text{ rad/seg} \quad (\text{C.3})$$

$$\lambda_b = \frac{V_b}{\omega_b} = 0.825 \text{ wb} \quad (\text{C.4})$$

onde I_b , V_b , ω_b , λ_b são os valores base utilizados para as variáveis corrente, tensão,

freqüência e fluxo do motor respectivamente.

O formato numérico utilizado na maior parte desta implementação para representar os valores das variáveis e parâmetros do motor foi o formato 4.12f. Neste formato, 4 bits são utilizados para representar a parte inteira do valor da variável ou parâmetro e 12 bits para a parte fracionária. A resolução deste formato é dada por:

$$\frac{1}{2^{12}} = 0,000244140625 \quad (\text{C.5})$$

Sendo assim, os números que podem ser representados com o formato 4.12f encontram-se na faixa de variação de -8 a 7.99975586 . A relação entre o valor real da variável ou parâmetro e a sua representação no formato 4.12f é mostrada na Figura C.4.

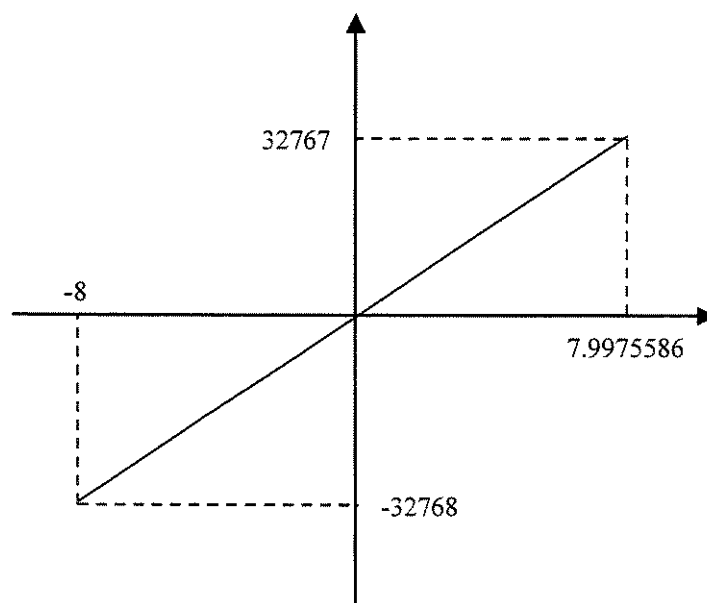


Figura C.4 – Representação da variável ou parâmetro no formato 4.12f.

A utilização do formato numérico 4.12f na implementação digital permite trabalhar no programa de controle com valores das variáveis do motor acima de 1 pu. Isso garante o funcionamento adequado do algoritmo e evita os problemas de saturação tanto nas condições de regime como durante o transitório onde em dependência do ponto de operação os valores das grandezas terminais do motor podem superar os valores nominais.

C.4 Medição e escalamento dos sinais de corrente

Para a implementação do algoritmo de controle é necessária a medição das correntes nas fases a e b do estator. Para isso foram utilizados sensores de efeito Hall. O circuito de condicionamento dos sinais de corrente está mostrado na figura C.5. A função deste circuito consiste simplesmente em transformar os sinais de corrente vindos do sensor e amostrados no início do período de PWM (a cada 300 μ s), com amplitude máxima limitada pela saturação do mesmo, em um sinal entre 0 e 5V, faixa de trabalho do conversor analógico-digital do DSP.

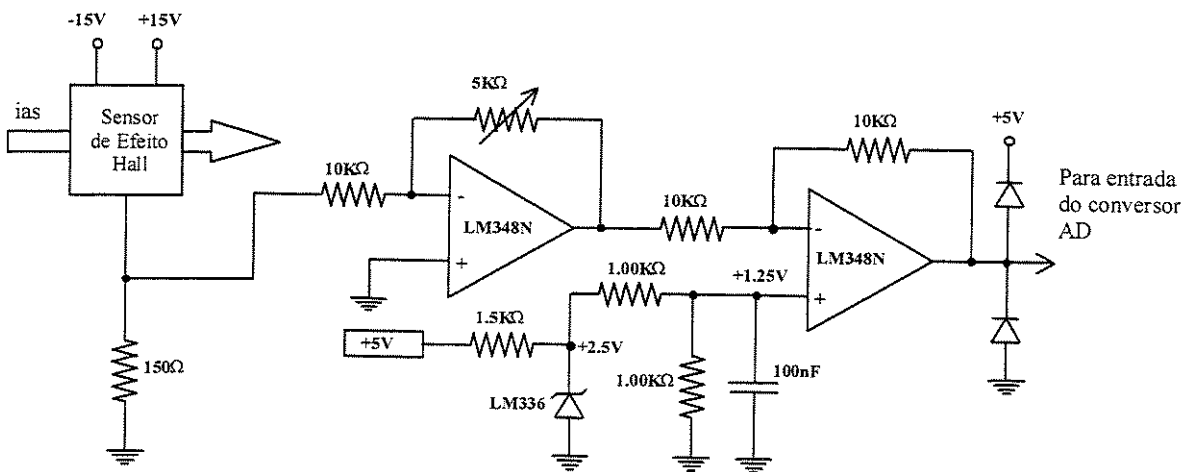


Figura C.5 - Circuito de acondicionamento dos sinais de corrente.

Uma vez que os sinais de corrente ingressam no DSP na faixa de 0 a 5V, o primeiro passo é retirar o nível DC destes para obter valores positivos e negativos das correntes. Dado que o conversor A/D do DSP tem uma resolução de 10 bits, após tirar o nível DC de 2.5V, as correntes são representadas na faixa de variação de -512 a 511 como mostrado na Figura C.6. O segundo passo é o escalamento destes sinais ou seja, representar estes sinais no sistema pu utilizando o formato numérico 4.12f.

Dado um valor quaisquer da corrente I , o valor desta, depois de retirado o nível DC, é dado por:

$$I' = \left(\frac{512}{I_{\max}} \right) I \quad (C.6)$$

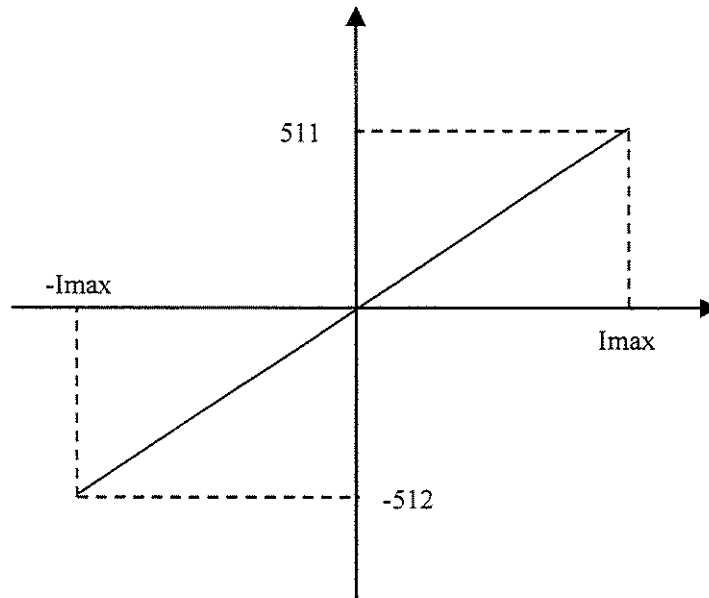


Figura C.6 – Representação dos sinais de corrente antes do escalamento.

onde I_{max} é o máximo valor mensurável da corrente de estator, o qual depende dos ganhos do circuito de condicionamento. Em nosso caso o valor de I_{max} é dado por:

$$I_{max} = 0.73 \cdot \sqrt{2} \cdot 5 \tag{C.7}$$

Para transformar o valor de I' ao sistema pu utilizamos uma constante K_{ipu} de multiplicação, ou seja:

$$I'' = K_{ipu} \cdot I' \tag{C.8}$$

onde I'' é a representação em pu da corrente I utilizando o formato numérico 4.12f.

O valor da constante K_{ipu} pode ser calculado considerando que quando a corrente medida se iguala à corrente de base I_b , o valor de I'' deve ser igual a 1 pu no formato 4.12f. No formato 4.12f o valor de 1 pu é representado por:

$$0 \ 0 \ 0 \ 1. \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 = 4096 \tag{C.9}$$

↓
ponto
decimal

Substituindo $I = I_b$ em (C.6) e logo substituindo o resultado de I' em (C.8) obtemos a seguinte expressão para o cálculo da constante K_{ipu} :

$$K_{ipu} = \frac{4096}{\left(\frac{512}{I_{\max}} \cdot I_b \right)} = 6.006 \Leftrightarrow 06017h \text{ (formato 4.12f)} \quad (\text{C.10})$$

C.5 Medição e escalamento da velocidade

Com o objetivo de comparar a velocidade mecânica real do motor com o valor obtido pelo algoritmo de estimação, a magnitude desta variável foi medida utilizando um encoder incremental de 1200 pulsos/rev. Os sinais **A** e **B** provenientes do encoder passam primeiramente por um circuito de desacoplamento e condicionamento e logo ingressam pela unidade QEP (Quadrature Encoder Pulse) do DSP. A unidade QEP faz a contagem dos flancos de subida e descida dos pulsos do encoder permitindo aumentar a resolução deste dispositivo até 4800 pulsos/rev (1200 pulsos/rev x 4 = 4800 pulsos/rev).

Para converter a informação da velocidade obtida a partir da unidade QEP ao sistema pu selecionamos a constante de multiplicação $K_{\omega pu}$ de maneira que quando a velocidade do motor se iguala à velocidade síncrona (1800 rpm), a representação desta variável no formato 4.12f seja igual a 1 pu, ou seja:

$$4096 = K_{\omega pu} \cdot np \quad (\text{C.11})$$

onde np é igual ao número de pulsos contados pela QEP no período de amostragem da malha de velocidade quando a velocidade do motor iguala-se à velocidade síncrona.

Normalmente na malha de velocidade do sistema de controle é utilizada uma frequência menor que aquela da malha de corrente. Isso ocorre porque a constante de tempo mecânica do motor é de uma ordem maior que as constantes elétricas. Nesta implementação foi utilizado um período de 3 msec para malha de velocidade. Sendo assim, o número de

pulos np é calculado como:

$$np = \frac{1800 \cdot 4800}{60} \cdot 3 \cdot 10^{-3} = 432 \text{ pulsos} \quad (C.12)$$

Substituindo o resultado anterior na equação (C.12) obtemos:

$$K_{\omega_{pu}} = \frac{4096}{432} = 9.481 \quad (C.13)$$

Dado que a faixa de variação do formato 4.12f é de -8 a 7.99975586, a constante $K_{\omega_{pu}}$ não pode ser representada neste formato. Para representar esta constante foi selecionado o formato 8.8f. Neste formato 8 bits são utilizados para representar a parte inteira do valor da variável ou parâmetro e 8 bits para a parte fracionária, obtendo-se uma faixa de variação de -128 a 127.99609375 como mostrado na Figura C.7. A resolução deste formato é dada por:

$$\frac{1}{2^8} = 0,00390625 \quad (C.14)$$

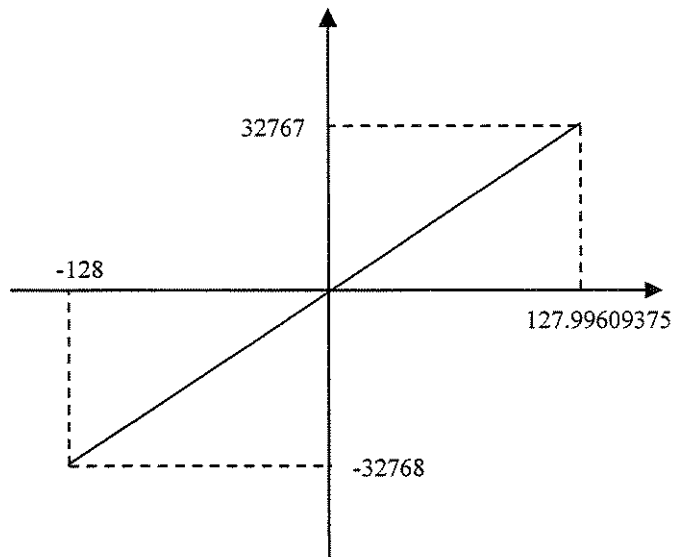


Figura C.7 – Representação da variável ou parâmetro no formato 8.8f.

O valor da constante $K_{\omega_{pu}}$ no formato 8.8f é calculado como segue:

$$K_{\omega_{pu}} = \frac{32767}{128} \cdot 9.481 = 27 \text{ (formato 8.8f)} \quad (C.15)$$

C.6 Implementação da modulação por vetores espaciais (SVM)

Para o controle do inversor foi utilizada a técnica de modulação por vetores espaciais (SVM) [54], dadas as vantagens deste método em relação aos métodos tradicionais de modulação. Quando comparada à modulação senoidal, esta estratégia de modulação apresenta níveis menores de distorção harmônica da corrente e um valor máximo da fundamental maior na tensão de saída [54,55].

Embora a formulação matemática da técnica SVM seja bastante simples, especial atenção deve ser dada à sua implementação digital. Neste sentido, a utilização de um processador com funções de modulação por largura de pulso (PWM) incorporadas por hardware simplifica bastante a implementação do algoritmo de SVM, liberando o tempo do processador para a execução das outras rotinas de controle. O DSP TS320F240 possui uma unidade dedicada para a geração dos sinais de PWM, sendo a única preocupação do programador calcular os tempos de chaveamento e carregar os registradores de comparação do DSP a cada ciclo de PWM para gerar o vetor de referência.

Esta seção trata da implementação da técnica de modulação SVM simétrica utilizada no acionamento proposto. Nas seções seguintes serão abordados aspectos relacionados ao escalamento das variáveis de entrada ao algoritmo SVM, determinação do setor do vetor de referência e o cálculo dos tempos para os sinais de PWM.

C.6.1 Escalamento dos sinais de entrada ao algoritmo SVM

O algoritmo SVM recebe como variáveis de entrada as referências das componentes $V_{s\alpha}^*$ e $V_{s\beta}^*$ do vetor tensão do estator no sistema estacionário α - β e o valor do tensão V_{DC} no barramento DC do inversor. Na saída do algoritmo são obtidos três sinais de PWM para o controle das chaves do inversor. Os valores de $V_{s\alpha}^*$ e $V_{s\beta}^*$ ingressam ao algoritmo no formato pu provenientes da transformação do sistema síncrono ao estacionário (transformação inversa de Park) a qual é realizada na saída dos reguladores PI da malha de

corrente do sistema de controle. O parâmetro V_{DC} é transformado ao sistema pu e expresso no formato numérico 4.12f através da seguinte expressão:

$$v_{DC} = \frac{V_{DC}}{V_b} = \frac{220 \cdot \sqrt{2}}{311.127} = 1 \Leftrightarrow 4096 \quad (\text{formato 4.12f}) \quad (C.16)$$

C.6.2 Determinação do setor do vetor tensão de referência

A Figura C.8 mostra a representação dos vetores fixos de tensão para as diferentes combinações dos estados das chaves do inversor. A área entre dois vetores consecutivos no plano é chamada de setor e, portanto, há 6 setores distintos. O vetor tensão de referência V^* pode ser representado como uma combinação linear dos vetores base que limitam o seu setor (vetores adjacentes) e dos vetores nulos. Na Figura C.8 os vetores base estão nomeados V_1, V_2, V_3, V_4, V_5 e V_6 , os vetores nulos que não aparecem na figura são $V_0 (000)$ e $V_7 (111)$.

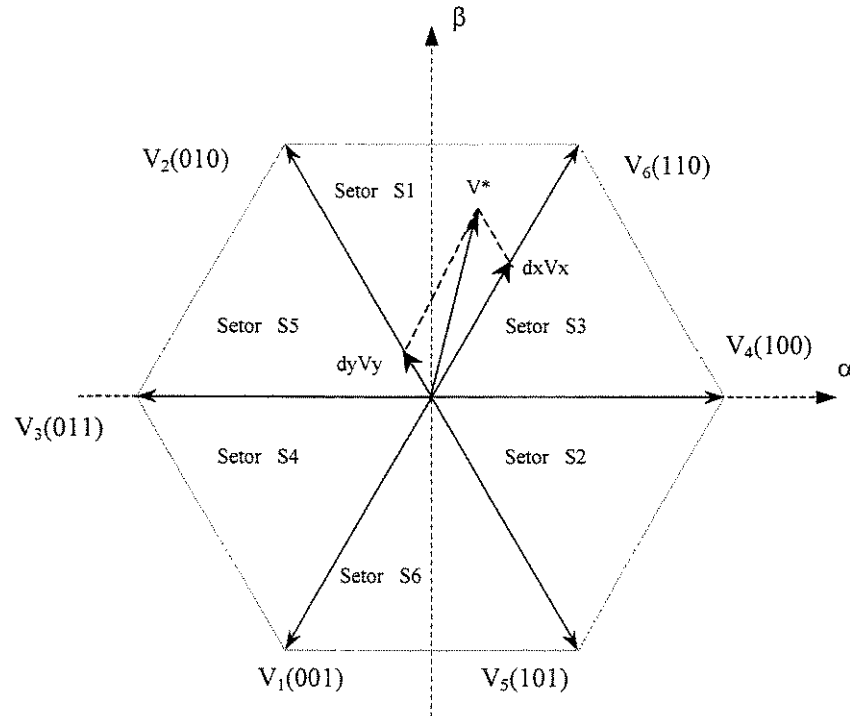


Figura C.8 – Representação dos vetores fixos de tensão para as diferentes combinações de estados das chaves no plano α, β .

O vetor de referência pode ser decomposto em função dos vetores base do setor e dos vetores nulos como segue:

$$V^* = d_x V_x + d_y V_y + d_z V_z \quad (C.17)$$

onde V_x e V_y são os vetores limites do setor onde encontra-se o vetor de referência V^* , V_z indica o vetor nulo aplicado e d_x , d_y e d_z as frações em relação a 1 ("duty rates") que representa o tempo em relação ao período de PWM que permanecem aplicados os vetores V_x , V_y e V_z de maneira a sintetizar o vetor de referência original.

Para o cálculo dos tempos dos sinais de PWM utilizando o algoritmo SVM o primeiro passo consiste em determinar o setor atual do vetor de referência V^* . Uma maneira simples de fazer isto minimizando o tempo de implementação é reportada em [56]. Para isto são definidas as variáveis V_{ref1} , V_{ref2} e V_{ref3} em função das componentes $V_{s\alpha}^*$ e $V_{s\beta}^*$ do vetor tensão de referência:

$$V_{ref1} = V_{s\beta}^* \quad (C.18)$$

$$V_{ref2} = \frac{1}{2} (\sqrt{3} V_{s\alpha}^* - V_{s\beta}^*) \quad (C.19)$$

$$V_{ref3} = \frac{1}{2} (-\sqrt{3} V_{s\alpha}^* - V_{s\beta}^*) \quad (C.20)$$

A partir de (C.18), (C.19) e (C.20) são calculados os coeficientes A, B e C, os quais podem assumir valor de 0 ou 1 em dependencia do sinal de V_{ref1} , V_{ref2} e V_{ref3} respectivamente. Para isto são aplicadas as seguintes regras:

IF $V_{ref1} > 0$ **THEN** A=1, **ELSE** A=0

IF $V_{ref2} > 0$ **THEN** B=1, **ELSE** B=0

IF $V_{ref3} > 0$ **THEN** C=1, **ELSE** C=0

Com os valores obtidos para os coeficientes A,B e C, o setor do vetor de referência é calculado através da seguinte expressão:

$$\text{setor} = A + 2B + 4C \tag{C.21}$$

C.6.3 Determinação dos tempos para os sinais de PWM

Uma vez determinado o setor do vetor de referência, o segundo passo consiste no calculo dos tempos t_1 e t_2 durante o qual serão aplicados os vetores base que limitam o setor em questão. A tabela C.1 mostra a definição dos tempos t_1 e t_2 para cada um dos setores do plano $\alpha\beta$, onde as variáveis X e Y que aparecem na tabela são definidas para um período T_{pwm} do sinal de PWM através das seguintes equações [56]:

$$X = \sqrt{3} \frac{T_{pwm}}{v_{DC}} v_{s\beta}^* \tag{C.22}$$

$$Y = \left(\frac{\sqrt{3}}{2} v_{s\beta}^* + \frac{3}{2} v_{s\alpha}^* \right) \frac{T_{pwm}}{v_{DC}} \tag{C.23}$$

$$Z = \left(\frac{\sqrt{3}}{2} v_{s\beta}^* - \frac{3}{2} v_{s\alpha}^* \right) \frac{T_{pwm}}{v_{DC}} \tag{C.24}$$

Tabela C.1- Definição dos tempos para os sinais de PWM

| Tempo \ Setor | 1 | 2 | 3 | 4 | 5 | 6 |
|---------------|---|----|----|----|----|----|
| t_1 | Z | Y | -Z | -X | X | -Y |
| t_2 | Y | -X | X | Z | -Y | -Z |

Se durante a execução do algoritmo a soma dos tempos t_1 e t_2 é maior que o período de PWM, o algoritmo estabelece a condição de saturação, sendo utilizadas então as seguintes equações para o calculo de t_1 e t_2 :

$$t_{lsat} = t_1 \frac{T_{pwm}}{t_1 + t_2} \tag{C.25}$$

$$t_{2\text{sat}} = t_2 \frac{T_{\text{pwm}}}{t_1 + t_2} \quad (\text{C.26})$$

Para obter os sinais de PWM simétricos do algoritmo SVM é necessário calcular os tempos de serviço t_a , t_b e t_c (“duty cycles”) destes sinais. Isto é feito através das seguintes expressões [56]:

$$t_a = \frac{T_{\text{pwm}} - t_1 - t_2}{2} \quad (\text{C.27})$$

$$t_b = t_a + t_1 \quad (\text{C.28})$$

$$t_c = t_b + t_2 \quad (\text{C.29})$$

Os valores de t_a , t_b e t_c são expressados então em unidades de contas de “timer” e carregados nos registradores de comparação CMPR1, CMPR2 e CMPR3 da unidade de PWM do DSP. Os valores carregados nestes registradores dependem do setor atual do vetor de referência como mostrado na tabela C.2. Uma vez feito isto a unidade de PWM do DSP gera por seus pinos de saída os 6 sinais de PWM (PWM $_x$, $x = 1..6$) necessários para comandar o inversor. A Figura C.9 mostra um exemplo dos sinais de PWM e tempos de serviço gerados considerando que o vetor de referência encontra-se no setor 3.

Tabela C.2- Registradores de comparação em função do setor e do tempo de serviço.

| Registrador \ Setor | 1 | 2 | 3 | 4 | 5 | 6 |
|---------------------|-------|-------|-------|-------|-------|-------|
| CMPR1 | t_b | t_b | t_a | t_c | t_c | t_b |
| CMPR2 | t_a | t_c | t_b | t_b | t_a | t_c |
| CMPR3 | t_c | t_b | t_c | t_a | t_b | t_a |

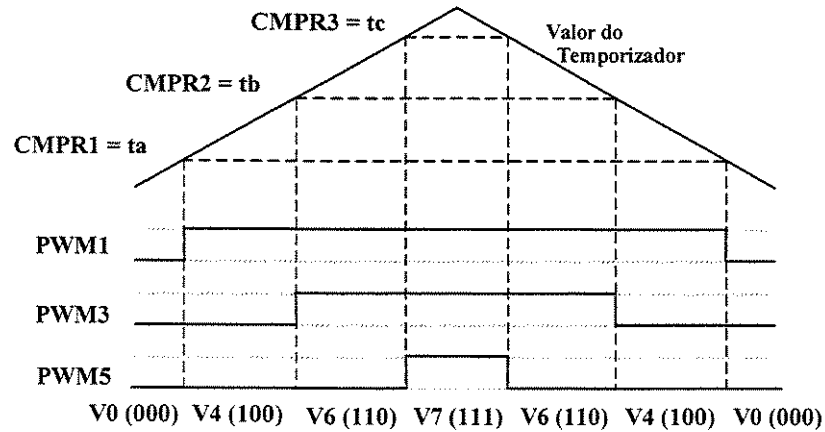


Figura C.9 – Sinais de PWM e tempos de serviço para vetor de referência no setor 3.

C.7 Transformações $(a,b,c) \rightarrow (\alpha,\beta)$, $(\alpha,\beta) \rightarrow (d,q)$ e $(d,q) \rightarrow (\alpha,\beta)$

A implementação do método de controle por orientação pelo campo exige transformações entre os sistemas de referência síncrono e estacionário das correntes e tensões do motor. Além disso, as correntes medidas do motor precisam ser transformadas ao sistema bifásico estacionário antes de ser processadas pelo algoritmo de controle. Estas transformações foram implementadas por *software* dentro do DSP. Os detalhes de sua implementação serão tratados nas próximas seções.

C.7.1 Transformação $(a,b,c) \rightarrow (\alpha,\beta)$

A transformação $(a,b,c) \rightarrow (\alpha,\beta)$ permite representar as correntes medidas no motor em um sistema de referência estacionário em relação ao estator. A partir das correntes medidas i_a e i_b a transformação para o sistema de referência (α,β) é feita através das seguintes equações:

$$i_{s\alpha} = i_a \tag{C.30}$$

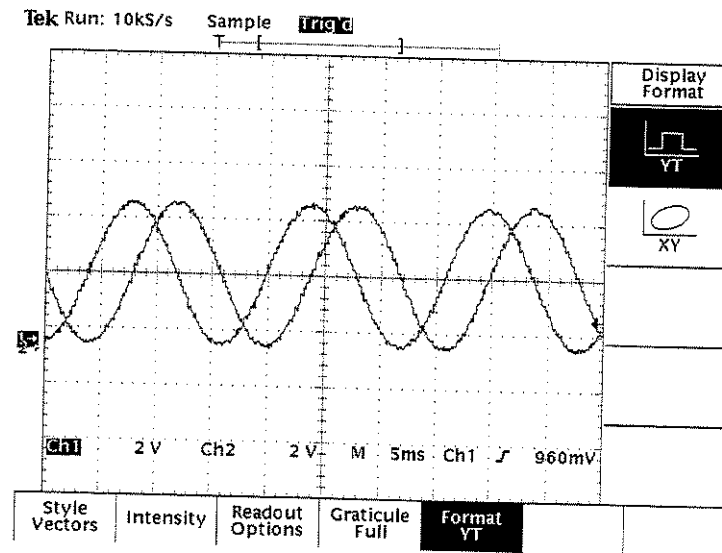
$$i_{s\beta} = \frac{1}{\sqrt{3}}(i_a + 2i_b) \tag{C.31}$$

As correntes i_a e i_b foram representadas no formato numérico 4.12f (seção C.4) e o

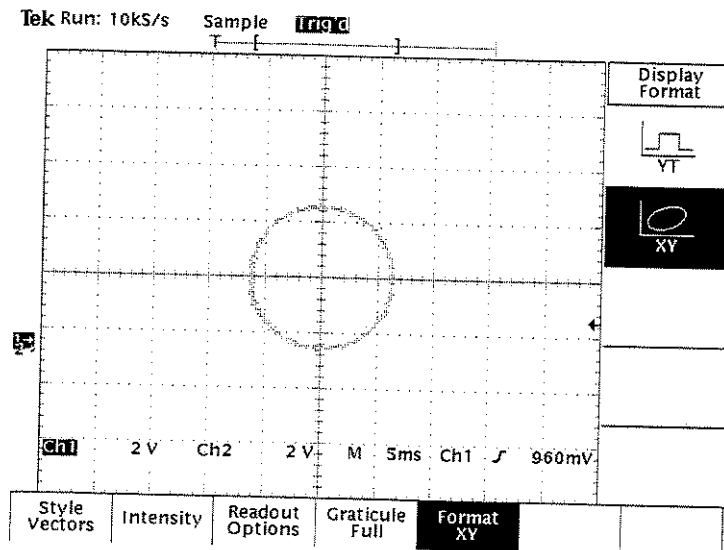
coeficiente $1/\sqrt{3}$ que aparece na equação (C.31) é expresso também neste formato da seguinte maneira:

$$\frac{1}{\sqrt{3}} = 0.5773503 \Leftrightarrow 2364 \text{ (formato 4.12f)} \quad (C.32)$$

Sendo assim, as correntes $i_{s\alpha}$ e $i_{s\beta}$ em pu são representadas no formato 4.12f para seu posterior tratamento pelo algoritmo de controle. A Figura C.10 mostra um exemplo dos resultados de implementação da transformação $(a,b,c) \rightarrow (\alpha,\beta)$.



(a)



(b)

Figura C.10 – Transformação $(a,b,c) \rightarrow (\alpha,\beta)$. (a) $i_{s\alpha}$, $i_{s\beta}$ (b) $i_{s\alpha}$ vs $i_{s\beta}$

C.7.2 Transformação $(\alpha, \beta) \rightarrow (d, q)$

A transformação $(\alpha, \beta) \rightarrow (d, q)$ é utilizada dentro do sistema de controle para representar as correntes de estator $i_{s\alpha}$ e $i_{s\beta}$, as quais encontram-se no sistema de referência estacionário, em um sistema de referência síncrono (sistema dq). Os eixos do sistema síncrono giram à frequência síncrona ω_e . A aplicação da transformação $(\alpha, \beta) \rightarrow (d, q)$ permite obter as componentes reais de fluxo e conjugado eletromagnético do vetor corrente de estator, as quais são comparadas com as respectivas componentes de referência na malha de fluxo e conjugado do sistema de controle. Isto é feito utilizando as seguintes equações:

$$i_{ds} = i_{s\alpha} \cos(\theta_e) + i_{s\beta} \sin(\theta_e) \quad (C.33)$$

$$i_{qs} = -i_{s\alpha} \sin(\theta_e) + i_{s\beta} \cos(\theta_e) \quad (C.34)$$

onde θ_e representa a posição espacial do vetor fluxo do rotor

Os valores de seno e do cosseno do ângulo θ_e são obtidos na saída do observador de fluxo de rotor e expressos no formato numérico 4.12f.

C.7.3 Transformação $(d, q) \rightarrow (\alpha, \beta)$

O algoritmo SVM recebe como sinais de entrada as componentes do vetor tensão de referência no sistema estacionário $\alpha\beta$. Para isso é necessário que as componentes V_{ds}^* e V_{qs}^* do vetor tensão obtidas nas saídas dos reguladores PI das malhas de fluxo e conjugado sejam transformadas para o sistema de referência estacionário. Nesse caso é utilizada a transformação $(d, q) \rightarrow (\alpha, \beta)$ representada pelas seguintes equações:

$$V_{s\alpha} = V_{ds}^* \cos(\theta_e) - V_{qs}^* \sin(\theta_e) \quad (C.35)$$

$$V_{s\beta} = V_{ds}^* \sin(\theta_e) + V_{qs}^* \cos(\theta_e) \quad (C.36)$$

onde V_{ds}^* e V_{qs}^* são as componentes do vetor tensão de referência no sistema síncrono.

Da mesma forma que no caso da transformação $(\alpha, \beta) \rightarrow (d, q)$, os valores de seno e do cosseno do ângulo θ_e são obtidos no formato 4.12f a partir do observador de fluxo de rotor.

C.8 Regulador Proporcional Integral (PI)

Na estrutura do controle por orientação direta do fluxo de rotor é utilizado um regulador PI na malha de velocidade, que produz na sua saída o valor da componente de corrente de estator associada à produção de conjugado eletromagnético. O regulador PI usa como entrada o erro entre a velocidade mecânica de referência da máquina e a velocidade estimada. Além disto, reguladores PI são também utilizados na malha de fluxo e de conjugado do sistema de controle para gerar as componentes do vetor tensão de referência no sistema síncrono. O modelo discreto do regulador PI, utilizado na implementação digital, é discutido a seguir.

C.8.1 Modelo discreto do regulador PI

Dada uma entrada $e(t)$ ao regulador PI, a saída $u(t)$ do regulador é dada por [57]:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t) dt \right] \quad (C.37)$$

onde K_p e T_i são o ganho proporcional e o tempo integral do regulador respectivamente.

Com o objetivo de obter o modelo discreto do regulador, o termo integral presente na equação (C.37) é aproximado por uma somatória utilizando integração retangular. Sendo assim, considerando T_0 como o período de amostragem dos sinais, obtemos a seguinte equação para o modelo discreto do regulador [58]:

$$u(k) = K_p \left[e(k) + \frac{T_0}{T_i} \sum_{i=0}^{k-1} e(i) \right] \quad (C.38)$$

onde k representa o k -ésimo instante de amostragem.

Do ponto de vista da implementação é mais conveniente a utilização de um algoritmo recursivo para o cálculo da saída do regulador, onde a saída $u(k)$ é calculada em função da saída anterior $u(k-1)$ e termos de correção. Para isto a partir de (C.38) $u(k-1)$ pode ser escrito como:

$$u(k-1) = K_p \left[e(k-1) + \frac{T_0}{T_i} \sum_{i=0}^{k-2} e(i) \right] \quad (C.39)$$

Subtraindo (C.39) a partir de (C.38) obtemos:

$$\Delta u(k) = u(k) - u(k-1) = q_0 e(k) + q_1 e(k-1) \quad (C.40)$$

onde:

$$q_0 = K_p \quad (C.41)$$

$$q_1 = -K_p \left[-1 + \frac{T_0}{T_i} \right] \quad (C.42)$$

Uma vez calculada a variação da saída do regulador $\Delta u(k)$ através de (C.40) e com a saída no instante anterior $u(k-1)$ armazenada em memória, a saída no regulador no instante k de amostragem é dada por:

$$u(k) = u(k-1) + \Delta u(k) \quad (C.43)$$

Os parâmetros q_0 e q_1 de cada um dos reguladores, como também as variáveis $e(k)$, $e(k-1)$, $u(k)$, $u(k-1)$ e $\Delta u(k)$ foram armazenadas em memória no formato numérico 4.12f.

C.9 Implementação do modelo discreto do motor

A implementação digital do algoritmo do observador de fluxo requer que as equações do modelo matemático do motor sejam discretizadas. Nesta implementação foi utilizada aproximação de primeira ordem para o cálculo das matrizes discretas da equação de estado, resultando para um período de amostragem T no seguinte modelo discreto:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C}_d \mathbf{x}(k) \end{aligned} \quad (\text{C.44})$$

onde:

$$\mathbf{A}_d = \mathbf{I} + \mathbf{A}\mathbf{T} = \begin{bmatrix} 1 - \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) T & 0 & \frac{L_m}{\sigma L_s L_r T_r} T & \frac{L_m}{\sigma L_s L_r} \omega_r T \\ 0 & 1 - \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right) T & -\frac{L_m}{\sigma L_s L_r} \omega_r T & \frac{L_m}{\sigma L_s L_r T_r} T \\ \frac{L_m}{T_r} T & 0 & 1 - \frac{T}{T_r} & -\omega_r T \\ 0 & \frac{L_m}{T_r} T & \omega_r T & 1 - \frac{T}{T_r} \end{bmatrix},$$

$$\mathbf{B}_d = \mathbf{B}\mathbf{T} = \begin{bmatrix} \frac{T}{\sigma L_s} & 0 \\ 0 & \frac{T}{\sigma L_s} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{C}_d = \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (\text{C.45})$$

Dado que as componentes do vetor de estado \mathbf{x} , da entrada \mathbf{u} e da saída \mathbf{y} são representadas no DSP no formato pu, os coeficientes das matrizes discretas \mathbf{A}_d e \mathbf{B}_d são modificados de maneira a representar a relação correta nas equações do modelo no formato pu. Sendo assim, são obtidas as seguintes equações para o modelo discreto do motor:

$$i_{s\alpha}(k+1) = \left(1 - \frac{T}{T_s} \right) i_{s\alpha}(k) + \frac{x_m T}{x_\sigma x_r T_r} \lambda_{r\alpha}(k) + \frac{x_m \omega_b \omega_r T}{x_\sigma x_r} \lambda_{r\beta}(k) + \frac{\omega_b T}{x_\sigma} v_{s\alpha}(k) \quad (\text{C.46})$$

$$i_{s\beta}(k+1) = \left(1 - \frac{T}{T_s} \right) i_{s\beta}(k) - \frac{x_m \omega_b \omega_r T}{x_\sigma x_r} \lambda_{r\alpha}(k) + \frac{x_m T}{x_\sigma x_r T_r} \lambda_{r\beta}(k) + \frac{\omega_b T}{x_\sigma} v_{s\beta}(k) \quad (\text{C.47})$$

$$\lambda_{r\alpha}(k+1) = \frac{x_m T}{T_r} i_{s\alpha}(k) + \left(1 - \frac{T}{T_r} \right) \lambda_{r\alpha}(k) - \omega_b \omega_r T \lambda_{r\beta}(k) \quad (\text{C.48})$$

$$\lambda_{r\beta}(k+1) = \frac{x_m T}{T_r} i_{s\beta}(k) + \omega_b \omega_r T \lambda_{r\alpha}(k) + \left(1 - \frac{T}{T_r} \right) \lambda_{r\beta}(k) \quad (\text{C.49})$$

onde:

$$x_r = \frac{\omega_b L_r}{Z_b}, \quad x_m = \frac{\omega_b L_m}{Z_b}, \quad x_\sigma = \frac{\sigma L_s \omega_b}{Z_b}, \quad T_s' = \frac{1}{\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r} \right)}, \quad Z_b = \frac{V_b}{I_b} \quad (C.50)$$

Com o objetivo de diminuir os erros de aproximação dados pela representação numérica no DSP de ponto fixo utilizado, foi escolhido o formato 1.15f para a implementação do modelo discreto do motor. Neste formato, 1 bit é utilizado para representar a parte inteira do valor da variável ou parâmetro e 15 bits para a parte fracionária, sendo a resolução deste formato igual a:

$$\frac{1}{2^{15}} = 0,00003051757 \quad (C.51)$$

Na tabela C.3 é mostrada a representação dos coeficientes das matrizes do modelo discreto no formato numérico 1.15f considerando um período de amostragem $T = 300 \mu s$. Para o caso da implementação do observador de fluxo com base no filtro de Kalman, onde o modelo de quinta ordem do motor é utilizado (equações 3.1 e 3.2), os elementos acrescentados às matrizes \mathbf{A}_d e \mathbf{B}_d são nulos com exceção do elemento a_{55} igual a:

$$a_{55} = 1 \Leftrightarrow 32767 \text{ (formato 1.15f)} \quad (C.52)$$

As componentes dos vetores de tensão e corrente de estator no sistema estacionário de referência são transformadas ao formato numérico 1.15f antes da sua entrada para o algoritmo do observador de fluxo. Por outro lado, as componentes estimadas do vetor fluxo de rotor e corrente de estator obtidas na saída do observador são transformadas para o formato 4.12f antes de serem processadas pelas demais rotinas do sistema de controle.

Tabela C.3- Representação dos coeficientes das matrizes do modelo discreto em 1.15f

| Coeficiente | Valor Real | Formato 1.15f |
|---|-------------------------------|------------------------|
| $a_{11} = a_{22} = 1 - \frac{T}{T_s}$ | 0.936342076 | 30681 |
| $a_{13} = a_{24} = \frac{x_m T}{x_\sigma x_r T_r}$ | 0.006150372 | 202 |
| $a_{14} = -a_{23} = \frac{x_m \omega_b \omega_r T}{x_\sigma x_r}$ | $0.760823644 \cdot \omega_r$ | $24930 \cdot \omega_r$ |
| $a_{31} = a_{42} = \frac{x_m T}{T_r}$ | 0.007263135 | 238 |
| $a_{33} = a_{44} = 1 - \frac{T}{T_r}$ | 0.998171485 | 32707 |
| $a_{34} = -a_{43} = -\omega_b \omega_r T$ | $-0.226196289 \cdot \omega_r$ | $-7412 \cdot \omega_r$ |
| $a_{12} = a_{21} = a_{32} = a_{41}$ | 0 | 0 |
| $b_{11} = b_{22} = \frac{\omega_b T}{x_\sigma}$ | 0.796653193 | 26104 |
| $b_{12} = b_{21} = b_{31} = b_{32} = b_{41} = b_{42}$ | 0 | 0 |

APÊNDICE D

Programa do sistema de controle por orientação direta do fluxo de rotor

```
=====
;
; Software do controle por orientação direta do fluxo de rotor para o TMS320F240 EVM
;
=====

                .include ".\c240app.h"
                .mmregs

=====
;
; Start
;
=====

                .globl _c_int0    ;Define _c_int0 como símbolo global
                .sect "vectors"

_c_int0 b        _c_int0          ;Vetor para interrupção de reset
_c_int1 b        _c_int1          ;
                b        _c_int2    ;Vetor para a interrupção de PWM (Timer1 underflow)
_c_int3 b        _c_int3          ;
_c_int4 b        _c_int4          ;
_c_int5 b        _c_int5          ;
_c_int6 b        _c_int6          ;
                .space 16*6        ;Reserve 6 words na tabela de interrupções

stack .usect "blockb2",15         ;Reserva memória de stack
dac_val .usect "blockb2",5        ;Reserva memória para o conversor DA

=====
;
; .include de tabelas (lookup table)
;
=====

                .sect "tabsqrt"
sqrttab .include sqrt.tab         ;Tabela de raiz quadrada
;
                .sect "tabluenb"
luenbtabs .include luenb.tab      ;Tabela de ganhos do observador de Luenberger
;
                .sect "tabslid"
slidtab .include slid.tab        ;Tabela de ganhos do observador em modos deslizantes

                .sect "table"
sintab .include sine.tab         ;Tabela de seno
```

```

;
; Definição de constantes
;
;=====
;
; .data
kal_fil      ;Apontador à página de variáveis do observador
u            ;Vetor de entrada
_u_al .word 0 ;ua
_u_be .word 0 ;ub
y          ;Vetor de saída
_i_al .word 0 ;ia
_i_be .word 0 ;ib

diff_f .word 30681, 0, 202, 0, 0 ;Coeficientes do modelo discreto para T=300us
        .word 0, 30681, 0, 202, 0
        .word 238, 0, 32707, 0, 0
        .word 0, 238, 0, 32707, 0
        .word 0, 0, 0, 0, 32767

_x      .word 0, 0, 0, 0, 0 ;Vetor de estado x(k,k)
Cx_1    .word 0, 0, 0, 0, 0 ;Cx_1=[x(1) x(2)]'
_x_1    .word 0, 0, 0, 0, 0 ;Vetor de estado x(k+1,k)

C4      .word 24930 ;[1,4] & [1,5] coef. de diff_f
C9      .word -7412 ;[3,4] & [3,5] coef. de diff_f
C5      .word 26104 ;Coef. wbase*T/xsig coeff
C8      .word 7412 ;Coef. [4,3] e [4,5]
C10     .word -24930 ;Coef. [2,3] e [2,5]
C       .word 32767, 0, 0, 0, 0 ;Matriz C_d
        .word 0, 32767, 0, 0, 0

P       .word 320, 0, 0, 0, 0 ;Prediction err cov mat P(k+1,k)
        .word 0, 320, 0, 0, 0
        .word 0, 0, 320, 0, 0
        .word 0, 0, 0, 320, 0
        .word 0, 0, 0, 0, 320

P_1     .word 0, 0, 0, 0, 0 ;Filter err cov mat P(k+1,k+1)
        .word 0, 0, 0, 0, 0
        .word 0, 0, 0, 0, 0
        .word 0, 0, 0, 0, 0
        .word 0, 0, 0, 0, 0

Q       .word 1000, 1000, 300, 300, 500 ;state noise cov matrix

R       .word 20000, 20000 ;Measure Noise Cov Matrix

RSLT    .space 25*16 ;Resultado de operação matricial (5x5)
_K      .word 0, 0
        .word 0, 0
        .word 0, 0
        .word 0, 0
        .word 0, 0

```

```

Kluebn .set      4095      ;Para a tabela do observador de Luneberger
Kslid  .set      4095      ;Para a tabela do observador em modos deslizantes
Ksqrt  .set      2731      ;Constante de multiplicação para tabela sqrt

```

Definição de variáveis

```

.bss    tmp,1          ;Variável temporária
.bss    tmp1,1         ;Variável temporária
.bss    daouttmp,1     ;Valor a ser enviado ao DA
.bss    ia,1           ;Corrente de fase ia
.bss    ib,1           ;Corrente de fase ib
.bss    ic,1           ;Corrente de fase ic
.bss    vab,1          ;Tensão de linha vab
.bss    vbc,1          ;Tensão de linha vbc
.bss    t1,1           ;Tempo t1 do SVM
.bss    t2,1           ;Tempo t2 do SVM
.bss    Vref1,1        ;Variável para calculo de setor
.bss    Vref2,1        ;Variável para calculo de setor
.bss    Vref3,1        ;Variável para calculo de setor
.bss    VDC,1          ;Tensão do link DC
.bss    taon,1         ; Instante de chaveamento para fase 1 (SVM)
.bss    tbon,1         ; Instante de chaveamento para fase 2 (SVM)
.bss    tcon,1         ; Instante de chaveamento para fase 3 (SVM)
.bss    iSalfa,1       ;Corrente  $i_{\alpha}$  real
.bss    iSbeta,1       ;Corrente  $i_{\beta}$  real
.bss    vSalfa,1       ;Tensão  $v_{\alpha}$  real
.bss    vSbeta,1       ;Tensão  $v_{\beta}$  real
.bss    vSal_ref,1     ;Tensão  $v_{\alpha}$  de referência
.bss    vSbe_ref,1     ;Tensão  $v_{\beta}$  de referência
.bss    iSdref,1       ;Corrente  $i_{sd}$  de referência
.bss    iSqref,1       ;Corrente  $i_{sq}$  de referência
.bss    iSd,1          ;Corrente  $i_{sd}$  real
.bss    iSq,1          ;Corrente  $i_{sq}$  real
.bss    vSdref,1       ;Tensão  $v_{sd}$  de referência
.bss    vSqref,1       ;Tensão  $v_{sq}$  de referência
.bss    epiq,1         ;Erro do regulador da corrente  $i_{sq}$ 
.bss    epid,1        ;Erro do regulador da corrente  $i_{sd}$ 
.bss    xiq,1          ;Parte integral do regulador da corrente  $i_{sq}$ 
.bss    xid,1          ;Parte integral do regulador da corrente  $i_{sd}$ 
.bss    n,1            ;Velocidade mecânica medida
.bss    n_ref,1        ;Velocidade mecânica de referência
.bss    epin,1         ;Erro do regulador de velocidade
.bss    xin,1          ;Parte integral do regulador da velocidade
.bss    X,1            ;Variável do algoritmo SVM
.bss    Y,1            ;Variável do algoritmo SVM
.bss    Z,1            ;Variável do algoritmo SVM
.bss    sector,1       ;Setor do vetor tensão de referência
.bss    sinTeta_cm,1   ;Valor de seno do ângulo de fluxo (modelo de corrente)
.bss    cosTeta_cm,1   ;Valor de cosseno do ângulo de fluxo (modelo de corrente)
.bss    i_mr,1         ;Corrente de magnetização (modelo de corrente)
.bss    fs,1           ;Frequência do vetor fluxo de rotor
.bss    Teta_cm,1     ;Ângulo do vetor fluxo de rotor (modelo de corrente)
.bss    VDCinvT,1     ; $VDCinv*(T/2)$  (utilizado no SVM)
.bss    tetaincr,1     ;Variável do modelo de corrente

```

```

.bss      Index,1           ;Apontador para tabela de seno
.bss      upi,1            ;Utilizada nos reguladores PI de velocidade e corrente
.bss      elpi,1          ;Limite de erro dos reguladores PI
.bss      encincr,1       ;Incremento nos pulsos do encoder
.bss      speedtmp,1      ;Armazena os incrementos nos pulsos do encoder
.bss      speedstep,1     ;Contador de períodos de amostragem
.bss      SPEEDSTEP,1    ;Utilizado para definir período da malha de velocidade
.bss      Kspeed,1       ;Conversão da velocidade ao formato 4.12f
.bss      Isqrefmin,1    ;Limite inferior da referência isq
.bss      Isqrefmax,1    ;Limite superior da referência isq
.bss      Kcurrent,1     ;Conversão da corrente ao formato 4.12f
.bss      Ktension,1    ;Conversão da tensão ao formato 4.12f
.bss      Kr,1           ;Constante T/Tr (modelo de corrente)
.bss      Kt,1           ;Constante 1/(Tr*wb) (modelo de corrente)
.bss      KA,1           ;Constante KA (modelo de corrente)
.bss      SQR3inv,1      ;Constante 1/sqrt(3)
.bss      SQR32,1       ;Constante sqrt(3)/2
.bss      SR8BIT,1      ;Constante para deslocar 8 bits
.bss      INV3,1        ;Constante 1/3
.bss      PWMPRD,1     ;Período de PWM
.bss      Ki,1          ;Parâmetro do PI de corrente
.bss      Kpi,1         ;Parâmetro do PI de corrente
.bss      Kcor,1        ;Parâmetro do PI de corrente
.bss      Kin,1         ;Parâmetro do PI de velocidade
.bss      Kpin,1        ;Parâmetro do PI de velocidade
.bss      Kcorn,1       ;Parâmetro do PI de velocidade
.bss      Vmin,1        ;Limite inferior da tensão de referência
.bss      Vmax,1        ;Limite superior da tensão de referência
.bss      iSal_est,1    ;Corrente  $i_{s\alpha}$  estimada, x(1)
.bss      iSbe_est,1    ;Corrente  $i_{s\beta}$  estimada, x(2)
.bss      psiRal_est,1  ;Componente de fluxo  $\lambda_{r\alpha}$  estimada, x(3)
.bss      psiRbe_est,1  ;Componente de fluxo  $\lambda_{r\beta}$  estimada, x(4)
.bss      n_est,1       ;Velocidade estimada, x(5) no formato 1.15f
.bss      psiR_est,1    ;Amplitude do fluxo de rotor estimado
.bss      sinTeta_est,1 ;Seno do ângulo de fluxo estimado
.bss      cosTeta_est,1 ;Cosseno do ângulo de fluxo estimado
.bss      psiRal_est1,1 ;Componente de fluxo  $\lambda_{r\alpha}$  estimada, x(3) (1.15f)
.bss      psiRbe_est1,1 ;Componente de fluxo  $\lambda_{r\beta}$  estimada, x(4) (1.15f)
.bss      psiR_est1,1   ;Amplitude do fluxo de rotor estimado (1.15f)
.bss      nmin,1       ;Limite inferior da velocidade estimada
.bss      nmax,1       ;Limite superior da velocidade estimada
.bss      kpmras,1     ;MRAS Proportional constant estimada
.bss      kimras,1     ;MRAS Integral constant
.bss      kcmras,1     ;MRAS Correction constant
.bss      copyal,1     ;Utilizada na estimação de velocidade
.bss      copybe,1     ;Utilizada na estimação de velocidade
.bss      eatual,1     ;Utilizada na estimação de velocidade
.bss      nestq4,1     ;Velocidade estimada no formato 4.12f

.text                                           ;link in "text section"
.copy "matGC24.inc"                            ;Cópia macros de operações com matrizes

```

```

nop
_c_int0:
;
;
;=====  

; Configurações gerais do DSP  

;=====  

clrc    xf

;=====  

; Desabilitar watchdog timer  

;=====  

ldp     #DP_PF1
splk    #006Fh, WD_CNTL
splk    #05555h, WD_KEY
splk    #0AAAAh, WD_KEY
splk    #006Fh, WD_CNTL

;=====  

; Inicializar administrador de eventos (event manager)  

; GPTimer 1 => Full PWM  

; Enable Timer 1==0 interrupt on INT2  

; All other pins are IO  

;=====  

ldp     #DP_PF1
splk    #00000010b,CKCR0
splk    #10110001b,CKCR1
splk    #10000011b,CKCR0
splk    #40C0h,SYSCR
lacc    SYSSR
and     #69FFh
sacl    SYSSR
lacc    #0004h
sacl    *
out     *,WSCR
zac
ldp     #DP_EV
sacl    GPTCON
sacl    T1CNT
sacl    T1CMP
sacl    T1PER
sacl    T1CON
sacl    T2CNT
sacl    T2CMP
sacl    T2PER
sacl    T2CON
sacl    T3CNT
sacl    T3CMP
sacl    T3PER
sacl    T3CON
sacl    COMCON
sacl    ACTR
sacl    SACTR
sacl    DBTCON
sacl    CMPR1
sacl    CMPR2
sacl    CMPR3

```

```
sacl    SCMPR1
sacl    SCMPR2
sacl    SCMPR3
sacl    CAPCON
sacl    CAPFIFO
sacl    FIFO1
sacl    FIFO2
sacl    FIFO3
sacl    FIFO4
```

Inicialização da unidade de PWM

```
splk    #666h,ACTR
splk    #100,CMPR1
splk    #200,CMPR2
splk    #300,CMPR3
splk    #0207h,COMCON
splk    #8207h,COMCON
splk    #3000,T1PER
splk    #0,T1CNT
splk    #0A800h,T1CON
lacc    T1CON
or      #40h
sacl    T1CON
```

Inicialização dos pinos de entrada –saída e registrador de stack

```
ldp    #DP_PF2
splk    #0FFFEh,OPCRA
splk    #0FFFFh,OPCRA
splk    #00F3h,OPCRB
lar     ar7,#79h
```

Inicialização da unidade QEP para leitura de velocidade

```
ldp    #DP_EV
splk    #0000h,T3CNT
splk    #00FFh,T3PER
splk    #9870h,T3CON
splk    #0E2F0h,CAPCON
```

Inicialização do conversor análogo – digital (A/D)

```
ldp    #DP_PF1
splk    #0003h,ADC_CNTL2
```

Inicialização de variáveis

```

ldp    #4
zac
sac1   Index
sac1   xid
sac1   xiq
sac1   xin
sac1   upi
sac1   elpi
sac1   Vref1
sac1   Vref2
sac1   Vref3
sac1   Teta_cm
sac1   Index
sac1   iSal_est
sac1   iSbe_est
sac1   n_est
sac1   psiRal_est
sac1   psiRbe_est
sac1   Vref1
sac1   Vref2
sac1   Vref3
sac1   da1
lacc   #1
sac1   da2
lacc   #2
sac1   da3
lacc   #3
sac1   da4
splk   #000Ah,SPEEDSTEP
splk   #097Bh,Kspeed
splk   #-3277,Isqrefmin
splk   #3277,Isqrefmax
splk   #0602h,Kcurrent
splk   #0AE8h,Ktension
splk   #0008h,Kr
splk   #0042h,Kt
splk   #049Ch,KA
splk   #093Dh,SQRT3inv
splk   #0DDBh,SQRT32
splk   #0100h,SR8BIT
splk   #0555h,INV3
splk   #3000,PWMPRD
splk   #0100h,Ki
splk   #01000h,Kpi
splk   #0100h,Kcor
splk   #0035h,Kin
splk   #482Bh,Kpin
splk   #000Bh,Kcorn
splk   #-1147,Vmin
splk   #1147,Vmax
lacc   SPEEDSTEP

```

```

sacl    speedstep
splk    #03ffh,fs
splk    #0199h,vSdref
splk    #0199h,vSqref
splk    #0800h,VDC
splk    #1770h,VDCinvT
splk    #00E4h,n_ref
splk    #0333h,iSdref
splk    #0000h,n_est
splk    #0000h,tpdo1
splk    #0FFFh,tpdo2
lacc    #1000h
sacl    sinTeta_est
lacc    #0
sacl    cosTeta_est
splk    #0800h,iSqref
setc    OVM
spm     0
setc    sxm

```

Inicialização das interrupções

```

ldp     #DP_EV
splk    #07FFh,IFRA
splk    #00FFh,IFRB
splk    #000Fh,IFRC
splk    #0200h,IMRA
splk    #0000h,IMRB
splk    #0000h,IMRC
ldp     #0h
lacc    #0FFh
sacl    IFR
lacc    #0000010b
sacl    IMR
ldp     #4
clrc    INTM

```

Loop de espera

```

LOOP
nop
nop
nop
nop
b      LOOP

```

Rotina de controle em sincronismo com a interrupção de PWM (Timer1 underflow)

_c_int2:

```
LDP    #DP_EV
SPLK  #0FFFFh,IFRA
```

Adquisição dos sinais de corrente e tensão pelo conversor A/D

```
ldp    #DP_PF1
splk   #0001100000100101b,ADC_CNTL1    ;Início de conversão de ia e ib
                                           ;ia por ADCIN3, ia por ADCIN11

bit    ADC_CNTL1,8
conversion
bcnd   conversion,tc
lacc   ADC_FIFO1,10                    ;formato 10.6f
ldp    #4
sach   tmp
;splk  #0199h,tmp
lacl   tmp
and    #3ffh
add    #000Ch                          ;Compensa offset do sensor
sacl   tmp
sub    #512                             ;Retira nível DC (2.5V)
sacl   tmp
spm    3                                ;PM=11, desloca 6 bits após multiplicação
lt     tmp
mpy    Kcurrent
pac
sfr
sfr
spm    0
sacl   ia                               ;ia em formato 4.12f

ldp    #DP_PF1
lacc   ADC_FIFO2,10
ldp    #4
sach   tmp
lacl   tmp
and    #3ffh
add    #000Ah                          ;Compensa offset do sensor
sub    #512                             ;Retira nível DC (2.5V)

sacl   tmp
spm    3                                ;PM=11, desloca 6 bits após multiplicação
lt     tmp
mpy    Kcurrent
pac
sfr
sfr
sacl   ib                               ;ib em formato 4.12f
spm    0                                ;PM=00
```

```

ldp    #DP_PF1
splk   #0001100001001001b,ADC_CNTL1    ;Início de conversão de vab e vbc
                                           ;vab por ADCIN5, vbc por ADCIN13
conversion1
bit    ADC_CNTL1,8
bcnd   conversion1,tc
lacc   ADC_FIFO1,10
ldp    #4
sach   tmp
lacl   tmp
and    #3ffh
add    #0000h    ;Compensa offset do sensor
sac1   tmp
sub    #512    ;Retira nível DC (2.5V)
sac1   tmp
spm    ;3 PM=11, desloca 6 bits após multiplicação
lt     tmp
mpy    Ktension
pac
sfr
sfr
spm    0
sac1   vab    ;vab em formato 4.12f

ldp    #DP_PF1
lacc   ADC_FIFO2,10
ldp    #4
sach   tmp
lacl   tmp
and    #3ffh
add    #000Bh    ;Compensa offset do sensor
sub    #512    ;Retira nível DC (2.5V)
sac1   tmp
spm    3
lt     tmp
mpy    Ktension
pac
sfr
sfr
sac1   vbc    ;vab em formato 4.12f
spm    0    ;PM=00

```

Transformação de Clarke

(a,b) -> (alfa,beta)

iSalfa = ia

iSbeta = (2 * ib + ia) / sqrt(3)

```

lacc   ia
sac1   iSalfa    ;iSalfa em formato 4.12f
add    ib
neg
sac1   ic
lacc   ib,1    ;iSbeta = (2 * ib + ia) / sqrt(3)
add    ia

```

```

sac1    tmp
lt      tmp
mpy     SQRT3inv          ;SQRT3inv = (1 / sqrt(3)) = 093dh
                               ;formato 4.12f = 0.577350269

pac
sach    iSbeta,4          ;iSbeta em formato 4.12f
lacc    iSbeta
sac1    _i_be,2           ;_i_be em formato 1.15f
lacc    iSalfa
sac1    _i_al,2          ;_i_al em formato 1.15f

```

```

;
; Transformação de Clarke
; (a,b) -> (alfa,beta)
; vSalfa = (2 * vab + vbc) / 3
; vSbeta = vbc / sqrt(3)
;

```

```

lt      vbc
mpy     SQRT3inv          ;SQRT3inv = (1 / sqrt(3)) = 093dh
                               ;formato 4.12f = 0.577350269

pac
sach    vSbeta,4          ;vSbeta em formato 4.12f
lacc    vab,1
add     vbc
sac1    tmp
lt      tmp
mpy     INV3              ;INV3 = (1/3) = 0555h
                               ;4.12 format = 0.333333333

pac
sach    vSalfa,4          ;vSalfa em formato 4.12f

```

```

;
; Medição e controle de velocidade
;

```

```

ldp    #DP_EV
lacc    T3CNT              ;Leitura dos pulsos do encoder
splk    #0000h,T3CNT
ldp    #4
sac1    encincr

```

```

;
; Calculo da velocidade a partir dos pulsos do encoder
;

```

```

lacc    speedstep
sub     #1
sac1    speedstep
bcnd    nocalc,GT
spm     3                  ;3 PM=11, desloca 6 bits após multiplicação
lt      speedtmp
mpy     Kspeed
pac
sfr
sfr

```

```

sac1    n
lacc    #0
sac1    speedtmp
lacc    SPEEDSTEP
sac1    speedstep
spm     0
;PM=00

```

Regulador PI de velocidade

```

lacc    n_ref
sub     nestq4
sac1    epin
lacc    xin,12
lt      epin
mpy     Kpin
apac
sach    upi,4
bit     upi,0
bcnd    upimagzeros,NTC
lacc    Isqrefmin
sub     upi
bcnd    neg_sat,GT
lacc    upi
b       limiters
neg_sat
lacc    Isqrefmin
b       limiters
upimagzeros
lacc    Isqrefmax
sub     upi
bcnd    pos_sat,LT
lacc    upi
b       limiters
pos_sat
lacc    Isqrefmax
limiters
sac1    iSqref
sub     upi
sac1    elpi
lt      elpi
mpy     Kcorn
pac
lt      epin
mpy     Kin
apac
add     xin,12
sach    xin,4
nocalc
lacc    speedtmp
add     encincr
sac1    speedtmp
;epin=n_ref-nestq4, formato 4.12f
;upi=xin+epin*Kpin, formato 4.12f
;If value >0 pular
;satuação negativa
;if upi<ISqrefmin pular para satuação
;satuação positiva
;if upi>ISqrefmax pular para satuação
;elpi=iSqref-upi, formato 4.12f
;xin=xin+epin*Kin+elpi*Kcorn, formato 4.12f

```

Calculo de sinTeta_cm, cosTeta_cm

```

mar    *,ar5
lt     Teta_cm
mpyu   SR8BIT
pac
sach   Index
lacl   Index
and    #0ffh
add    #sintab
sac1   tmp
lar    ar5,tmp
lacl   *
sac1   sinTeta_cm           ;sine Teta_cm em formato 4.12f
lacl   Index               ;cos(teta)=sin(teta+90º)
add    #40h                 ;90º = 40h elementos da tabela de seno
and    #0ffh
add    #sintab
sac1   tmp
lar    ar5,tmp
lacc   *
sac1   cosTeta_cm          ;cosine Teta_cm em formato 4.12f

```

Transformação de Park

(alfa, beta)->(d,q)

$iSd = iSalfa \cdot \cos(Teta_est) + iSbeta \cdot \sin(Teta_est)$

$iSq = -iSalfa \cdot \sin(Teta_est) + iSbeta \cdot \cos(Teta_est)$

```

zac
lt     iSbeta
mpy    sinTeta_est
lta    iSalfa
mpy    cosTeta_est
mpya   sinTeta_est
sach   iSd,4                ;iSd em formato 4.12f
lacc   #0
lt     iSbeta
mpys   cosTeta_est
apac
sach   iSq,4                ;iSq em formato 4.12f

```

Modelo de corrente (utilizado para comparar com o observador)

```

lacc   iSdref
sub    i_mr
sac1   tmp
lt     tmp
mpy    Kr
pac
sach   tmp,4

```

```

    lacc    tmp
    add    i_mr
    sacl   i_mr
    bcnd   i_mrnotzero,NEQ
    lacc   #0
    sacl   tmp
    b      i_mrzero
; i_mr=i_mr+Kr*(iSd-i_mr), formato 4.12 f

i_mrnotzero
*** division (iSq/i_mr)
    lacc   i_mr
    bcnd   i_mrzero,EQ
    sacl   tmp1
    lacc   iSqref
    abs
    sacl   tmp
    lacc   tmp,12
    rpt    #15
    subc   tmp1
    sacl   tmp
    lacc   iSqref
    bcnd   iSqpos,GT
    lacc   tmp
    neg
    sacl   tmp
; tmp=iSq/i_mr, 4.12 format

iSqpos
i_mrzero
*** END division ***
    lt     tmp
    mpy    Kt
    pac
    sach   tmp,4
    lacc   tmp
    add    n
    sacl   fs
    lacc   fs
    abs
    sacl   tmp
    lt     tmp
    mpy    KA
    pac
    sach   tetaincr,4
    bit    fs,0
    bcnd   fs_neg,TC
    lacl   tetaincr
    adds   Teta_cm
    sacl   Teta_cm
    b      fs_pos

fs_neg
    lacl   Teta_cm
    subs   tetaincr
    sacl   Teta_cm
; Teta_cm=Teta_cm+K*fs=Teta_cm+tetaincr
; (0;360)<->(0;65535)

fs_pos

```



```

=====
;
; Regulador PI para a componente isq
; (iSq,iSqref)->(vSqref)
;
=====

    lacc    iSqref
    sub     iSq
    sac1    epiq                                ;epiq=iSqref-iSq, formato 4.12f
    lacc    xiq,12
    lt      epiq
    mpy     Kpi
    apac
    sach    upi,4                                ;upi=xiq+epiq*Kpi, formato 4.12f
    bit     upi,0
    bcnd    upimagzeroq,NTC
    lacc    Vmin
    sub     upi
    bcnd    neg_satq,GT                          ;if upi<Vmin pular para saturação
    lacc    upi
    b       limiterq
neg_satq
    lacc    Vmin                                ;set ACC to neg saturation
    b       limiterq
upimagzeroq
    lacc    Vmax
    sub     upi
    bcnd    pos_satq,LT                          ;if upi>Vmax pular para saturação
    lacc    upi
    b       limiterq
pos_satq
    lacc    Vmax
limiterq
    sac1    vSqref
    sub     upi
    sac1    elpi                                ;elpi=vSqref-upi, formato 4.12f
    lt      elpi
    mpy     Kcor
    pac
    lt      epiq
    mpy     Ki
    apac
    add     xiq,12
    sach    xiq,4                                ;xiq=xiq+epiq*Ki+elpi*Kcor, formato 4.12 f
=====

```

```

=====
;
; Regulador PI para a componente isd
; (iSd,iSdref)->(vSdref)
;
=====

```

```

    lacc    iSdref
    sub     iSd
    sac1    epid                                ;epid=iSdref-iSd, formato 4.12f
    lacc    xid,12
    lt      epid
    mpy     Kpi

```

```

apac
sach    upi,4                                ;upi=xid+epid*Kpi, formato 4.12f
bit     upi,0
bcnd    upimagzerod,NTC
lacc    Vmin
sub     upi
bcnd    neg_satd,GT                          ;if upi<Vmin pular para saturação
lacc    upi
b       limiterd
neg_satd
lacc    Vmin
b       limiterd
upimagzerod
lacc    Vmax
sub     upi
bcnd    pos_satd,LT                          ;if upi>Vmax pular para saturação
lacc    upi
b       limiterd
pos_satd
lacc    Vmax
limiterd
sac1    vSdref
sub     upi
sac1    elpi                                ;elpi=vSdref-upi, formato 4.12f
lt      elpi
mpy     Kcor
pac
lt      epid
mpy     Ki
apac
add     xid,12
sach    xid,4                                ;xid=xid+epid*Ki+elpi*Kcor, formato 4.12 f

```

Transformação inversa de Park

(d,q) -> (alfa,beta)

$vSbe_ref = vSqref * \cos(Teta_est) + vSdref * \sin(Teta_est)$

$vSal_ref = -vSqref * \sin(Teta_est) + vSdref * \cos(Teta_est)$

```

lacc    #0
lt      vSdref
mpy     sinTeta_est
lta     vSqref
mpy     cosTeta_est
mpya    sinTeta_est
sach    vSbe_ref,4                          ;vSbe_ref=vSqref*cosTeta_est+vSdref*sinTeta_est
lacc    #0
lt      vSdref
mpys    cosTeta_est
apac
sach    vSal_ref,4                          ;vSal_ref=vSdref*cosTeta_est-vSqref*sinTeta_est
lacc    vSal_ref
sac1    _u_al,2                             ;_u_al em formato 1.15f

```



```

no4   lacc   sector
      sub   #5
      bcnd  no5,NEQ                ;setor 5
      lacc  X
      sacl  t1                    ;t1=X
      lacc  Y
      neg   sacl  t2                ;t2=-Y
      b     t1t2out
no5   lacc  Y                      ;setor 5
      neg   sacl  t1                ;t1=-Y
      lacc  Z
      neg   sacl  t2                ;t2=-Z
t1t2out
      lacc  t1                      ;if t1+t2>PWMPRD -> saturação de t1 e t2
      add  t2
      sacl  tmp
      sub  PWMPRD
      bcnd  nosaturation,LT,EQ
***   t1 and t2 saturation
      lacc  PWMPRD,15              ;PWMPRD/(t1+t2)
      rpt  #15
      subc tmp
      sacl  tmp
      lt   tmp                      ;Calculo da saturação de t1 e t2
      mpy  t1                      ;t1 =t1*(PWMPRD/(t1+t2))
      pac
      sach  t1,1
      mpy  t2                      ;t2 =t2*(PWMPRD/(t1+t2))
      pac
      sach  t2,1
nosaturation

;Calculo de taon,tbon e tcon

      lacc  PWMPRD
      sub  t1
      sub  t2                      ;taon=(PWMPRD-t1-t2)/2
      sfr
      sacl  taon
      add  t1                      ;tbon=taon+t1
      sacl  tbon
      add  t2                      ;tcon=tbon+t2
      sacl  tcon

;Atualização dos registradores de comparação do DSP

      lacc  sector
      sub  #1
      bcnd  nosect1,NEQ
      bldd  tbon,#CMPR1            ;sector 1
      bldd  taon,#CMPR2
      bldd  tcon,#CMPR3

```

```

    b      inikal
nosect1
    lacl   sector
    sub    #2
    bcnd   nosect2,NEQ
    bldd   taon,#CMPR1           ;sector 2
    bldd   tcon,#CMPR2
    bldd   tbon,#CMPR3
    b      inikal
nosect2
    lacl   sector
    sub    #3
    bcnd   nosect3,NEQ
    bldd   taon,#CMPR1           ;sector 3
    bldd   tbon,#CMPR2
    bldd   tcon,#CMPR3
    b      inikal
nosect3
    lacl   sector
    sub    #4
    bcnd   nosect4,NEQ
    bldd   tcon,#CMPR1           ;sector 4
    bldd   tbon,#CMPR2
    bldd   taon,#CMPR3
    b      inikal
nosect4
    lacl   sector
    sub    #5
    bcnd   nosect5,NEQ
    bldd   tcon,#CMPR1           ;sector 5
    bldd   taon,#CMPR2
    bldd   tbon,#CMPR3
    b      inikal
nosect5
    bldd   tbon,#CMPR1           ;sector 6
    bldd   tcon,#CMPR2
    bldd   taon,#CMPR3

```

Observador de fluxo de rotor

variáveis estimadas iSal_est,iSbe_est,psiRal_est,psiRbe_est
n_est (quando utilizado algoritmo EKF)

```

;atualização da matriz diff_f
;diff_f=
;[1-T/Ts2,0,(xH*T)/(xsig*xR*TR),((xH*wBase*T)/(xsig*xR))*x(5),((xH*wBase*T)/(xsig*xR))*x(4);
;0,1-T/Ts2,-((xH*wBase*T)/(xsig*xR))*x(5),(xH*T)/(xsig*xR*TR),-((xH*wBase*T)/(xsig*xR))*x(3);
; (xH*T)/TR, 0, 1-T/TR,-wBase*T*x(5),-wBase*T*x(4);
; 0, (xH*T)/TR, wBase*T*x(5),1-T/TR, wBase*T*x(3);
; 0, 0, 0, 0, 1];

```

inikal

```

lar    AR2,#n_est           ;Não utilizar para algoritmo EKF
lar    AR2,#_x+4           ;load in AR4 the address of _x+4=x(5)
lar    AR3,#diff_f+3      ;load in AR3 the address of A(k)[1,4]
lar    AR6,#C4             ;load in AR6 the address of C4
mar    *,AR2               ;ARP=2
lt     *-,AR6              ;Treg=x(5), AR2->x(4), ARP=6
mpy    *+,AR3              ;Preg=C4*x(5), AR6->C9, ARP=3
pac                    ;Acc=Preg
sach   *,1                 ;A(k)[1,4]=C4*x(5) in Q15 format
neg                    ;Acc= -Preg
adrk   4                   ;add 4 to the address loaded in AR3 <-> AR3->A(k)[2,3]
sach   *,1,AR6             ; A(k)[2,3]=-C4*x(5) in Q15 format, ARP=6 (N.B. AR6->C9)
mpy    *-,AR3              ;Preg=C9*x(5)=-C8*x(5), ARP=3
pac                    ;Acc=Preg
adrk   6                   ;add 6 to the address loaded in AR3 <-> AR3->A(k)[3,4]
sach   *,1                 ;A(k)[3,4]=C9*x(5)=-C8*x(5) in Q15 format
neg                    ;Acc= -Preg
adrk   4                   ;add 4 to the address loaded in AR3 <-> AR3->A(k)[4,3]
sach   *,1,AR6             ;A(k)[4,3]=C8*x(5) in Q15 format, ARP=2
lt     *-,AR6              ;Treg=x(4), AR2->x(3), ARP=6
mpy    *-,AR3              ;Preg=C9*x(4)=-C8*x(4), AR6->C4, ARP=3
pac                    ;Acc=Preg
sbrk   3                   ;sub 3 to the address loaded in AR3 <-> AR3->A(k)[3,5]
sach   *,1,AR6             ;A(k)[3,5]=-C8*x(4) in Q15 format, ARP=6
mpy    *-,AR3              ;Preg=x(4)*C4, ARP=3
pac                    ;Acc=Preg
sbrk   10                  ;sub 10 to the address loaded in AR3 <-> AR3->A(k)[1,5]
sach   *,1,AR6             ;A(k)[1,5]=C4*x(4) in Q15 format, ARP=2
lt     *                    ;Treg=x(3)
lar    AR6,#C10            ;Load in AR6 the address of C10
mar    *,AR6               ;ARP=6
mpy    *-,AR3              ;Preg=x(3)*C10=-C4*x(3), AR6->C8, ARP=3
pac                    ;Acc=Preg
adrk   5                   ;add 5 to the address loaded in AR3 <-> AR3->A(k)[2,5]
sach   *,1,AR6             ;A(k)[2,5]=C10*x(3)=-C4*x(3) in Q15 format, ARP=6
mpy    *-,AR3              ;Preg=x(3)*C8, ARP=3
pac                    ;Acc=Preg
adrk   10                  ;add 10 to the address loaded in AR3 <-> AR3->A(k)[4,5]
sach   *,1                 ;A(k)[4,5]=C8*x(3) in Q15 format.

```

```

; x_1 = x(k/k-1)
;x_1=[diff_f(1,1)*x(1)+diff_f(1,3)*x(3)+diff_f(1,4)*x(4);
; diff_f(2,2)*x(2)+diff_f(2,3)*x(3)+diff_f(2,4)*x(4);
; diff_f(3,1)*x(1)+diff_f(3,3)*x(3)+diff_f(3,4)*x(4);
; diff_f(4,2)*x(2)+diff_f(4,3)*x(3)+diff_f(4,4)*x(4);
; diff_f(5,5)*x(5)]...
; +[((wBase*T)/xsig)*u(1);((wBase*T)/xsig)*u(2); 0; 0; 0];

```

```

spm    1                   ;set default left shift on Preg for Q15 calculations
setc   ovm
lar    AR4,#_x_1           ;load in AR4 the address of x_1 (=x(k/k-1))
lar    AR2,#_x             ;load in AR2 the address of x (=x(k-1/k-1))
lar    AR3,#diff_f         ;load in AR3 the address of diff_f
lar    AR6,#C5             ;load in AR6 the address of C5

```

```

mar    *,AR2                ;ARP=2
lt     *+                  ;Treg=x(1), AR2->x(2)
mar    *+,AR3              ;AR2->x(3), ARP=3
mpy    *+                  ;Preg=x(1)*A(k)[1,1], AR3-> A(k)[1,2]
mar    *+,AR2              ;AR3-> A(k)[1,3], ARP=2
ltp    *+,AR3              ;Acc=Preg, Treg=x(3), AR2->x(4), ARP=3
mpy    *+,AR2              ;Preg=x(3)*A(k)[1,3], AR3-> A(k)[1,4], ARP=2
lta    *+,AR3              ;Acc=Acc+Preg, Treg=x(4), AR2->x(5), ARP=3
mpy    *+,AR2              ;Preg=x(4)* A(k)[1,4], AR3-> A(k)[1,5], ARP=2
lar    AR2,#u              ;load in AR2 the address of u(1)
lta    *,AR6               ;Acc=Acc+Preg, Treg=u(1), ARP=6
mpy    *,AR4               ;Preg=C5*u(1), ARP=4
apac                   ;Acc=Acc+Preg
sach   *+,AR2              ;store in x_1(1), AR4->x_1(2), ARP=2

lar    AR2,#_x+1           ;load in AR2 the address of x(2)
lar    AR3,#diff_f+6      ;load in AR3 the address of A(k)[2,2]
lt     *+,AR3              ;Treg=x(2), AR2->x(3), ARP=3
mpy    *+,AR2              ;Preg=x(2)*A(k)[2,2], AR3->A(k)[2,3], ARP=2
ltp    *+,AR3              ;Acc=Preg, Treg=x(3), AR2->x(4), ARP=3
mpy    *+,AR2              ;Preg=x(3)*A(k)[2,3], AR3->A(k)[2,4], ARP=2
lta    *+,AR3              ;Acc=Acc+Preg, Treg=x(4), AR2->x(5), ARP=3
mpy    *+,AR2              ;Preg=x(4)*A(k)[2,4], AR3->A(k)[2,5], ARP=2
lar    AR2,#u+1           ;load in AR2 the address of u(2)
lta    *,AR6               ;Acc=Acc+Preg, Treg=u(2), ARP=6
mpy    *,AR4               ;Preg=C5*u(2), ARP=4
apac                   ;Acc=Acc+Preg
sach   *+,AR2              ;store in x_1(2), AR4->x_1(3), ARP=2

lar    AR2,#_x            ;load in AR2 the address of x(1)
lar    AR3,#diff_f+10     ;load in AR3 the address of A(k)[3,1]
mar    *,AR2               ;ARP=2
lt     *+                  ;Treg=x(1), AR2->x(2), ARP=3
mar    *+,AR3              ;AR2->x(3), ARP=3
mpy    *+                  ;Preg=x(1)*A(k)[3,1], AR3-> A(k)[3,2]
mar    *+,AR2              ;AR3->A(k)[3,3], ARP=2
ltp    *+,AR3              ;Acc=Preg, Treg=x(3), AR2->x(4), ARP=3
mpy    *+,AR2              ;Preg=x(3)*A(k)[3,3], AR3->A(k)[3,4], ARP=2
lta    *+,AR3              ;Acc=Acc+Preg, Treg=x(4), AR2->x(5), ARP=3
mpy    *+,AR4              ;Preg=x(4)*A(k)[3,4], AR3->A(k)[3,5], ARP=4
apac                   ;Acc=Acc+Preg
sach   *+,AR2              ;store in x_1(3), AR4->x_1(4), ARP=2

lar    AR2,#_x+1           ;load in AR2 the address of x(2)
lar    AR3,#diff_f+16     ;load in AR3 the address of A(k)[4,2]
lt     *+,AR3              ;Treg=x(2), AR2->x(3), ARP=3
mpy    *+,AR2              ;Preg=x(2)*A(k)[4,2], AR3-> A(k)[4,3], ARP=2
ltp    *+,AR3              ;Acc=Preg, Treg=x(3), AR2->x(4), ARP=3
mpy    *+,AR2              ;Preg=x(3)*A(k)[4,3], AR3->A(k)[4,4], ARP=2
lta    *+,AR3              ;Acc=Acc+Preg, Treg=x(4), AR2->x(5), ARP=3
mpy    *+,AR4              ;Preg=x(4)*A(k)[4,4], AR3->A(k)[4,5], ARP=4
apac                   ;Acc=Acc+Preg
sach   *+,AR2              ;store in x_1(4), AR4->x_1(5), ARP=2

lar    AR2,#_x+4           ;load in AR2 the address of x(5)

```



```

lar    AR3,#diff_f+24           ;load in AR3 the address of A(k)[5,5]
lt     *,AR3                    ;Treg=x(5), ARP=3
mpy    *,AR4                    ;Preg=x(5)*A(k)[5,5]
pac                                         ;Acc=Preg
sach   *,AR2                    ;store in A(k)[5,5]
spm    0
clrc   ovm

```

;Para o caso do observador com base no filtro de Kalman utilizar o bloco a seguir!!!

;P₁= prediction error covariance matrix = P(k/k-1)

;P = filter error covariance matrix = P(k-1/k-1)

;P₁=diff_f*P*diff_f +Q

```

mmtfraG    P, diff_f, RSLT, 5,5,5           ;RSLT=P*diff_f
mmfraG     diff_f, RSLT, P_1, 5,5,5         ;P_1=diff_f*P*diff_f
madiG      P_1,Q,5                           ;P_1= diff_f*P*diff_f+Q

```

;Kalman gain matrix calculation

;K= P₁ * C' * inv(C*P₁*C' +R);

```

mmtfraG    P_1, C, RSLT, 5,5,2             ;RSLT=P_1*C'
mmfraG     C, RSLT, _K, 2,5,2              ;_K is a temp matrix, _K=C*P_1*C'
madiG      _K, R, 2                          ;_K=_K+R
madjG      _K, P                             ;P is a temp matrix, P=adj(_K)
mmfraG     RSLT, P, _K, 5, 2, 2            ;_K=RSLT*P=P_1*C'*adj(C*P_1*C'+R)
detG       P, RSLT                          ;RSLT=det(P)
mdsclG     _K, RSLT, _K, 5, 2, RSLT+1      ;_K=1/det(P)* P_1*C'*adj(C*P_1*C'+R)

```

;state vector estimation

;x₁= state vector prediction = x(k/k-1)

;x = state vector estimation

;x = x₁ + K*(y-C*x₁)

```

msubG      y, Cx_1, P, 2,1                  ;P is a temp matrix, P= y-C*x_1
mmfraG     _K, P, RSLT, 5,2,1              ;RSLT=_K*P=_K*(y-C*x_1)
maddG      _x_1, RSLT, _x, 5,1             ;_x= x_1 + K*(y-C*x_1)

```

;Filter error covariance matrix

;P=P₁-K*C*P₁;

```

mmfraG     _K, C, RSLT, 5, 2, 5            ;RSLT=_K*C
mmfraG     RSLT, P_1, P, 5, 5, 5          ;P=_K*C*P_1
msubG      P_1,P,P,5,5                     ;P=P_1-P

```

;Para o caso do observador de Luenberger utilizar o bloco a seguir!!!

```

zac
ldp    #4
lar    AR2,#_K
lacc   n
abs
sac1   ntest
lt     ntest
mpy    #Klueb
pac
sach   Index
lacc   Index
sfl

```

```

sfl
add      #luenbtabs      ;Acc=Index+luenbtabs table address
sac1     tmp             ;tmp= Index+luenbtabs table address
lar      AR5,tmp         ;load in AR5 the content of tmp
mar      *,AR5
lacc     *+,AR2
sac1     *
adrk     3
sac1     *-,AR5

```

```

;Testar sinal da velocidade
lacc     n
bcnd     velo_posi,GEQ
lacc     *+,AR2
neg
sac1     *-
neg
sac1     *,AR5
b        otro_coef
velo_posi
lacc     *+,AR2
sac1     *-
neg
sac1     *,AR5
otro_coef
lacc     *+,AR2
adrk     3
sac1     *
adrk     3
sac1     *-,AR5

lacc     n
bcnd     velo_posi1,GEQ
lacc     *,AR2
neg
sac1     *-
neg
sac1     *
velo_posi1
lacc     *,AR2
sac1     *-
neg
sac1     *

```

;Para o caso do observador em modos deslizantes utilizar o bloco a seguir!!!

```

zac
ldp     #4
lar      AR2,#_K+4
lacc     n
abs
sac1     ntest
lt       ntest
mpy     #Kslid

```

```

pac
sach Index
lacc Index
sfl
sfl
add #slidtab ;Acc=Index+slidtab table address
sac1 tmp ;tmp= Index+slidtab table address
lar AR5,tmp ;load in AR5 the content of tmp

```

;Testar sinal da velocidade

```

lacc n
bcnd velo_posi,GEQ
mar *,AR5
lacc *+,AR2
sac1 *+,AR5
lacc *+,AR2
neg
sac1 *+,AR5
lacc *+,AR2
neg
sac1 *+,AR5
lacc *,AR2
sac1 *

```

velo_posi

```

mar *,AR5
lacc *+,AR2
sac1 *+,AR5
lacc *+,AR2
sac1 *+,AR5
lacc *+,AR2
sac1 *+,AR5
lacc *,AR2
sac1 *

```

;Vetor de estado estimado (válido para os três observadores)

```

ldp #4
lar AR2,#_x
mar *,AR2
lacc * ;is $\alpha$  estimado em formato 1.15f
sfr
sfr
sac1 iSal_est ;is $\alpha$  estimado em formato 4.12f

lar AR2,#_x+1
lacc * ;is $\beta$  estimado em formato 1.15f
sfr
sfr
sac1 iSbe_est ;is $\beta$  estimado em formato 4.12f

lar AR2,#_x+2
lacc * ; $\lambda$ r $\alpha$  estimado em formato 1.15f
sac1 psiRal_est1
sfr
sfr

```

```

sfr
sac1    psiRal_est           ;λrα estimado em formato 4.12f

lar     AR2,#_x+3
lacc   *                     ;λrβ estimado em formato 1.15f
sac1    psiRbe_est1
sfr
sfr
sfr
sac1    psiRbe_est           ;λrβ estimado em formato 4.12f
lar     AR2,#_x+4
    
```

;Para o caso do observador com base no filtro de Kalman utilizar as 4 instruções a seguir para retornar a velocidade estimada!!!

```

lacc   *                     ;ωr estimada em formato 1.15f
sfr
sfr
sac1    n_est                ;ωr estimada em formato 4.12f
    
```

Calculo da amplitude do vetor de fluxo de rotor λ_r

```

mar     *,AR5                 ;ARP->AR5
spm     2                     ;automatic <<2 for 4.12 multiplications
mpy     #0                    ;Preg is cleared to zero
zac     ;Acc is cleared to zero
sqra    psiRal_est           ;Preg=(psiRal_est)2
pac     ;Acc=(psiRal_est)2 in 4.12 format (spm=2)
sach    tmp                  ;tmp=(psiRal_est)2
sqra    psiRbe_est           ;Preg=(psiRbe_est)2
pac     ;Acc=(psiRbe_est)2 in 4.12 format (spm=2)
sach    tmp1                 ;tmp1=(psiRbe_est)2
lacc    tmp1
add     tmp
sac1    tmp                  ;tmp=(psiRal_est)2 + (psiRbe_est)2 still in 4.12f
spm     0                     ;no automatic shift anymore
lt      tmp                  ;Treg=tmp
mpy     #Ksqrt               ;Preg=tmp*Ksqrt
pac     ;Acc=Preg,
sach    Index                ;store the two interesting digits into Index (->8.8f)
lacc    Index                ;Acc=Index
add     #sqrttab             ;Acc=Index+sqrt table address
sac1    tmp                  ;tmp= Index+sqrt table address
lar     AR5,tmp              ;load in AR5 the content of tmp
lacc   *                     ;Acc= sqrt(psiRal_est2+psiRbe_est2) in 8.8 format
sac1    psiR_est,4           ;psiR_est=sqrt(psiRal_est2+psiRbe_est2)
sac1    psiR_est1,7         ;psiR_est1=sqrt(psiRal_est2+psiRbe_est2)
    
```

```

=====
;
;
; Calculo de cosTeta_est e sinTeta_est
;
=====

    lacc    psiR_est          ;Acc=psiR_est
    bcnd   modnotzero,NEQ
    lacc   #0
    sacl   tmp
    b      modzero

modnotzero
    lacc   psiRal_est1      ;If rotor flux module < 0
                                ;Acc=psiRal_est1 in 1.15 format
    abs
                                ;Acc=1/2psiRal_est1 1/2, unsigned numbers division
    sacl   tmp
                                ;tmp=1/2psiRal_est1 1/2
    lacc   tmp,15
                                ;store 1/2psiRal_est1 1/2 in high side of Acc
    rpt    #15
    subc   psiR_est1        ;AccL=psiRal_est1/psiR_est1, res in 1.15 format
    sfr
    sfr
    sfr
                                ;three right shifts -> division result in 4.12 format
    sacl   cosTeta_est      ;store 4.12 res (=psiRal_est1/psiR_est1) in cosTeta_est
    lacc   psiRal_est1
    bcnd   cospos,GT
    lacc   cosTeta_est
                                ;If psiRal_est1 < 0 then cosTeta_est < 0
    neg

modzero
    sacl   cosTeta_est

cospos
    lacc   psiR_est          ;cosTeta_est=psiRal_est1/psiR_est1, 4.12 format
    bcnd   modnotzero1,NEQ
    lacc   #0
    sacl   tmp
    b      modzero1

modnotzero1
    lacc   psiRbe_est1      ;If rotor flux module < 0
                                ;Acc=psiRbe_est1 b comp of the rotor flux in 1.15 format
    abs
                                ;Acc=1/2psiRbe_est1 1/2, unsigned numbers division
    sacl   tmp
                                ;tmp=1/2psiRbe_est1 1/2
    lacc   tmp,15
                                ;store 1/2psiRbe_est1 1/2 in high side of Acc
    rpt    #15
    subc   psiR_est1        ;AccL=psiRbe_est1/psiR_est1, res in 1.15 format
    sfr
    sfr
    sfr
                                ;three right shifts -> division result in 4.12 format
    sacl   sinTeta_est      ;store 4.12 res (=psiRbe_est1/psiR_est1) in sinTeta_est
    lacc   psiRbe_est1
    bcnd   sinpos,GT
    lacc   sinTeta_est
                                ;If psiRbe_est1 < 0 then sinTeta_est < 0
    neg

modzero1
    sacl   sinTeta_est
                                ;sinTeta_est=psiRbe_est1/psiR_est1, 4.12 format
sinpos

```

Estimação da velocidade para o caso do observador de Luneberger

```

lacc    iSalfa
sub     iSal_est
sac1    eiS_al
lt      psiRbe_est
mpy     eiS_al
pac
sach    eiS_al,4
lacc    iSbeta
sub     iSbe_est
sac1    eiS_be
lt      psiRal_est
mpy     eiS_be
pac
sach    eiS_be,4
lacc    eiS_al
sub     eiS_be
sac1    ep_lb
; Init PI
lacc    xlb,12
lt      ep_lb
mpy     kpmras
apac
sach    uplb,4                ;upi=xid+epid*Kpi, formato 4.12f

bit     uplb,0
bcnd    uplbmagzero,NTC
lacc    nmin
sub     uplb
bcnd    neglb_sat,GT        ;if upi<nmin pular para saturação
lacc    uplb
b       limiterlb
neglb_sat
lacc    nmin
b       limiterlb
uplbmagzero
lacc    nmax
sub     uplb
bcnd    poslb_sat,LT        ;if upi>nmax pular para saturação
lacc    uplb
b       limiterlb
poslb_sat
lacc    nmax
limiterlb
sac1    n_est,2
sub     uplb
sac1    eiplb
lt      elplb
mpy     kcmras
pac
lt      ep_lb
mpy     kimras

```

```

apac
add      xlb,12
sach     xlb,4           ;xid=xid+epid*Ki+elpi*Kcor, formato 4.12 f
lacc     n_est         ;or estimada em formato 1.15f
sfr
sfr
sac1     nestq4        ;or estimada em formato 4.12f

```

Estimação da velocidade para o caso do observador em modos deslizantes

```

lacc     psiRal_est
sac1     copyal
lacc     psiRbe_est
sac1     copybe
lacc     iSal_est
sub      iSalfa
bcnd     seg_coef,GT
lacc     copybe
neg
sac1     copybe
seg_coef
lacc     iSbe_est
sub      iSbeta
bcnd     calc_erro,GT
lacc     copyal
neg
sac1     copyal
calc_erro
lacc     copyal
sub      copybe
sac1     eatual
; Init PI
lt       eatual
mpy      kimras
pac
sach     eatual,4
lacc     nestq4
add      eatual
sac1     nestq4
lacc     nestq4
sub      nmax
bcnd     poslb_sat,GT
lacc     nestq4
sub      nmin
bcnd     neglb_sat,LT
b        finmras
poslb_sat
lacc     nmax
sac1     nestq4
b        finmras
neglb_sat
lacc     nmin
sac1     nestq4

```

finmras

```

lacc    nestq4                ;or estimada em formato 4.12f
sac1    n_est,2              ;or estimada em formato 1.15f
    
```

Enviar variáveis desejadas ao conversor D/A

dacout

```

ldp     #4
lacc    n_ref
sfr
add     #800h
and     #0fffh
sac1    daouttmp
out     daouttmp,DAC0_VAL
    
```

```

lacc    n
sfr
add     #800h
and     #0fffh
sac1    daouttmp
out     daouttmp,DAC1_VAL
out     daouttmp,DAC_VAL
    
```

```

clrc    INTM
ret
    
```

Final da rotina de controle

```

;
;
;      Controle fuzzy de chattering – utilizar para o observador em modos deslizantes
;      fuzzyTECH 4.22 MCU-320 Edition --- Preassembler
;
;      code generation date: Nov 22 18:14:25 2000 --- project: CHATTER
;
;-----
;
;      .mmregs
;      .ds

_lv0_derro      .word 00000H
_lv1_erro      .word 00000H
_lv2_ganho     .word 00000H

;----- RAM fuzzy variables -----
fuzvals        .space 000D0H

;----- ROM -----
;      .ps

;----- term definition -----

tpts:
.word 00000H, 00000H, 00000H, 00008H
.word 00000H, 00008H, 04000H, 00008H
.word 04000H, 00008H, 08000H, 00008H
.word 08000H, 00008H, 0C000H, 0000AH
.word 0BFFFH, 00008H, 0FFFFH, 00000H
.word 00000H, 00000H, 00000H, 00007H
.word 00000H, 00007H, 04939H, 00009H
.word 04CCDH, 0000AH, 08000H, 0000AH
.word 08114H, 0000AH, 0B447H, 00007H
.word 0B333H, 00007H, 0FFFFH, 00000H

;----- xcom table (defuzzification) -----

xcom:
.word 00000H, 08000H, 0FCB7H

;----- fast rule access table -----

frat0:
.word 00002H, 00014H, 00014H, 00014H, 00014H, 00014H

;----- rule table(s) -----

rt0:
.byte 019H, 000H
.byte 001H, 001H, 005H, 00AH
.byte 001H, 001H, 006H, 00AH
.byte 001H, 001H, 007H, 00AH
.byte 001H, 001H, 008H, 00AH
.byte 001H, 001H, 009H, 00AH
.byte 001H, 001H, 005H, 00AH
.byte 001H, 001H, 006H, 00AH
.byte 001H, 001H, 007H, 00AH
.byte 001H, 001H, 008H, 00AH

```

```

.byte 001H, 001H, 009H, 00CH
.byte 001H, 001H, 005H, 00AH
.byte 001H, 001H, 006H, 00AH
.byte 001H, 001H, 007H, 00BH
.byte 001H, 001H, 008H, 00CH
.byte 001H, 001H, 009H, 00CH
.byte 001H, 001H, 005H, 00AH
.byte 001H, 001H, 006H, 00CH
.byte 001H, 001H, 007H, 00CH
.byte 001H, 001H, 008H, 00CH
.byte 001H, 001H, 009H, 00CH
.byte 001H, 001H, 005H, 00CH
.byte 001H, 001H, 006H, 00CH
.byte 001H, 001H, 007H, 00CH
.byte 001H, 001H, 008H, 00CH
.byte 001H, 001H, 009H, 00CH

```

```

;----- fuzzy controller function -----

```

```

_chatter:

```

```

;----- fuzzification -----

```

```

    ldp    #fuzptr
    lacc   #fuzvals
    sacl   fuzptr
    lacc   #tpts
    sacl   tpptr

    ldp    #_lv0_derro
    lacl   _lv0_derro
    ldp    #crisp
    sacl   crisp
    lacl   #04H
    sacl   itcnt
    call   flmss

    ldp    #_lv1_erro
    lacl   _lv1_erro
    ldp    #crisp
    sacl   crisp
    lacl   #04H
    sacl   itcnt
    call   flmss

```

```

;----- inference -----

```

```

    lacc   #rt0
    sacl   rtptr
    lacc   #fuzvals
    sacl   pfuzvals
    sacl   fuzptr
    lacc   #frat0
    sacl   fratptr
    lacl   #4
    sacl   noit
    call   Min

```

```

;----- defuzzification -----
    lacl    #0H
    sacl    _invalidflags
    lacc    #fuzvals
    add     #10
    sacl    fuzptr
    lacc    #xcom
    sacl    xcomptr

    ldp     #_lv2_ganho
    lacc    #08000H
    sacl    _lv2_ganho
    ldp     #defuzz
    lacc    #_lv2_ganho
    sacl    defuzz
    lacl    #02H
    sacl    otcnt
    call    com
    ret

                                ;end of fuzzy controller

_initchatter:
    ldp     #fuzvals
    Lacc    #fuzvals
    add     #0AH
    samm    AR2
    mar     *,AR2
    lacl    #0H
    rpt     02H
    sacl    *+
    ret

                                ;initialisation

    .copy  pubdec.asm
    .copy  flmss.asm
    .copy  min.asm
    .copy  com.asm

;data size knowledge base (bytes):
;RAM:    32 00020H
;ROM:    200 000C8H
;TOTAL:  232 000E8H
;

```

Tabela de seno com 256 elementos $[0;2\pi]$
Formato numérico: 4.12f Arquivo: sine.tab

.word 0,101,201,301,401,501,601,700,799,897
.word 995,1092,1189,1285,1380,1474,1567,1660,1751,1842
.word 1931,2019,2106,2191,2276,2359,2440,2520,2598,2675
.word 2751,2824,2896,2967,3035,3102,3166,3229,3290,3349
.word 3406,3461,3513,3564,3612,3659,3703,3745,3784,3822
.word 3857,3889,3920,3948,3973,3996,4017,4036,4052,4065
.word 4076,4085,4091,4095,4096,4095,4091,4085,4076,4065
.word 4052,4036,4017,3996,3973,3948,3920,3889,3857,3822
.word 3784,3745,3703,3659,3612,3564,3513,3461,3406,3349
.word 3290,3229,3166,3102,3035,2967,2896,2824,2751,2675
.word 2598,2520,2440,2359,2276,2191,2106,2019,1931,1842
.word 1751,1660,1567,1474,1380,1285,1189,1092,995,897
.word 799,700,601,501,401,301,201,101,0,65435
.word 65335,65235,65135,65035,64935,64836,64737,64639,64541,64444
.word 64347,64251,64156,64062,63969,63876,63785,63694,63605,63517
.word 63430,63345,63260,63177,63096,63016,62938,62861,62785,62712
.word 62640,62569,62501,62434,62370,62307,62246,62187,62130,62075
.word 62023,61972,61924,61877,61833,61791,61752,61714,61679,61647
.word 61616,61588,61563,61540,61519,61500,61484,61471,61460,61451
.word 61445,61441,61440,61441,61445,61451,61460,61471,61484,61500
.word 61519,61540,61563,61588,61616,61647,61679,61714,61752,61791
.word 61833,61877,61924,61972,62023,62075,62130,62187,62246,62307
.word 62370,62434,62501,62569,62640,62712,62785,62861,62938,63016
.word 63096,63177,63260,63345,63430,63517,63605,63694,63785,63876
.word 63969,64062,64156,64251,64347,64444,64541,64639,64737,64836
.word 64935,65035,65135,65235,65335,65435

Tabela de raiz quadrada com 256 elementos $[0;1.5]$
Formato numérico: 8.8f Arquivo: sqrt.tab

.word 0,20,28,34,39,44,48,52,55,59,62,65,68,71,73,76
.word 78,81,83,85,88,90,92,94,96,98,100,102,104,106,107,109
.word 111,113,114,116,118,119,121,122,124,125,127,128,130,131,133,134
.word 136,137,139,140,141,143,144,145,147,148,149,151,152,153,154,156
.word 157,158,159,160,162,163,164,165,166,167,169,170,171,172,173,174
.word 175,176,177,179,180,181,182,183,184,185,186,187,188,189,190,191
.word 192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,206
.word 207,208,209,210,211,212,213,214,215,216,216,217,218,219,220,221
.word 222,223,223,224,225,226,227,228,229,229,230,231,232,233,234,234
.word 235,236,237,238,238,239,240,241,242,242,243,244,245,246,246,247
.word 248,249,249,250,251,252,252,253,254,255,255,256,257,258,258,259
.word 260,261,261,262,263,264,264,265,266,267,267,268,269,269,270,271
.word 272,272,273,274,274,275,276,276,277,278,279,279,280,281,281,282
.word 283,283,284,285,285,286,287,287,288,289,289,290,291,291,292,293
.word 293,294,295,295,296,297,297,298,298,299,300,300,301,302,302,303
.word 304,304,305,305,306,307,307,308,309,309,310,310,311,312,312,313

```

=====
;
;   Arquivo de comando do “Linker” (link.cmd)
;
=====

-C
-stack 100h
/*----- */
/*MEMORY SPECIFICATION */
/*Block B0 is configured as data memory (CNFD) */
/*and MP/MC=1 */
/*(microprocessor mode). Note that data memory */
/*locations 6h--5Fh */
/*and 80h--1FFh are not configured. */
/*----- */

MEMORY
{
    PAGE 0:
    FLASH_VEC    : origin = 0h,    length = 40h
    FLASH        : origin = 40h,    length = 3FC0h
    PAGE 1:
    REGS         : origin = 0h,    length = 60h
    BLK_B22      : origin = 60h,    length = 20h
    BLK_B0       : origin = 200h,   length = 100h
    BLK_B1       : origin = 300h,   length = 100h
    EXT_DATA     : origin = 8000h,  length = 1000h
}

/*----- */
/* SECTIONS ALLOCATION*/
/*----- */

SECTIONS
{
    vectors :    {} >    FLASH_VEC PAGE 0
    .text   :    {} >    FLASH PAGE 0
    .cinit  :    {} >    FLASH PAGE 0
    .switch :    {} >    FLASH PAGE 0
    blockb2 :    {} >    BLK_B22
    .bss    :    {} >    BLK_B0 PAGE 1
    .data   :    {} >    BLK_B0 PAGE 1
    table   :    {} >    BLK_B1 PAGE 1
    .sysmem :    {} >    EXT_DATA PAGE 1
    .const  :    {} >    EXT_DATA PAGE 1
    .stack  :    {} >    EXT_DATA PAGE 1
    tabsqrt :    {} >    EXT_DATA PAGE 1
    tabluenb : {} >    EXT_DATA PAGE 1
    tabslid :    {} >    EXT_DATA PAGE 1
}

```

Obs.: As seções de memória “tabluenb” e “tabslid” são utilizadas para a alocação das tabelas com a matriz de ganho dos observadores de Luenberger e em modos deslizantes, respectivamente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] F. Blaschke, "The principle of field orientation as applied to the new transvector closed loop control system for rotating field machines," *Siemens Rev.*, no. 39, pp. 217-220, 1972.
- [2] J. Holtz, "Speed estimation and sensorless control of AC drives," *Proc. of IEEE Industrial Electronics Society Conference (IECON)*, pp. 649-654, 1993.
- [3] C. Ilas, A. Bettini, L. Feraris, G. Griva, F. Profumo, "Comparison of different schemes without shaft sensors for field oriented control drives," *Proc. of IEEE Industrial Electronics Society Conference (IECON)*, pp. 1579-1588, 1994.
- [4] P. Vas, *Sensorless vector and direct torque control*, Oxford University Press, 1998.
- [5] R. B. Giménez, G. M. Asher, M. Sumner, K. J. Bradley, "Dynamic performance limitations for mras based sensorless induction motor drives. part I: Stability analysis for the closed loop drive," *IEE Proc. Electr. Power Appl.*, 143(2), pp. 113-122, March 1996.
- [6] R. B. Giménez, G. M. Asher, M. Sumner, K. J. Bradley, "Dynamic performance limitations for mras based sensorless induction motor drives. part II: Online parameter tuning and dynamic performance studies," *IEE Proc. Electr. Power Appl.*, 143(2), pp. 123-134, March 1996.
- [7] C. Shauder, "Adaptive speed identification for vector control of induction motors without rotational transducers," *IEEE Transactions on Industry Applications*, vol. 28, no. 5, pp. 1054-1061, September/October 1992.
- [8] H. Tajima, Y. Hori, "Speed sensorless field orientation control of the induction machine," *Proc. of IEEE Industry Applications Society Annual Meeting*, pp. 385-391, 1991.
- [9] A. Bellini, G. Figalli, "A bilinear observer of the state of induction machine," *Ricerche di Automatica*, vol. 19, no. 1, pp. 70-85, June 1978.
- [10] A. Bellini, G. Figalli, G. Ulivi, "Realization of a bilinear observer of the induction machine," *Proc. of Int. Conf. Electrical Variable Speed Drives (London)*, pp. 175-178, 1979.

-
- [11] S. R. Sanders, "State Estimation in Induction Machines," S. M. thesis, EECS Dept., M.I.T., Cambridge, MA, June 1985.
- [12] G. C. Verghese, S. R. Sanders, "Observers for flux estimation in induction machines," *IEEE Transactions on Industrial Electronics*, vol. 35, no. 1, pp. 85-94, February 1988.
- [13] A. Bellini, G. Figalli, "Analysis and design of a microcomputer-based observer for an induction machine," *Automatica*, vol. 24, no. 4, pp. 549-555, 1988.
- [14] H. Kubota, K. Matsuse, T. Nakano, "New adaptive flux observer of induction motor for wide speed range motor drives," *Proc. of IEEE Industrial Electronics Society Conference (IECON)*, pp. 921-926, 1990.
- [15] G. Yang, T. H. Chin, "Adaptive-speed identification scheme for a vector-controlled speed sensorless inverter-induction motor drive," *IEEE Transactions on Industry Applications*, vol. 29, no. 4, pp. 820-825, Jul/Aug 1993.
- [16] G. C. D. Sousa, A. P. S. Rabello, "Application of the Luenberger observer to the sensorless speed control of an induction motor drive: analysis and implementation," *Proc. of Brazilian Power Electronics Conference IEEE Industrial Electronics Society Conference (COBEP)*, 1999.481, 1999.
- [17] H. Kubota, K. Matsuse, T. Nakano, "DSP-based speed adaptive flux observer of induction motor," *IEEE Transactions on Industry Applications*, vol. 29, no. 2, pp. 344-348, March/April 1993.
- [18] H. Kubota, K. Matsuse, "Speed sensorless field-oriented control of induction motor with rotor resistance adaptation," *IEEE Transactions on Industry Applications*, vol. 30, no. 5, pp. 1219-1224, September/October 1994.
- [19] C. P. Bottura, J. L. Silvino, P. de Resende, "A flux observer for induction machines based on a time-variant discrete model," *IEEE Transactions on Industry Applications*, vol. 29, no. 2, pp. 349-354, March/April 1993.
- [20] F. Profumo, M. Abrate, G. Griva, A. Tenconi, "High speed sensorless fuzzy-like Luenberger observer," *Proc. of Power Electronics Specialist Conference (PESC)*, pp. 477-481, 1999.

- [21] F. Profumo, G. Griva, A. Tenconi, M. Abrate, L. Ferraris, "Stability analysis of Luenberger observers for speed sensorless high performance spindle drives," *Proc. of European Conference on Power Electronics (EPE)*, 1999.
- [22] J. Maes, J. Melkebeek, "Adaptive flux observer for sensorless induction motor drives with enhanced dynamic performance," *Proc. of European Conference on Power Electronics (EPE)*, 1999.
- [23] J. Maes, J. Melkebeek, "Improved adaptive flux observer for wide speed range sensorless induction motor drives," *Proc. of Electromotion Conference*, 1999.
- [24] H. Hoffman, S. Sanders, "Speed-sensorless vector torque control of induction machines using a two-time-scale approach," *IEEE Transactions on Industry Applications*, vol. 34, no. 1, pp. 169-177, Jan./Feb. 1998.
- [25] G. Henneberger, B. J. Brunsbach, Th. Klepsch, "Field oriented control of synchronous and asynchronous drives without mechanical sensors using a Kalman filter," *Proc. of European Conference on Power Electronics (EPE)*, pp. 664-671, 1991.
- [26] Y. R. Kim, S. K. Sul, M. H. Park, "Speed sensorless vector control of induction motor using extended Kalman filter," *IEEE Transactions on Industry Applications*, vol. 30, no. 5, pp. 1225-1233, September/October 1994.
- [27] Th. Von Raumer, J. M. Dion, L. Dugard, J. L. Thomas, "Applied nonlinear control of an induction motor using digital signal processing," *IEEE Transactions on Control Systems Technology*, vol. 2, no 4, pp. 327-335, December 1994.
- [28] C. M. Lee, C. L. Chen, "Observer-based speed estimation method for sensorless vector control of induction motors," *IEE Proc. on Control Theory and Applications*, vol. 145, no 3, pp. 359-363, May 1998.
- [29] H. Hilairer, F. Auger, C. Darengosse, "Two efficient Kalman filters for flux and velocity estimation of induction motors," *Proc. of Power Electronics Specialist Conference (PESC)*, 2000.
- [30] S. Sangwongwanich, S. Doki, T. Yonemoto, S. Okuma, "Adaptive sliding observers for direct field-oriented control of induction motor," *Proc. of International Power Electronics Conference (IPEC)*, pp. 1235-1242, 1990.

- [31] S. Sangwongwanich, T. Yonemoto, S. Okuma, "Time domain characteristics of sliding observer for flux estimation of induction motor and its implementation," *Proc. of Power Electronics Specialist Conference (PESC)*, pp. 592-595, 1991.
- [32] S. Doki, S. Sangwongwanich, S. Okuma, "Implementation of speed sensorless field oriented vector control using adaptive sliding observer," *Proc. of IEEE Industrial Electronics Society Conference (IECON)*, pp. 453-458, 1992.
- [33] A-S. A. Luiz, M. M. Stopa, W. L. F. Lopes, S. R. Silva, B. R. Menezes, "A discrete sliding observer for an induction motor drive," *Proc. of Brazilian Power Electronics Conference IEEE Industrial Electronics Society Conference (COBEP)*, 1999.
- [34] A. Benchaib, A. Rachid, and E. Audrezet, "Sliding Mode Input-Output Linearization and Field Orientation for Real-Time Control of Induction Motors," *IEEE Transactions on Power Electronics*, vol. 14, no 1, pp. 3-13, January 1999.
- [35] V. I. Utkin, "Variable structure systems with sliding modes," *IEEE Transactions on Automatic Control*, vol. AC-22, no 2, pp. 212-222, April 1977.
- [36] V. I. Utkin, "Sliding mode control design principles and applications to electric drives," *IEEE Trans. Ind. Electron.*, vol. 40, no 1, pp. 23-36, 1993.
- [37] U. Itkis, *Control Systems of Variable Structure*, New York: Wiley, 1976.
- [38] D. E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, New York, USA, Addison-Wesley, January 1989.
- [39] P. Pillay, R. Nolan, T. Haque, "Application of Genetic Algorithms to Motor Parameter Determination for Transient Torque Calculations", *IEEE Transactions on Industry Applications*, vol. 33, no 5, pp. 1273-1282, Sep./Oct. 1997.
- [40] P. Pillay, V. Levin, P. Otaduy, J. Kueck, "In - Situ induction Motor Efficiency Determination Using the Genetic Algorithm", *IEEE Transactions on Energy Conversion*, vol. 13, no 4, pp. 326-333, Dec. 1998.
- [41] E. B. Souza, A M. N. Lima, C. B. Jacobina, "Characterization of Induction Machines with a Genetic Algorithm", Conf. Rec. IEEE-IEMDC'99, Seattle, USA, May 1999, pp. 446-448.

- [42] L. Rosell, E. Bim, "A Genetic Algorithms Approach for Adaptive Field Oriented Control of Induction Motor Drives ", Conf. Rec. IEEE- IEMDC'99, Seattle, USA, May 1999, pp. 643-645.
- [43] A. F. Filippov, "Application of theory of differential equations with discontinuous righthand side to nonlinear control problems", in *Proc. 1st IFAC World Congress*, vol. 3, Izd. ANSSSR, 1960.
- [44] V. A. Utkin, "Equations of sliding mode in discontinuous systems, vols. 1,2", *Automation and Remote Control*, vol. 44, no 12, pp. 1897-1907, 1971, no 2, pp. 211-219, 1972.
- [45] J. Y. Hung, W. Gao, J. C. Hung, "Variable Structure Control: A Survey," *IEEE Transactions on Industrial Electronics*, vol. 40, no. 1, pp. 2-22, February 1993.
- [46] V. I. Utkin, *Sliding Modes in Control Optimization*, Springer-Verlag Press, 1992.
- [47] Z. M. A. Peixoto, "Desenvolvimento de observadores a modos deslizantes aplicados à estimação de posição, velocidade e força contra eletromotriz em máquinas síncronas a ímãs permanentes", Dissertação de Mestrado, PPGEE/UFMG, Julho/95.
- [48] E. Ho et al., "Control Dynamics of Speed Drive Systems Using Sliding Mode Controllers with Integral Compensation", *IEEE Transactions on Industry Applications*, vol. 27, no 5, pp. 883-892, 1991.
- [49] A. Kawamura et al., "Chattering Reduction of Disturbance Observer Based Sliding Mode Control", *IEEE Transactions on Industry Applications*, vol. 30, no 2, pp. 456-461, 1994.
- [50] P. Kackroo, M. Tomizuka, "Chattering reduction and error convergence in the sliding-mode control of a class of nonlinear systems", *IEEE Transactions on Automatic Control*, vol. 41, no 7, pp. 1063-1068, July 1996.
- [51] M. Tursini, R. Petrella, F. Parasiliti, "Adaptive Sliding-Mode Observer for Speed-Sensorless Control of Induction Motors", *IEEE Transactions on Industry Applications*, vol. 36, no 5, pp. 1380-1387, Sep./Oct. 2000.
- [52] R. Palm, "Robust Control by fuzzy sliding mode", *Automatica*, vol. 30, no 9, pp. 1429-1437, Sep. 1994.

-
- [53] M. De Neyer, R. Gorez, "Use of fuzzy concepts in adaptive sliding mode control", *Proc. of IEEE Conference on Systems, Man, and Cybernetics*, vol. 1, pp. 443-447, San Antonio, USA, 1994.
- [54] J. Holtz, "Pulsewidth Modulation for Electronic Power Conversion", *Proceedings of the IEE*, Vol. 82, Nº 8, August 1994.
- [55] Heinz W., Van. D. B.e Hans S. M, "Analisis and Realization of a Pulse width Modulator Based on Voltage Space Vectors", *IEEE Transactions on Industry Applications*, Vol. 24, pp. 142-150, January/February 1988.
- [56] Texas Instruments, "Implementation of a Speed Field Orientated Control of Three Phase AC Induction Motor using TMS320F240", Literature number: BPRA076, March 1998.
- [57] K. Ogata, *Engenharia de Controle Moderno*, Prentice Hall do Brasil, 1993.
- [58] R. Isermann, *Digital Control Systems*, Springer-Verlag Press, 1989.