

Universidade Estadual de Campinas  
Faculdade de Engenharia Elétrica e de Computação

**Controlador Nebuloso de Arquitetura Simplificada  
Estudo de Caso:  
Acionamento de Motor de Indução Trifásico**

**Autor: Mário Eduardo Bordon**

**Orientador: Oséas Valente de Avilez Filho**

**Tese de Doutorado** apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: **Eletrônica, Microeletrônica e Optoeletrônica.**

Banca Examinadora

Edwin Avolio, Dr. . . . . . FEB/UNESP  
Elnatan Chagas Ferreira, Dr. . . . . . FEEC/UNICAMP  
Ivan Nunes da Silva, Dr. . . . . . EESC/USP  
José Carlos Rossi, Dr. . . . . . FEIS/UNESP  
Oséas Valente de Avilez Filho, Dr. . . . . . FEEC/UNICAMP

Campinas, SP  
Outubro de 2004

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

B644c Bordon, Mário Eduardo  
Controlador nebuloso de arquitetura simplificada estudo de caso: acionamento de motor de indução trifásico / Mário Eduardo Bordon. --Campinas, SP: [s.n.], 2004.

Orientador: Oséas Valente de Avilez Filho.  
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Controle digital. 2. Controle em tempo real. 3. Lógica difusa. 4. Redes neurais (Computação). 5. Motores elétricos de indução. I. Avilez Filho, Oséas Valente de. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

*Dedico este trabalho à Comunidade Jesus Maria José e minha Família, em especial à Lucia, Filipe e Isabela cujas orações, alegria e amor incondicional, têm sido decisivos em todos os momentos de minha vida.*



*Este trabalho não teria sido possível sem a  
ajuda e contribuição de várias pessoas:*

*Oséas Valente de Avilez Filho*

*Edwin Avolio e Ivo Reis Fontes*

*Maurício José Bordon*

*Alceu Ferreira Alves*

*Todos os Amigos do*

*Dee-Feb-Unesp*



## *Resumo*

O objetivo deste trabalho foi o desenvolvimento de um controlador digital simples e eficiente, utilizando conceitos de lógica nebulosa, aplicado no acionamento de um motor de indução trifásico. Trata-se de um trabalho de engenharia aplicada, que apresenta o projeto de um controlador nebuloso de arquitetura simplificada, que emprega uma estrutura padronizada para representação das funções de pertinência e permite efetuar a ponderação dos termos lingüísticos. Para avaliar o desempenho deste controlador, foi implementado um sistema para acionamento do motor de indução, com frequência de operação controlada e limitação de corrente, capaz de gerar em tempo real, um perfil de frequência adequado, sempre que um novo valor para frequência de operação seja estabelecido. No projeto deste controlador, optou-se pela utilização de um sistema digital de baixo custo, baseado em microcontroladores de 8 bits, mesmo assim, os resultados obtidos superaram as expectativas, comprovando a viabilidade operacional deste controlador nebuloso de arquitetura simplificada, para acionamento do motor de indução, evitando que a corrente ultrapasse o limite estipulado e impedindo o desligamento desnecessário do sistema.

***Palavras-Chave:*** Controle Digital, Lógica Nebulosa, Motor de Indução e Redes Neurais.

## *Abstract*

This work presents a new approach of a simple and efficient digital controller, which uses the fuzzy logic concepts to drive a three-phase induction motor in a practical engineering application. In order to represent the membership functions and carry out the weighting up of linguistic terms a custom structure was used in the fuzzy logic controller. The performance evaluation of the digital controller was verified using a new strategy of induction motor drive for speed control and input current restriction. This fuzzy logic controller with simplified architecture applied in a real time speed control assures the current restriction and the normal operation of the induction motor. The control strategy operates in closed loop and has a dynamic gain adjustment that provides the soft start requirements. The closed loop frequency control allows to adjust the desired value of the rotation frequency, while the closed loop current control assures that current doesn't cross the limit stipulated for motor drive. The versatility of this digital controller based on low cost 8 bits microcontrollers can be verified from the results presented. The accuracy and time response could be compared to the other conventional systems using scalar or vector techniques.

***Key-Words:*** Digital Control, Fuzzy Logic, Induction Motors and Neural Networks.



# Sumário

<b>Sumário .....</b>	<b>ix</b>
<b>Lista de Figuras .....</b>	<b>xiii</b>
<b>Lista de Tabelas .....</b>	<b>xvii</b>
<b>Trabalhos Publicados .....</b>	<b>xix</b>
<b>Capítulo 1 .....</b>	<b>1</b>
<b>Contexto e Motivação .....</b>	<b>1</b>
1.1 Introdução .....	1
1.2 Objetivo do Trabalho .....	5
1.3 Apresentação .....	7
1.3.1 Descrição dos Capítulos .....	7
1.3.2 Descrição dos Apêndices .....	8
<b>Capítulo 2 .....</b>	<b>9</b>
<b>Controlador Nebuloso de Arquitetura Simplificada .....</b>	<b>9</b>
2.1 Introdução .....	9
2.2 Variáveis de Entrada .....	10
2.3 Variável de Saída .....	12
2.4 Procedimentos de Fuzzificação .....	15
2.4.1 Fuzzificação com Inferência Direta .....	16
2.4.2 Fuzzificação com Inferência Indireta .....	17
2.5 Procedimentos de Defuzzificação .....	18
2.5.1 Defuzzificação com Inferência Direta .....	19
2.5.2 Defuzzificação com Inferência Indireta .....	22
2.6 Simulação Computacional .....	25
2.6.1 Resultados de Simulação Computacional .....	26
2.6.2 Controle Nebuloso com Inferência Direta .....	27
2.6.3 Controle Nebuloso com Inferência Indireta .....	31
2.7 Controlador Neuro-Nebuloso .....	35
2.7.1 Operador de Agregação .....	36
2.7.2 Controle Direto com Função Quadrática .....	38
2.7.3 Controle Direto com Função Exponencial .....	40
2.7.4 Controle Indireto com Função Quadrática .....	42
2.7.5 Controle Indireto com Função Exponencial .....	44
<b>Capítulo 3 .....</b>	<b>47</b>
<b>Acionamento de Máquinas Elétricas de Indução .....</b>	<b>47</b>
3.1 Introdução .....	47
3.2 Motores de Indução Trifásicos .....	48
3.3 Descrição do Sistema de Controle .....	50
3.3.1 Unidade de Controle .....	51
3.3.2 Circuito Gerador de Pulsos .....	51
3.3.3 Inversor de Tensão Trifásico .....	52
3.3.4 Motor de Indução Trifásico .....	52
3.4 Estratégias de Controle .....	53
3.4.1 Controle Proporcional Integral Derivativo .....	54
3.4.2 Controle Nebuloso com Inferência Direta .....	56

---

3.4.3 Controle Nebuloso com Inferência Indireta .....	59
3.4.4 Estratégia de Controle Neuro-Nebulosa .....	62
3.5 Simulação Computacional.....	66
3.5.1 Simulação do Motor de Indução.....	67
3.5.2 Simulação do Motor de Indução com Inversor .....	69
3.5.3 Simulação do Sensor de Corrente.....	71
3.5.4 Simulação do Sensor de Corrente com Inversor .....	72
3.5.5 Motor de Indução com Perfil de Frequência Senoidal .....	73
3.5.6 Sensor de Corrente com Perfil de Frequência Senoidal .....	74
<b>Capítulo 4.....</b>	<b>75</b>
<b>Implementação do Sistema de Controle.....</b>	<b>75</b>
4.1 Introdução.....	75
4.2 Descrição do Sistema de Controle .....	76
4.2.1 Inversor de Tensão .....	76
4.2.2 Motor de Indução .....	76
4.2.3 Gerador Síncrono.....	76
4.2.4 Sensor de Corrente.....	77
4.2.5 Torquímetro Rotativo .....	77
4.2.6 Tacômetro Digital.....	77
4.3 Circuitos Condicionadores de Sinais.....	77
4.3.1 Sinal de Controle .....	78
4.3.2 Sinal de Conjugado.....	78
4.3.3 Sinal de Corrente .....	79
4.3.4 Sinal de Frequência .....	79
4.4 Sistema de Aquisição de Dados .....	80
4.4.1 Placa de Aquisição de Dados .....	80
4.4.2 Sinais de Entrada e Saída.....	81
4.5 Apresentação de Resultados.....	82
4.5.1 Motor de Indução e Gerador Síncrono .....	82
4.5.2 Controle Proporcional Integral Derivativo .....	87
4.5.3 Controle Nebuloso com Inferência Direta.....	96
4.5.4 Controle Nebuloso com Inferência Indireta .....	105
4.5.5 Estratégia de Controle Neuro-Nebulosa.....	114
<b>Capítulo 5.....</b>	<b>123</b>
<b>Análise de Resultados e Conclusão.....</b>	<b>123</b>
5.1 Introdução.....	123
5.2 Análise de Resultados .....	125
5.2.1 Tempo de Resposta do Motor de Indução.....	125
5.2.2 Comportamento do Sistema de Controle.....	127
5.3 Conclusão .....	130
5.3.1 Estratégias de Controle.....	131
5.3.2 Trabalhos Futuros .....	132
<b>Apêndice 1.....</b>	<b>133</b>
<b>Lógica Nebulosa .....</b>	<b>133</b>
1.1 Introdução.....	133
1.2 Fundamentos da Lógica Nebulosa .....	134
1.3 Descrição de Sistemas e Processos .....	141

---

---

1.4 Sistemas de Controle Nebulosos.....	147
1.4.1 Interface de Fuzzificação .....	148
1.4.2 Base de Conhecimento.....	148
1.4.3 Base de Dados.....	148
1.4.4 Procedimento de Inferência .....	149
1.4.5 Interface de Defuzzificação .....	149
1.4.6 Operador de Agregação .....	149
1.5 Projeto de Controladores Nebulosos.....	149
<b>Apêndice 2.....</b>	<b>151</b>
<b>Redes Neurais Artificiais.....</b>	<b>151</b>
2.1 Introdução .....	151
2.2 Elemento Linear Adaptativo.....	151
2.3 Cerebellar Model Articulation Controller.....	154
2.4 Transformações Presentes no CMAC.....	155
2.4.1 Transformação $S \rightarrow M$ .....	155
2.4.2 Transformação $M \rightarrow A$ .....	157
2.4.3 Transformação $A \rightarrow Y^c$ .....	158
2.5 Treinamento do CMAC .....	158
2.6 Requisitos de Memória .....	159
2.7 CMAC Alternativo .....	160
2.8 CMAC Pesos Discretos .....	162
<b>Apêndice 3.....</b>	<b>165</b>
<b>Máquinas Elétricas de Indução.....</b>	<b>165</b>
3.1 Introdução .....	165
3.2 Motores de Indução Trifásicos .....	165
3.3 Modelo Matemático do Motor de Indução .....	166
3.4 Conversor de Frequência .....	169
3.5 Modelo Matemático do Conversor .....	169
<b>Referências Bibliográficas .....</b>	<b>173</b>
<b>Bibliografia Complementar.....</b>	<b>177</b>

---



## Lista de Figuras

<b>Sumário .....</b>	<b>ix</b>
<b>Lista de Figuras .....</b>	<b>xiii</b>
<b>Lista de Tabelas .....</b>	<b>xvii</b>
<b>Trabalhos Publicados .....</b>	<b>xix</b>
<b>Capítulo 1 .....</b>	<b>1</b>
<b>Contexto e Motivação .....</b>	<b>1</b>
Figura 1.1 Variável Lingüística Temperatura.....	2
<b>Capítulo 2 .....</b>	<b>9</b>
<b>Controlador Nebuloso de Arquitetura Simplificada.....</b>	<b>9</b>
Figura 2.1 Funções de Pertinência da Entrada (x) .....	11
Figura 2.2 Funções de Pertinência da Entrada (y) .....	11
Figura 2.3 Pesos Associados à Variável de Saída .....	13
Figura 2.4 Funções de Pertinência da Variável de Saída.....	13
Figura 2.5 Controle com Inferência Direta (1º Caso).....	15
Figura 2.6 Controle com Inferência Indireta (2º Caso) .....	15
Figura 2.7 Gráfico dos Pesos com Função Linear (1º Caso).....	20
Figura 2.8 Variáveis Cartesianas com Função Linear (1º Caso) .....	21
Figura 2.9 Ações de Controle com Função Linear (1º Caso) .....	21
Figura 2.10 Gráfico dos Pesos com Função Linear (2º Caso).....	23
Figura 2.11 Variáveis Cartesianas com Função Linear (2º Caso) .....	24
Figura 2.12 Ações de Controle com Função Linear (2º Caso) .....	24
Figura 2.13 Resposta do Sistema de Segunda Ordem h(t) .....	26
Figura 2.14 Controle com Inferência Direta (1º Caso).....	27
Figura 2.15 Resposta do Sistema (1º Caso, $K_p=1.0$ ).....	28
Figura 2.16 Ações de Controle (1º Caso, $K_p=1.0$ ) .....	28
Figura 2.17 Resposta do Sistema (1º Caso, $K_p=2.0$ ).....	29
Figura 2.18 Ações de Controle (1º Caso, $K_p=2.0$ ) .....	29
Figura 2.19 Resposta do Sistema (1º Caso, $K_p=3.0$ ).....	30
Figura 2.20 Ações de Controle (1º Caso, $K_p=3.0$ ) .....	30
Figura 2.21 Controle com Inferência Indireta (2º Caso) .....	31
Figura 2.22 Resposta do Sistema (2º Caso, $K_p=0.5$ ).....	32
Figura 2.23 Ações de Controle (2º Caso, $K_p=0.5$ ) .....	32
Figura 2.24 Resposta do Sistema (2º Caso, $K_p=1.0$ ).....	33
Figura 2.25 Ações de Controle (2º Caso, $K_p=1.0$ ) .....	33
Figura 2.26 Resposta do Sistema (2º Caso, $K_p=1.5$ ).....	34
Figura 2.27 Ações de Controle (2º Caso, $K_p=1.5$ ) .....	34
Figura 2.28 Procedimento de Treinamento.....	36
Figura 2.29 Procedimento de Estimação .....	36
Figura 2.30 Memória do Controlador Neuro-Nebuloso .....	37
Figura 2.31 Arquitetura do Controlador Neuro-Nebuloso.....	37
Figura 2.32 Pesos com Função Quadrática (1º Caso).....	38
Figura 2.33 Rede Neural com Função Quadrática (1º Caso).....	39
Figura 2.34 Ações de Controle com Função Quadrática (1º Caso) .....	39
Figura 2.35 Pesos com Função Exponencial (1º Caso) .....	40

---

Figura 2.36 Rede Neural com Função Exponencial (1º Caso).....	41
Figura 2.37 Ações de Controle com Função Exponencial (1º Caso).....	41
Figura 2.38 Pesos com Função Quadrática (2º Caso).....	42
Figura 2.39 Rede Neural com Função Quadrática (2º Caso).....	43
Figura 2.40 Ações de Controle com Função Quadrática (2º Caso).....	43
Figura 2.41 Pesos com Função Exponencial (2º Caso).....	44
Figura 2.42 Rede Neural com Função Exponencial (2º Caso).....	45
Figura 2.43 Ações de Controle com Função Exponencial (2º Caso).....	45
<b>Capítulo 3.....</b>	<b>47</b>
<b>Acionamento de Máquinas Elétricas de Indução.....</b>	<b>47</b>
Figura 3.1 Diagrama do Sistema de Controle.....	50
Figura 3.2 Estratégia de Controle PID.....	55
Figura 3.3 Controle Nebuloso com Inferência Direta.....	56
Figura 3.4 Funções de Pertinência das Entradas (1º Caso).....	57
Figura 3.5 Funções de Pertinência da Variável de Saída (1º Caso).....	58
Figura 3.6 Controle Nebuloso com Inferência Indireta.....	59
Figura 3.7 Funções de Pertinência das Entradas (2º Caso).....	60
Figura 3.8 Funções de Pertinência da Variável de Saída (2º Caso).....	61
Figura 3.9 Estratégia de Controle Neuro-Nebulosa.....	62
Figura 3.10 Funções de Pertinência das Entradas.....	63
Figura 3.11 Conjunto de Pesos com Função Não-Linear.....	64
Figura 3.12 Rede Neural Artificial com Função Não-Linear.....	65
Figura 3.13 Ações de Controle com Função Não-Linear.....	65
Figura 3.14 Corrente de Partida do Motor de Indução.....	68
Figura 3.15 Conjugado de Partida do Motor de Indução.....	68
Figura 3.16 Tensão e Corrente de Regime do Motor de Indução.....	68
Figura 3.17 Corrente de Partida do Motor de Indução com Inversor.....	70
Figura 3.18 Conjugado de Partida do Motor de Indução com Inversor.....	70
Figura 3.19 Tensão e Corrente de Regime do Motor de Indução com Inversor.....	70
Figura 3.20 Resposta de Sensor de Corrente na Partida.....	71
Figura 3.21 Resposta de Sensor de Corrente em Regime.....	71
Figura 3.22 Resposta de Sensor de Corrente na Partida com Inversor.....	72
Figura 3.23 Resposta de Sensor de Corrente em Regime com Inversor.....	72
Figura 3.24 Perfil de Frequência Senoidal com Inversor.....	73
Figura 3.25 Tensão e Corrente do Motor de Indução com Inversor.....	73
Figura 3.26 Tensão de Entrada do Motor de Indução com Inversor.....	74
Figura 3.27 Corrente de Entrada do Motor de Indução com Inversor.....	74
<b>Capítulo 4.....</b>	<b>75</b>
<b>Implementação do Sistema de Controle.....</b>	<b>75</b>
Figura 4.1 Diagrama em Blocos do Sistema de Controle.....	75
Figura 4.2 Corrente de Partida e Frequência de Rotação (Ensaio PD).....	83
Figura 4.3 Conjugado de Carga e Frequência de Rotação (Ensaio PD).....	83
Figura 4.4 Corrente de Partida: Valor Instantâneo e Valor Eficaz (Ensaio CE).....	84
Figura 4.5 Corrente de Regime: Valor Instantâneo e Valor Eficaz (Ensaio CE).....	84
Figura 4.6 Corrente de Partida e Frequência de Rotação (Ensaio PDex).....	85
Figura 4.7 Conjugado de Carga e Frequência de Rotação (Ensaio PDex).....	85
Figura 4.8 Corrente de Partida: Valor Instantâneo e Valor Eficaz (Ensaio CEex).....	86

---

---

Figura 4.9 Corrente de Regime: Valor Instantâneo e Valor Eficaz (Ensaio CEex).....	86
Figura 4.10 Sistema de Controle Proporcional Integral Derivativa.....	87
Figura 4.11 Resultado de Simulação com Matlab (Ensaio PID08).....	89
Figura 4.12 Resultado Prático obtido em Laboratório (Ensaio PID08).....	89
Figura 4.13 Resultado de Simulação com Matlab (Ensaio PID12).....	90
Figura 4.14 Resultado Prático obtido em Laboratório (Ensaio PID12).....	90
Figura 4.15 Resultado de Simulação com Matlab (Ensaio PID16).....	91
Figura 4.16 Resultado Prático obtido em Laboratório (Ensaio PID16).....	91
Figura 4.17 Resultado de Simulação com Matlab (Ensaio PID08ex).....	93
Figura 4.18 Resultado Prático obtido em Laboratório (Ensaio PID08ex).....	93
Figura 4.19 Resultado de Simulação com Matlab (Ensaio PID12ex).....	94
Figura 4.20 Resultado Prático obtido em Laboratório (Ensaio PID12ex).....	94
Figura 4.21 Resultado de Simulação com Matlab (Ensaio PID16ex).....	95
Figura 4.22 Resultado Prático obtido em Laboratório (Ensaio PID16ex).....	95
Figura 4.23 Sistema de Controle Nebuloso com Inferência Direta.....	96
Figura 4.24 Resultado de Simulação com Matlab (Ensaio NiD08).....	98
Figura 4.25 Resultado Prático obtido em Laboratório (Ensaio NiD08).....	98
Figura 4.26 Resultado de Simulação com Matlab (Ensaio NiD12).....	99
Figura 4.27 Resultado Prático obtido em Laboratório (Ensaio NiD12).....	99
Figura 4.28 Resultado de Simulação com Matlab (Ensaio NiD16).....	100
Figura 4.29 Resultado Prático obtido em Laboratório (Ensaio NiD16).....	100
Figura 4.30 Resultado de Simulação com Matlab (Ensaio NiD08ex).....	102
Figura 4.31 Resultado Prático obtido em Laboratório (Ensaio NiD08ex).....	102
Figura 4.32 Resultado de Simulação com Matlab (Ensaio NiD12ex).....	103
Figura 4.33 Resultado Prático obtido em Laboratório (Ensaio NiD12ex).....	103
Figura 4.34 Resultado de Simulação com Matlab (Ensaio NiD16ex).....	104
Figura 4.35 Resultado Prático obtido em Laboratório (Ensaio NiD16ex).....	104
Figura 4.36 Sistema de Controle Nebuloso com Inferência Indireta.....	105
Figura 4.37 Resultado de Simulação com Matlab (Ensaio NiI08).....	107
Figura 4.38 Resultado Prático obtido em Laboratório (Ensaio NiI08).....	107
Figura 4.39 Resultado de Simulação com Matlab (Ensaio NiI12).....	108
Figura 4.40 Resultado Prático obtido em Laboratório (Ensaio NiI12).....	108
Figura 4.41 Resultado de Simulação com Matlab (Ensaio NiI16).....	109
Figura 4.42 Resultado Prático obtido em Laboratório (Ensaio NiI16).....	109
Figura 4.43 Resultado de Simulação com Matlab (Ensaio NiI08ex).....	111
Figura 4.44 Resultado Prático obtido em Laboratório (Ensaio NiI08ex).....	111
Figura 4.45 Resultado de Simulação com Matlab (Ensaio NiI12ex).....	112
Figura 4.46 Resultado Prático obtido em Laboratório (Ensaio NiI12ex).....	112
Figura 4.47 Resultado de Simulação com Matlab (Ensaio NiI16ex).....	113
Figura 4.48 Resultado Prático obtido em Laboratório (Ensaio NiI16ex).....	113
Figura 4.49 Sistema com Estratégia de Controle Neuro-Nebulosa.....	114
Figura 4.50 Resultado de Simulação com Matlab (Ensaio NeN08).....	116
Figura 4.51 Resultado Prático obtido em Laboratório (Ensaio NeN08).....	116
Figura 4.52 Resultado de Simulação com Matlab (Ensaio NeN12).....	117
Figura 4.53 Resultado Prático obtido em Laboratório (Ensaio NeN12).....	117
Figura 4.54 Resultado de Simulação com Matlab (Ensaio NeN16).....	118
Figura 4.55 Resultado Prático obtido em Laboratório (Ensaio NeN16).....	118

---

---

Figura 4.56 Resultado de Simulação com Matlab (Ensaio NeN08ex).....	120
Figura 4.57 Resultado Prático obtido em Laboratório (Ensaio NeN08ex) .....	120
Figura 4.58 Resultado de Simulação com Matlab (Ensaio NeN12ex).....	121
Figura 4.59 Resultado Prático obtido em Laboratório (Ensaio NeN12ex) .....	121
Figura 4.60 Resultado de Simulação com Matlab (Ensaio NeN16ex).....	122
Figura 4.61 Resultado Prático obtido em Laboratório (Ensaio NeN16ex) .....	122
<b>Capítulo 5.....</b>	<b>123</b>
<b>Análise de Resultados e Conclusão.....</b>	<b>123</b>
Figura 5.1 Tempo de Resposta do Motor de Indução com Inversor de Tensão.....	126
Figura 5.2 Comportamento do Sistema de Controle com Corrente de Partida 16A .....	127
Figura 5.3 Comportamento do Sistema de Controle com Corrente de Partida 12A .....	128
Figura 5.4 Precisão do Sistema para Diferentes Estratégias de Controle .....	129
<b>Apêndice 1.....</b>	<b>133</b>
<b>Lógica Nebulosa .....</b>	<b>133</b>
Figura 1.1 Funções de Pertinência .....	135
Figura 1.2 Variável Lingüística Temperatura .....	138
Figura 1.3 Regra de Inferência Composicional.....	140
Figura 1.4 Mecanismo de Inferência.....	141
Figura 1.5 Função Descrita por Equação Matemática .....	141
Figura 1.6 Função Descrita por Regras Lingüísticas Exatas.....	142
Figura 1.7 Função Descrita por Regras Lingüísticas Nebulosas.....	145
Figura 1.8 Função Descrita por Rede Neural Artificial .....	146
Figura 1.9 Estrutura Básica de um Controlador Nebuloso.....	148
<b>Apêndice 2.....</b>	<b>151</b>
<b>Redes Neurais Artificiais.....</b>	<b>151</b>
Figura 2.1 Elemento Linear Adaptativo Padrão.....	151
Figura 2.2 Funções Típicas de Ativação de Saída .....	152
Figura 2.3 Exemplo de Elemento Linear Adaptativo.....	153
Figura 2.4 Hiperplano Separador .....	153
Figura 2.5 Modelo Simplificado do Cerebelo.....	155
Figura 2.6 Funções de Quantização .....	156
Figura 2.7 Conjunto S e Região Ativa para $S=(X_0, Y_0)$ .....	160
<b>Apêndice 3.....</b>	<b>165</b>
<b>Máquinas Elétricas de Indução .....</b>	<b>165</b>
Figura 3.1 Diagrama Fasorial.....	166
<b>Referências Bibliográficas.....</b>	<b>173</b>
<b>Bibliografia Complementar .....</b>	<b>177</b>

---

## *Lista de Tabelas*

<b>Sumário .....</b>	<b>ix</b>
<b>Lista de Figuras .....</b>	<b>xiii</b>
<b>Lista de Tabelas .....</b>	<b>xvii</b>
<b>Trabalhos Publicados .....</b>	<b>xix</b>
<b>Capítulo 1 .....</b>	<b>1</b>
<b>Contexto e Motivação .....</b>	<b>1</b>
<b>Capítulo 2 .....</b>	<b>9</b>
<b>Controlador Nebuloso de Arquitetura Simplificada .....</b>	<b>9</b>
Tabela 2.1 Representação Binária das Variáveis de Entrada .....	10
Tabela 2.2 Representação Binária da Variável de Saída .....	12
Tabela 2.3 Valor dos Pesos Associados à Variável de Saída .....	13
Tabela 2.4 Fuzzificação com Inferência Direta .....	16
Tabela 2.5 Fuzzificação com Inferência Indireta .....	17
Tabela 2.6 Defuzzificação da Variável de Saída .....	18
Tabela 2.7 Termos Lingüísticos com Inferência Direta .....	19
Tabela 2.8 Ações de Controle com Inferência Direta .....	20
Tabela 2.9 Termos Lingüísticos com Inferência Indireta .....	22
Tabela 2.10 Ações de Controle com Inferência Indireta .....	23
Tabela 2.11 Ações de Controle com Função Quadrática .....	38
Tabela 2.12 Ações de Controle com Função Exponencial .....	40
Tabela 2.13 Ações de Controle com Função Quadrática .....	42
Tabela 2.14 Ações de Controle com Função Exponencial .....	44
<b>Capítulo 3 .....</b>	<b>47</b>
<b>Acionamento de Máquinas Elétricas de Indução .....</b>	<b>47</b>
Tabela 3.1 Fuzzificação com Inferência Direta .....	57
Tabela 3.2 Ações de Controle com Inferência Direta .....	58
Tabela 3.3 Fuzzificação com Inferência Indireta .....	60
Tabela 3.4 Ações de Controle com Inferência Indireta .....	61
Tabela 3.5 Fuzzificação com Controle Neuro-Nebuloso .....	63
Tabela 3.6 Ações de Controle com Função Não-Linear .....	64
<b>Capítulo 4 .....</b>	<b>75</b>
<b>Implementação do Sistema de Controle .....</b>	<b>75</b>
Tabela 4.1 Características da Placa de Aquisição de Dados .....	81
Tabela 4.2 Características dos Sinais de Entrada e Saída .....	81
Tabela 4.3 Ensaios com Motor de Indução e Gerador Síncrono .....	82
Tabela 4.4 Ensaios com Controle Proporcional Integral Derivativo .....	87
Tabela 4.5 Ensaios com Controle Nebuloso com Inferência Direta .....	96
Tabela 4.6 Ensaios com Controle Nebuloso com Inferência Indireta .....	105
Tabela 4.7 Ensaios com Estratégia de Controle Neuro-Nebulosa .....	114
<b>Capítulo 5 .....</b>	<b>123</b>
<b>Análise de Resultados e Conclusão .....</b>	<b>123</b>
Tabela 5.1 Resolução da Placa de Aquisição de Dados .....	123
Tabela 5.2 Representação das Variáveis de Entrada do Controlador .....	124
Tabela 5.3 Representação das Variáveis de Saída do Controlador .....	124

<b>Apêndice 1</b> .....	<b>133</b>
<b>Lógica Nebulosa</b> .....	<b>133</b>
<b>Apêndice 2</b> .....	<b>151</b>
<b>Redes Neurais Artificiais</b> .....	<b>151</b>
Tabela 2.1 Função Reproduzida.....	153
<b>Apêndice 3</b> .....	<b>165</b>
<b>Máquinas Elétricas de Indução</b> .....	<b>165</b>
<b>Referências Bibliográficas</b> .....	<b>173</b>
<b>Bibliografia Complementar</b> .....	<b>177</b>

---

## ***Trabalhos Publicados***

- [01] M. E. BORDON, I. N. DA SILVA, A. N. DE SOUZA, “Design of a Fuzzy Controller with Simplified Architecture”, **Proceedings of the VI Brazilian Symposium on Neural Networks**, Rio de Janeiro, Brasil, 2000.
- [02] M. E. BORDON, I. N. DA SILVA, E. AVOLIO, “Design of a Neuro-Fuzzy Controller with Simplified Architecture”, **International Journal of Neural Systems**, 11, (3), 229-235, 2001.
- [03] M. E. BORDON, E. AVOLIO, I. N. DA SILVA, A. N. DE SOUZA, “Design of a Simplified Neuro-Fuzzy System Based on CMAC Architectures”, **Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics**, Orlando, USA, 2001.
- [04] M. E. BORDON, O. V. AVILEZ FILHO, E. AVOLIO, “A New Strategy of Induction Motor Drive Using Fuzzy Logic for Real Time Speed Control and Input Current Restriction”, **Anais da VI Conferência Internacional de Aplicações Industriais**, Joinville, Brasil, 2004.



---

## Capítulo 1

### Contexto e Motivação

#### 1.1 Introdução

A lógica nebulosa foi desenvolvida por *Zadeh* durante a década de **1960**. Seu artigo publicado em 1965 pela Universidade da Califórnia, em Berkeley, revolucionou o conceito de lógica com a criação dos sistemas nebulosos (ZADEH, 1965).

Os recursos tecnológicos disponíveis não eram capazes de automatizar as atividades relacionadas a problemas que compreendessem informações ambíguas, não passíveis de serem processadas através da lógica booleana. Era preciso algo mais do que simplesmente dois valores possíveis, dois estados diferentes não eram suficientemente representativos.

O trabalho desenvolvido por *Zadeh* combina lógica multivalorada, teoria probabilística, inteligência artificial e redes neurais artificiais para que possa representar o pensamento humano, ou seja, ligar a lingüística e a inteligência humana, pois muitos conceitos contendo informações vagas e incertas, são melhores definidos por palavras do que por matemática.

A lógica nebulosa é a lógica que suporta os modos de raciocínio que são aproximados ao invés de exatos, ou ainda, de forma mais abrangente, pode ser aceita como a melhor maneira de representação do raciocínio humano, que é parcial e aproximado em sua essência.

Os sistemas nebulosos utilizam técnicas para o tratamento de informações qualitativas de uma forma rigorosa. Derivada do conceito de conjuntos nebulosos, a lógica nebulosa constitui a base para o desenvolvimento de métodos e algoritmos de modelagem e controle de processos, permitindo a redução da complexidade de projeto e implementação, tornando-se uma alternativa para a solução de problemas de controle e modelagem de sistemas.

Esta forma de estruturação do raciocínio é capaz de tomar decisões racionais mesmo estando em um ambiente de incertezas e imprecisões, onde dados desta natureza e até mesmo conflitantes são tratados. A tecnologia advinda da teoria dos conjuntos nebulosos e da lógica nebulosa tem permitido a aplicação destas técnicas em diversas áreas do conhecimento, tendo como exemplos os sistemas de controle nebulosos, os quais podem possibilitar um controle mais acurado, além de um desempenho estável e robusto (LEE, 1990).

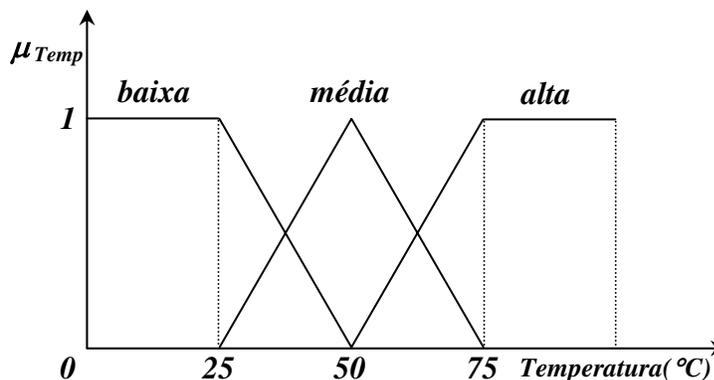
Através da lógica nebulosa, um sistema pode ser representado através de conceitos imprecisos tais como: *rápido*, *baixo* e *pequeno*. Essa característica é responsável pela facilidade na extração do conhecimento de um sistema a partir de observações realizadas sobre este.

Além disso, na lógica nebulosa existe a possibilidade de se empregar probabilidades lingüísticas, tais como: *provável*, *altamente provável* e *improvável*, interpretadas como dados nebulosos manipuladas pela aritmética nebulosa. Em contraste com a lógica clássica, o conceito de possibilidade é interpretado usando subconjuntos nebulosos no universo dos reais.

Na teoria de conjuntos clássica, um elemento pertence ou não pertence um conjunto, enquanto que na teoria de conjuntos nebulosa, sugere-se que alguns elementos são mais membros de um conjunto do que outros. O índice de pertinência  $\mu$  é usado para caracterizar o quanto cada elemento pertence ao conjunto em questão, podendo assumir qualquer valor no intervalo de  $0$  a  $1$ , sendo que o valor  $0$  indica uma completa exclusão e um valor  $1$  representa uma completa inclusão, ou seja, a função característica é contínua no seu domínio (ZADEH, 1973).

Para expressar conceitos através da linguagem natural é comum o uso de elementos qualitativos ao invés de valores quantitativos. Estes conceitos podem ser capturados pela definição de variáveis lingüísticas. Uma variável lingüística tem por característica assumir valores dentro de um conjunto de termos lingüísticos, ou seja, palavras ou frases. Assim, ao invés de assumir instâncias numéricas, estas variáveis assumem instâncias lingüísticas.

Por exemplo, uma variável lingüística *Temperatura* poderá assumir como valor um dos termos do conjunto  $\{baixa, média, alta\}$ . Para se atribuir um significado aos termos lingüísticos, associa-se a cada um destes termos um conjunto nebuloso definido sobre um universo de discurso comum. A *Figura 1.1* ilustra os termos lingüísticos da variável lingüística *Temperatura*, onde  $\mu_{Temp}$  representa o índice de pertinência de cada termo lingüístico.



*Figura 1.1 Variável Lingüística Temperatura*

A forma mais comum de expressar o conhecimento especialista é por meio de regras do tipo condição-ação. Neste caso, um conjunto de condições que descreve uma parcela observável das entradas de um sistema é associado com uma ação de controle que irá manter ou levar o sistema para a condição de operação desejada.

Tipicamente, uma condição é uma proposição lingüística relacionada com o valor de alguma variável de entrada, como por exemplo *temperatura muito alta*. Da mesma maneira, uma típica ação de controle é uma descrição lingüística do tipo *aumente a vazão*.

A idéia de representar o conhecimento por meio de um conjunto de termos lingüísticos associados às variáveis de entrada e saída de um processo é absorvida nos sistemas de inferência nebulosa. As ações de controle são expressas de forma similar para cada variável de controle, onde regras do tipo *se-então* são freqüentemente usadas e chamadas de declarações condicionais nebulosas ou simplesmente regras nebulosas (PEDRYCZ & GOMIDE, 1998).

Nos sistemas que utilizam inferência nebulosa, os valores de entrada são normalizados e convertidos para uma representação nebulosa. Este procedimento é denominado de *fuzzificação* das variáveis de entrada. As regras do sistema nebuloso serão executadas em paralelo, criando uma região nebulosa para cada variável de saída. A criação desta região nebulosa a partir das regras ativas do sistema é conhecida como procedimento de agregação.

Depois do processamento das regras de inferência, as regiões nebulosas são convertidas em valores não nebulosos, determinando o valor de cada variável de saída. O procedimento de se converter uma região nebulosa em um número real é conhecido por *defuzzificação*.

Em uma base de regras, quando mais de uma regra é acionada, as contribuições das diversas regras são combinadas pelo operador de agregação. Depois do processo de agregação, tem-se como resultado um conjunto nebuloso representativo da saída do sistema de inferência nebulosa, no entanto, este conjunto nebuloso deve ser convertido para um valor real.

A ação de controle resultante é obtida através da união de todas as regras nebulosas ativas. O valor final, ou seja, aquele que será apresentado pelas saídas do sistema nebuloso, será determinado pelo procedimento de *defuzzificação*. Existem diversos métodos para se realizar a *defuzzificação*, tais como, a média dos máximos, o centro de massa e o centro de área.

Durante a fase de desenvolvimento um sistema de inferência nebulosa depende da especificação de alguns parâmetros operacionais. Dentre os parâmetros relacionados com a sua estrutura pode-se citar o número de funções de pertinência para cada entrada e para cada saída, o número de regras nebulosas, bem como a definição de cada uma das regras.

---

A determinação destes parâmetros pode ser realizada pelo uso de associações empíricas, no entanto, muitos processos não fornecem subsídios adequados à determinação correta dos parâmetros estruturais do sistema nebuloso a ser desenvolvido.

De modo geral, os sistemas de inferência nebulosa possuem propriedades peculiares que os tornam atrativos em diversas aplicações. Além de serem capazes de incorporar conceitos lingüísticos, os sistemas de inferência nebulosa são também aproximadores universais, capazes de mapearem as relações entre as entradas e saídas de um dado sistema.

Essas duas propriedades podem ser utilizadas para diferenciar dois tipos principais de sistemas de inferência nebulosa (GUILLAUME, 2001).

O primeiro tipo de sistema de inferência nebulosa possui seu foco direcionado para a habilidade de modelar a linguagem natural inerente à lógica nebulosa. Este tipo de sistema de inferência nebulosa contém regras nebulosas definidas a partir de uma base de conhecimento especialista. Tais sistemas são conhecidos como sistemas nebulosos especialistas ou sistemas de controle nebulosos. (MANDANI & ASSILIAN, 1975).

O uso desse primeiro tipo de sistema de inferência nebulosa, devido á sua estruturação, oferece robustez adequada aos propósitos de controle ou mesmo modelagem, pois os mesmos permitem a incorporação gradual de regras à sua base de conhecimento sem haja perda de generalidade ou inconsistência no resultado final.

Outro tipo de sistema de inferência nebulosa pode ser identificado como sendo aqueles que incorporam o treinamento fazendo uso de um conjunto de dados onde estão paralelizadas as variáveis de entrada com suas respectivas saídas. Desta forma, pode-se observar que este segundo tipo de sistema de inferência nebulosa é definido com base nos dados coletados do sistema ao qual se deseja modelar (TAKAGI & SUGENO, 1985).

*Takagi e Sugeno* estão entre os primeiros pesquisadores a proporem novas estruturas de sistemas de inferência nebulosa adequadas à utilização de algoritmos de treinamento e sintonia, baseados em padrões de ajuste para sua definição. Os métodos de ajuste de sistemas de inferência nebulosa são compostos de duas etapas. A primeira etapa é responsável pela otimização estrutural e a segunda etapa é dedicada ao ajuste paramétrico do sistema nebuloso.

A otimização estrutural do sistema de inferência nebulosa pode abordar aspectos como a determinação do número de funções de pertinência para cada entrada, bem como sua topologia. A maioria dos métodos de otimização estrutural foca a geração da base de regras a qual pode ser decomposta em dois passos principais: indução de regras e redução da base de regras.

---

Além da otimização estrutural, um sistema de inferência nebulosa possui muitos outros parâmetros que podem ser ajustados como os parâmetros das funções de pertinência de entrada e a ponderação de cada regra nebulosa. Este procedimento de ajuste de parâmetros é conhecido como otimização paramétrica de sistemas de inferência nebulosa.

Mais recentemente, podem-se observar expressivos avanços no desenvolvimento de metodologias de otimização estrutural. Destacam-se as técnicas de otimização que empregam ferramentas computacionais como algoritmos genéticos e redes neurais artificiais, resultando em modelos híbridos neuro-nebulosos (GUILLAUME, 2001).

## 1.2 *Objetivo do Trabalho*

O objetivo deste trabalho foi o desenvolvimento de um controlador nebuloso simples e eficiente para aplicações industriais. Durante o projeto deste controlador optou-se pela utilização de um sistema digital de baixo custo baseado em microcontroladores de 8 bits, levando-se em conta todas as limitações impostas pela arquitetura destes microcontroladores. Mesmo assim, os resultados práticos obtidos neste trabalho foram além da expectativa comprovando a viabilidade operacional deste controlador nebuloso de arquitetura simplificada.

A estrutura simplificada do controlador nebuloso considerando microcontroladores com arquitetura de 8 bits, determinou a execução de procedimentos padronizados de *fuzzificação* das variáveis de entrada e *defuzzificação* da variável de saída com as ações de controle previamente calculadas e armazenadas em memória. Um operador de agregação desenvolvido especificamente para este controlador nebuloso determina o valor de todas as ações de controle.

Trata-se de uma expressão matemática capaz de mapear superfícies planas que podem ser armazenadas em *64 kbytes* de memória. Cada posição de memória pode ser endereçada pela combinação binária gerada a partir dos valores digitais das variáveis de entrada  $x$  e  $y$ . O valor armazenado em cada posição de memória corresponde à variável de saída  $z$ , constituindo uma estrutura de dados do tipo  $z=f(x,y)$ , onde  $z$  representa uma ação de controle usada na composição da variável de controle, segundo um procedimento de inferência previamente estabelecido.

Portanto, o tempo de resposta deste controlador nebuloso de arquitetura simplificada, operando em malha fechada depende de uma simples operação de acesso à memória de dados, onde estão armazenadas todas as possíveis ações de controle, geradas a partir de procedimentos de inferência nebulosa baseada em conhecimento especialista.

Para verificar a eficiência do controlador nebuloso foi implementado em laboratório um sistema digital para acionamento de um motor de indução com controle de frequência de rotação e limitação de corrente em tempo real. O objetivo principal desta aplicação foi o estudo de novas estratégias de controle digital para acionamento de motores de indução com perfil de frequência gerado em tempo real e independente do modelo matemático do motor de indução.

Informando ao sistema de controle o valor da frequência de rotação e o valor limite da corrente de entrada do motor de indução, a estratégia de controle determina em tempo real, o perfil de frequência adequado, garante que o limite de corrente não seja ultrapassado e ainda evita o desligamento desnecessário do sistema durante o acionamento do motor de indução.

Para tanto, são necessárias duas malhas de controle, uma para o controle da frequência de rotação e outra para o controle da corrente de entrada. Devido à utilização de duas malhas de controle digital existe a necessidade da composição de duas ações de controle provenientes de duas malhas de controle diferentes a fim de se obter uma ação de controle combinada.

Uma única ação de controle pode ser obtida por meio de operações de inferência a partir de informações referentes ao erro relativo à frequência de rotação e ao erro relativo à corrente de entrada do motor de indução, no entanto, a superfície a ser mapeada não é plana. Considerando que a expressão matemática usada no procedimento de *defuzzificação* atua como um mecanismo de interpolação linear, então, não é possível mapear superfícies não-planas.

Uma rede neural artificial do tipo *CMAC* (Cerebellar Model Articulation Controller) foi utilizada como mecanismo de interpolação não-linear capaz de mapear superfícies não-planas geradas a partir de procedimentos de inferência nebulosa. O uso da arquitetura *CMAC* foi muito interessante pois esta rede neural artificial pode ser mapeada na mesma estrutura de memória usada no controlador nebuloso desenvolvido neste trabalho (ALBUS, 1975).

Portanto, uma rede neural artificial usada como um mecanismo de interpolação é capaz de mapear em memória as ações de controle combinadas substituindo duas malhas de controle por apenas uma. Usar uma rede neural artificial com arquitetura *CMAC* como um mecanismo de interpolação pode não ser viável do ponto de vista computacional mas, além de atuar como um mecanismo de interpolação esta rede neural artificial pode ser usada como um mecanismo de inferência transformando este controlador digital em um controlador neuro-nebuloso.

Por outro lado, uma rede neural artificial do tipo *CMAC* também pode ser usada como um mecanismo de estimação capaz de estimar a frequência de escorregamento de um motor de indução a partir de medidas de corrente, tensão e frequência, efetuadas durante o acionamento.

---

---

## 1.3 Apresentação

A apresentação deste trabalho é feita através de capítulos e apêndices. Nos capítulos, estão descritos os procedimentos adotados durante o projeto do controlador nebuloso, definição das estratégias de controle e implementação do sistema de controle para acionamento de um motor de indução trifásico. Nos apêndices, estão detalhados conceitos importantes de lógica nebulosa, redes neurais artificiais, máquinas elétricas de indução e simulação computacional.

Trata-se de um trabalho de engenharia aplicada que utiliza conhecimentos específicos de diversas áreas com o objetivo de desenvolver um controlador digital simples e eficiente visando aplicações industriais. Para tanto, foram usadas teorias amplamente conhecidas de várias áreas da engenharia elétrica, tais como: sistemas eletrônicos de medida e controle, automação eletrônica de processos industriais, máquinas elétricas de indução, sistemas microprocessados, redes neurais artificiais, lógica nebulosa, tratamento de sinais, aquisição dados e programação.

### 1.3.1 Descrição dos Capítulos

No **Capítulo 1**, apresenta-se uma descrição sucinta do trabalho com destaque para os motivos que levaram à sua realização considerando o projeto de um controlador nebuloso de arquitetura simplificada e sua aplicação em diferentes estratégias de controle para acionamento de motores de indução trifásicos, discutindo como e porque a lógica nebulosa e as redes neurais artificiais foram utilizadas durante o desenvolvimento deste trabalho.

No **Capítulo 2**, apresenta-se o projeto detalhado do controlador nebuloso de arquitetura simplificada, o tratamento das variáveis de entrada e saída e os procedimentos de **fuzzificação** e **defuzzificação**. Para validar o projeto do controlador apresenta-se uma simulação computacional utilizando uma função de transferência de segunda ordem e, no final do capítulo, apresenta-se a rede neural artificial **CMAC** atuando como um operador de agregação.

No **Capítulo 3**, apresenta-se o projeto de um sistema de controle para acionamento do motor de indução considerando diferentes estratégias de controle e a simulação computacional dos modelos matemáticos do motor de indução trifásico e do sensor de corrente. Neste trabalho, foram implementadas e devidamente simuladas as estratégias de controle: proporcional integral derivativa, controle nebuloso direto e indireto e controle neuro-nebuloso.

No **Capítulo 4**, apresenta-se a descrição do sistema de controle digital construído em laboratório, o projeto dos circuitos condicionadores de sinais e do sistema de aquisição de dados. Os resultados obtidos durante os ensaios do motor de indução acoplado a um gerador síncrono e acionado por um inversor de tensão do tipo **PWM** (Pulse Width Modulation) sujeito a diferentes estratégias de controle, também são apresentados neste capítulo.

No **Capítulo 5**, apresenta-se uma análise completa dos resultados obtidos considerando o desempenho das diferentes estratégias de controle comparando resultados práticos e simulados, destacando vantagens e desvantagens do sistema de controle proposto. A conclusão do trabalho e a perspectiva de novos estudos visando a implementação de um mecanismo para estimação da frequência de escorregamento, também são apresentados neste capítulo.

### **1.3.2 Descrição dos Apêndices**

No **Apêndice 1**, apresenta-se o conceito de lógica nebulosa e sua aplicação na descrição de sistemas e processos em geral. Os fundamentos da lógica nebulosa e os conceitos de interface de **fuzzificação**, base de conhecimento, base de dados, procedimentos de inferência, interface de **defuzzificação** e operadores de agregação, são apresentados. Todos os assuntos abordados neste apêndice foram extraídos de um artigo publicado por (GOMIDE & GUDWIN, 1994).

No **Apêndice 2**, apresenta-se o conceito de redes neurais artificiais tipo **FeedForward**, **Hopfield** e **CMAC** (Cerebellar Model Articulation Controller). Fundamentos sobre os elementos lineares adaptativos, as funções típicas de ativação e os métodos de treinamento baseados em mínimos quadrados, são apresentados. Um estudo detalhado das redes neurais artificiais do tipo **CMAC** foi extraído da dissertação de mestrado publicada por (BORDON & TEIXEIRA, 1995).

No **Apêndice 3**, apresenta-se o conceito de máquinas elétricas de indução e conversores de frequência, o modelo matemático de um motor de indução baseado nas equações de **Park** e o modelo matemático de um inversor de tensão do tipo **PWM** (Pulse Width Modulation) baseado em séries de **Fourier**. O modelo de um motor de indução trifásico acionado por um inversor de tensão foi extraído da dissertação de mestrado publicada por (FONTES & SZAJNER, 1993).

## Capítulo 2

# Controlador Nebuloso de Arquitetura Simplificada

### 2.1 Introdução

O tratamento de um sistema de controle através de equações diferenciais que descrevem o comportamento estático ou dinâmico deste sistema representa uma ferramenta matemática de suma importância. No entanto, é muito difícil identificar as equações diferenciais que descrevem com exatidão o comportamento de sistemas complexos com múltiplas variáveis pois, segundo *Zadeh*, quando a complexidade de um sistema aumenta a possibilidade de descrever este sistema através de equações matemáticas diminui. Uma alternativa viável é o uso de mecanismos de inferência baseados em regras de controle ativadas a partir de informações consistentes, advindas de termos lingüísticos associados às variáveis de entrada do sistema (ZADEH, 1973).

As regras de controle são estabelecidas a partir de um conjunto de condições que descrevem o comportamento do sistema e através de uma base de conhecimento consistente é possível determinar uma relação precisa entre variáveis de entrada e ações de controle. Deste modo, as variáveis de entrada são desmembradas em conjuntos discretos de termos lingüísticos, devidamente ponderados, estabelecendo um significado preciso às situações de entrada. As ações de controle, decorrentes do conjunto de termos lingüísticos associados às variáveis de entrada, são obtidas a partir de procedimentos de inferência e regras simbólicas da inteligência artificial devidamente interpretada segundo a lógica nebulosa (GOMIDE & GUDWIN, 1994).

Uma base de conhecimento consistente determina todas as regras de controle a serem implementadas de modo eficaz e preciso. Se a base de conhecimento não é consistente, é preciso recorrer a mecanismos capazes de incorporar novos conhecimentos. As redes neurais artificiais são usadas na formação de bases de conhecimento consistentes e podem ser incorporadas aos sistemas de controle baseados em lógica nebulosa. Muitas vezes, as redes neurais são utilizadas para acrescentar novas informações à base de conhecimento tendo como parâmetro algum índice de mérito que deve ser otimizado. Um controlador neuro-nebuloso representa uma alternativa viável para descrição de sistemas complexos pois as limitações de ambas as abordagens são tratadas de modo a criar sistemas mais eficientes (OLIVEIRA et al., 1994).

Apresenta-se a seguir o projeto de um controlador nebuloso de uso geral com arquitetura simplificada, que utiliza um algoritmo com rotinas específicas para aquisição de dados e controle. Trata-se de um sistema digital que pode ser implementado em microcomputador equipado com placa de aquisição de dados ou através de um circuito eletrônico dedicado construído a partir de microcontroladores e conversores de 8, 12 ou 16 bits.

## 2.2 Variáveis de Entrada

A interface de *fuzzificação* recebe as variáveis analógicas de entrada geradas por um elemento sensor e um circuito de referência e fornece as variáveis digitais equivalentes usando um conversor analógico digital de 8 bits. As variáveis de entrada  $x$  e  $y$  do controlador nebuloso codificadas através de um único byte podem variar dentro do intervalo discreto [0:255].

A partir das variáveis de entrada  $x$  e  $y$ , é possível determinar os termos lingüísticos e os índices de pertinência dos conjuntos nebulosos de entrada. Os conjuntos nebulosos de entrada são padronizados segundo uma representação binária adequada que associa a cada termo lingüístico de entrada um peso  $w$  inteiro positivo, conforme indicado na *Tabela 2.1*.

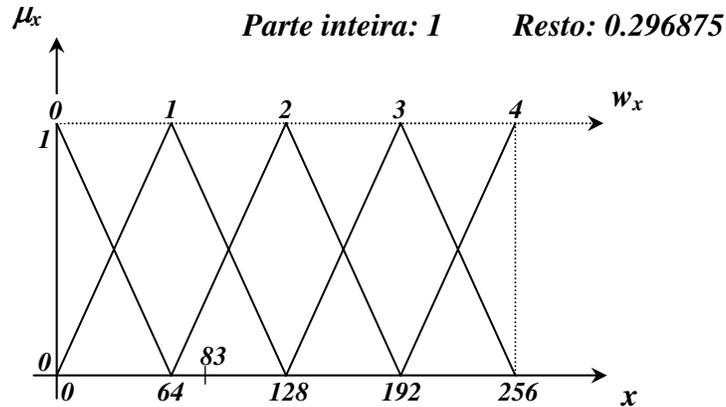
**Tabela 2.1 Representação Binária das Variáveis de Entrada**

<i>Fator</i> $f_x$ ou $f_y$	<i>Peso</i> $w_x$ ou $w_y$	<i>Índice de Pertinência</i> $\mu_x$ ou $\mu_y$	<i>Representação Binária</i>
256	[0:1]	[0:255]÷256	.#####
128	[0:2]	[0:127]÷128	#####
64	[0:4]	[0:63]÷64	##.#####
32	[0:8]	[0:31]÷32	###.#####
16	[0:16]	[0:15]÷16	####.#####
8	[0:32]	[0:7]÷8	#####.###
4	[0:64]	[0:3]÷4	#####.##
2	[0:128]	[0:1]÷2	#####.#

Os fatores  $f_x$  e  $f_y$  atuam como divisores das variáveis de entrada  $x$  e  $y$  e através de uma simples operação de divisão ocorre o processo de *fuzzificação*. O quociente indica o(s) peso(s) associado(s) ao conjunto lingüístico de entrada, enquanto que o resto indica o(s) índice(s) de pertinência correspondente(s). Considerando fatores  $f_x$  e  $f_y$  iguais a 64, os índices de pertinência  $\mu_x$  e  $\mu_y$  serão digitalizados com 6 bits significativos e resolução igual a 0.015625 unidades.

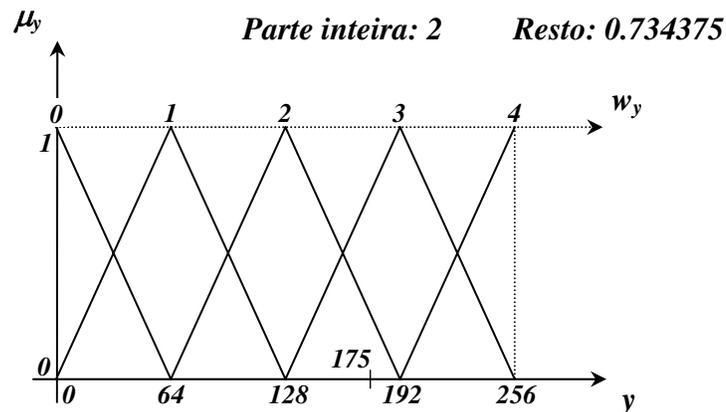
Como exemplo, a **Figura 2.1** apresenta a variável de entrada  $x$  e a **Figura 2.2** apresenta a variável de entrada  $y$ , ambas variando no intervalo discreto  $[0:255]$ , com pesos  $w_x$  e  $w_y$  variando no intervalo discreto  $[0:4]$  e índices de pertinência  $\mu_x$  e  $\mu_y$  variando no intervalo contínuo  $[0:1]$ . Para uma entrada  $x=83$  e uma entrada  $y=175$ , obtêm-se os seguintes resultados:

$$\frac{x}{f_x} = \frac{83}{64} = 1.296875 = (01.010011)_2$$



Para a entrada  $x=83$ , estão ativos os termos lingüísticos de peso  $w_{xa}=1$ , com índice de pertinência  $\mu_{xa}=0.703125$  e, peso  $w_{xb}=2$ , com índice de pertinência  $\mu_{xb}=0.296875$ .

$$\frac{y}{f_y} = \frac{175}{64} = 2.734375 = (10.101111)_2$$



Para a entrada  $y=175$ , estão ativos os termos lingüísticos de peso  $w_{ya}=2$ , com índice de pertinência  $\mu_{ya}=0.265625$  e, peso  $w_{yb}=3$ , com índice de pertinência  $\mu_{yb}=0.734375$ .

Portanto, a arquitetura simplificada deste controlador nebuloso permite o tratamento das variáveis de entrada segundo um padrão lógico de representação que não pode ser modificado. Assim, o universo de discurso de cada variável de entrada o conjunto de termos lingüísticos e até mesmo os conjuntos nebulosos são elementos estáticos do controlador nebuloso.

## 2.3 Variável de Saída

A interface de *defuzzificação* gera a variável de saída  $z$  a partir de um conjunto de regras de controle decorrentes de operações de inferência. A variável de saída  $z$  do controlador nebuloso codificada através de um único byte pode variar dentro do intervalo discreto  $[-127:127]$ .

As regras de controle e as respectivas ações de controle são determinadas a partir dos conjuntos nebulosos de entrada considerando uma representação binária que associa a cada termo lingüístico de entrada, um peso  $w$  inteiro positivo ou negativo, conforme a *Tabela 2.2*.

*Tabela 2.2 Representação Binária da Variável de Saída*

<i>Fator</i> $f_z$	<i>Peso</i> $w_z$	<i>Índice de Pertinência</i> $\mu_z$	<i>Representação Binária</i>
128	$[-1:1]$	$[0:127] \div 128$	$\pm\#\#\#\#\#\#$
64	$[-2:2]$	$[0:63] \div 64$	$\pm\#\#\#\#\#\#$
32	$[-4:4]$	$[0:31] \div 32$	$\pm\#\#\#\#\#\#$
16	$[-8:8]$	$[0:15] \div 16$	$\pm\#\#\#\#\#\#$
8	$[-16:16]$	$[0:7] \div 8$	$\pm\#\#\#\#\#\#$
4	$[-32:32]$	$[0:3] \div 4$	$\pm\#\#\#\#\#\#$
2	$[-64:64]$	$[0:1] \div 2$	$\pm\#\#\#\#\#\#$

O fator  $f_z$  define o ganho do controlador nebuloso e seu valor depende dos fatores  $f_x$  e  $f_y$  definidos durante os procedimentos de *fuzzificação*. Para fatores  $f_x$  e  $f_y$  iguais a 64 e pesos  $w_x$  e  $w_y$  variando dentro do intervalo discreto  $[0:4]$ , o valor do fator  $f_z$  será igual a 32 com peso  $w_z$  variando dentro no intervalo discreto  $[-4:4]$ . Assim, os pesos  $w_z$  associados às regras de controle podem assumir valores inteiros positivos, negativos e nulos, conforme a *Tabela 2.3*.

A cada par ordenado  $(w_x, w_y)$  está associado um valor de peso  $w_z = f(w_x, w_y)$ , neste caso, são considerados 25 pares ordenados com os respectivos pesos associados às regras de controle: 10 valores positivos, 10 valores negativos e 5 valores iguais a zero. A *Figura 2.3* mostra os pesos associados às regras de controle, decorrentes de operações lógicas de inferência.

Tabela 2.3 Valor dos Pesos Associados à Variável de Saída

Valores Positivos:	Valores Negativos:
$w_z(0,4) = 4$	$w_z(4,0) = -4$
$w_z(0,3) = w_z(1,4) = 3$	$w_z(3,0) = w_z(4,1) = -3$
$w_z(0,2) = w_z(1,3) = w_z(2,4) = 2$	$w_z(2,0) = w_z(3,1) = w_z(4,2) = -2$
$w_z(0,1) = w_z(1,2) = w_z(2,3) = w_z(3,4) = 1$	$w_z(1,0) = w_z(2,1) = w_z(3,2) = w_z(4,3) = -1$
Valores Iguais a Zero:	
$w_z(0,0) = w_z(1,1) = w_z(2,2) = w_z(3,3) = w_z(4,4) = 0$	

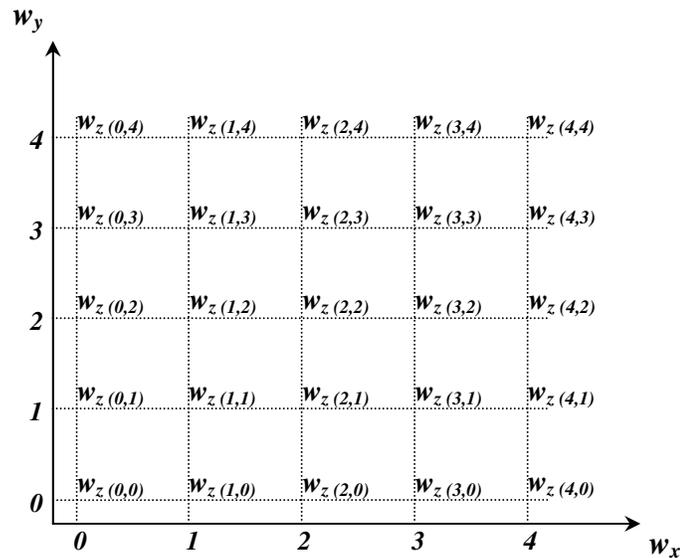


Figura 2.3 Pesos Associados à Variável de Saída

Considerando fator  $f_z$  igual a 32, o índice de pertinência  $\mu_z$  será digitalizado com 5 bits na parte fracionária, 2 bits na parte inteira, 1 bit de sinal e resolução igual a **0.03125** unidades, para uma variável de saída  $z$  dentro do intervalo  $[-127;127]$ , conforme indicado na **Figura 2.4**.

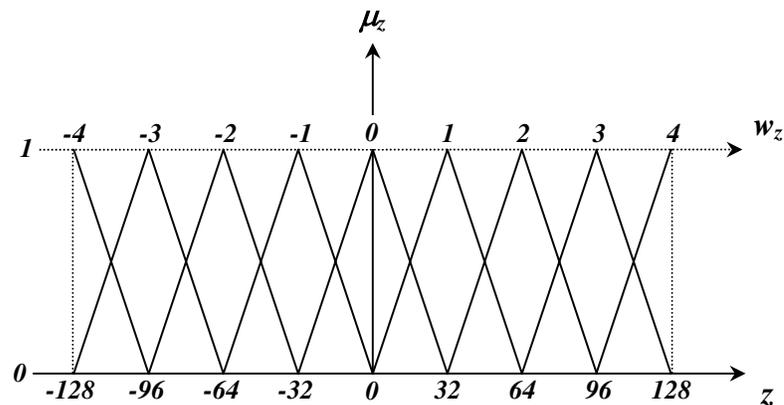


Figura 2.4 Funções de Pertinência da Variável de Saída

A partir de cálculos matemáticos elementares é possível obter para cada combinação de entrada, o valor da variável de saída  $z$  correspondente. A equação (2.1) representa os cálculos matemáticos elementares utilizados durante os procedimentos de *defuzzificação*.

$$\frac{z}{f_z} = (\mu_{xa} + \mu_{ya})w_{z(w_{xa}, w_{ya})} + (\mu_{xb})w_{z(w_{xb}, w_{ya})} + (\mu_{yb})w_{z(w_{xa}, w_{yb})} - w_{z(w_{xa}, w_{ya})} \quad (2.1)$$

Com auxílio de procedimentos simplificados de *fuzzificação* é possível determinar os pesos associados aos termos lingüísticos ativos, com seus respectivos índices de pertinência e determinar as regras de controle ativas, decorrentes de operações básicas de inferência.

Para uma entrada  $x$  igual a **83**, estão ativos os termos lingüísticos de peso  $w_{xa}=1$ , com índice de pertinência  $\mu_{xa}=0.703125$  e peso  $w_{xb}=2$ , com índice de pertinência  $\mu_{xb}=0.296875$ .

Para uma entrada  $y$  igual a **175**, estão ativos os termos lingüísticos de peso  $w_{ya}=2$ , com índice de pertinência  $\mu_{ya}=0.265625$  e peso  $w_{yb}=3$ , com índice de pertinência  $\mu_{yb}=0.734375$ .

Com  $x$  igual a **83** e  $y$  igual a **175**, estão ativas regras de controle com os seguintes pesos:

$$w_{z(w_{xa}, w_{ya})} = w_{z(1,2)} = 1$$

$$w_{z(w_{xb}, w_{ya})} = w_{z(2,2)} = 0$$

$$w_{z(w_{xa}, w_{yb})} = w_{z(1,3)} = 2$$

O fator  $f_z$  igual a **32** define uma ação de controle  $Ac$  dentro do intervalo  $[-127:127]$ . Substituindo os valores na equação (2.1), é possível determinar o valor de uma ação de controle.

$$\frac{z}{f_z} = (0.703125 + 0.265625)w_{z(1,2)} + (0.296875)w_{z(2,2)} + (0.734375)w_{z(1,3)} - w_{z(1,2)}$$

$$\frac{z}{f_z} = (0.96875)(1) + (0.296875)(0) + (0.734375)(2) - (1) = 1.4375$$

$$Ac = 1.4375 \times 32 = 46$$

Cada ação de controle  $Ac$  alterada por uma constante proporcional  $Kp$ , pode ser utilizada na composição da variável de controle  $Vc$ , conforme indicado na equação (2.2).

$$Vc_{(n)} = Vc_{(n-1)} + Ac_{(n)}Kp \quad (2.2)$$

A variável de controle  $Vc$  deve ser representada com pelo menos **16** bits, para garantir a precisão do controlador, mesmo usando um conversor digital analógico de **8** ou **12** bits.

## 2.4 Procedimentos de Fuzzificação

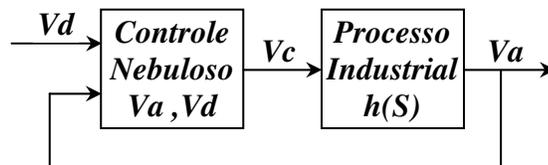
Durante os procedimentos de *fuzzificação*, o controlador nebuloso toma as variáveis de entrada, faz um escalonamento para condicionar os valores de entrada, transformando números em conjuntos normalizados de modo que possam se tornar instâncias de variáveis lingüísticas.

O projeto de um controlador nebuloso depende da definição de alguns parâmetros por parte do projetista. Dado um processo, alguns parâmetros são fixos dentro de condições normais de operação sendo que outros precisam ser alterados periodicamente. Os parâmetros fixos são chamados de parâmetros estruturais e os variáveis são chamados parâmetros ajustáveis.

A arquitetura simplificada do controlador nebuloso desenvolvido neste trabalho permite estruturar automaticamente os conjuntos nebulosos de entrada. Os procedimentos de *fuzzificação* são simplificados, os limites de atuação do controlador padronizados e todas as ações de controle previamente calculadas. A representação binária de entrada determina a precisão do controlador nebuloso e depende do circuito digital utilizado para sua implementação.

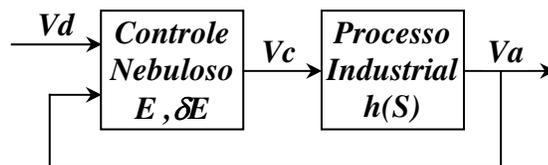
Durante os procedimentos de *fuzzificação* foram considerados dois casos distintos para configuração do sistema de controle em função das variáveis de entrada.

No primeiro caso, utiliza-se a inferência direta pois as variáveis de entrada são obtidas diretamente, a partir do elemento sensor e do circuito de referência.



**Figura 2.5 Controle com Inferência Direta (1º Caso)**

No segundo caso, utiliza-se a inferência indireta pois as variáveis de entrada são obtidas indiretamente, através do cálculo do erro fundamental e do gradiente do erro.



**Figura 2.6 Controle com Inferência Indireta (2º Caso)**

### 2.4.1 Fuzzificação com Inferência Direta

Neste caso, as variáveis de entrada do sistema de controle são:  $Va$  valor atual, obtido a partir de um elemento sensor e  $Vd$  valor desejado, obtido através de um circuito de referência.

Estas variáveis podem ser digitalizadas e expressas no formato binário, usando apenas um byte sem bit de sinal, podendo variar de  $0.0$  a  $127.5$  com resolução mínima de  $0.5$  unidades.

A combinação binária  $(00000000)_2$  corresponde ao valor mínimo:  $0.0$ .

A combinação binária  $(11111111)_2$  corresponde ao valor máximo:  $127.5$ .

Considerando-se que as variáveis de entrada  $Va$  e  $Vd$  são representadas pelas variáveis cartesianas  $x$  e  $y$  e que os fatores  $f_x$  e  $f_y$  são iguais a  $64$ , deve-se então, estabelecer cinco termos lingüísticos para descrever cada variável de entrada com os respectivos pesos  $w_x$  e  $w_y$  associados a cada termo lingüístico, conforme indicado na *Tabela 2.4*.

*Tabela 2.4 Fuzzificação com Inferência Direta*

<i>Variável de Entrada Va ou Vd</i>	<i>Variável Cartesiana x ou y</i>	<i>Termo Lingüístico Va ou Vd</i>	<i>Peso Associado w<sub>x</sub> ou w<sub>y</sub></i>
[0:31.5]	[0:63]	<i>Muito Baixo</i>	<i>0</i>
[0:63.5]	[0:127]	<i>Baixo</i>	<i>1</i>
[32:95.5]	[64:191]	<i>Médio</i>	<i>2</i>
[64:127.5]	[128:255]	<i>Alto</i>	<i>3</i>
[96:127.5]	[192:255]	<i>Muito Alto</i>	<i>4</i>

Por exemplo, para as variáveis de entrada  $Va$  ou  $Vd$  iguais a  $50$ , estão ativos os termos lingüísticos *Baixo* e *Médio*, sendo que a variável cartesiana  $x$  ou  $y$  equivalente é igual a  $100$ .

O procedimento de *fuzzificação* determina quais são os pesos  $w_x$  ou  $w_y$  associados aos termos lingüísticos que estão ativos e seus respectivos índices de pertinência  $\mu_x$  ou  $\mu_y$ .

Através de uma simples operação de divisão ( $100 \div 64$ ) igual a  $1.5625$ , sabe-se que estão ativos os termos lingüísticos de peso  $w_a$  igual a  $1$ , com índice de pertinência  $\mu_a$  igual a  $0.4375$  e peso  $w_b$  igual a  $2$ , com índice de pertinência  $\mu_b$  igual a  $0.5625$ .

### 2.4.2 Fuzzificação com Inferência Indireta

Neste caso, as variáveis de entrada do sistema de controle são:  $E$  erro fundamental, dado pela diferença algébrica  $Vd-Va$  e  $\delta E$  gradiente do erro, dado pela diferença algébrica  $E_{(n)}-E_{(n-1)}$ .

Estas variáveis podem ser digitalizadas e expressas no formato binário, usando apenas um byte com um bit de sinal, podendo variar de  $-127$  a  $127$  com resolução mínima de  $1$  unidade.

A combinação binária  $(11111111)_2$  corresponde ao valor mínimo:  $-127$ .

A combinação binária  $(01111111)_2$  corresponde ao valor máximo:  $127$ .

Considerando-se que as variáveis de entrada  $E$  e  $\delta E$  são representadas pelas variáveis cartesianas  $x$  e  $y$  e que os fatores  $f_x$  e  $f_y$  são iguais a  $64$ , deve-se então, estabelecer cinco termos lingüísticos para descrever cada variável de entrada com os respectivos pesos  $w_x$  e  $w_y$  associados a cada termo lingüístico, conforme indicado na *Tabela 2.5*.

*Tabela 2.5 Fuzzificação com Inferência Indireta*

<i>Variável de Entrada E ou <math>\delta E</math></i>	<i>Variável Cartesiana x ou y</i>	<i>Termo Lingüístico E ou <math>\delta E</math></i>	<i>Peso Associado <math>w_x</math> ou <math>w_y</math></i>
$[-127:-64]$	$[-127:-64]$	<i>Muito Negativo</i>	$-2$
$[-127:0]$	$[-127:0]$	<i>Negativo</i>	$-1$
$[-63:63]$	$[-63:63]$	<i>Zero</i>	$0$
$[0:127]$	$[0:127]$	<i>Positivo</i>	$1$
$[64:127]$	$[64:127]$	<i>Muito Positivo</i>	$2$

Por exemplo, para as variáveis de entrada  $E$  ou  $\delta E$  iguais a  $-28$ , estão ativos os termos lingüísticos *Negativo* e *Zero*, sendo que a variável cartesiana  $x$  ou  $y$  equivalente é igual a  $-28$ .

O procedimento de *fuzzificação* determina quais são os pesos  $w_x$  ou  $w_y$  associados aos termos lingüísticos que estão ativos e seus respectivos índices de pertinência  $\mu_x$  ou  $\mu_y$ .

Através de uma simples operação de divisão  $(-28 \div 64)$  igual a  $-0,4375$ , sabe-se que estão ativos os termos lingüísticos de peso  $w_a$  igual a  $0$ , com índice de pertinência  $\mu_a$  igual a  $0.5625$  e peso  $w_b$  igual a  $-1$ , com índice de pertinência  $\mu_b$  igual a  $0.4375$ .

## 2.5 Procedimentos de Defuzzificação

Durante o processo de *defuzzificação*, procedimentos de inferência processam os dados nebulosos de entrada, junto com as regras de controle, de modo a inferir as ações de controle nebuloso, aplicando o operador de agregação adequado. A interface de *defuzzificação* transforma regras de controle nebulosas em ações de controle não-nebulosas.

Estas ações de controle derivam de um conjunto de regras de controle decorrentes de operações de inferência obtidas a partir de uma base de conhecimento consistente. Uma base de conhecimento consiste de um conjunto de regras de controle decorrentes de operações básicas de inferência que podem caracterizar uma estratégia de controle e suas metas.

Para dar início ao procedimento de *defuzzificação* é preciso estabelecer um conjunto de regras de controle nebuloso capaz de descrever com exatidão as ações de controle como uma função das variáveis de entrada  $x$  e  $y$  do controlador nebuloso.

O procedimento de *defuzzificação* apresentado na *Tabela 2.6*, permite determinar todos os valores da variável de saída  $z$  a partir dos pesos  $w_x$ ,  $w_y$  e  $w_z$  associados aos termos lingüísticos ativos e seus respectivos índices de pertinência  $\mu_x$ ,  $\mu_y$  e  $\mu_z$ .

Considerando o fator  $f_z$  igual a 32, através da equação (2.1) é possível calcular todos os valores atribuídos à variável de saída  $z$  que corresponde a uma ação de controle  $Ac$ , utilizada na composição da variável de controle  $Vc$ , conforme indicado na equação (2.2).

*Tabela 2.6 Defuzzificação da Variável de Saída*

<i>Ação de Controle</i> $Ac$	<i>Variável de Saída</i> $(z \div f_z)$	<i>Termo Lingüístico</i> $Ac$	<i>Peso Associado</i> $w_z$
[-127:-96]	[-3.96875:-3]	<i>Subtrair 4</i>	-4
[-127:-64]	[-2.96875:-2]	<i>Subtrair 3</i>	-3
[-95:-32]	[-1.96875:-1]	<i>Subtrair 2</i>	-2
[-63:0]	[-0.96875:0]	<i>Subtrair 1</i>	-1
[-31:31]	[-1.96875:1.96875]	<i>Manter</i>	0
[0:63]	[0:1.96875]	<i>Somar 1</i>	1
[32:95]	[1:1.96875]	<i>Somar 2</i>	2
[64:127]	[2:3.96875]	<i>Somar 3</i>	3
[96:127]	[3:3.96875]	<i>Somar 4</i>	4

### 2.5.1 Defuzzificação com Inferência Direta

Neste caso, o conjunto de regras e ações de controle é definido a partir das variáveis de entrada  $Va$  e  $Vd$ . A informação apresentada na **Tabela 2.7** corresponde ao conjunto de termos lingüísticos associados às variáveis de entrada  $Va$  e  $Vd$  e às ações de controle  $Ac$ , resultantes da aplicação de todas as combinações de entrada possíveis.

**Tabela 2.7 Termos Lingüísticos com Inferência Direta**

Valor Atual $Va$	$w_x$	Valor Desejado $Vd$	$w_y$	Ação de Controle $Ac$	$w_z$
Muito Baixo	0	Muito Baixo	0	Manter	0
Muito Baixo	0	Baixo	1	Somar 1	1
Muito Baixo	0	Médio	2	Somar 2	2
Muito Baixo	0	Alto	3	Somar 3	3
Muito Baixo	0	Muito Alto	4	Somar 4	4
Baixo	1	Muito Baixo	0	Subtrair 1	-1
Baixo	1	Baixo	1	Manter	0
Baixo	1	Médio	2	Somar 1	1
Baixo	1	Alto	3	Somar 2	2
Baixo	1	Muito Alto	4	Somar 3	3
Médio	2	Muito Baixa	0	Subtrair 2	-2
Médio	2	Baixo	1	Subtrair 1	-1
Médio	2	Médio	2	Manter	0
Médio	2	Alto	3	Somar 1	1
Médio	2	Muito Alto	4	Somar 2	2
Alto	3	Muito Baixo	0	Subtrair 3	-3
Alto	3	Baixo	1	Subtrair 2	-2
Alto	3	Médio	2	Subtrair 1	-1
Alto	3	Alto	3	Manter	0
Alto	3	Muito Alto	4	Somar 1	1
Muito Alto	4	Muito Baixo	0	Subtrair 4	-4
Muito Alto	4	Baixo	1	Subtrair 3	-3
Muito Alto	4	Médio	2	Subtrair 2	-2
Muito Alto	4	Alto	3	Subtrair 1	-1
Muito Alto	4	Muito Alto	4	Manter	0

**Muito Baixo** → **MB**

**Baixo** → **B**

**Médio** → **M**

**Alto** → **A**

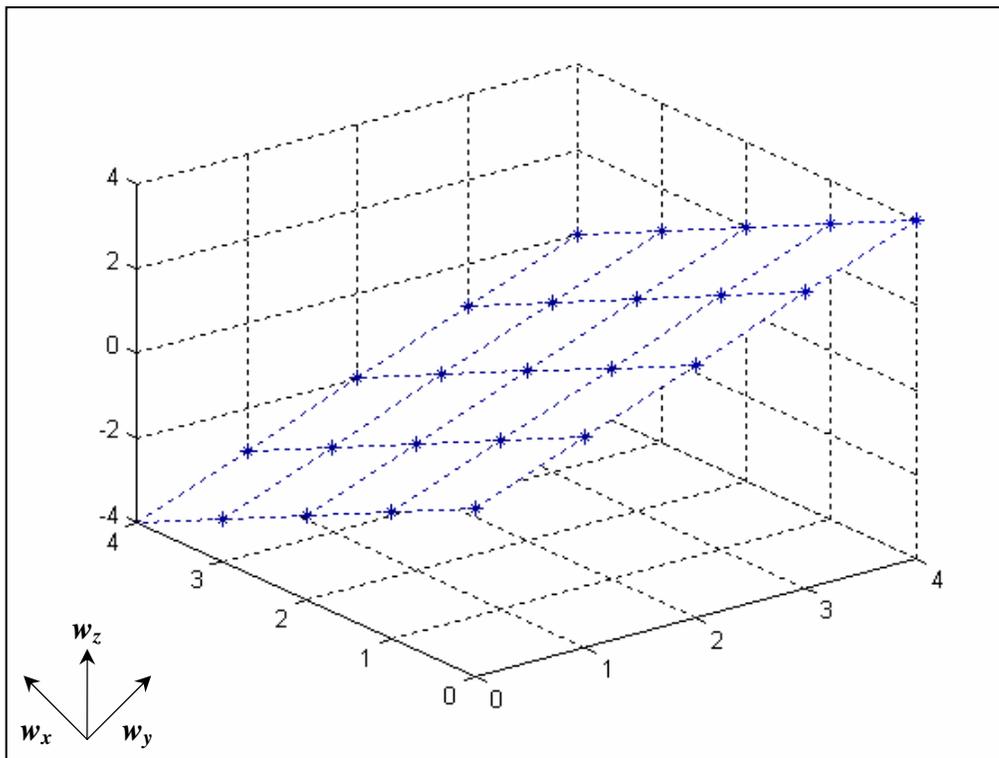
**Muito Alto** → **MA**

Conforme a **Tabela 2.8**, todos os pesos associados às ações de controle com inferência direta podem ser mapeados através da representação gráfica indicada na **Figura 2.7**.

**Tabela 2.8 Ações de Controle com Inferência Direta**

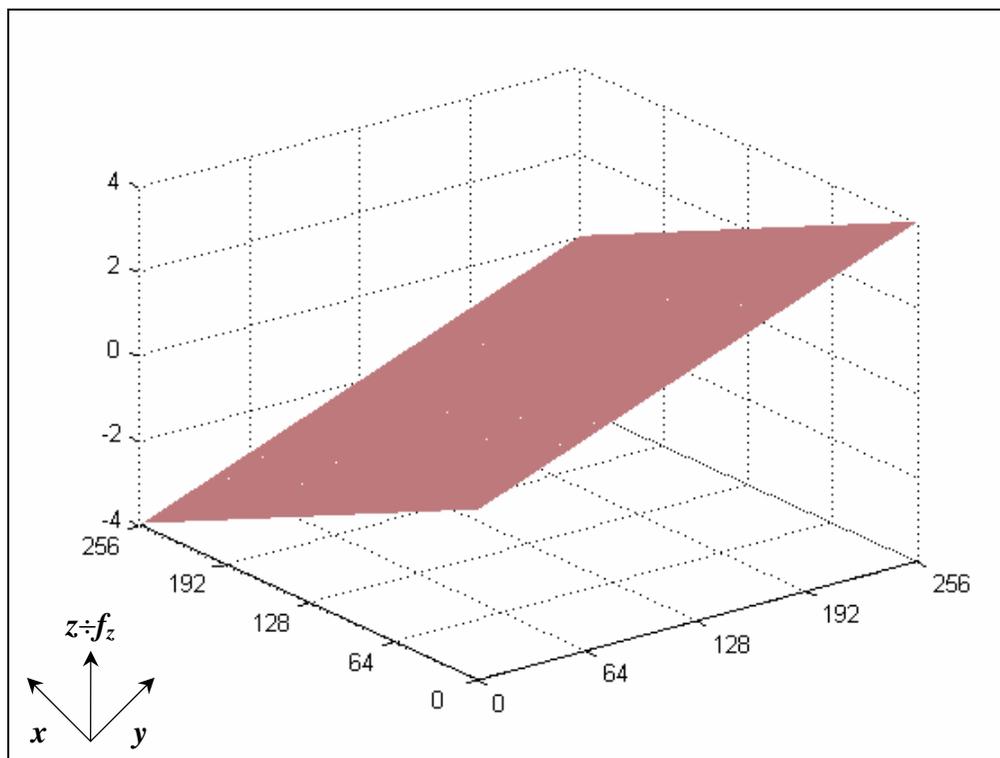
<b>Ac</b>			<b>Vd</b>					
			<b>y</b>	<b>0</b>	<b>64</b>	<b>128</b>	<b>192</b>	<b>256</b>
			<b>w<sub>y</sub></b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>Va</b>	<b>x</b>	<b>w<sub>x</sub></b>	<b>w<sub>z</sub></b>	<b>MB</b>	<b>B</b>	<b>M</b>	<b>A</b>	<b>MA</b>
	<b>0</b>	<b>0</b>	<b>MB</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
	<b>64</b>	<b>1</b>	<b>B</b>	<b>-1</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>
	<b>128</b>	<b>2</b>	<b>M</b>	<b>-2</b>	<b>-1</b>	<b>0</b>	<b>1</b>	<b>2</b>
	<b>192</b>	<b>3</b>	<b>A</b>	<b>-3</b>	<b>-2</b>	<b>-1</b>	<b>0</b>	<b>1</b>
	<b>256</b>	<b>4</b>	<b>MA</b>	<b>-4</b>	<b>-3</b>	<b>-2</b>	<b>-1</b>	<b>0</b>

**Controle Nebuloso com Função Linear**

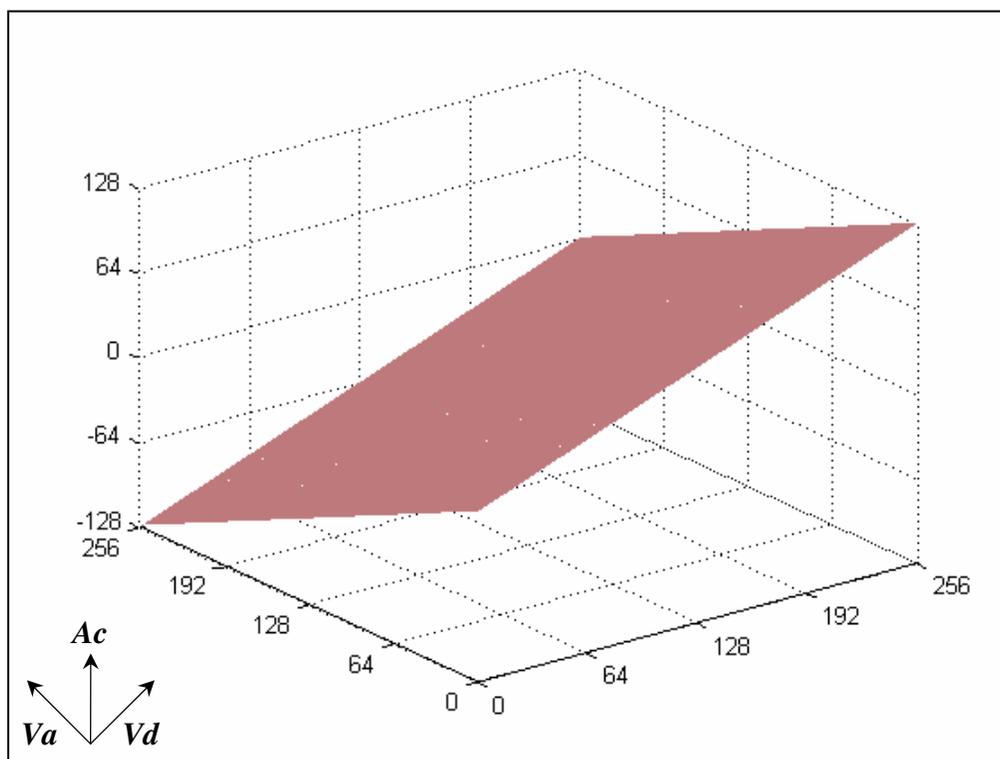


**Figura 2.7 Gráfico dos Pesos com Função Linear (1º Caso)**

Os valores exatos da variável de saída ( $z \neq f_z$ ) e das ações de controle **Ac**, previamente calculados e armazenados em **64 kbytes** de memória, podem ser mapeados pela representação gráfica indicada na **Figura 2.8** e **Figura 2.9**, respectivamente. Um valor atribuído à variável de saída  $z$  corresponde a uma ação de controle **Ac** usada na composição da variável de controle **Vc**.



*Figura 2.8 Variáveis Cartesianas com Função Linear (1º Caso)*



*Figura 2.9 Ações de Controle com Função Linear (1º Caso)*

### 2.5.2 Defuzzificação com Inferência Indireta

Neste caso, o conjunto de regras e ações de controle é definido a partir das variáveis de entrada  $E$  e  $\delta E$ . A informação apresentada na **Tabela 2.9** corresponde ao conjunto de termos lingüísticos associados às variáveis de entrada  $E$  e  $\delta E$  e às ações de controle  $Ac$ , resultantes da aplicação de todas as combinações de entrada possíveis.

**Tabela 2.9 Termos Lingüísticos com Inferência Indireta**

<b>Erro Fundamental</b> $E$	$w_x$	<b>Gradiente do Erro</b> $\delta E$	$w_y$	<b>Ação de Controle</b> $Ac$	$w_z$
<i>Muito Negativo</i>	-2	<i>Muito Negativo</i>	-2	<i>Subtrair 4</i>	-4
<i>Muito Negativo</i>	-2	<i>Negativo</i>	-1	<i>Subtrair 3</i>	-3
<i>Muito Negativo</i>	-2	<i>Zero</i>	0	<i>Subtrair 2</i>	-2
<i>Muito Negativo</i>	-2	<i>Positivo</i>	1	<i>Subtrair 1</i>	-1
<i>Muito Negativo</i>	-2	<i>Muito Positivo</i>	2	<i>Manter</i>	0
<i>Negativo</i>	-1	<i>Muito Negativo</i>	-2	<i>Subtrair 3</i>	-3
<i>Negativo</i>	-1	<i>Negativo</i>	-1	<i>Subtrair 2</i>	-2
<i>Negativo</i>	-1	<i>Zero</i>	0	<i>Subtrair 1</i>	-1
<i>Negativo</i>	-1	<i>Positivo</i>	1	<i>Manter</i>	0
<i>Negativo</i>	-1	<i>Muito Positivo</i>	2	<i>Somar 1</i>	1
<i>Zero</i>	0	<i>Muito Negativo</i>	-2	<i>Subtrair 2</i>	-2
<i>Zero</i>	0	<i>Negativo</i>	-1	<i>Subtrair 1</i>	-1
<i>Zero</i>	0	<i>Zero</i>	0	<i>Manter</i>	0
<i>Zero</i>	0	<i>Positivo</i>	1	<i>Somar 1</i>	1
<i>Zero</i>	0	<i>Muito Positivo</i>	2	<i>Somar 2</i>	2
<i>Positivo</i>	1	<i>Muito Negativo</i>	-2	<i>Subtrair 1</i>	-1
<i>Positivo</i>	1	<i>Negativo</i>	-1	<i>Manter</i>	0
<i>Positivo</i>	1	<i>Zero</i>	0	<i>Somar 1</i>	1
<i>Positivo</i>	1	<i>Positivo</i>	1	<i>Somar 2</i>	2
<i>Positivo</i>	1	<i>Muito Positivo</i>	2	<i>Somar 3</i>	3
<i>Muito Positivo</i>	2	<i>Muito Negativo</i>	-2	<i>Manter</i>	0
<i>Muito Positivo</i>	2	<i>Negativo</i>	-1	<i>Somar 1</i>	1
<i>Muito Positivo</i>	2	<i>Zero</i>	0	<i>Somar 2</i>	2
<i>Muito Positivo</i>	2	<i>Positivo</i>	1	<i>Somar 3</i>	3
<i>Muito Positivo</i>	2	<i>Muito Positivo</i>	2	<i>Somar 4</i>	4

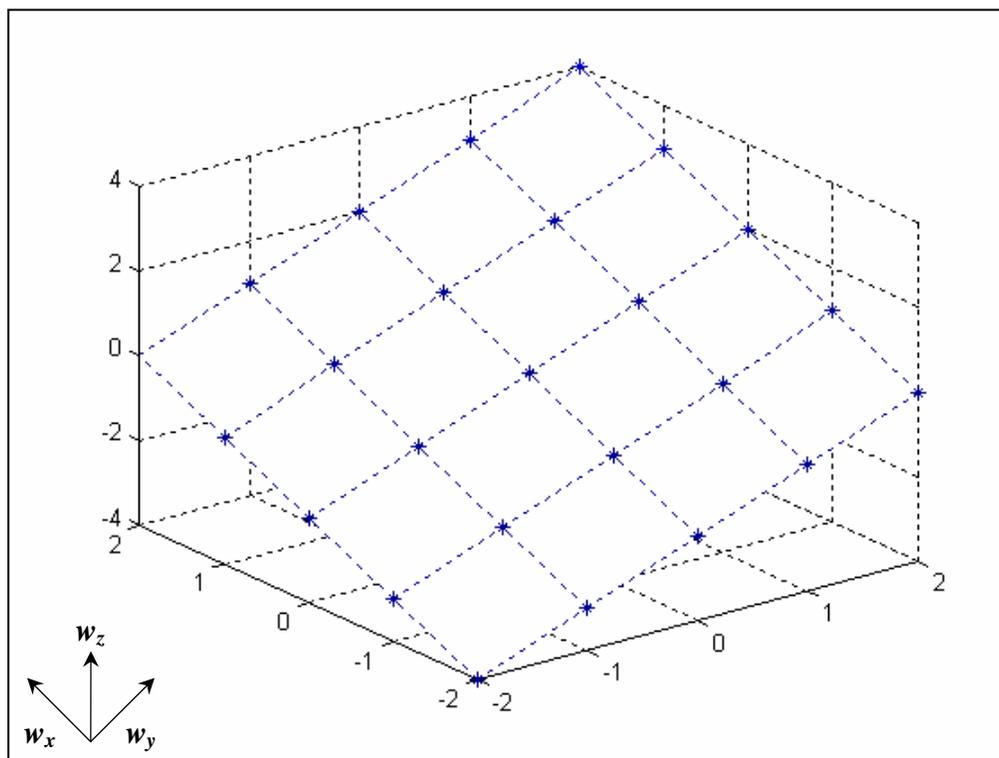
*Muito Negativo* → *MN*  
*Negativo* → *N*  
*Zero* → *Z*  
*Positivo* → *P*  
*Muito Positivo* → *MP*

Conforme a **Tabela 2.10** todos os pesos associados às ações de controle com inferência indireta podem ser mapeados através da representação gráfica indicada na **Figura 2.10**.

**Tabela 2.10 Ações de Controle com Inferência Indireta**

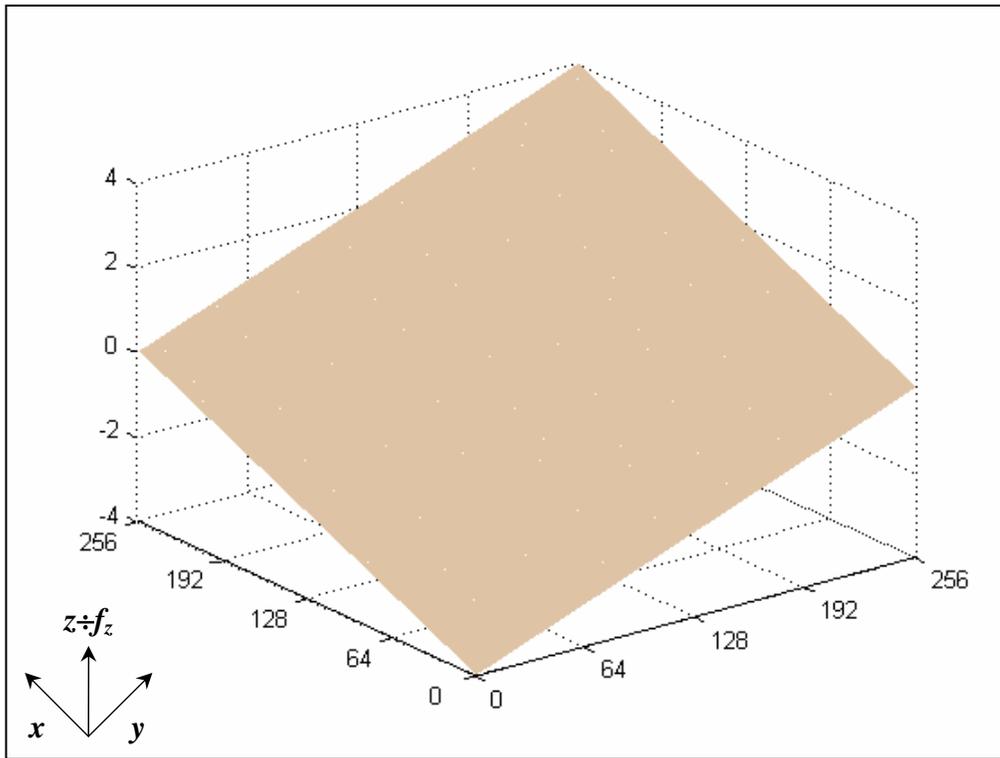
$Ac$			$\delta E$					
			$y$	$-128$	$-64$	$0$	$64$	$128$
			$w_y$	$-2$	$-1$	$0$	$1$	$2$
$E$	$x$	$w_x$	$w_z$	$MN$	$N$	$Z$	$P$	$MP$
	$-128$	$-2$	$MN$	$-4$	$-3$	$-2$	$-1$	$0$
	$-64$	$-1$	$N$	$-3$	$-2$	$-1$	$0$	$1$
	$0$	$0$	$Z$	$-2$	$-1$	$0$	$1$	$2$
	$64$	$1$	$P$	$-1$	$0$	$1$	$2$	$3$
$128$	$2$	$MP$	$0$	$1$	$2$	$3$	$4$	

**Controle Nebuloso com Função Linear**

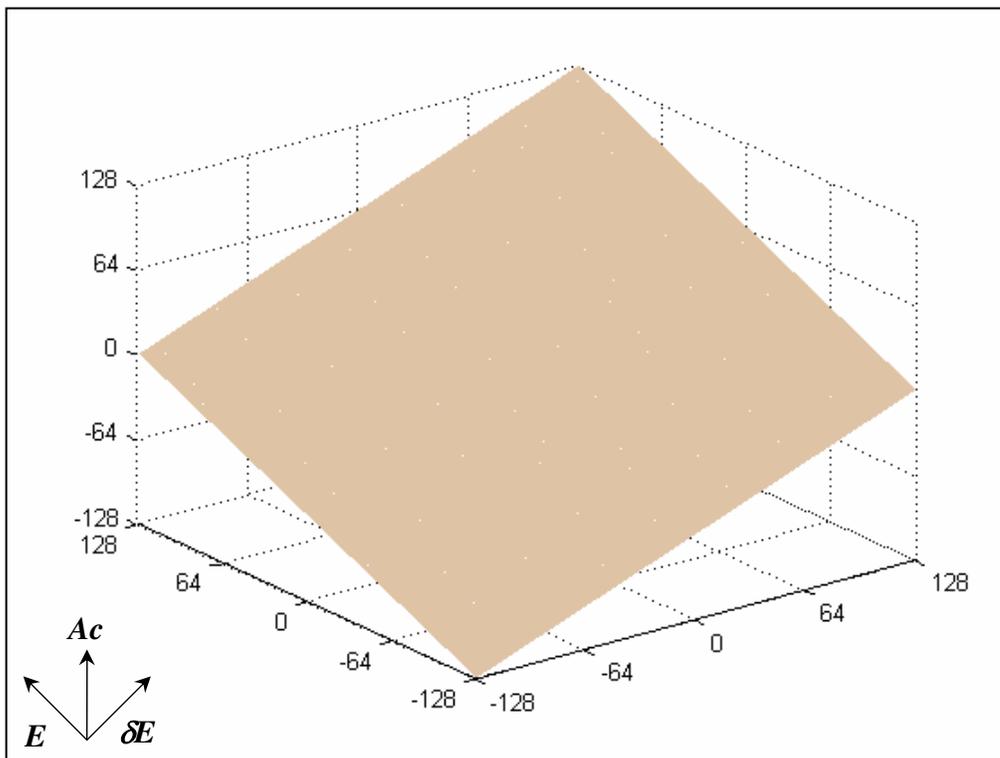


**Figura 2.10 Gráfico dos Pesos com Função Linear (2º Caso)**

Os valores exatos da variável de saída ( $z=f_z$ ) e das ações de controle  $Ac$ , previamente calculados e armazenados em **64 kbytes** de memória, podem ser mapeados pela representação gráfica indicada na **Figura 2.11** e **Figura 2.12**, respectivamente. O valor atribuído à variável de saída  $z$  corresponde a uma ação de controle  $Ac$  usada na composição da variável de controle  $Vc$ .



*Figura 2.11 Variáveis Cartesianas com Função Linear (2º Caso)*



*Figura 2.12 Ações de Controle com Função Linear (2º Caso)*

## 2.6 Simulação Computacional

Para simular uma aplicação prática do controlador nebuloso de arquitetura simplificada, foi usado um aplicativo computacional elaborado com o programa *MATLAB™*. Este aplicativo computacional inclui detalhes específicos de um sistema de controle digital implementado com circuitos integrados microcontroladores de 8 bits da família *Intel-8051*.

As operações lógicas e aritméticas são efetuadas considerando as limitações impostas pela arquitetura de 8 bits dos microcontroladores da família *Intel-8051*. Assim, a estrutura básica do controlador nebuloso representada genericamente por uma função do tipo  $z=f(x,y)$ , pode ser implementada através de memórias semicondutoras com capacidade para armazenar 64 kbytes de informação, utilizando um barramento de endereço de 16 bits.

O endereçamento da memória é feito diretamente através das variáveis de entrada  $x$  e  $y$ , codificadas usando um byte para cada variável, segundo a representação binária apresentada na *Tabela 2.1*. A variável de entrada  $x$  fornece o byte menos significativo e a variável de entrada  $y$  fornece o byte mais significativo. Os valores atribuídos à variável de saída  $z$  são representados pela informação binária armazenada na memória endereçada por  $x$  e  $y$ .

Considerando que a variável de saída  $z$  pode assumir valores positivos ou negativos, a variável de controle  $Vc$  depende do comportamento dinâmico da variável de saída  $z=f(x,y)$ . Como cada valor da variável de saída  $z$  corresponde a uma ação de controle  $Ac$ , usada na composição da variável de controle  $Vc$ , é possível implementar um controlador nebuloso com ação de controle do tipo *PI* (Proporcional Integral) ou *PID* (Proporcional Integral Derivativa).

Como uma ação de controle em malha fechada se resume a uma simples operação de acesso à memória, o tempo de amostragem não é limitado pelo tempo de execução do algoritmo de controle, pois o mesmo pode ser executado na mesma frequência de operação do sistema de aquisição de dados ou dos conversores analógico digital e digital analógico.

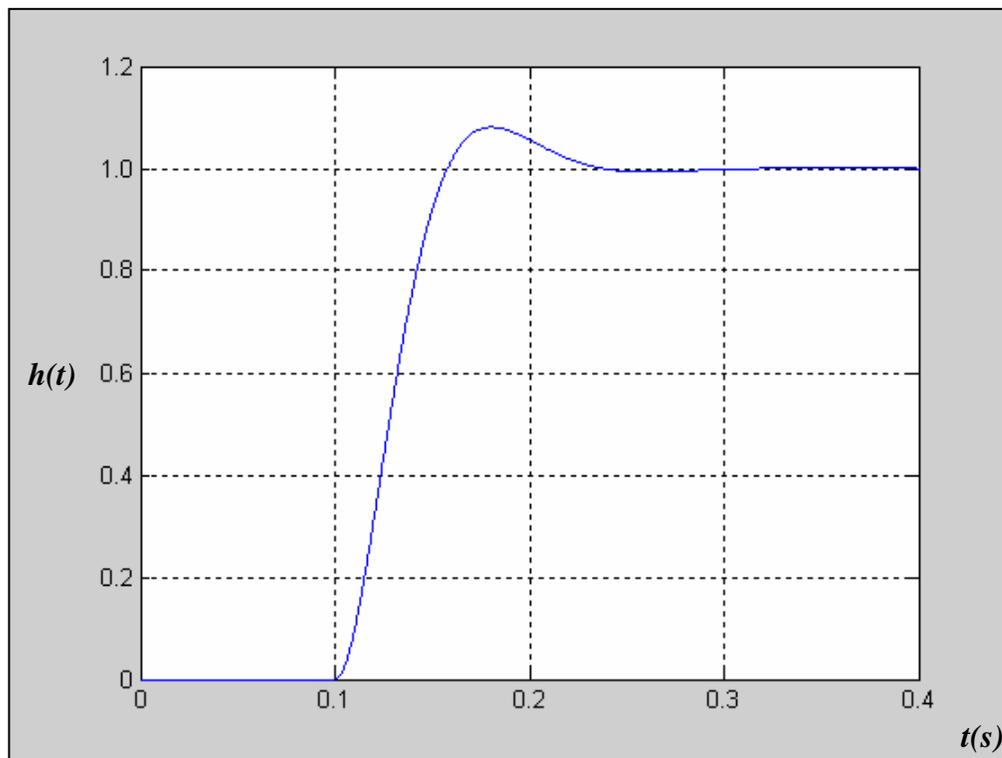
Para demonstrar a funcionalidade deste controlador, foram levantados alguns resultados de aplicação, obtidos através de simulações feitas em microcomputador tendo como parâmetro de referência uma função de transferência de segunda ordem. Durante a simulação computacional, o controlador nebuloso opera com 1.0ms de tempo de amostragem, no entanto, este controlador pode operar com tempo de amostragem de 100µs.

### 2.6.1 Resultados de Simulação Computacional

A equação (2.3) mostra a função de transferência  $h(S)$  usada nas simulações. Trata-se de um sistema de segunda ordem com parâmetros  $\omega_n=100$  e  $2\zeta\omega_n=125$  devidamente ajustados para permitir uma operação em malha fechada com tempo de amostragem igual a  $1.0ms$ .

$$h(S) = \frac{\omega_n^2}{S^2 + 2\zeta\omega_n S + \omega_n^2} \quad (2.3)$$

A **Figura 2.14** apresenta a resposta do sistema de segunda ordem  $h(t)$  para uma entrada do tipo degrau unitário aplicada no instante de  $100ms$  com constante de amortecimento  $\zeta=0.625$ , tempo de estabilização de  $150ms$  e sobre-sinal menor que  $10\%$  do valor de regime permanente.

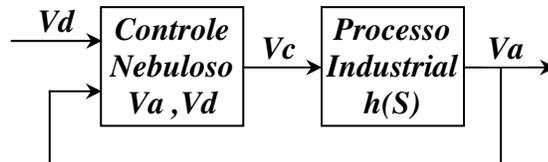


**Figura 2.13** Resposta do Sistema de Segunda Ordem  $h(t)$

Apresentam-se a seguir os resultados de simulação utilizando o programa **MATLAB™**. Os gráficos mostram a resposta do sistema e as ações de controle correspondentes para ambos os casos e diferentes valores da constante proporcional  $Kp$ , conforme a equação (2.2).

### 2.6.2 Controle Nebuloso com Inferência Direta

Neste caso, as variáveis de entrada do controlador nebuloso são:  $Va$  valor atual, obtido através de um elemento sensor e  $Vd$  valor desejado, obtido através de um circuito de referência, caracterizando um sistema de controle do tipo *PI*, conforme indicado na *Figura 2.14*.



*Figura 2.14 Controle com Inferência Direta (1º Caso)*

Apresentam-se a seguir os resultados obtidos através de simulação computacional. Neste caso, o erro de regime corresponde à resolução mínima de **0.5** unidades, conforme *Tabela 2.4*.

A *Figura 2.15* apresenta a resposta do sistema operando em malha fechada utilizando controle nebuloso com inferência direta e constante proporcional  $Kp=1.0$ . O tempo de resposta do sistema é menor que **100ms** e o máximo sobre-sinal é maior que **10%** do valor desejado  $Vd$ .

A *Figura 2.16* mostra as ações de controle decorrentes da operação em malha fechada considerando uma constante proporcional  $Kp=1.0$ . Assim, para  $Va=0$  e  $Vd=64$  corresponde uma ação de controle  $Ac=1.0$  e para  $Va=32$  e  $Vd=64$  corresponde uma ação de controle  $Ac=0.5$ .

A *Figura 2.17* apresenta a resposta do sistema operando em malha fechada utilizando controle nebuloso com inferência direta e constante proporcional  $Kp=2.0$ . O tempo de resposta do sistema é menor que **100ms** e o máximo sobre-sinal é maior que **20%** do valor desejado  $Vd$ .

A *Figura 2.18* mostra as ações de controle decorrentes da operação em malha fechada considerando uma constante proporcional  $Kp=2.0$ . Assim, para  $Va=0$  e  $Vd=64$  corresponde uma ação de controle  $Ac=2.0$  e para  $Va=32$  e  $Vd=64$  corresponde uma ação de controle  $Ac=1.0$ .

A *Figura 2.19* apresenta a resposta do sistema operando em malha fechada utilizando controle nebuloso com inferência direta e constante proporcional  $Kp=3.0$ . O tempo de resposta do sistema é menor que **100ms** e o máximo sobre-sinal é maior que **30%** do valor desejado  $Vd$ .

A *Figura 2.20* mostra as ações de controle decorrentes da operação em malha fechada considerando uma constante proporcional  $Kp=3.0$ . Assim, para  $Va=0$  e  $Vd=64$  corresponde uma ação de controle  $Ac=3.0$  e para  $Va=32$  e  $Vd=64$  corresponde uma ação de controle  $Ac=1.5$ .

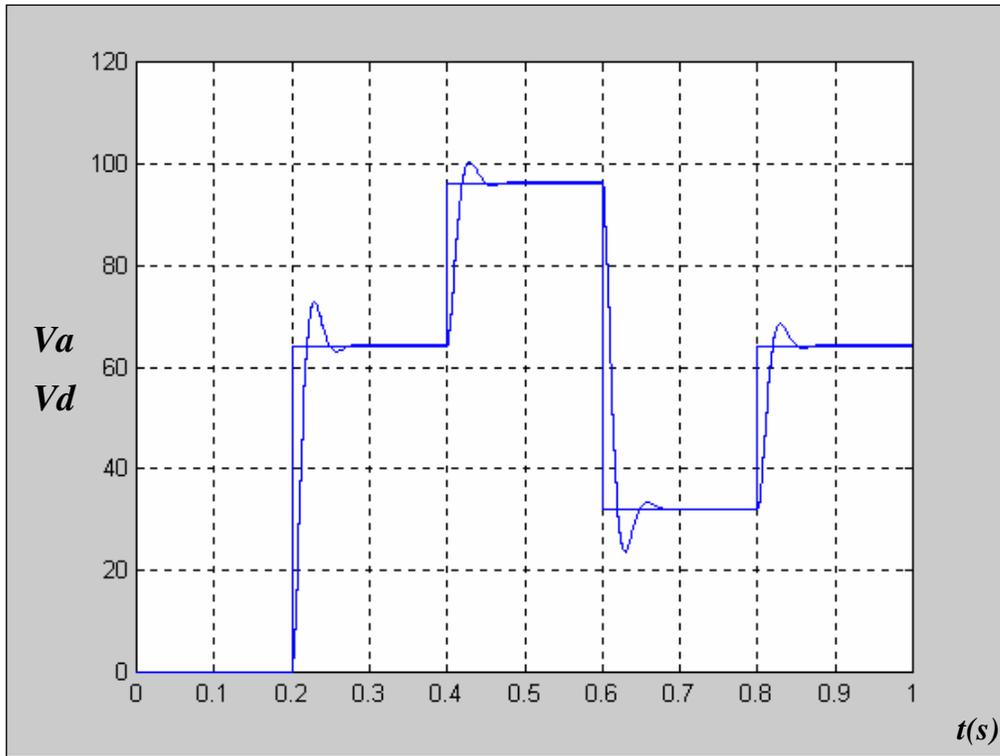


Figura 2.15 Resposta do Sistema (1º Caso,  $K_p=1.0$ )

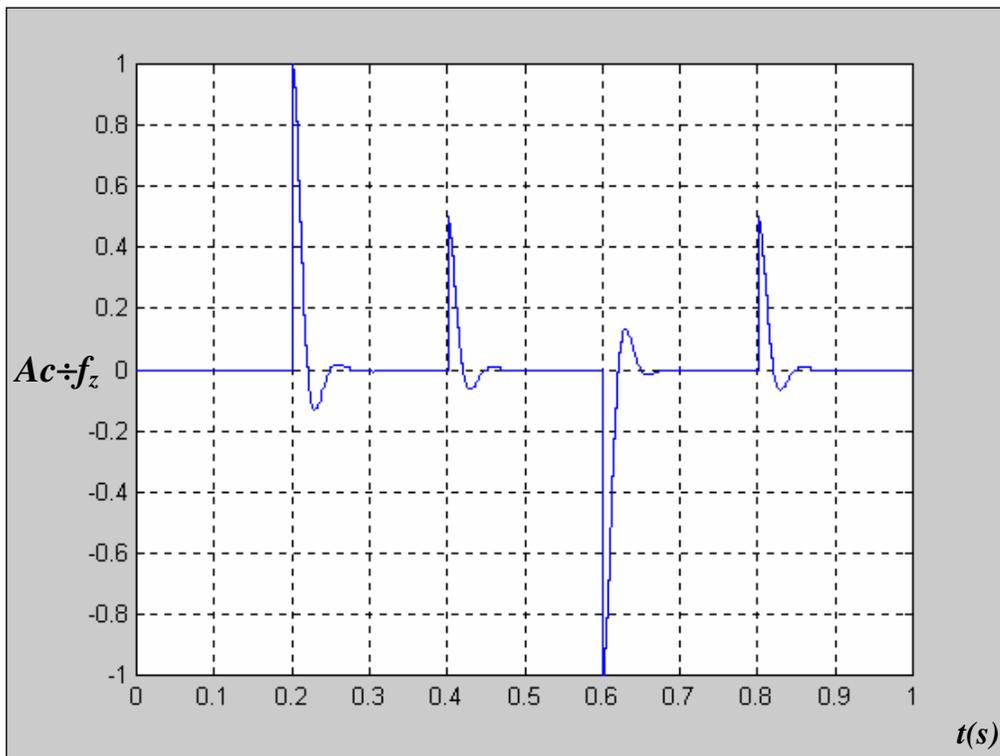


Figura 2.16 Ações de Controle (1º Caso,  $K_p=1.0$ )

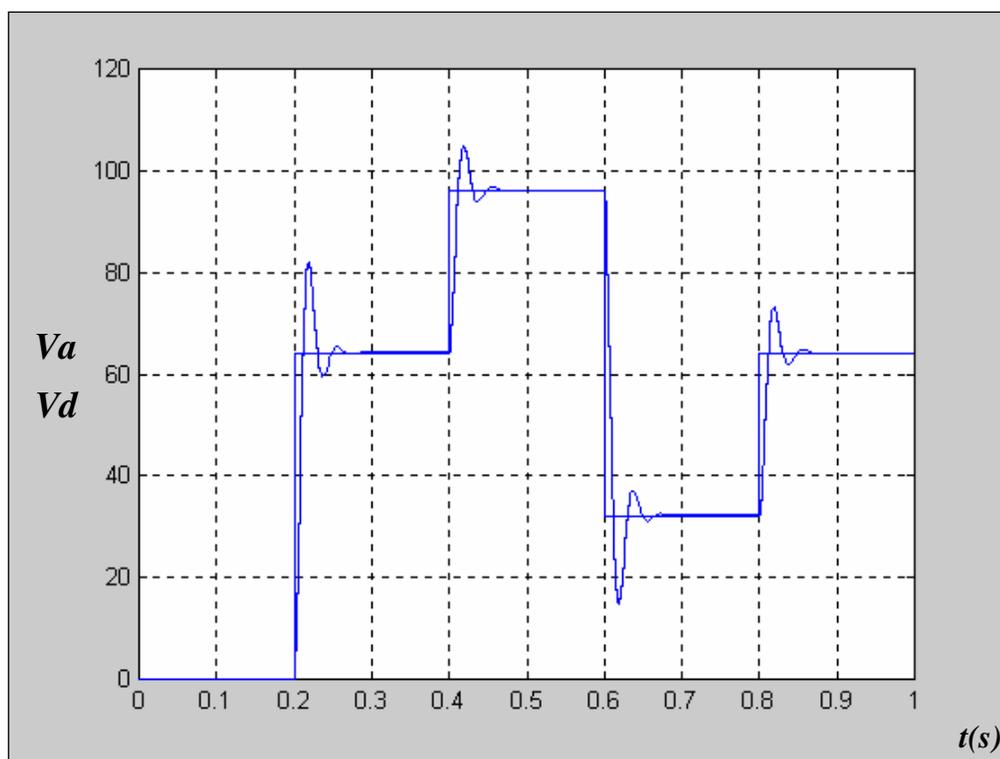


Figura 2.17 Resposta do Sistema (1<sup>o</sup> Caso,  $K_p=2.0$ )

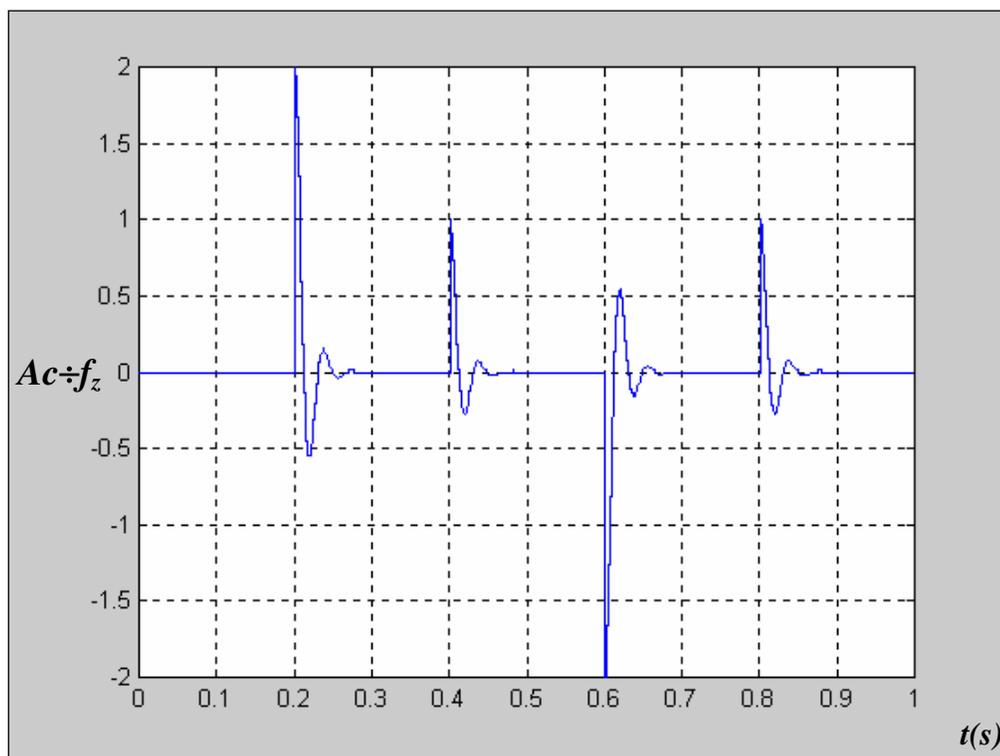


Figura 2.18 Ações de Controle (1<sup>o</sup> Caso,  $K_p=2.0$ )

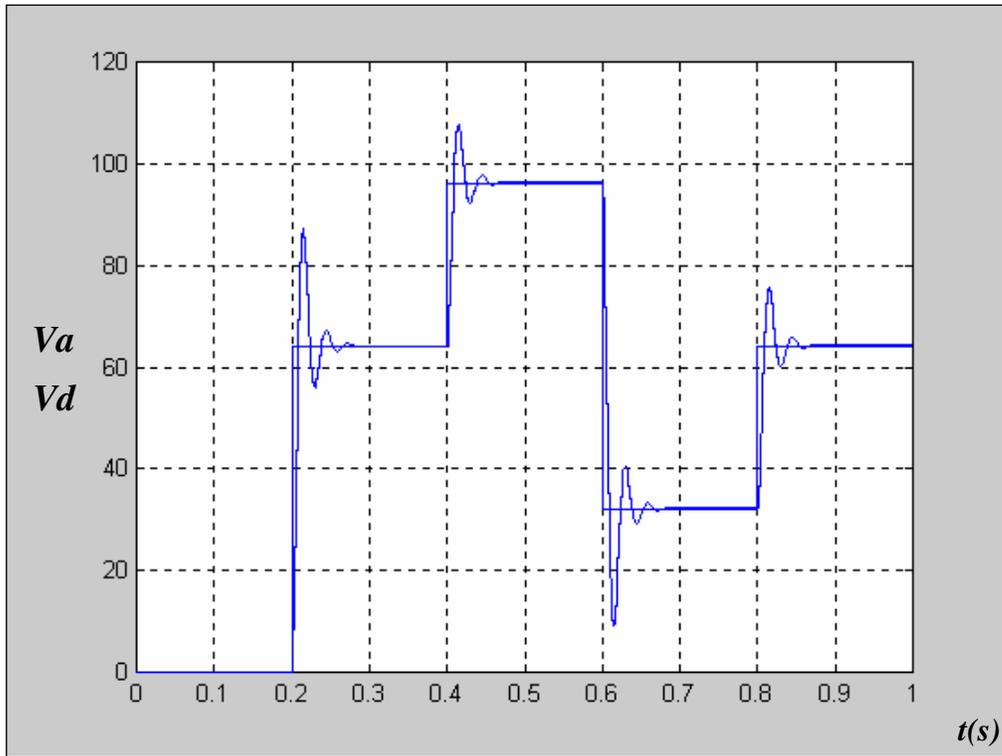


Figura 2.19 Resposta do Sistema (1º Caso,  $K_p=3.0$ )

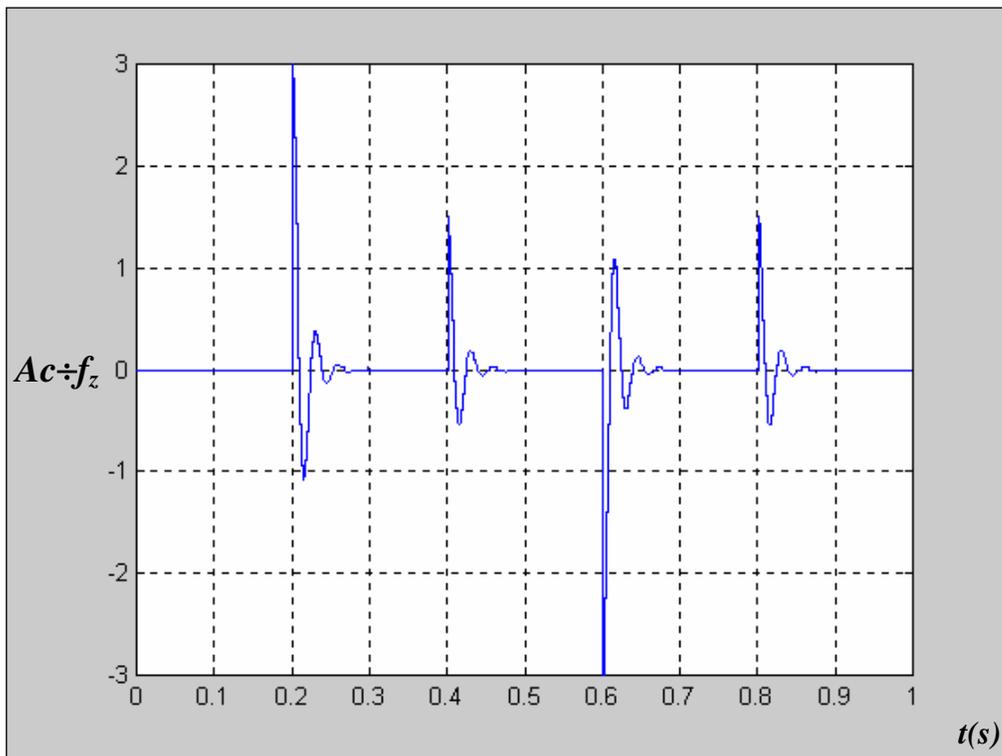
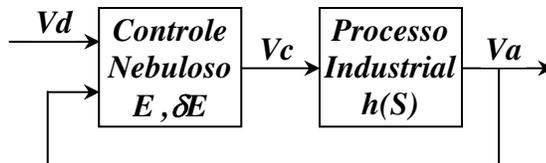


Figura 2.20 Ações de Controle (1º Caso,  $K_p=3.0$ )

### 2.6.3 Controle Nebuloso com Inferência Indireta

Neste caso, as variáveis de entrada do controle nebuloso são:  $E$  erro fundamental, dado pela diferença algébrica  $Vd-Va$  e  $\delta E$  gradiente do erro, dado pela diferença algébrica  $E_{(n)}-E_{(n-1)}$ , caracterizando um sistema de controle do tipo  $PID$ , conforme indicado na **Figura 2.21**.



**Figura 2.21** Controle com Inferência Indireta (2º Caso)

Apresentam-se a seguir os resultados obtidos através de simulação computacional. Neste caso, o erro de regime corresponde à resolução mínima de  $1$  unidade, conforme **Tabela 2.5**.

A **Figura 2.22** apresenta a resposta do sistema operando em malha fechada utilizando controle nebuloso com inferência indireta e constante proporcional  $Kp=0.5$ . O tempo de resposta do sistema é menor que  $100ms$  e o máximo sobre-sinal é menor que  $1\%$  do valor desejado  $Vd$ .

A **Figura 2.23** mostra as ações de controle decorrentes da operação em malha fechada considerando uma constante proporcional  $Kp=0.5$ . Assim, para  $E=64$  e  $\delta E=64$  corresponde uma ação de controle  $Ac=1.0$  e para  $E=32$  e  $\delta E=32$  corresponde uma ação de controle  $Ac=0.5$ .

A **Figura 2.24** apresenta a resposta do sistema operando em malha fechada utilizando controle nebuloso com inferência direta e constante proporcional  $Kp=1.0$ . O tempo de resposta do sistema é menor que  $100ms$  e o máximo sobre-sinal é maior que  $5\%$  do valor desejado  $Vd$ .

A **Figura 2.25** mostra as ações de controle decorrentes da operação em malha fechada considerando uma constante proporcional  $Kp=1.0$ . Assim, para  $E=64$  e  $\delta E=64$  corresponde uma ação de controle  $Ac=2.0$  e para  $E=32$  e  $\delta E=32$  corresponde uma ação de controle  $Ac=1.0$ .

A **Figura 2.26** apresenta a resposta do sistema operando em malha fechada utilizando controle nebuloso com inferência direta e constante proporcional  $Kp=1.5$ . O tempo de resposta do sistema é menor que  $100ms$  e o máximo sobre-sinal é maior que  $10\%$  do valor desejado  $Vd$ .

A **Figura 2.27** mostra as ações de controle decorrentes da operação em malha fechada considerando uma constante proporcional  $Kp=1.5$ . Assim, para  $E=64$  e  $\delta E=64$  corresponde uma ação de controle  $Ac=3.0$  e para  $E=32$  e  $\delta E=32$  corresponde uma ação de controle  $Ac=1.5$ .

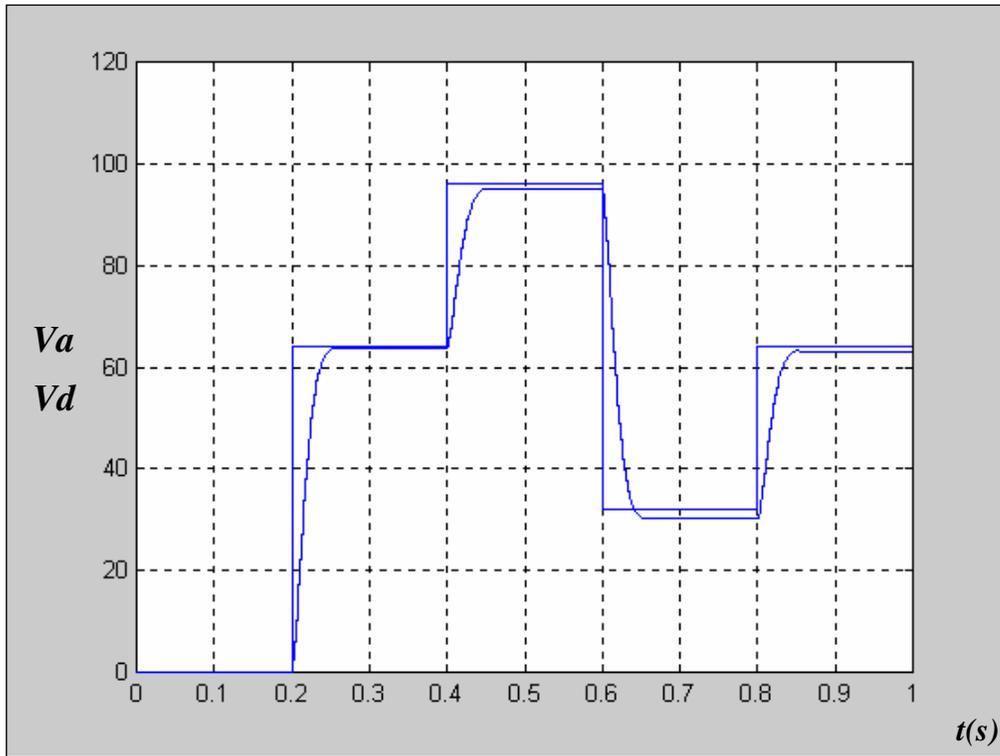


Figura 2.22 Resposta do Sistema (2º Caso,  $K_p=0.5$ )

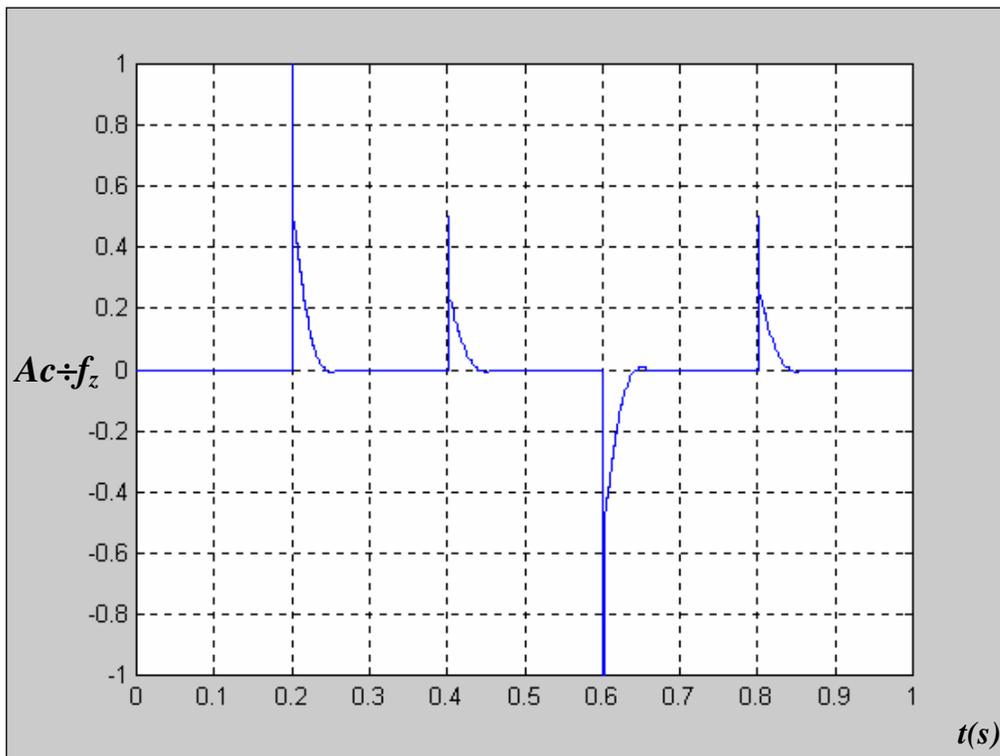


Figura 2.23 Ações de Controle (2º Caso,  $K_p=0.5$ )

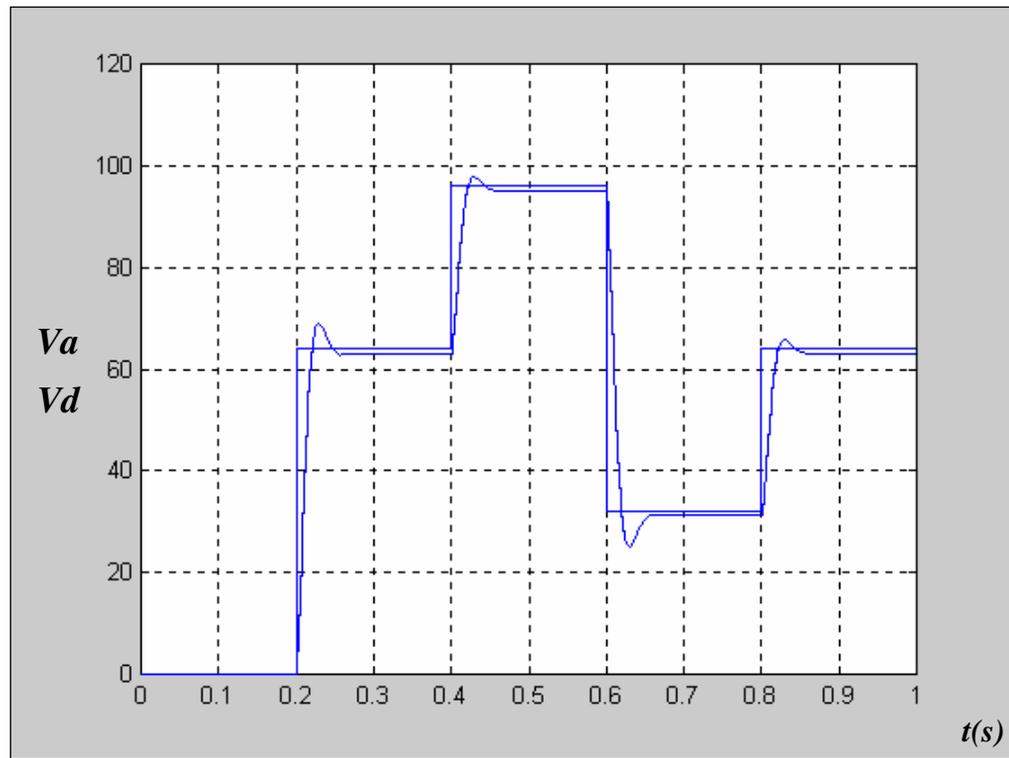


Figura 2.24 Resposta do Sistema (2<sup>a</sup> Caso,  $K_p=1.0$ )

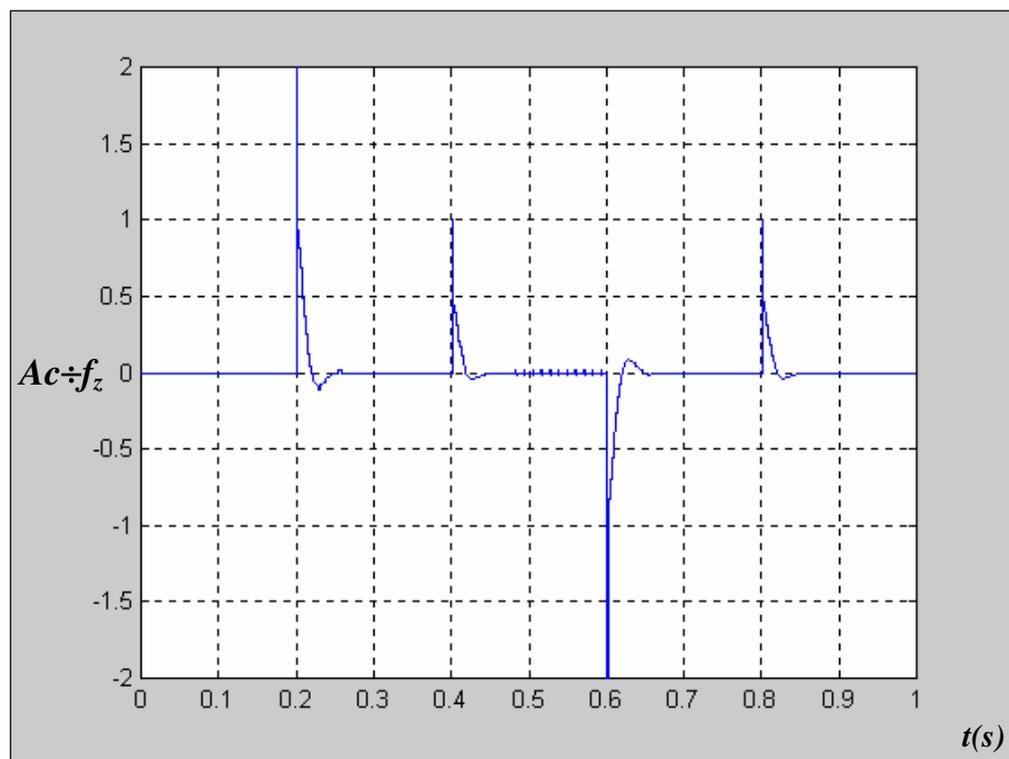


Figura 2.25 Ações de Controle (2<sup>a</sup> Caso,  $K_p=1.0$ )

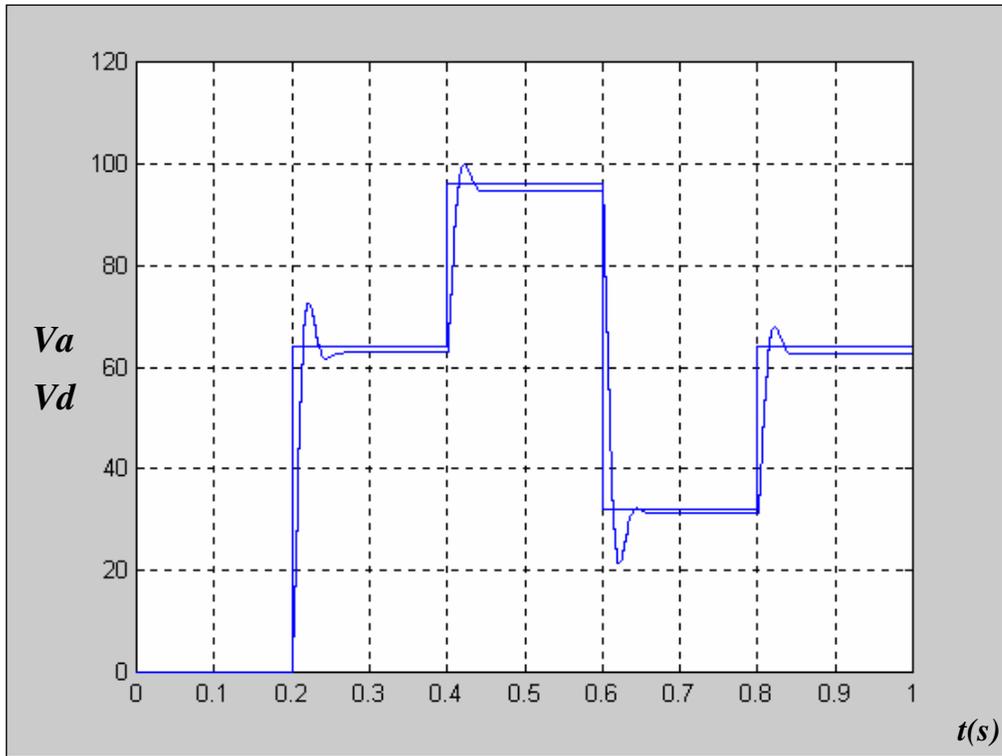


Figura 2.26 Resposta do Sistema (2º Caso,  $K_p=1.5$ )

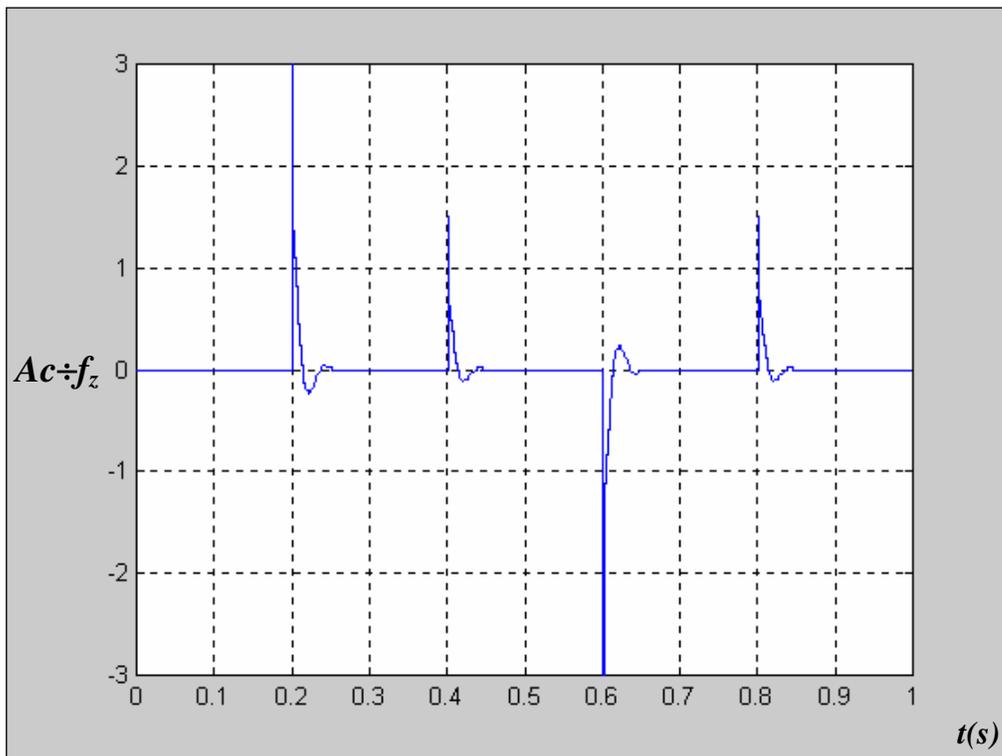


Figura 2.27 Ações de Controle (2º Caso,  $K_p=1.5$ )

## 2.7 Controlador Neuro-Nebuloso

Atualmente, a possibilidade de manusear sistemas de controle complexos, com auxílio de controladores baseados em lógica nebulosa, combinados com as redes neurais artificiais que possuem capacidade de treinamento e adaptação, tem propiciado o desenvolvimento de novos sistemas e controladores neuro-nebulosos. Este trabalho apresenta uma estrutura simplificada de um controlador nebuloso, com possibilidade real de combinação com uma rede neural artificial baseada no **CMAC** (Cerebellar Model Articulation Controller) (ALBUS, 1975).

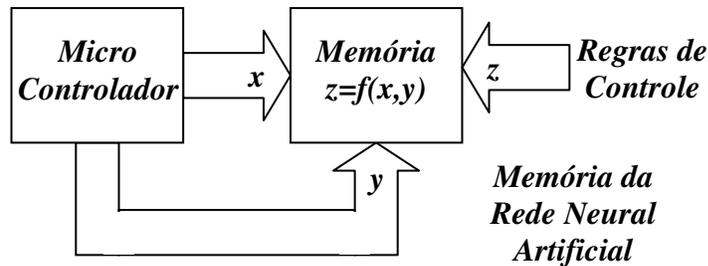
Considerando que os procedimentos de *fuzzificação* e *defuzzificação* são executados antecipadamente, então, o controlador neuro-nebuloso deve operar com informações previamente armazenadas em memória, deste modo, a determinação de uma ação de controle se resume a uma simples consulta de dados armazenados em memória. Portanto, a arquitetura deste controlador admite o uso de uma rede neural artificial capaz de armazenar em memória, as informações relativas às ações de controle. A estrutura mais adequada é uma rede neural baseada no **CMAC**, capaz de armazenar dados numa estrutura do tipo  $z=f(x,y)$  (KRAFT & CAMPAGNA, 1990).

Para que a rede neural do tipo **CMAC** possa armazenar dados em uma estrutura do tipo  $z=f(x,y)$  é necessário considerar uma abordagem discreta para representação dos pesos utilizados no armazenamento de dados, visando uma redução da memória física requerida pela rede neural. Considerando a representação binária padrão, usada no controlador nebuloso apresentado neste trabalho, é possível implementar uma rede neural do tipo **CMAC** com pesos discretos, obtendo um controlador neuro-nebuloso que combina a inferência da lógica nebulosa com a capacidade de treinamento e adaptação das redes neurais artificiais (BORDON & TEIXEIRA, 1995).

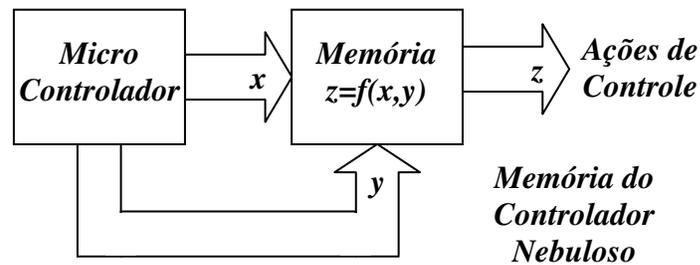
Com auxílio de memórias semicondutoras, é possível armazenar as ações de controle decorrentes da ativação das regras de controle. Considerando as ações de controle devidamente armazenadas em memória, as mesmas podem ser modificadas ou otimizadas, com auxílio de procedimentos de treinamento e adaptação, inerentes às redes neurais artificiais. Dado um conjunto de regras consistente, uma operação em malha fechada se resume a um simples acesso para leitura de uma única posição de memória (BORDON et al., 2001).

Apresenta-se a seguir o projeto de um controlador neuro-nebuloso de uso geral com arquitetura simplificada. Trata-se de um sistema digital implementado em microcomputador com placa de aquisição dados ou através de circuitos integrados microcontroladores.

A estrutura básica do controlador neuro-nebuloso pode ser implementada com circuitos integrados microcontroladores de 8 bits da família *Intel-8051*. Seu princípio de funcionamento consiste basicamente de procedimentos de treinamento, estimação e controle em malha fechada. A *Figura 2.28* representa o procedimento de treinamento que assimila todas as regras de controle enquanto a *Figura 2.29* representa o procedimento de estimação que gera as ações de controle.



*Figura 2.28 Procedimento de Treinamento*



*Figura 2.29 Procedimento de Estimação*

### 2.7.1 Operador de Agregação

Considerando um procedimento de *defuzzificação* padronizado, para uma dada função  $z=f(x,y)$  representando uma superfície genérica gerada por uma função não-linear, não é possível utilizar o operador de agregação representado pela equação (2.1).

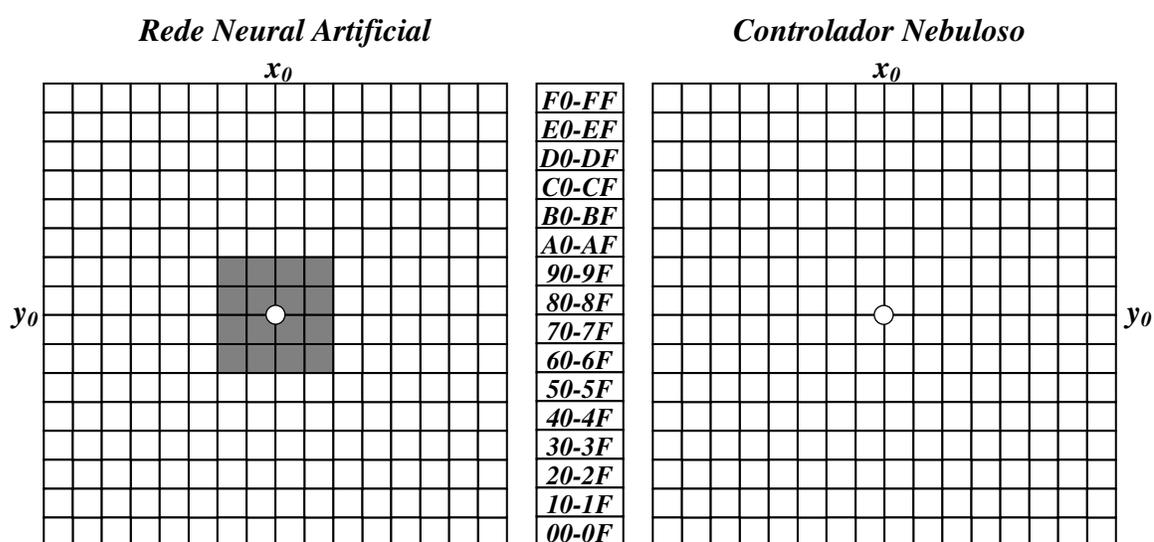
Para um dado conjunto de pesos associados a uma superfície não-linear, outro operador de agregação deve ser utilizado, neste caso, emprega-se uma rede neural artificial baseada no *CMAC*, que pode ser implementada na mesma estrutura  $z=f(x,y)$ , usada no controlador nebuloso de arquitetura simplificada desenvolvido neste trabalho.

Para estimar todas as ações de controle  $z=f(x,y)$  é preciso executar o procedimento de estimação característico do *CMAC*, no entanto, antes do procedimento de estimação, é necessário recorrer ao processo de treinamento, que demanda um custo computacional elevado, devido à estrutura de armazenamento usada no *CMAC*.

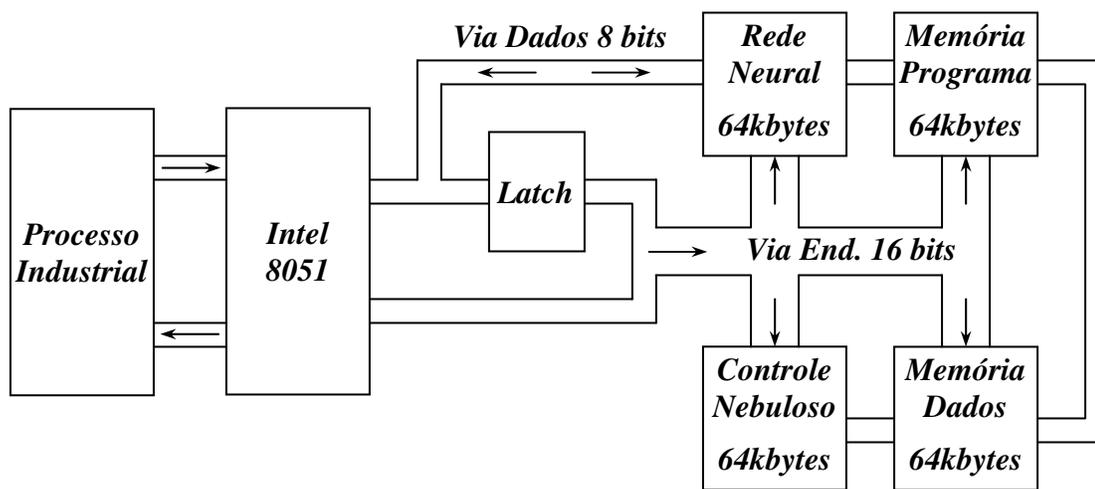
Durante o processo de treinamento todas as regras de controle serão armazenadas numa estrutura do tipo  $z=f(x,y)$  que constitui a rede neural artificial. Com as informações previamente armazenadas na rede neural, durante o processo de estimação, todas as ações de controle serão estimadas e armazenadas numa estrutura do tipo  $z=f(x,y)$  que constitui o controlador nebuloso.

A **Figura 2.30** apresenta a estrutura de memória do controlador neuro-nebuloso com a rede neural artificial associada ao controlador nebuloso de arquitetura simplificada.

A **Figura 2.31** apresenta a arquitetura básica do controlador neuro-nebuloso utilizando microcontroladores da família *Intel-8051* e memórias semicondutoras de **64 kbytes**.



**Figura 2.30** Memória do Controlador Neuro-Nebuloso



**Figura 2.31** Arquitetura do Controlador Neuro-Nebuloso

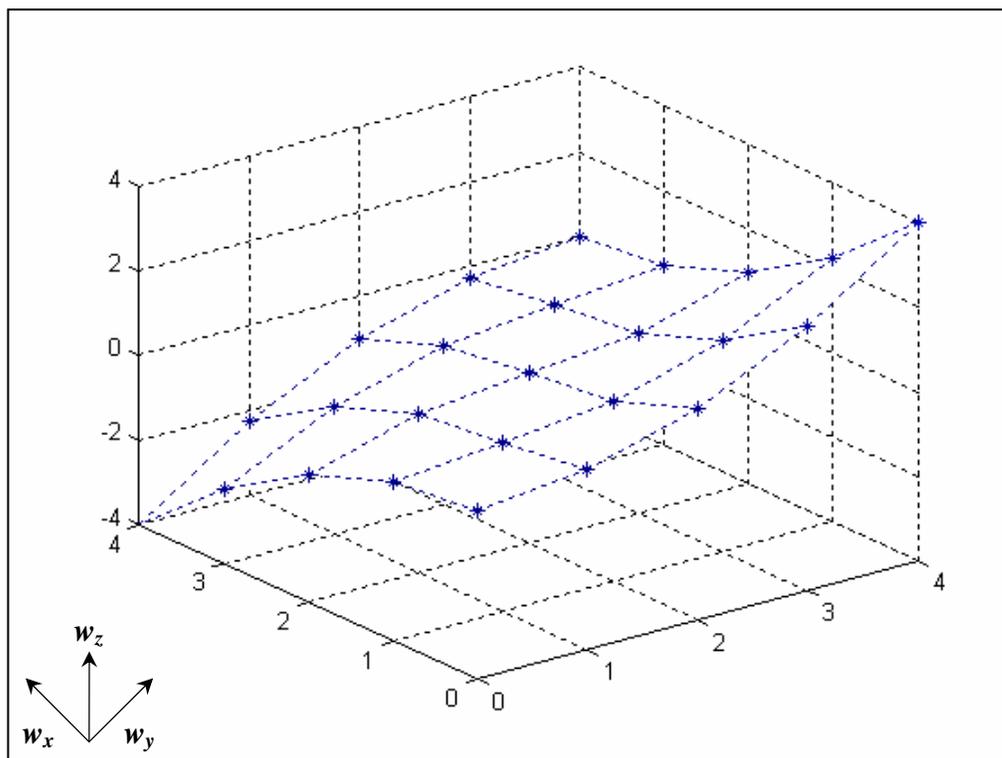
### 2.7.2 Controle Direto com Função Quadrática

A **Tabela 2.11** apresenta os pesos associados às ações de controle  $Ac$  mapeados através da representação gráfica da **Figura 2.32**, usando função quadrática e inferência direta.

**Tabela 2.11** Ações de Controle com Função Quadrática

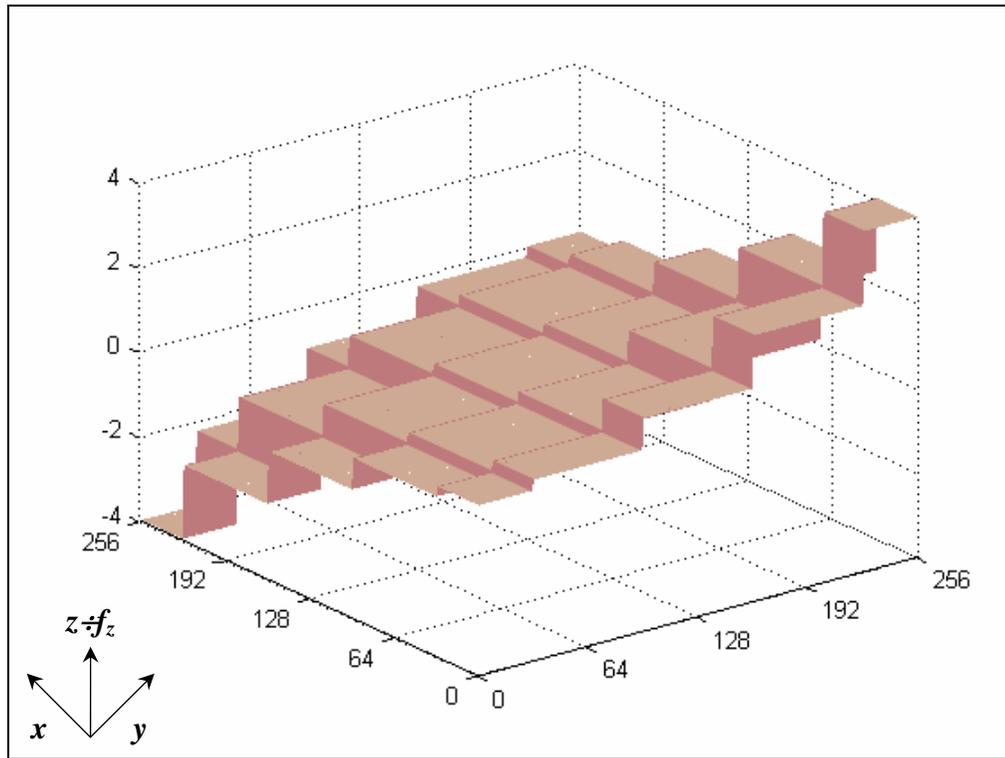
$Ac$			$Vd$					
			$y$	0	64	128	192	256
			$w_y$	0	1	2	3	4
$Va$	$x$	$w_x$	$w_z$	$MB$	$B$	$M$	$A$	$MA$
	0	0	$MB$	0.00	0.25	1.00	2.25	4.00
	64	1	$B$	-0.25	0.00	0.25	1.00	2.25
	128	2	$M$	-1.00	-0.25	0.00	0.25	1.00
	192	3	$A$	-2.25	-1.00	-0.25	0.00	0.25
	256	4	$MA$	-4.00	-2.25	-1.00	-0.25	0.00

**Controle Neuro-Nebuloso com Inferência Direta**

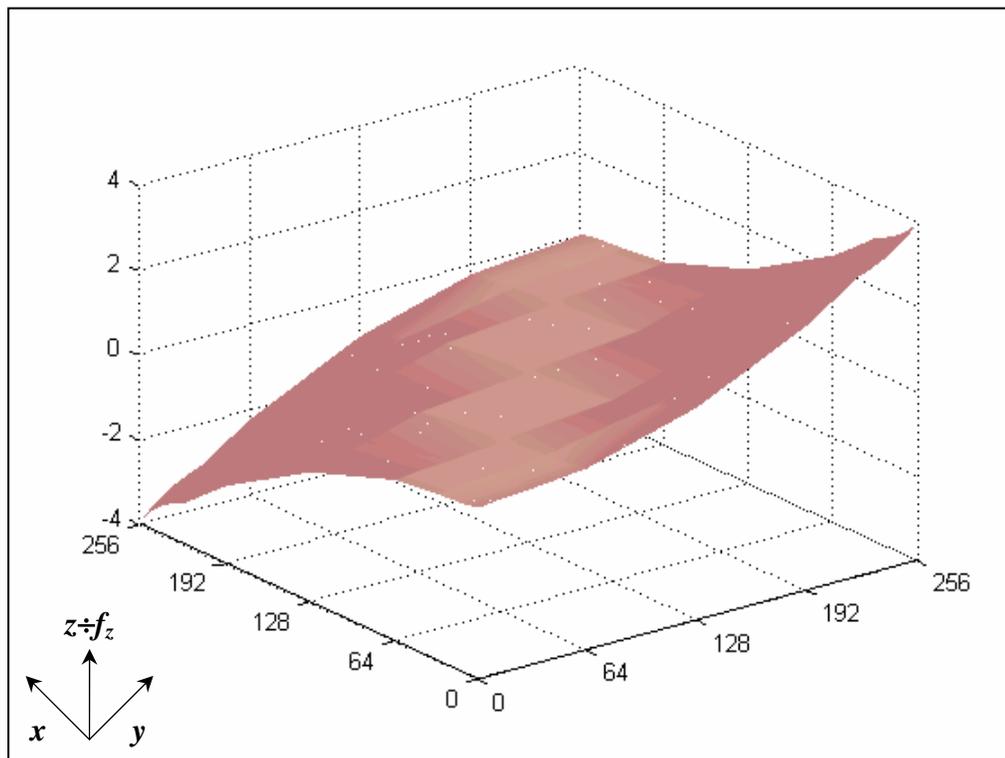


**Figura 2.32** Pesos com Função Quadrática (1<sup>o</sup> Caso)

A **Figura 2.33** apresenta a rede neural **CMAC** atuando como um operador de agregação, enquanto que a **Figura 2.34** mostra as ações de controle decorrentes do processo de estimação.



**Figura 2.33 Rede Neural com Função Quadrática (1º Caso)**



**Figura 2.34 Ações de Controle com Função Quadrática (1º Caso)**

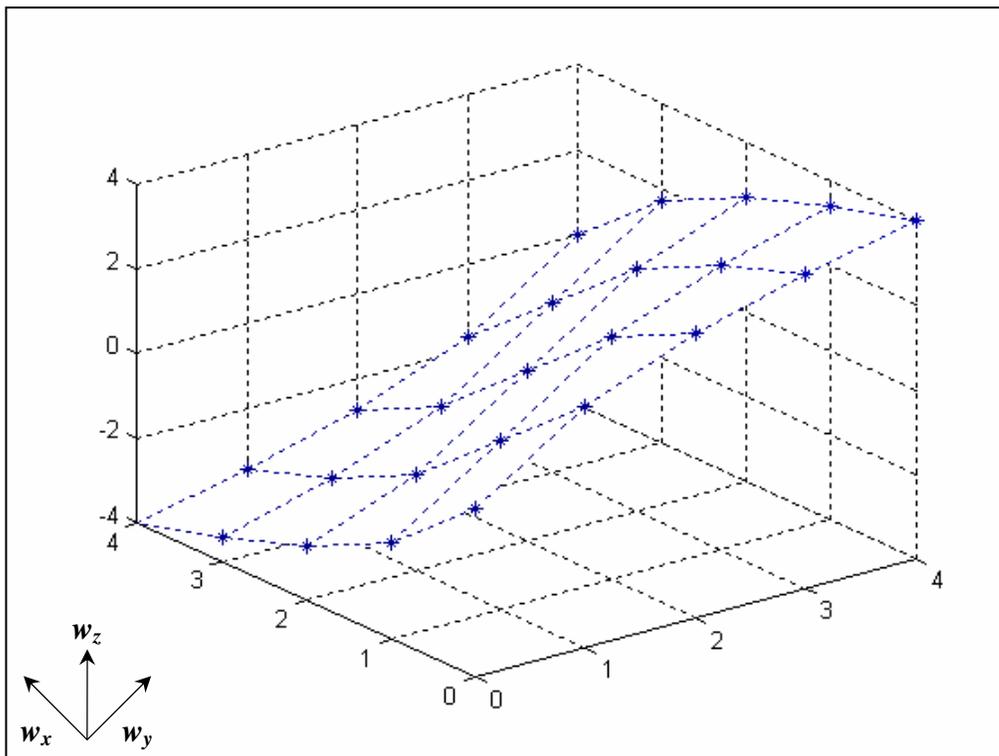
### 2.7.3 Controle Direto com Função Exponencial

A **Tabela 2.12** apresenta os pesos associados às ações de controle  $Ac$  mapeados através da representação gráfica da **Figura 2.35**, usando função exponencial e inferência direta.

**Tabela 2.12** Ações de Controle com Função Exponencial

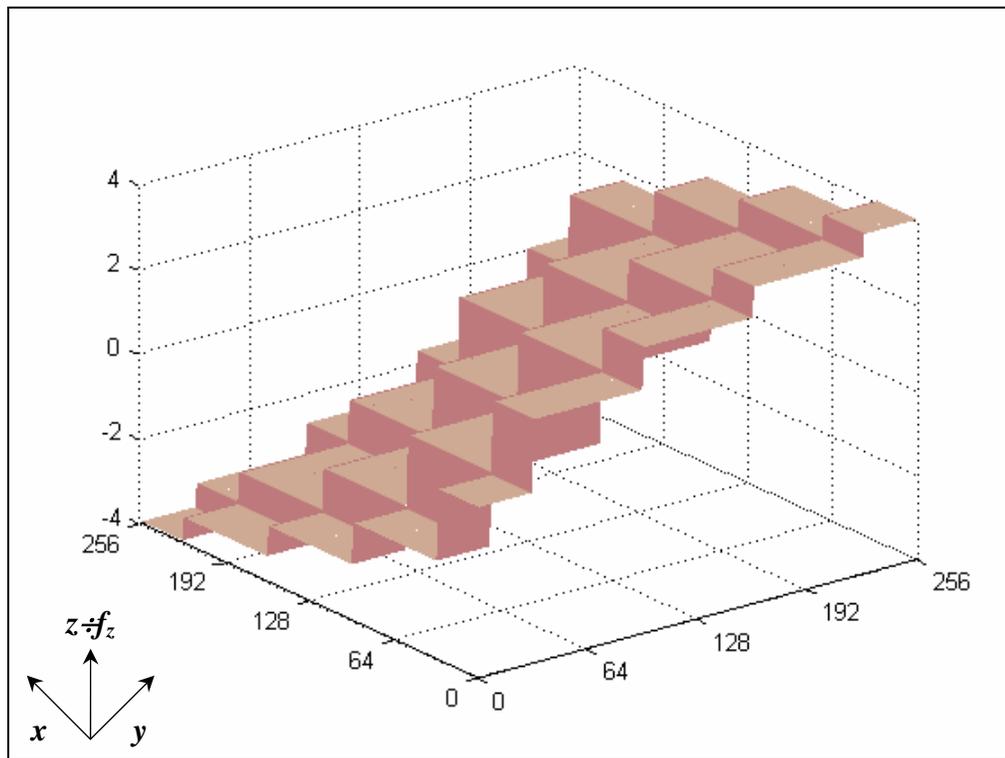
$Ac$			$Vd$					
			$y$	0	64	128	192	256
			$w_y$	0	1	2	3	4
$Va$	$x$	$w_x$	$w_z$	$MB$	$B$	$M$	$A$	$MA$
	0	0	$MB$	0.00	1.72	2.73	3.44	4.00
	64	1	$B$	-1.72	0.00	1.72	2.73	3.44
	128	2	$M$	-2.73	-1.72	0.00	1.72	2.73
	192	3	$A$	-3.44	-2.73	-1.72	0.00	1.72
	256	4	$MA$	-4.00	-3.44	-2.73	-1.72	0.00

**Controle Neuro-Nebuloso com Inferência Direta**

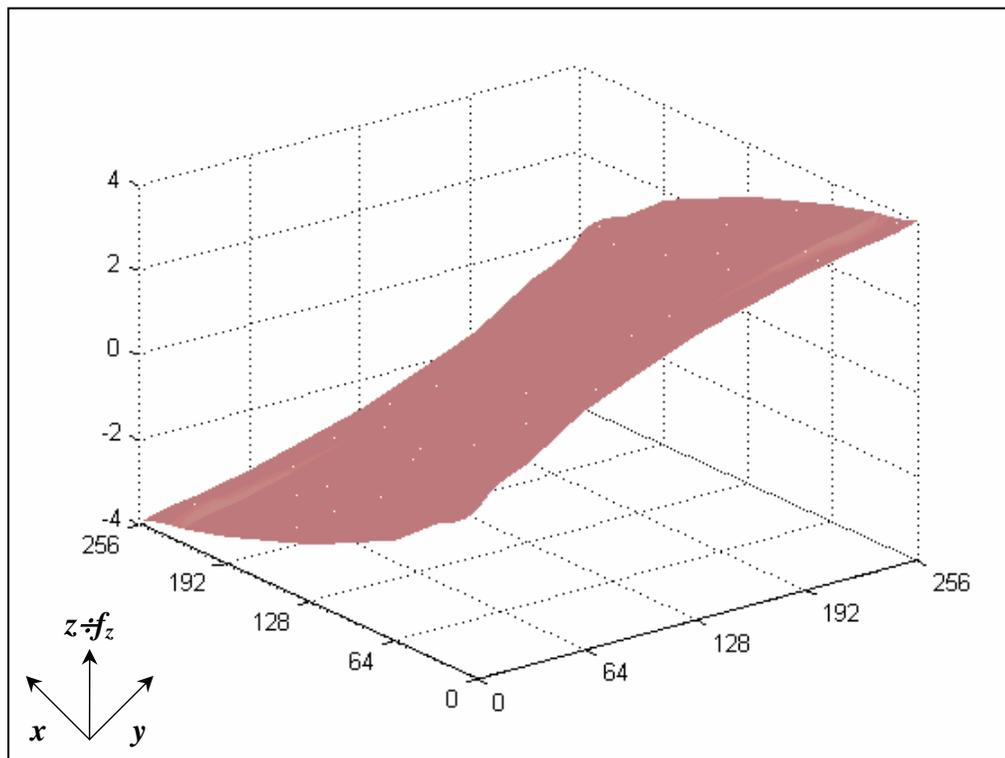


**Figura 2.35** Pesos com Função Exponencial (1º Caso)

A **Figura 2.36** apresenta a rede neural **CMAC** atuando como um operador de agregação, enquanto que a **Figura 2.37** mostra as ações de controle decorrentes do processo de estimação.



*Figura 2.36 Rede Neural com Função Exponencial (1º Caso)*



*Figura 2.37 Ações de Controle com Função Exponencial (1º Caso)*

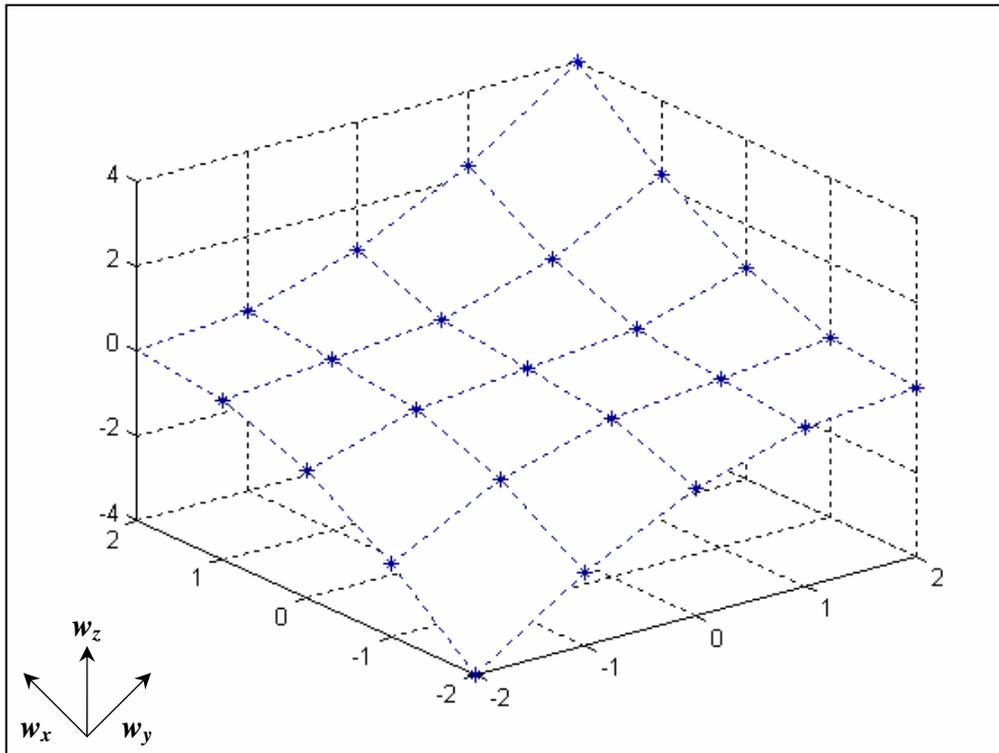
### 2.7.4 Controle Indireto com Função Quadrática

A **Tabela 2.13** apresenta os pesos associados às ações de controle  $Ac$  mapeados através da representação gráfica da **Figura 2.38**, usando função exponencial e inferência direta.

**Tabela 2.13** Ações de Controle com Função Quadrática

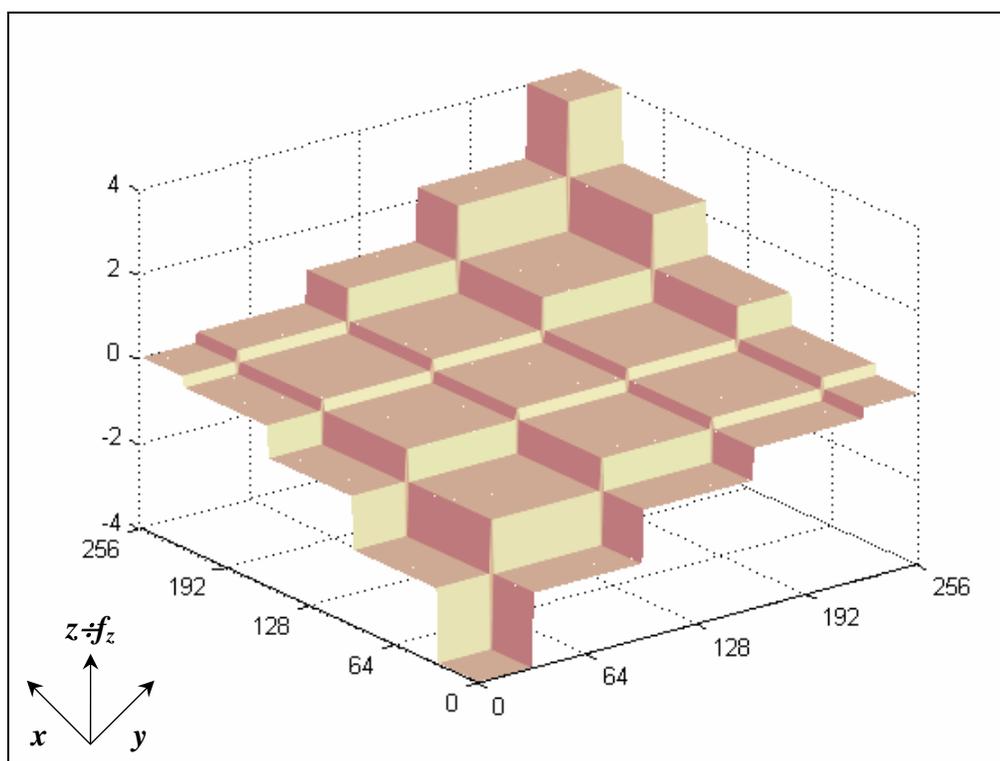
$Ac$			$\delta E$					
			$y$	-128	-64	0	64	128
$E$	$x$	$w_x$	$w_z$	$MN$	$N$	$Z$	$P$	$MP$
	-128	-2	$MN$	-4.00	-2.25	-1.00	-0.25	0.00
	-64	-1	$N$	-2.25	-1.00	-0.25	0.00	0.25
	0	0	$Z$	-1.00	-0.25	0.00	0.25	1.00
	64	1	$P$	-0.25	0.00	0.25	1.00	2.25
	128	2	$MP$	0.00	0.25	1.00	2.25	4.00

**Controle Neuro-Nebuloso com Inferência Indireta**

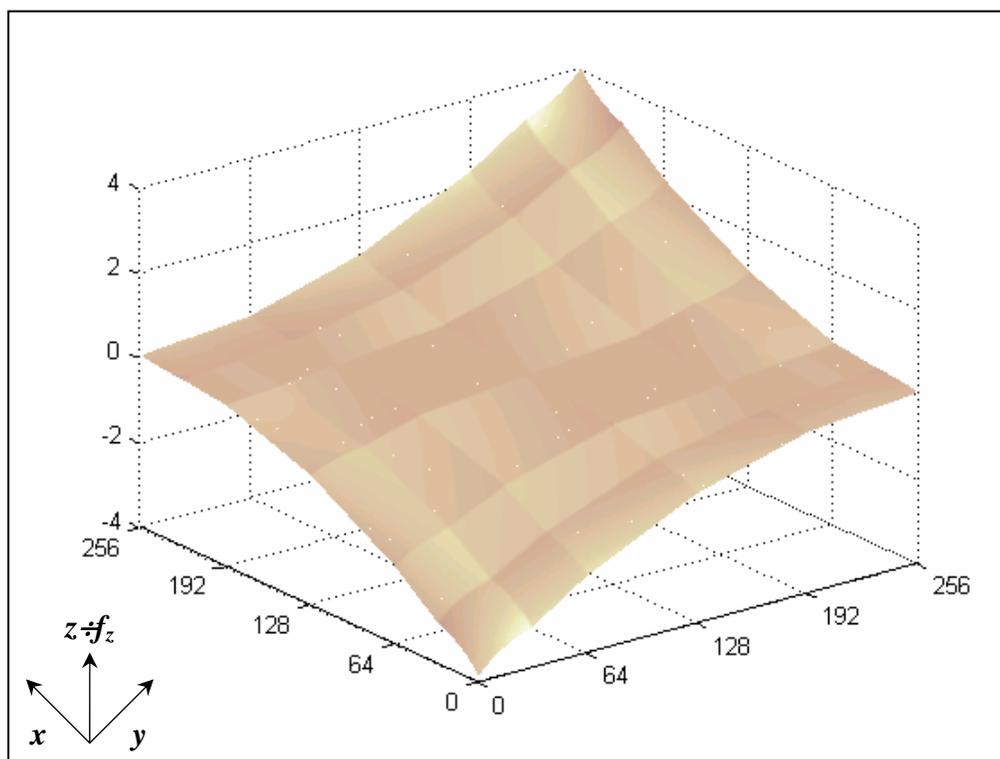


**Figura 2.38** Pesos com Função Quadrática (2<sup>o</sup> Caso)

A **Figura 2.39** apresenta a rede neural **CMAC** atuando como um operador de agregação, enquanto que a **Figura 2.40** mostra as ações de controle decorrentes do processo de estimação.



**Figura 2.39 Rede Neural com Função Quadrática (2º Caso)**



**Figura 2.40 Ações de Controle com Função Quadrática (2º Caso)**

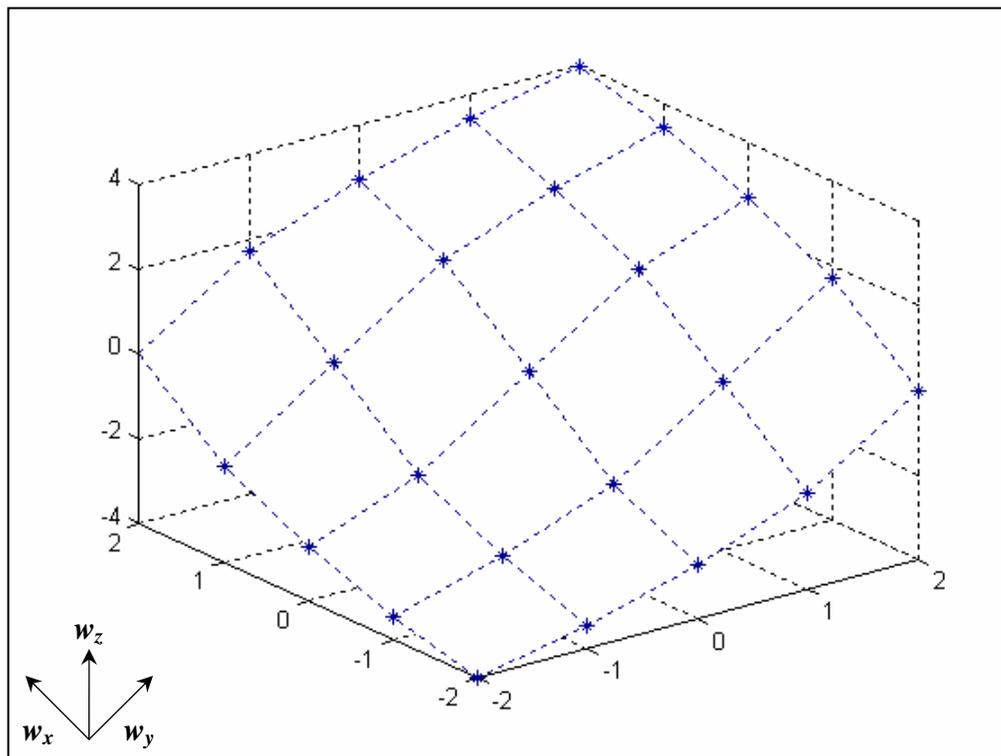
### 2.7.5 Controle Indireto com Função Exponencial

A **Tabela 2.14** apresenta os pesos associados às ações de controle  $Ac$  mapeados através da representação gráfica da **Figura 2.41**, usando função exponencial e inferência direta.

**Tabela 2.14** Ações de Controle com Função Exponencial

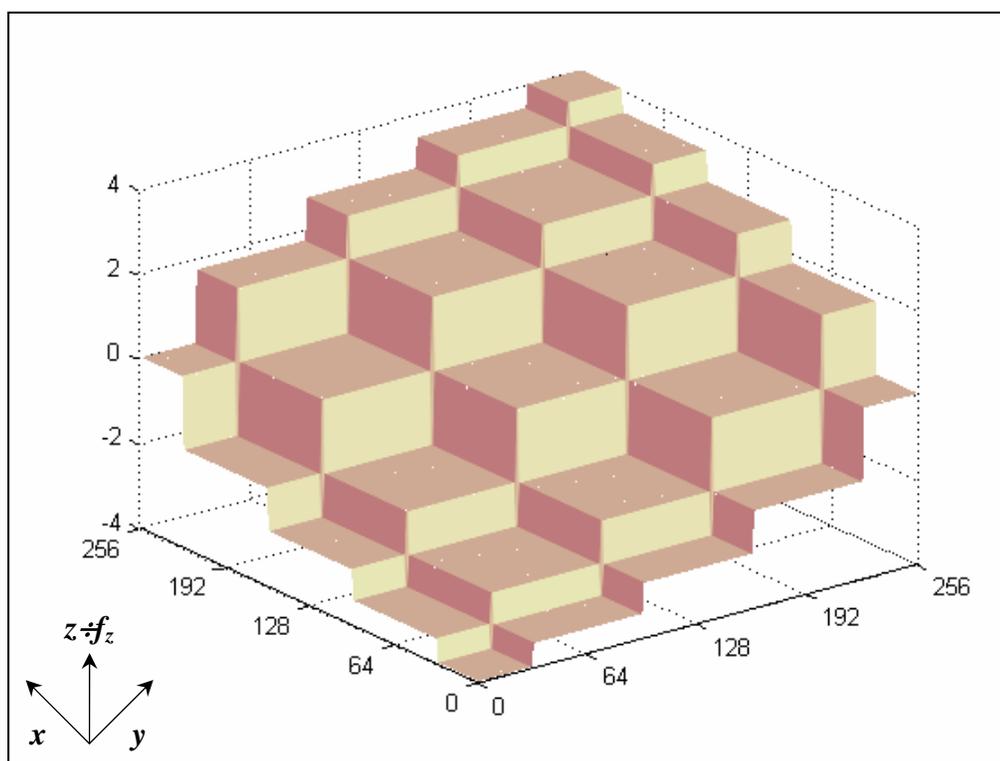
$Ac$			$\delta E$					
			$y$	-128	-64	0	64	128
$E$	$x$	$w_x$	$w_z$	$MN$	$N$	$Z$	$P$	$MP$
	-128	-2	$MN$	-4.00	-3.44	-2.73	-1.72	0.00
	-64	-1	$N$	-3.44	-2.73	-1.72	0.00	1.72
	0	0	$Z$	-2.73	-1.72	0.00	1.72	2.73
	64	1	$P$	-1.72	0.00	1.72	2.73	3.44
	128	2	$MP$	0.00	1.72	2.73	3.44	4.00

**Controle Neuro-Nebuloso com Inferência Indireta**

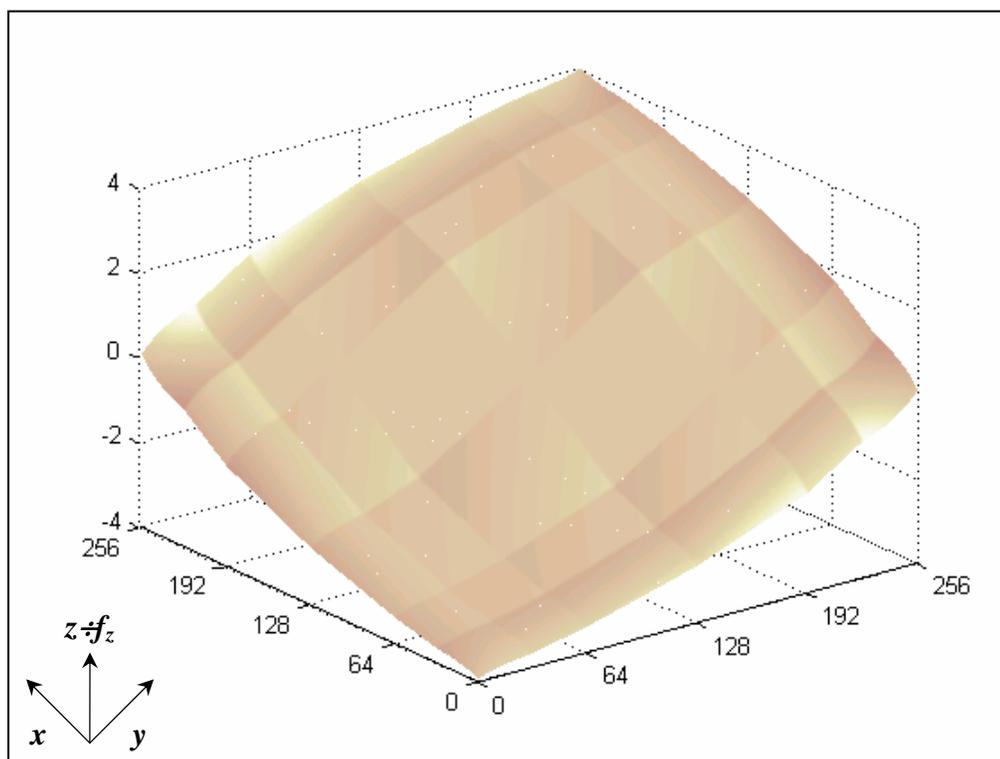


**Figura 2.41** Pesos com Função Exponencial (2º Caso)

A **Figura 2.42** apresenta a rede neural **CMAC** atuando como um operador de agregação, enquanto que a **Figura 2.43** mostra as ações de controle decorrentes do processo de estimação.



*Figura 2.42 Rede Neural com Função Exponencial (2º Caso)*



*Figura 2.43 Ações de Controle com Função Exponencial (2º Caso)*



## ***Capítulo 3***

# ***Acionamento de Máquinas Elétricas de Indução***

### ***3.1 Introdução***

Com o advento da microeletrônica, pode-se perceber que os sistemas de controle estão mais eficientes e confiáveis devido aos componentes eletrônicos de última geração. Atualmente, é possível implementar um sistema de controle digital, capaz de executar algoritmos complexos atuando em tempo real. Da mesma forma, a teoria de controle digital permitiu que as técnicas de ajuste de parâmetros desenvolvidas anteriormente, sejam utilizadas na prática propiciando bons resultados nas implementações microcontroladas (KALMAN, 1958).

Na indústria, diversos processos empregam máquinas e ferramentas cuja característica principal é o acionamento com velocidade variável. Na sua maioria, os acionamentos industriais usam máquinas elétricas que são mais resistentes, mais confiáveis e possuem custos reduzidos. Para controlar a velocidade destes acionamentos, muitos métodos têm sido investigados, indo desde sistemas mecânicos e pneumáticos até sistemas elétricos e eletrônicos que utilizam motores de corrente contínua ou motores de indução (SMITH & DAVISON, 1972).

Tradicionalmente, o controle preciso de velocidade dos acionamentos era feito através de máquinas de corrente contínua, empregando um conjunto de três máquinas elétricas. Com o advento da eletrônica de potência, este arranjo pôde ser substituído por uma única máquina de corrente contínua. Nas aplicações com controle de velocidade empregam-se motores de corrente contínua com excitação independente, cuja velocidade pode ser controlada variando-se a tensão de alimentação do circuito de armadura (KRISHNAN & RAMASWAMI, 1974).

O uso de motores de indução representa uma grande vantagem, tendo em vista seu custo de manutenção reduzido, além de serem normalizados e poderem ser utilizados em ambientes agressivos, onde o uso de motores convencionais de corrente contínua implicaria em custos mais elevados. Sempre foi desejo dos fabricantes e usuários, controlar a velocidade dos motores de indução por meios elétricos mas as técnicas de acionamento dos motores de indução somente tornaram-se viáveis, graças ao desenvolvimento tecnológico alcançado na produção de semicondutores de potência e circuitos integrados dedicados (KRAUSE & LIPO, 1969).

Os sistemas de controle utilizados no acionamento de motores de indução baseiam-se em modelos pré-determinados, onde os parâmetros elétricos da máquina de indução são bem conhecidos. Normalmente, estes modelos são estáticos pois o algoritmo de controle não está preparado para adaptar-se às mudanças de carga ou às alterações de parâmetros, quando o motor de indução está sujeito a diferentes regimes de operação (NELSON et al., 1969).

Genericamente, o acionamento do motor de indução, operando em velocidade variável, exige conhecimento preciso de variáveis e parâmetros da máquina elétrica para garantir respostas dinâmicas rápidas e precisas, além de proporcionar um bom desempenho em regime estacionário. O emprego de técnicas de acionamento de alto desempenho para motores de indução é um procedimento desejável do ponto de vista da aplicação e sua implementação em tempo real tem sido viabilizada devido à evolução das teorias de controle (JOETTEN & MAEDER, 1983).

### ***3.2 Motores de Indução Trifásicos***

Geralmente, as máquinas e ferramentas usadas nos processos industriais, são acionadas por motores de indução trifásicos. Neste tipo de máquina elétrica, para uma variação contínua e precisa da velocidade, a frequência e a tensão de entrada devem ser variáveis, neste caso, os circuitos inversores devem ser usados para sintetizar um sistema trifásico de tensão e frequência variáveis, mantendo-se constante a relação entre tensão e frequência, para que o fluxo magnético presente no entreferro do motor de indução, seja mantido constante. Nesta condição de operação, as translações obtidas nas curvas de conjugado ocorrem ao mesmo tempo em que se preserva o conjugado máximo disponível no eixo do motor de indução (LIPO, 1971).

Como o motor de indução apresenta parâmetros variantes no tempo a manutenção do nível de eficiência do sistema de controle depende do uso de técnicas de estimação e ajuste de parâmetros. Atualmente, significativos avanços têm ocorrido no desenvolvimento de sistemas de controle que usam técnicas de estimação e ajuste de parâmetros, otimizando o desempenho dos acionamentos que empregam motores de indução (MATSUO & LIPO, 1985).

Devido ao comportamento não-linear do motor de indução existe uma forte motivação para o uso de sistemas baseados em lógica nebulosa e redes neurais artificiais na implementação de novas estratégias de controle. Um controlador neuro-nebuloso agrega a lógica nebulosa e as redes neurais artificiais em um único dispositivo eletrônico capaz de controlar sistemas com comportamento dinâmico e característica não-linear, propiciando um alto grau de desempenho.

O objetivo principal deste trabalho é o desenvolvimento de sistemas de controle para acionamento de máquinas e ferramentas industriais. Dentre esses sistemas de controle destaca-se o controle digital de velocidade de motores de indução trifásicos utilizando técnicas especiais baseadas em lógica nebulosa e redes neurais artificiais. As técnicas especiais utilizadas neste trabalho permitem controlar a velocidade do motor de indução sujeito a diferentes condições de carga, sem conhecimento prévio do modelo matemático correspondente.

Este trabalho considera as estratégias de controle utilizadas em acionamentos industriais, mais precisamente, as técnicas de acionamento de motores de indução que utilizam inversores de tensão baseados em modulação do tipo *PWM* (Pulse Width Modulation). Envolve ainda o estudo de novas estratégias de controle digital baseado em métodos automáticos de ajuste de parâmetros, proporcionando ao algoritmo de controle uma estrutura dinâmica capaz de adaptar-se às diversas situações de funcionamento dos motores de indução trifásicos (ADAMS & FOX, 1972).

Para tanto, foi desenvolvido um sistema digital dedicado ao controle dos inversores de tensão do tipo *PWM*, constituído por circuitos eletrônicos conectados a um microcomputador, responsável pela supervisão e operação do sistema. Com os recursos disponíveis no laboratório foi possível implementar um conjunto para ensaios e testes operacionais, visando a avaliação do comportamento do sistema usando técnicas convencionais de controle de velocidade de motores de indução acionados por inversores de tensão do tipo *PWM* (STARR & LOON, 1980).

Com os resultados obtidos nos testes e ensaios operacionais, foi possível avaliar o uso de novas estratégias de controle digital baseadas em técnicas de ajuste de parâmetros. Inicialmente, o estudo das novas estratégias de controle foi feito através de simulação computacional usando o programa *MATLAB™*. A partir da simulação computacional foi obtido um conjunto preliminar de informações com o qual chegou-se à solução mais adequada dentro das características do sistema de acionamento e controle proposto, viabilizando o projeto de um controlador digital de velocidade para motores de indução trifásicos (KRAUSE & THOMAS, 1965).

A implementação deste sistema de controle digital envolveu a utilização de diversos componentes, circuitos eletrônicos e programas aplicativos, requerendo um novo conjunto para ensaios e testes operacionais, dada a complexidade característica dos sistemas digitais de controle de motores de indução, envolvendo aquisição de dados em tempo real. Portanto, este trabalho vem contribuir com os meios necessários para o desenvolvimento de estratégias de controle direcionadas para utilização de microcontroladores de 8 bits, empregando novas metodologias de controle digital adequadas às características do processo a ser controlado.

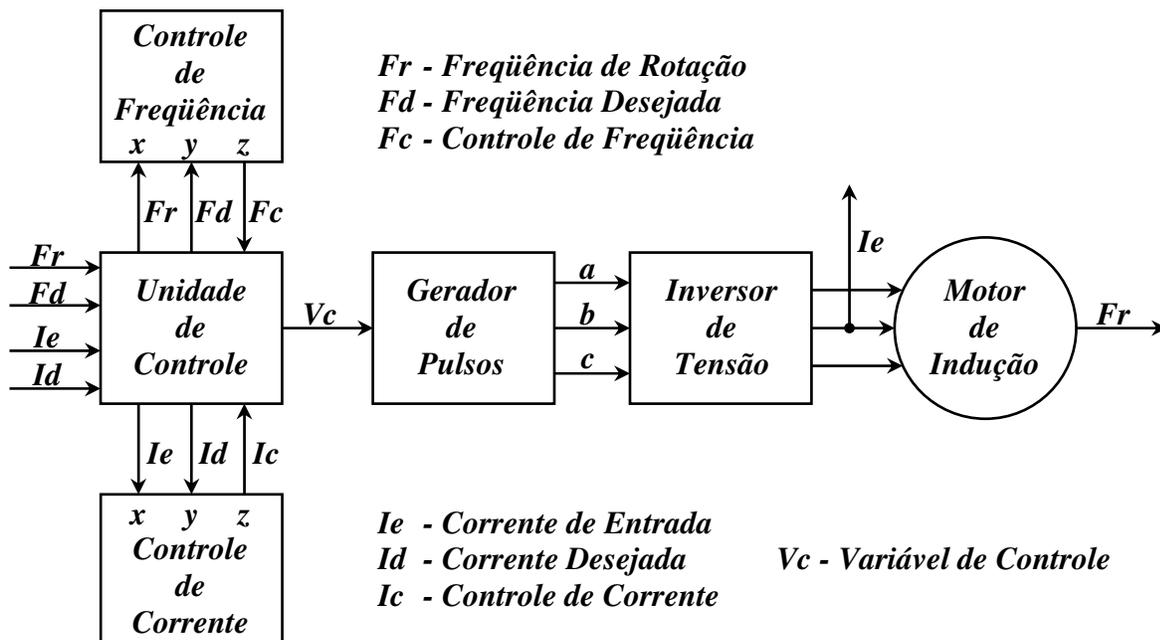
---

O uso de microcontroladores tem como objetivo a simplificação do hardware e software, normalmente estruturados a partir de microcomputadores, o que torna o sistema de controle mais caro e complexo enquanto que os microcontroladores permitem a implementação de um sistema mais simples e menos dispendioso. Por outro lado, o uso de microcontroladores traz uma série de limitações, reduzindo a capacidade de processamento e armazenamento de dados. Com o uso de várias estruturas em paralelo pode-se melhorar sensivelmente a capacidade de processamento e armazenamento disponível nos microcontroladores (ATHANI & DESHPANDE, 1980).

### 3.3 Descrição do Sistema de Controle

Todas as estratégias de controle para o acionamento de motores de indução trifásicos utilizadas neste trabalho, foram implementadas no mesmo sistema de controle digital. Diferentes algoritmos são utilizados para fechar as malhas de controle de frequência e controle de corrente.

Os componentes que fazem parte do sistema de controle apresentado no diagrama em blocos da **Figura 3.1** podem ser melhor compreendidos com uma breve descrição de cada um indicando suas principais características e limitações, evidenciando o princípio de funcionamento de cada elemento e apresentando as alternativas de implementação do sistema de controle.



**Figura 3.1** Diagrama do Sistema de Controle

### 3.3.1 *Unidade de Controle*

Esta unidade é responsável pela implementação das estratégias de controle, executando algoritmos específicos e previamente estabelecidos, propiciando o gerenciamento efetivo de todas as tarefas necessárias para o acionamento de motores de indução trifásicos.

A unidade de controle pode ser implementada a partir de um microcomputador usando linguagem de programação adequada ou através de um circuito dedicado composto de memórias semicondutoras e um microcontrolador de 8 bits da família *Intel-8051*.

Esta unidade possui elementos para aquisição de sinais analógicos e processamento de sinais digitais. Os sinais mais importantes são: a frequência desejada  $F_d$ , a frequência de rotação  $F_r$ , a corrente desejada  $I_d$ , a corrente de entrada  $I_e$  e as variáveis de controle  $F_c$ ,  $I_c$  e  $V_c$ .

A unidade de controle gerencia efetivamente duas malhas fechadas: a malha de controle de frequência, responsável pelo ajuste de velocidade do motor de indução e a malha de controle de corrente, responsável pela limitação de corrente de entrada do motor de indução.

A malha principal é a malha de controle de frequência sendo que a malha de controle de corrente, atua como um elemento de ajuste dinâmico de ganho, regulando a ação da malha de controle de frequência. A limitação de corrente não determina o desligamento do sistema de acionamento pois não se trata de um simples circuito de proteção. Trata-se de um elemento capaz de controlar a corrente máxima de operação do motor de indução trifásico.

### 3.3.2 *Circuito Gerador de Pulsos*

O gerador de pulsos pode ser implementado com o circuito integrado *Philips-4752*, que utiliza o processo de modulação por largura de pulsos em dupla borda. Trata-se de circuito eletrônico puramente digital que permite um controle total da tensão de alimentação e da frequência de operação, com uma frequência de chaveamento que é sempre um múltiplo inteiro da frequência sintetizada pelo circuito inversor de tensão (STARR & LOON, 1980).

Como resultado prático, pode-se obter excelente balanceamento de frequência e tensão, reduzida perda nos motores de indução e otimização do conteúdo harmônico. O controle é feito através do ajuste da frequência de dois sinais digitais: um sinal responsável pelo valor da tensão nominal e outro responsável pelo valor da frequência sintetizada pelo inversor *PWM*.

### 3.3.3 Inversor de Tensão Trifásico

Um conversor de frequência é o equipamento mais adequado para controlar a velocidade dos motores de indução. Através dos inversores é possível alimentar este tipo de máquina elétrica com tensão e frequência variáveis, mantendo constante a relação tensão/frequência.

O circuito inversor de tensão é alimentado por uma fonte de tensão primária obtida a partir de um retificador não-controlado com filtro capacitivo. Possui uma configuração em ponte trifásica e emprega modulação por largura de pulsos de borda dupla (HOLTZ, 1994).

Normalmente, este circuito utiliza dispositivos semicondutores do tipo **IGBT** (Insulated Gate Bipolar Transistor) para o chaveamento e opera-se em modo contínuo sendo que, a qualquer instante existem sempre três dispositivos de chaveamento em estado de condução.

Pode-se utilizar na implementação de um sistema de controle e acionamento de motores de indução, um circuito inversor de tensão comercial, com entrada de controle independente, para que o controlador possa atuar efetivamente sem nenhum tipo de interferência.

### 3.3.4 Motor de Indução Trifásico

O motor de indução trifásico usado neste trabalho possui os seguintes dados de catálogo: potência nominal  $P_n=1.0\text{CV}$ , tensão nominal  $V_n=220\text{V}$ , conjugado nominal  $T_n=4.0\text{Nm}$ , corrente nominal  $I_n=3.8\text{A}$ , frequência nominal  $F_n=60\text{Hz}$  e momento de inércia  $J_o=0.0015\text{kgm}^2$ .

Trata-se de uma máquina elétrica de indução trifásica fabricada pela empresa **Weg** com rotor do tipo gaiola de esquilo, modelo **80-1190** de **4** pólos, isolamento classe **B**, proteção **IP-54**, categoria **N**, fator de potência **0.75**, fator de serviço **1.15** e rendimento de **70%**.

Os parâmetros do circuito elétrico equivalente, obtidos em laboratório, são os seguintes: resistência do estator  $R_s=9.8\Omega$ , resistência do rotor  $R_r=5.7\Omega$ , indutância do estator  $L_s=12.54\text{mH}$ , indutância do rotor  $L_r=18.71\text{mH}$  e indutância de magnetização  $L_m=312.75\text{mH}$ . Estes parâmetros foram levantados através de ensaio em vazio e ensaio com rotor bloqueado e seus valores foram utilizados nas simulações computacionais apresentadas neste trabalho.

Quando um motor de indução é alimentado por um circuito inversor de tensão trifásico, mantendo-se constante a relação tensão/frequência, é possível controlar a velocidade de operação deste tipo de máquina elétrica, utilizando um perfil de velocidade adequado.

### 3.4 Estratégias de Controle

O presente trabalho apresenta uma metodologia de ajuste dinâmico de velocidade de um motor de indução trifásico, acionado por um inversor de tensão do tipo *PWM*, utilizando controle digital e planejamento de ganho. Trata-se de um sistema de controle que opera em malha fechada, monitorando em tempo real a frequência de rotação do motor de indução e procurando minimizar ou anular as diferenças em relação à frequência de operação desejada.

Este sistema de controle utiliza uma variável auxiliar para determinar as condições de operação do motor de indução e estabelecer o ajuste automático de um parâmetro de controle, relacionado com o planejamento de ganho em malha fechada.

A variável auxiliar utilizada é a corrente de entrada do motor de indução e para efeito de planejamento de ganho, a corrente de entrada é sempre comparada com um múltiplo da corrente nominal do motor de indução visando à limitação da corrente de entrada.

O acionamento em malha fechada permite que um novo e adequado perfil de frequência seja obtido sempre que um novo valor de frequência desejada seja estabelecido, pois as condições de operação do motor de indução trifásico, podem ser estimadas a partir de informações precisas obtidas através de aquisição de dados em tempo real.

Deste modo, pode-se a qualquer instante de operação do motor de indução, aumentar ou diminuir a frequência desejada, promover consideráveis variações de carga, limitando a corrente de entrada do motor de indução, mantendo o erro de regime praticamente nulo.

Deve-se notar que as estratégias de controle apresentadas neste trabalho, não utilizam modelos matemáticos para determinar as condições de operação do motor de indução, muito pelo contrário, necessitam apenas de informações relativas aos dados de placa do motor de indução.

Como o valor eficaz da corrente de entrada do motor de indução, decorre de uma medida auxiliar, uma estrutura computacional baseada em redes neurais artificiais pode armazenar as informações relativas ao valor da corrente de entrada, durante um processo de treinamento e, posteriormente, estas informações podem ser resgatadas durante um processo de estimação.

A utilização de um sistema de controle digital baseado em aquisição de dados em tempo real, representa uma solução interessante do ponto de vista prático, mas acaba sendo um pouco mais complexo, quando comparado com as soluções normalmente adotadas para acionamento e controle de motores de indução, usando inversores de tensão do tipo *PWM*.

Este trabalho procura apresentar novas alternativas de controle digital de velocidade de motores de indução, inserindo inovações no que diz respeito às possibilidades de controle em malha fechada, usando lógica nebulosa para formulação de uma estratégia de controle alternativa que possa substituir a estratégia de controle **PID** (KRIKELIS & FASSOIS, 1984).

Os algoritmos de controle podem ser implementados usando linguagem de máquina de um microcontrolador de 8 bits da família **Intel-8051** ou ainda usando linguagem de alto nível de um microcomputador tipo **PC** equipado com placa de aquisição de dados.

As limitações decorrentes da digitalização das variáveis de entrada e saída são inerentes e dizem respeito à precisão numérica e ao fundo de escala uma vez que estes valores dependem do número de bits utilizados na representação numérica das grandezas físicas envolvidas.

Apresentam-se a seguir as estratégias de controle digital proporcional integral derivativa e as novas estratégias de controle nebulosas e neuro-nebulosas desenvolvidas neste trabalho, visando à implementação de um sistema de controle digital, simples e eficiente.

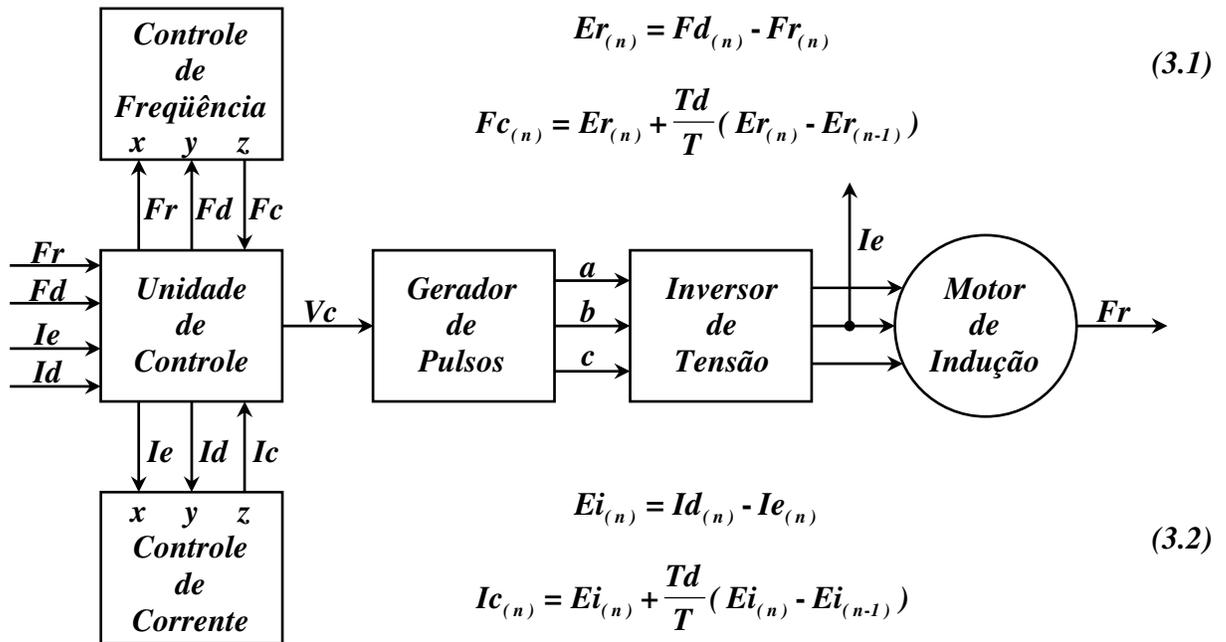
### **3.4.1 Controle Proporcional Integral Derivativo**

A estratégia de controle **PID** (Proporcional Integral Derivativo) determina a variável de controle **Vc** que ajusta a frequência da tensão sintetizada pelo circuito inversor de tensão **PWM**. O ganho de malha fechada deste sistema de controle é ajustado periodicamente em função da corrente de entrada do motor de indução, conforme indicado na **Figura 3.2**.

O algoritmo de controle **PID** efetua os cálculos das parcelas de controle proporcional, integral e derivativa. Sabendo que o tempo de amostragem interfere no ganho do controlador, o mesmo foi ajustado com base na dinâmica do sistema de acionamento. Os tempos **Ti** e **Td** foram mantidos iguais a **I** para garantir a compatibilidade com as demais estratégias de controle.

O erro de frequência **Er** é obtido a partir da diferença entre a frequência desejada **Fd** e a frequência de rotação **Fr** medida no eixo do motor de indução e, em seguida, calculam-se as parcelas de controle proporcional e derivativa. Somando-se estas duas parcelas, obtém-se uma variável de controle **Fc**, responsável pelo ajuste da frequência de rotação **Fr**.

O erro de corrente **Ei** é obtido a partir da diferença entre a corrente desejada **Id** e a corrente de entrada **Ie** medida na alimentação do motor de indução e, em seguida, calculam-se as parcelas de controle proporcional e derivativa. Somando-se estas duas parcelas, obtém-se uma variável de controle **Ic**, responsável pela limitação da corrente de entrada **Ie**.



**Figura 3.2 Estratégia de Controle PID**

A frequência de rotação  $Fr$  do motor de indução é obtida em tempo real com auxílio de um tacômetro digital. Através de um circuito conversor de frequência para tensão é possível obter um sinal analógico proporcional à frequência de rotação do motor de indução.

A corrente de entrada  $Ie$  do motor de indução é obtida em tempo real com auxílio de um sensor de efeito *Hall*. Através de um circuito somador retificador de precisão é possível obter um sinal analógico proporcional à corrente de entrada do motor de indução.

A variável de controle  $Fc$  responsável pelo ajuste da frequência de rotação é dada pela equação (3.1). A variável de controle  $Ic$  responsável pela limitação da corrente de entrada é dada pela equação (3.2). A variável de controle  $Vc$  responsável pelo fechamento das duas malhas de controle é dada pela equação (3.3) que representa a ação integral do controle *PID*.

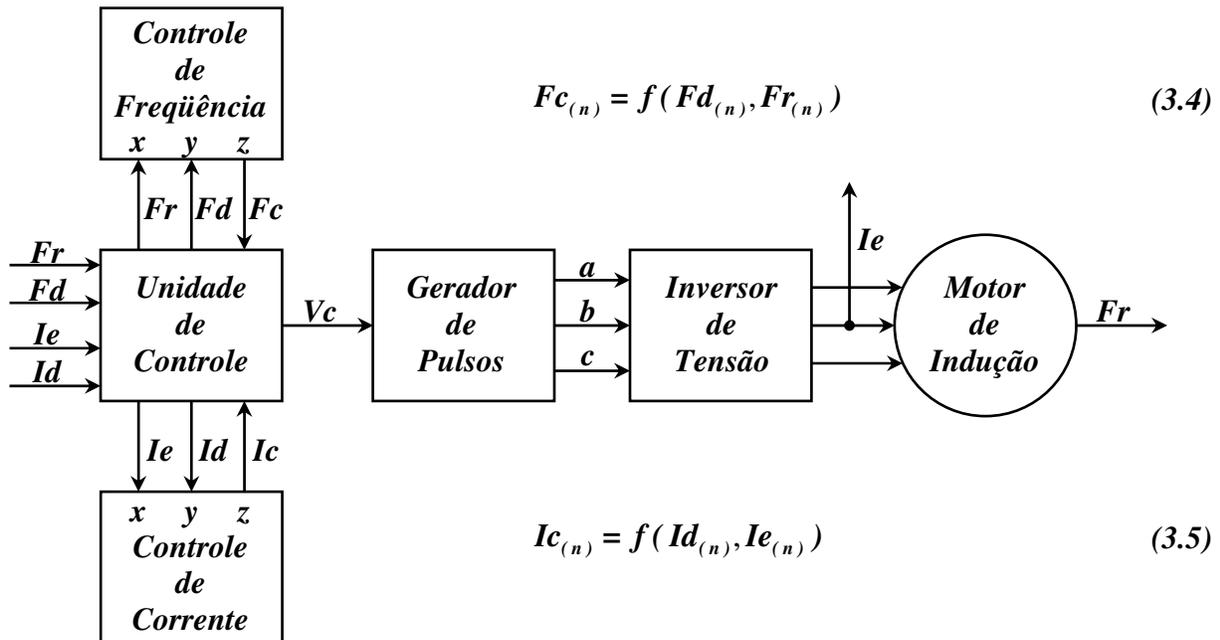
$$Vc_{(n)} = Vc_{(n-1)} + \frac{T}{Ti} (Fc_{(n)} \times Ic_{(n)}) Kp \quad (3.3)$$

Os tempos de integração  $Ti$  e derivação  $Td$ , o tempo de amostragem  $T$  e a constante de proporcionalidade  $Kp$ , determinados segundo métodos clássicos de ajuste de parâmetros, podem interferir no comportamento estático e dinâmico do sistema de acionamento do motor de indução alterando os valores de erro de regime, tempo de resposta e sobre-sinal.

### 3.4.2 Controle Nebuloso com Inferência Direta

A estratégia de controle nebuloso com inferência direta gera a variável de controle  $Vc$ , responsável pelo ajuste de frequência da tensão sintetizada pelo circuito inversor de tensão **PWM**. O ganho de malha fechada deste sistema de controle é ajustado periodicamente em função da corrente de entrada do motor de indução trifásico, conforme indicado na **Figura 3.3**.

O algoritmo de controle de inferência direta executa os procedimentos de **fuzzificação** das variáveis de entrada  $Fr$ ,  $Fd$ ,  $Ie$ ,  $Id$  e **defuzzificação** das variáveis de saída  $Fc$  e  $Ic$ . O ajuste do controlador é obtido através de procedimentos básicos de inferência utilizados na formulação de regras de controle consistentes e depende do tempo de amostragem  $T$ .



**Figura 3.3** Controle Nebuloso com Inferência Direta

O algoritmo de controle de frequência de inferência direta utiliza as variáveis de entrada: frequência desejada  $Fd$  e frequência de rotação  $Fr$ , gerando a variável de controle  $Fc$ .

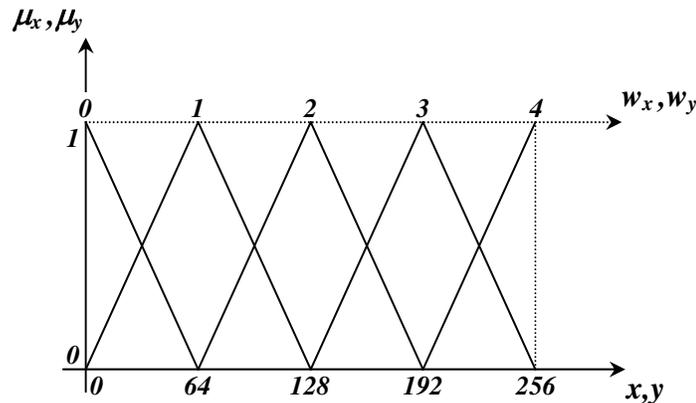
O algoritmo de controle de corrente de inferência direta utiliza as variáveis de entrada: corrente desejada  $Id$  e corrente de entrada  $Ie$ , gerando a variável de controle  $Ic$ .

$$Vc_{(n)} = Vc_{(n-1)} + (Fc_{(n)} \times Ic_{(n)})Kp \quad (3.6)$$

Considerando a representação binária apresentada na *Tabela 2.1*, para  $f_x$  e  $f_y$  iguais a **64**, a *fuzzificação* usando inferência direta é feita como indicado na *Tabela 3.1* e *Figura 3.4*.

**Tabela 3.1 Fuzzificação com Inferência Direta**

Entradas		Variável	Termo	Peso
<i>Fr</i> ou <i>Fd</i> [Hz]	<i>Ie</i> ou <i>Id</i> [A]	Cartesiana <i>x</i> ou <i>y</i>	Lingüístico Associado	Associado $w_x$ ou $w_y$
[0:16]	[0:4]	[0:64]	<i>Muito Baixo</i>	0
[0:32]	[0:8]	[0:128]	<i>Baixo</i>	1
[16:48]	[4:12]	[64:192]	<i>Médio</i>	2
[32:64]	[8:16]	[128:256]	<i>Alto</i>	3
[48:64]	[12:16]	[192:256]	<i>Muito Alto</i>	4



**Figura 3.4 Funções de Pertinência das Entradas (1º Caso)**

Com a variável de entrada *Fr* igual a **20Hz**, temos a variável *x* igual a **80**. Através de uma simples divisão ( $x \div f_x$ ), com  $f_x$  igual a **64**, define-se que estão ativos os termos lingüísticos:

**Baixo** com peso  $w_{xa}$  igual a **1**, com índice de pertinência  $\mu_{xa}$  igual a **0.75**.

**Médio** com peso  $w_{xb}$  igual a **2**, com índice de pertinência  $\mu_{xb}$  igual a **0.25**.

Pois,  $(80 \div 64)$  é igual a **1.25**.

Com a variável de entrada *Fd* igual a **60Hz**, temos a variável *y* igual a **240**. Através de uma simples divisão ( $y \div f_y$ ), com  $f_y$  igual a **64**, define-se que estão ativos os termos lingüísticos:

**Alto** com peso  $w_{ya}$  igual a **3**, com índice de pertinência  $\mu_{ya}$  igual a **0.25**.

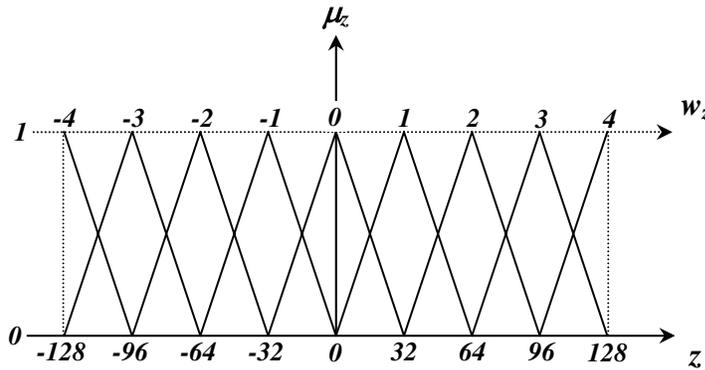
**Muito Alto** com peso  $w_{yb}$  igual a **4**, com índice de pertinência  $\mu_{yb}$  igual a **0.75**.

Pois,  $(240 \div 64)$  é igual a **3.75**.

Considerando a representação binária indicada na **Tabela 2.2**, para um fator  $f_z$  igual a 32, a **defuzzificação** com inferência direta é feita como indicado na **Tabela 3.2** e **Figura 3.5**.

**Tabela 3.2** Ações de Controle com Inferência Direta

$Fc, Ic$			$Fd, Id$					
			$y$	0	64	128	192	256
			$w_y$	0	1	2	3	4
$Fr, Ir$	$x$	$w_x$	$w_z$	MB	B	M	A	MA
	0	0	MB	0	1	2	3	4
	64	1	B	-1	0	1	2	3
	128	2	M	-2	-1	0	1	2
	192	3	A	-3	-2	-1	0	1
	256	4	MA	-4	-3	-2	-1	0



**Figura 3.5** Funções de Pertinência da Variável de Saída (1º Caso)

Para  $x$  igual a 80 e  $y$  igual a 240 estão ativas as regras com os seguintes pesos  $w_z$ :

$$w_{z(w_{xa}, w_{ya})} = w_{z(1,3)} = 2$$

$$w_{z(w_{xb}, w_{ya})} = w_{z(2,3)} = 1$$

$$w_{z(w_{xa}, w_{yb})} = w_{z(1,4)} = 3$$

Através da equação (2.1) pode-se determinar o valor da variável de saída ( $z \div f_z$ ).

$$\frac{z}{f_z} = (\mu_{xa} + \mu_{ya})w_{z(w_{xa}, w_{ya})} + (\mu_{xb})w_{z(w_{xb}, w_{ya})} + (\mu_{yb})w_{z(w_{xa}, w_{yb})} - w_{z(w_{xa}, w_{ya})}$$

$$(0.75 + 0.25)(2) + (0.25)(1) + (0.75)(3) - (2) = 2.5$$

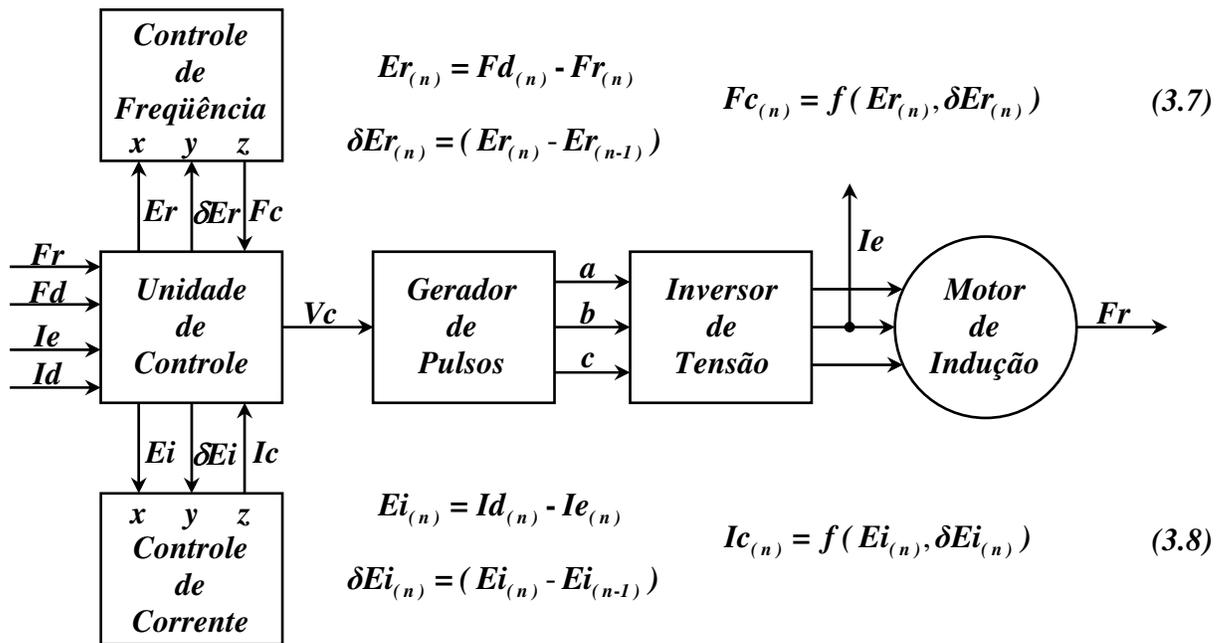
Para um fator  $f_z$  igual a 32 pode-se calcular o valor da variável de controle  $Fc$ .

$$Fc = 2.5 \times 32 = 80$$

### 3.4.3 Controle Nebuloso com Inferência Indireta

A estratégia de controle nebuloso com inferência indireta gera a variável de controle  $Vc$ , responsável pelo ajuste de frequência da tensão sintetizada pelo circuito inversor de tensão *PWM*. O ganho de malha fechada deste sistema de controle é ajustado periodicamente em função da corrente de entrada do motor de indução trifásico, conforme indicado na *Figura 3.6*.

O algoritmo de controle de inferência indireta executa procedimentos de *fuzzificação* das variáveis de entrada  $Er$ ,  $\delta Er$ ,  $Ei$ ,  $\delta Ei$  e *defuzzificação* das variáveis de saída  $Fc$  e  $Ic$ . O ajuste do controlador é obtido através de procedimentos básicos de inferência utilizados na formulação de regras de controle consistentes e depende do tempo de amostragem  $T$ .



*Figura 3.6 Controle Nebuloso com Inferência Indireta*

O algoritmo de controle de frequência de inferência indireta usa as variáveis de entrada: erro de frequência  $Er$  e gradiente do erro de frequência  $\delta Er$ , gerando a variável de controle  $Fc$ .

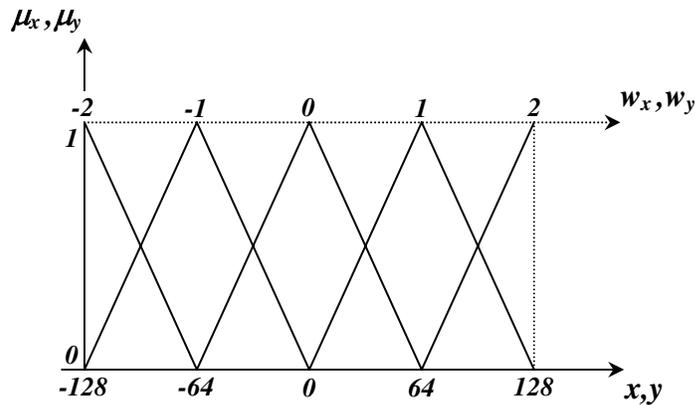
O algoritmo de controle de corrente de inferência indireta usa as variáveis de entrada: erro de corrente  $Ei$  e gradiente do erro de corrente  $\delta Ei$ , gerando a variável de controle  $Ic$ .

$$Vc_{(n)} = Vc_{(n-1)} + (Fc_{(n)} \times Ic_{(n)})Kp \quad (3.9)$$

Considerando a representação binária apresentada na *Tabela 2.1*, para  $f_x$  e  $f_y$  iguais a **64**, a *fuzzificação* usando inferência indireta é feita como indicado na *Tabela 3.3* e *Figura 3.7*.

*Tabela 3.3 Fuzzificação com Inferência Indireta*

<i>Entradas</i>		<i>Variável Cartesiana x ou y</i>	<i>Termo Lingüístico Associado</i>	<i>Peso Associado <math>w_x</math> ou <math>w_y</math></i>
<i>Er ou <math>\delta Er</math> [Hz]</i>	<i>Ei ou <math>\delta Ei</math> [A]</i>			
<b>[-64;-32]</b>	<b>[-16;-8]</b>	<b>[-128;-64]</b>	<b>Muito Negativo</b>	<b>-2</b>
<b>[-64;0]</b>	<b>[-16;0]</b>	<b>[-128;0]</b>	<b>Negativo</b>	<b>-1</b>
<b>[-32;32]</b>	<b>[-8;8]</b>	<b>[-64;64]</b>	<b>Zero</b>	<b>0</b>
<b>[0;64]</b>	<b>[0;16]</b>	<b>[0;128]</b>	<b>Positivo</b>	<b>1</b>
<b>[32;64]</b>	<b>[8;16]</b>	<b>[64;128]</b>	<b>Muito Positivo</b>	<b>2</b>



*Figura 3.7 Funções de Pertinência das Entradas (2º Caso)*

Com a variável de entrada *Er* igual a **24Hz**, temos a variável *x* igual a **48**. Através de uma simples divisão ( $x \div f_x$ ), com  $f_x$  igual a **64**, define-se que estão ativos os termos linguísticos:

**Zero** com peso  $w_{xa}$  igual a **0**, com índice de pertinência  $\mu_{xa}$  igual a **0.25**.

**Positivo** com peso  $w_{xb}$  igual a **1**, com índice de pertinência  $\mu_{xb}$  igual a **0.75**.

Pois,  $(48 \div 64)$  é igual a **0.75**.

Com a variável de entrada  $\delta Er$  igual a **-8Hz**, temos a variável *y* igual a **-16**. Através de uma simples divisão ( $y \div f_y$ ), com  $f_y$  igual a **64**, define-se que estão ativos os termos linguísticos:

**Negativo** com peso  $w_{ya}$  igual a **-1**, com índice de pertinência  $\mu_{ya}$  igual a **0.25**.

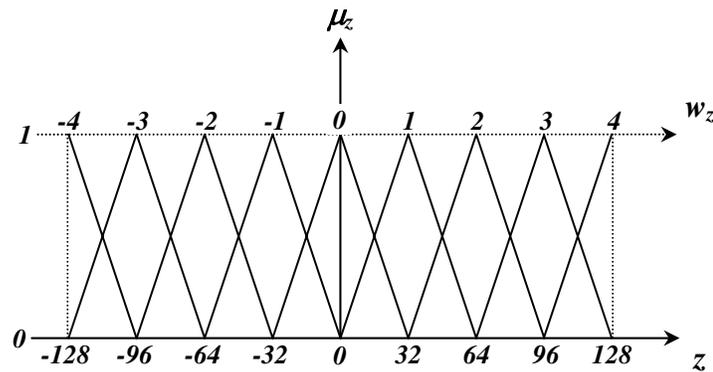
**Zero** com peso  $w_{yb}$  igual a **0**, com índice de pertinência  $\mu_{yb}$  igual a **0.75**.

Pois,  $(-16 \div 64)$  é igual a **-0.25**.

Considerando a representação binária indicada na **Tabela 2.2**, para um fator  $f_z$  igual a 32, a **defuzzificação** com inferência indireta é feita como indicado na **Tabela 3.4** e **Figura 3.8**.

**Tabela 3.4** Ações de Controle com Inferência Indireta

$Fc, Ic$			$\delta Er, \delta Ei$					
			$y$	-128	-64	0	64	128
			$w_y$	-2	-1	0	1	2
$Er, Ei$	$x$	$w_x$	$w_z$	MN	N	Z	P	MP
	-128	-2	MN	-4	-3	-2	-1	0
	-64	-1	N	-3	-2	-1	0	1
	0	0	Z	-2	-1	0	1	2
	64	1	P	-1	0	1	2	3
	128	2	MP	0	1	2	3	4



**Figura 3.8** Funções de Pertinência da Variável de Saída (2º Caso)

Para  $x$  igual a 80 e  $y$  igual a 240 estão ativas as regras com os seguintes pesos  $w_z$ :

$$w_{z(w_{xa}, w_{ya})} = w_{z(0, -1)} = -1$$

$$w_{z(w_{xb}, w_{ya})} = w_{z(1, -1)} = 0$$

$$w_{z(w_{xa}, w_{yb})} = w_{z(0, 0)} = 0$$

Através da equação (2.1) pode-se determinar o valor da variável de saída ( $z \div f_z$ ).

$$\frac{z}{f_z} = (\mu_{xa} + \mu_{ya}) w_{z(w_{xa}, w_{ya})} + (\mu_{xb}) w_{z(w_{xb}, w_{ya})} + (\mu_{yb}) w_{z(w_{xa}, w_{yb})} - w_{z(w_{xa}, w_{ya})}$$

$$(0.25 + 0.25)(-1) + (0.75)(0) + (0.75)(0) - (-1) = 0.5$$

Para um fator  $f_z$  igual a 32 pode-se calcular o valor da variável de controle  $Fc$ .

$$Fc = 0.5 \times 32 = 16$$

### 3.4.4 Estratégia de Controle Neuro-Nebulosa

Devido ao uso de duas malhas de controle, é necessário fazer uma composição de duas ações de controle  $F_c$  e  $I_c$ , a fim de se obter uma única variável de controle  $V_c$ . Uma rede neural do tipo *CMAC* pode ser usada como mecanismo de inferência capaz de mapear em memória a composição de todas as ações de controle necessárias para o acionamento do motor de indução, substituindo as duas malhas de controle por apenas uma, capaz de gerar as ações de controle  $Ac$  em função do erro de frequência  $Er$  e do erro de corrente  $Ei$ .

O algoritmo de controle neuro-nebuloso, executa os procedimentos de *fuzzificação* das duas variáveis de entrada  $Er$ ,  $Ei$  e *defuzzificação* da variável de saída  $Ac$ . O ajuste do controlador neuro-nebuloso é obtido através de procedimentos básicos de inferência utilizados na formulação de regras de controle consistentes e depende do tempo de amostragem  $T$ .

A estratégia de controle neuro-nebulosa, utiliza as ações de controle  $Ac$ , para gerar uma variável de controle  $V_c$ , responsável pelo ajuste de frequência da tensão sintetizada pelo inversor de tensão *PWM*. O ganho de malha fechada deste sistema de controle é ajustado periodicamente em função da corrente de entrada do motor de indução, conforme indicado na *Figura 3.9*.

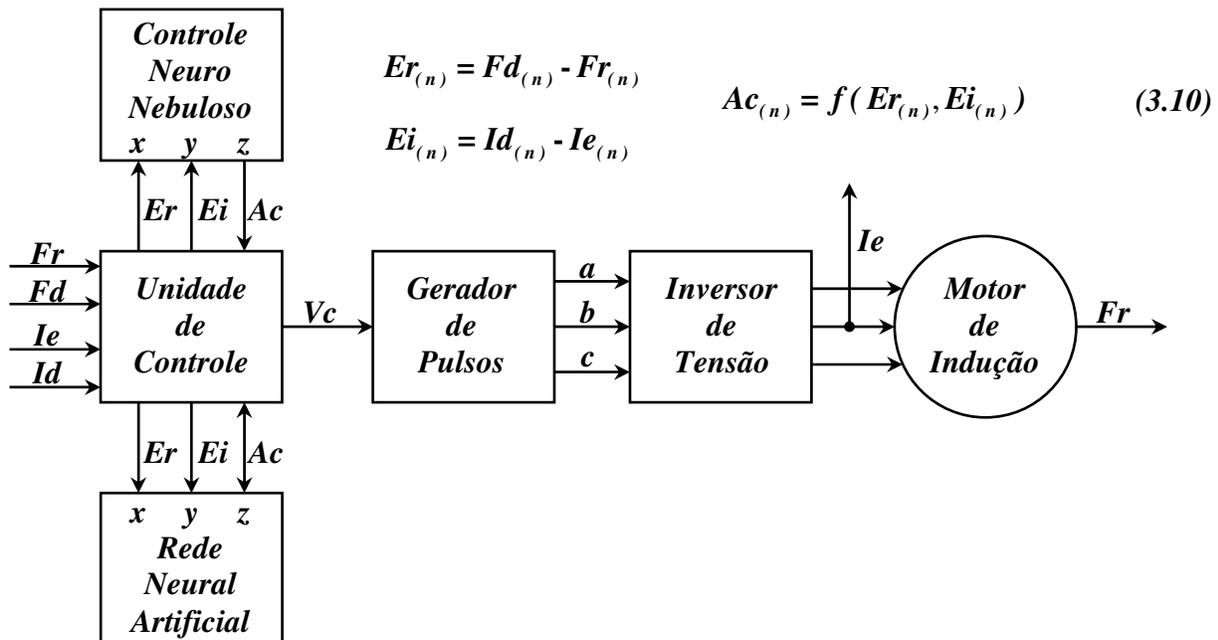
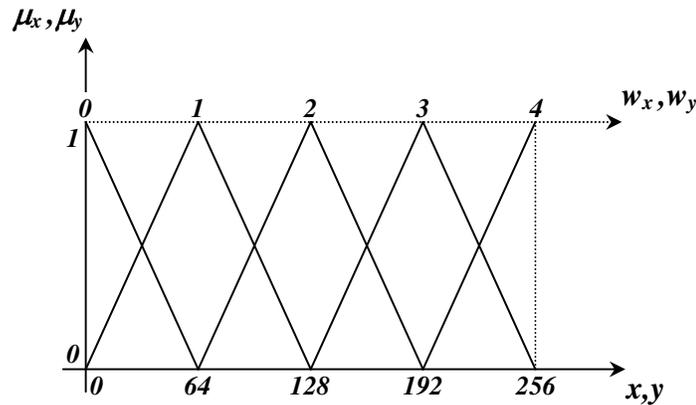


Figura 3.9 Estratégia de Controle Neuro-Nebulosa

Considerando a representação binária apresentada na *Tabela 2.1*, para  $f_x$  e  $f_y$  iguais a 64, a *fuzzificação* com controle neuro-nebuloso foi feita como indicado na *Tabela 3.5* e *Figura 3.10*.

*Tabela 3.5 Fuzzificação com Controle Neuro-Nebuloso*

<i>Entradas</i>		<i>Variável Cartesiana</i> <i>x ou y</i>	<i>Termo Linguístico Associado</i>	<i>Peso Associado</i> $w_x$ ou $w_y$
<i>Abs(Er)</i> [Hz]	<i>Abs(Ei)</i> [A]			
[0:16]	[0:4]	[0:64]	<i>Muito Baixo</i>	0
[0:32]	[0:8]	[0:128]	<i>Baixo</i>	1
[16:48]	[4:12]	[64:192]	<i>Médio</i>	2
[32:64]	[8:16]	[128:256]	<i>Alto</i>	3
[48:64]	[12:16]	[192:256]	<i>Muito Alto</i>	4



*Figura 3.10 Funções de Pertinência das Entradas*

As variáveis de entrada  $Er$ ,  $Ei$  e a variável de saída  $Ac$ , são consideradas em termos absolutos para garantir a precisão do controlador neuro-nebuloso. Portanto, o sinal  $Sc$  das ações de controle  $Ac$  depende do sinal do erro de frequência  $Sr$  e do sinal do erro de corrente  $Si$ .

$$Sc_{(n)} = (Sr_{(n)} \times Si_{(n)}) \quad (3.11)$$

O valor da variável de controle  $Vc$  é obtido periodicamente, a partir do valor absoluto das ações de controle  $Ac$ , conforme indicado na equação (3.12).

$$Vc_{(n)} = Vc_{(n-1)} + (Sc_{(n)} \times Ac_{(n)})Kp \quad (3.12)$$

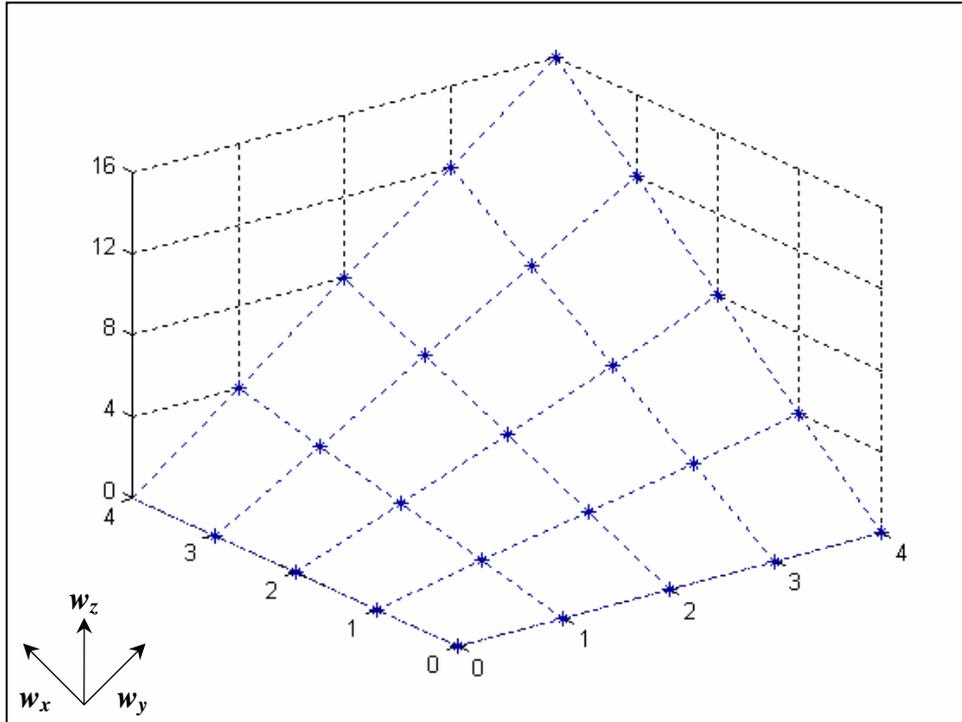
O sinal  $Sc$  das ações de controle  $Ac$ , é determinado pelo sinal resultante da combinação lógica dos sinais  $Sr$  e  $Si$ , determinado através da equação (3.11).

Apresenta-se na **Tabela 3.6** a composição de todas as ações de controle necessárias para o acionamento do motor de indução usando uma única malha de controle. Considera-se apenas o valor absoluto das ações de controle  $Ac$  em função do erro de frequência  $Er$  e erro de corrente  $Ei$ . Os valores dos pesos  $w_z$  associados à variável de saída  $z$  foram determinados através de operações de inferência baseadas em conhecimento especialista, conforme indicado na **Figura 3.11**.

**Tabela 3.6** Ações de Controle com Função Não-Linear

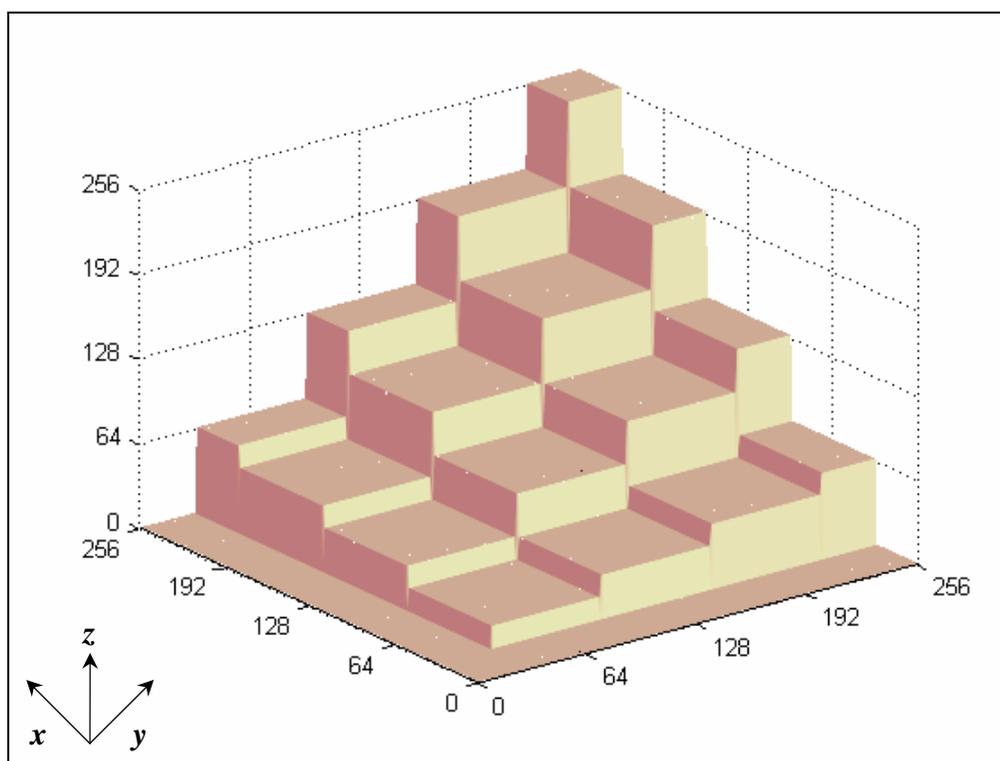
$Abs(Ac)$			$Abs(Ei)$					
			$y$	0	64	128	192	256
			$w_y$	0	1	2	3	4
$Abs(Er)$	$x$	$w_x$	$w_z$	$MN$	$N$	$Z$	$P$	$MP$
	0	0	$MN$	0	0	0	0	0
	64	1	$N$	0	1	2	3	4
	128	2	$Z$	0	2	4	6	8
	192	3	$P$	0	3	6	9	12
	256	4	$MP$	0	4	8	12	16

**Controle Neuro-Nebuloso**

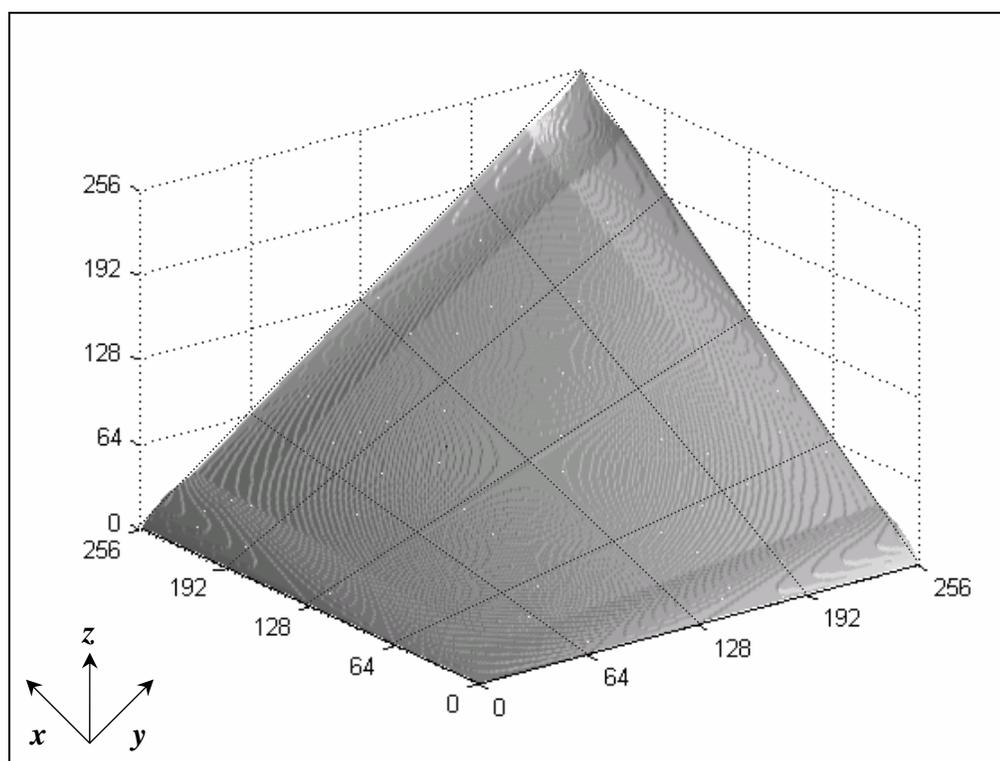


**Figura 3.11** Conjunto de Pesos com Função Não-Linear

A **Figura 3.12** apresenta a rede neural **CMAC** atuando como um operador de agregação, enquanto que a **Figura 3.13** mostra as ações de controle decorrentes do processo de estimação.



**Figura 3.12** Rede Neural Artificial com Função Não-Linear



**Figura 3.13** Ações de Controle com Função Não-Linear

### 3.5 *Simulação Computacional*

Para avaliar as estratégias de controle desenvolvidas neste trabalho, foram elaborados modelos matemáticos detalhados e precisos do motor de indução, do inversor de tensão trifásico e dos algoritmos de controle. Estes modelos foram implementados em microcomputador usando um aplicativo computacional, elaborado com o programa *MATLAB™*, para simulação do sistema de controle operando em malha fechada com alteração periódica de parâmetros.

Este aplicativo computacional inclui os detalhes específicos de um sistema de controle digital implementado com microcontroladores de 8 bits da família *Intel-8051*. As operações lógicas e aritméticas são efetuadas considerando as limitações impostas pela arquitetura de 8 bits. A estrutura básica do controlador nebuloso representada genericamente por uma função  $z=f(x,y)$ , pode ser implementada em memórias semicondutoras com capacidade para armazenar *64 kbytes*, usando barramento de dados de 8 bits e barramento de endereços de *16* bits.

O modelo dinâmico do motor de indução apresentado no *Apêndice 3* utiliza um sistema de equações diferenciais não-lineares, pois o efeito de acoplamento entre as fases do estator e do rotor do motor de indução depende da posição angular do rotor em relação ao estator. Este efeito de acoplamento torna os coeficientes das equações diferenciais variáveis.

Para evitar um sistema de equações diferenciais com coeficientes variáveis pode-se usar uma transformação de coordenadas através de um sistema de referência adequado que torna os coeficientes das equações diferenciais constantes. O modelo dinâmico do motor de indução adota um sistema de referência *d-q* girando na velocidade do campo girante, usando as transformações de coordenadas baseadas nas equações de *Park* (KRAUSE & THOMAS, 1965).

No modelo dinâmico do inversor as tensões de fase são representadas através do produto algébrico de tensões obtidas para operação em onda quadrada e um trem de pulsos de amplitude unitária. Os pulsos são distribuídos simetricamente em relação a componente fundamental usando modulação *PWM* com borda dupla e taxa de pulsos variável (ADAMS & FOX, 1972).

Neste método, são estabelecidos pulsos de controle cuja forma de onda é caracterizada pela utilização de ângulos de comutação pré-determinados de modo que o conteúdo harmônico possa ser controlado, eliminando-se os elementos indesejáveis. No modelo dinâmico do inversor a geração dos pulsos de controle obedece às relações entre frequência sintetizada e frequência portadora utilizadas no circuito integrado *Philips-4752* (STARR & LOON, 1980).

### 3.5.1 Simulação do Motor de Indução

Os modelos matemáticos para simulação computacional, foram elaborados segundo o desenvolvimento apresentado no *Apêndice 3* e implementados em microcomputador usando um aplicativo computacional desenvolvido com o programa *MATLAB™*.

O motor de indução trifásico usado neste trabalho possui os seguintes dados de catálogo: potência nominal  $P_n=1.0\text{Cv}$ , tensão nominal  $V_n=220\text{V}$ , conjugado nominal  $T_n=4.0\text{Nm}$ , corrente nominal  $I_n=3.8\text{A}$ , frequência nominal  $F_n=60\text{Hz}$  e momento de inércia  $J_o=0.0015\text{kgm}^2$ .

Trata-se de uma máquina elétrica de indução trifásica fabricada pela empresa *Weg* com rotor do tipo gaiola de esquilo, modelo *80-1190* de 4 pólos, isolamento classe *B*, proteção *IP-54*, categoria *N*, fator de potência *0.75*, fator de serviço *1.15* e rendimento de *70%*.

Os parâmetros do circuito elétrico equivalente, obtidos em laboratório, são os seguintes: resistência do estator  $R_s=9.8\Omega$ , resistência do rotor  $R_r=5.7\Omega$ , indutância do estator  $L_s=12.54\text{mH}$ , indutância do rotor  $L_r=18.71\text{mH}$  e indutância de magnetização  $L_m=312.75\text{mH}$ .

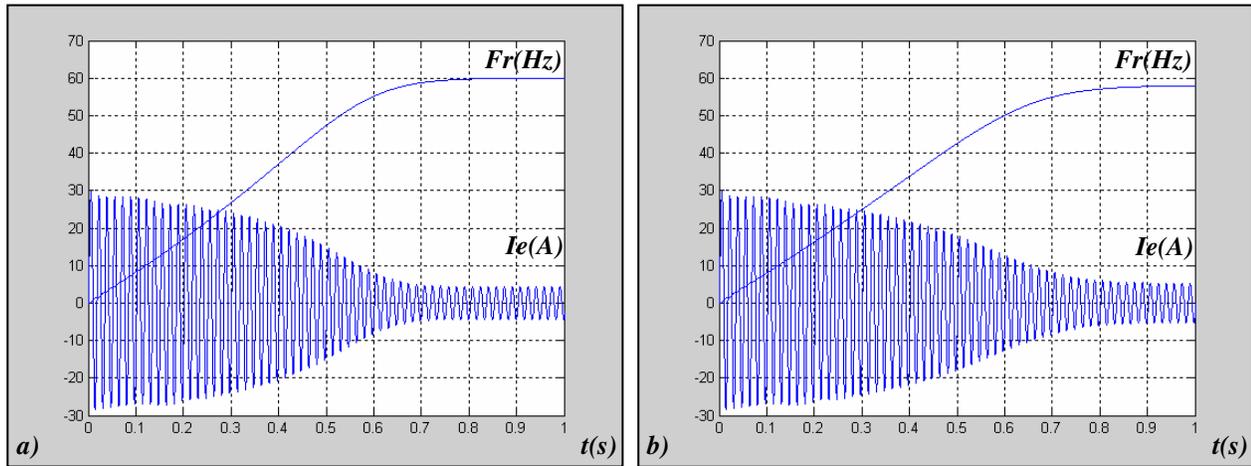
Um gerador síncrono conectado a uma carga resistiva representada por um conjunto de lâmpadas incandescentes capazes de dissipar até *750W* de potência ativa é usado como uma carga mecânica acoplada ao eixo do motor de indução. Com este gerador síncrono é possível obter um conjugado de carga  $T_c$  diretamente proporcional à frequência de rotação  $F_r$ .

Trata-se de um alternador trifásico fabricado pela empresa *Kohlbach*, modelo *FK600* de 4 pólos, isolamento classe *B*, proteção *IP-21*. Os dados de catálogo do gerador síncrono trifásico são os seguintes: tensão nominal de *220V*, corrente nominal *10.5A*, potência nominal de *4.0kVA*, fator de potência *0.85*, frequência nominal *60Hz* e rendimento de *85%*.

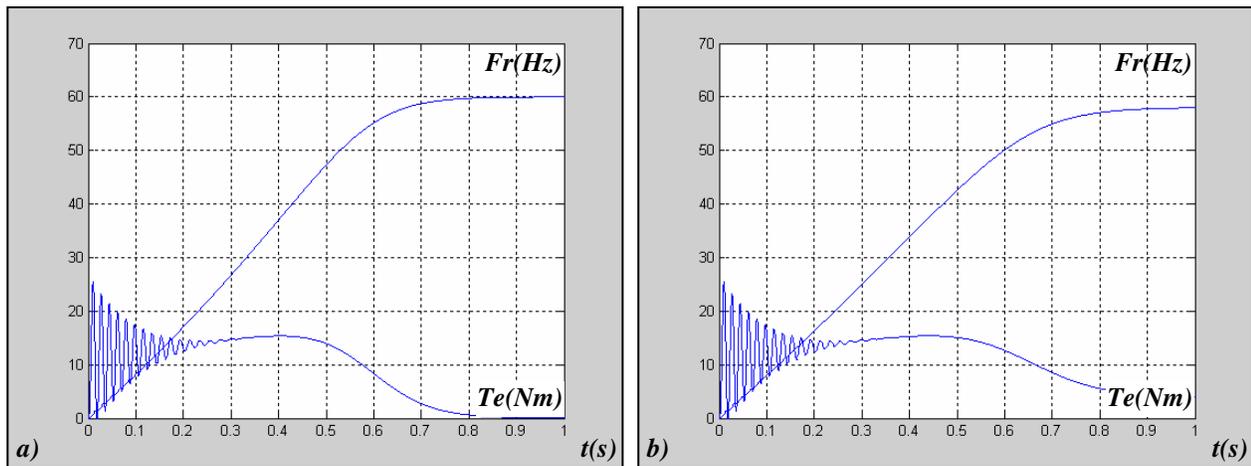
Este gerador síncrono possui momento de inércia  $J_c=0.045\text{kgm}^2$  e gera um conjugado de carga  $T_c$  controlado através da corrente de excitação do circuito de campo do gerador. A corrente de excitação  $I_c$  é ajustada para proporcionar um conjugado de carga  $T_c$  equivalente ao conjugado nominal do motor de indução, operando com frequência nominal de *60Hz*.

Apresentam-se a seguir os resultados da simulação computacional do motor de indução ligado direto na rede de alimentação e acoplado ao gerador síncrono. Considera-se para efeito de simulação duas situações distintas para operação do conjunto motor-gerador.

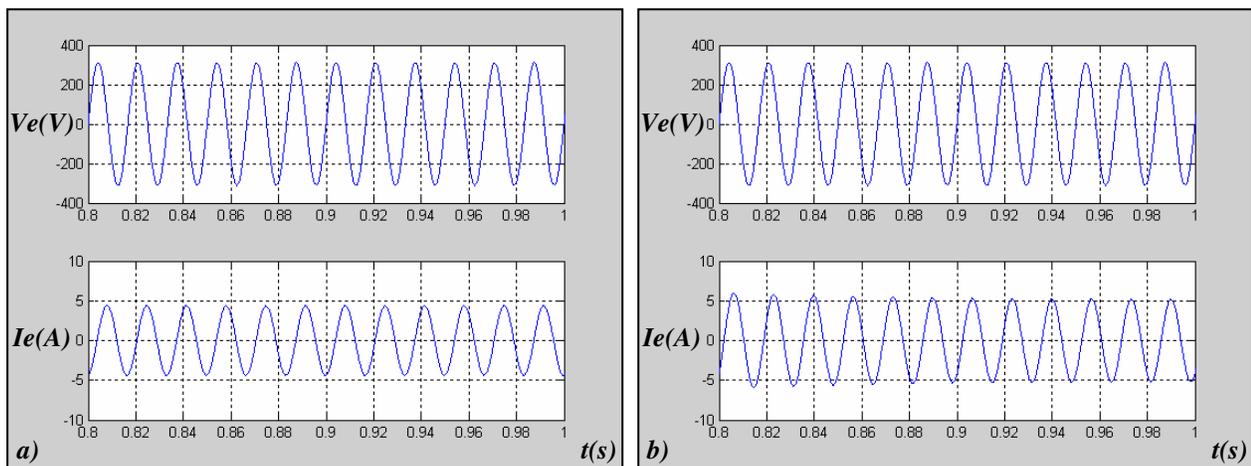
- a) gerador síncrono operando *sem* corrente de excitação no circuito de campo.
- b) gerador síncrono operando *com* corrente de excitação no circuito de campo.



**Figura 3.14** Corrente de Partida do Motor de Indução



**Figura 3.15** Conjugado de Partida do Motor de Indução



**Figura 3.16** Tensão e Corrente de Regime do Motor de Indução

### 3.5.2 Simulação do Motor de Indução com Inversor

Os modelos matemáticos para simulação computacional foram elaborados segundo o desenvolvimento apresentado no *Apêndice 3* e implementados em microcomputador usando um aplicativo computacional desenvolvido com o programa *MATLAB™*.

O modelo matemático do inversor de tensão usado neste trabalho é baseado no circuito integrado *Philips-4752*, especialmente desenvolvido para o acionamento de motores de indução trifásicos, capaz de gerar os pulsos para chaveamento dos dispositivos semicondutores.

O circuito integrado *Philips-4752* utiliza uma modulação por largura de pulsos de borda dupla fornecendo um conjunto de pulsos complementares aos seis elementos do circuito inversor com uma seqüência de fase apropriada capaz de produzir uma tensão trifásica simétrica.

Trata-se de um circuito digital onde a freqüência de chaveamento é sempre um múltiplo inteiro da freqüência sintetizada pelo inversor. Isto resulta em um excelente balanceamento de fase e tensão reduzindo as perdas nos motores de indução trifásicos.

O modelo matemático desenvolvido neste trabalho reproduz as principais características operacionais do inversor de tensão utilizado na implementação prática do sistema de controle e acionamento do motor de indução trifásico acoplado ao gerador síncrono.

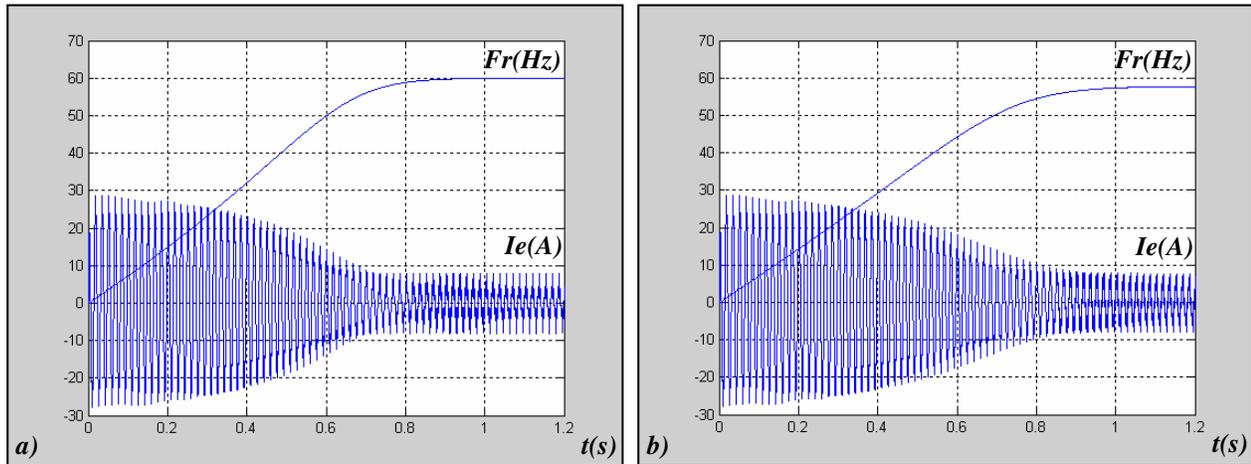
No laboratório foi utilizado um inversor de tensão do tipo *PWM*, fabricado pela empresa *Telemecanique*, modelo *ALTIVAR*, com uma tensão nominal de *220V*, potência nominal *3.0kW*, freqüência nominal *60Hz* e freqüência de saída variável de *7Hz* até *85Hz*.

Nos ensaios realizados em laboratório, este inversor de tensão foi capaz de suportar uma corrente de partida de até *16A*, acionando o conjunto motor-gerador descrito anteriormente, como pode ser verificado nos resultados práticos apresentados no *Capítulo 4*.

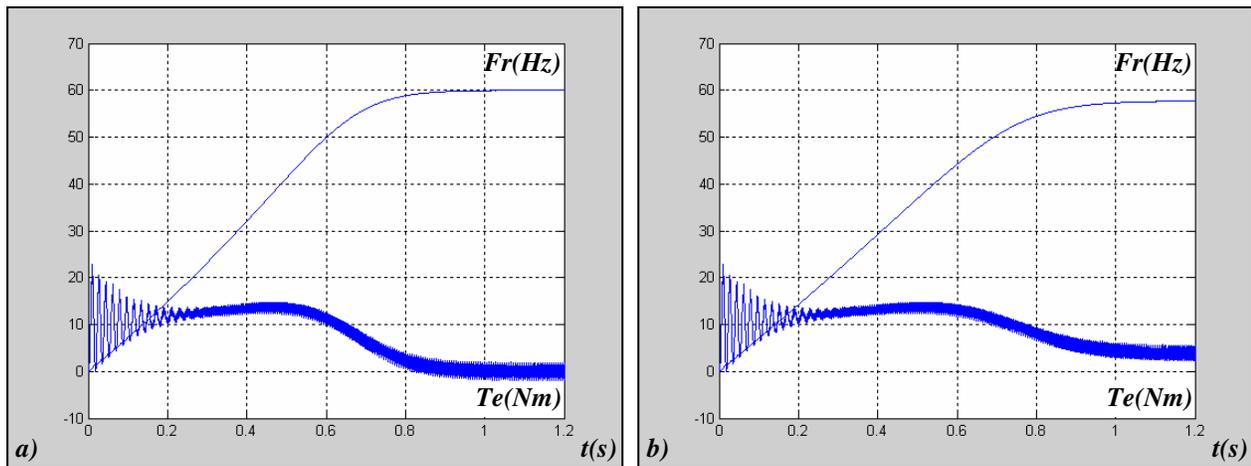
Para efeito de comparação a simulação computacional do inversor de tensão admite a possibilidade de realizar uma partida em *60Hz*. Na prática esta operação não pode ser realizada pois o inversor não suporta a corrente máxima de partida do motor de indução.

Apresentam-se a seguir os resultados da simulação computacional do motor de indução acoplado ao gerador síncrono e acionado pelo inversor de tensão. Durante a operação do conjunto motor-gerador duas situações distintas foram consideradas:

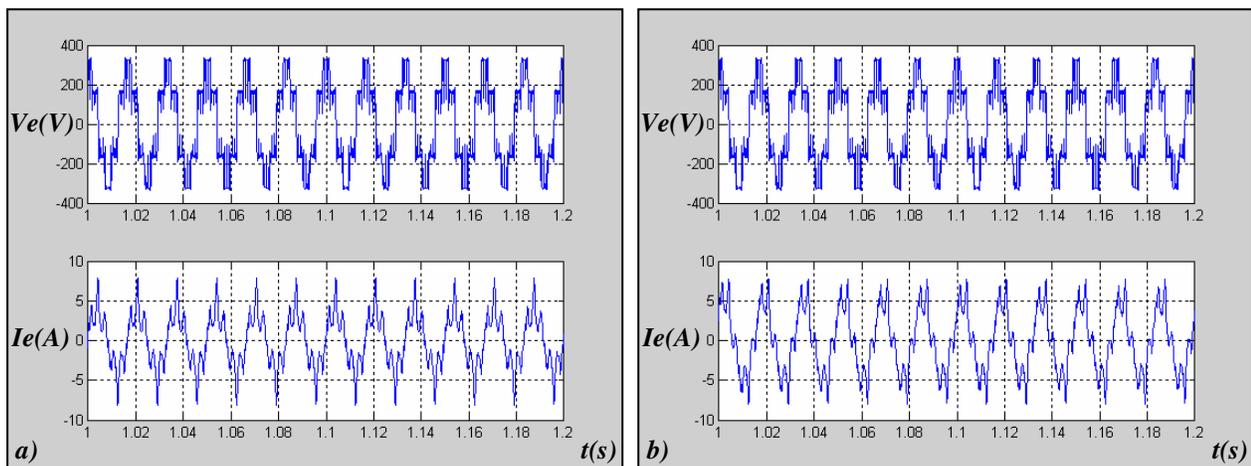
- a) gerador síncrono operando *sem* corrente de excitação no circuito de campo.
- b) gerador síncrono operando *com* corrente de excitação no circuito de campo.



**Figura 3.17** Corrente de Partida do Motor de Indução com Inversor



**Figura 3.18** Conjugado de Partida do Motor de Indução com Inversor



**Figura 3.19** Tensão e Corrente de Regime do Motor de Indução com Inversor

### 3.5.3 Simulação do Sensor de Corrente

Todas as estratégias de controle implementadas neste trabalho dependem do sensor de corrente, o qual fornece um sinal analógico equivalente ao valor eficaz da corrente de fase do motor de indução. Este sinal é obtido através de um circuito eletrônico muito simples, composto de sensores de efeito *Hall*, retificadores de precisão usando amplificadores operacionais e filtros capacitivos. Apresentam-se a seguir alguns resultados da simulação computacional do sensor de corrente, considerando o motor de indução ligado à rede de alimentação.

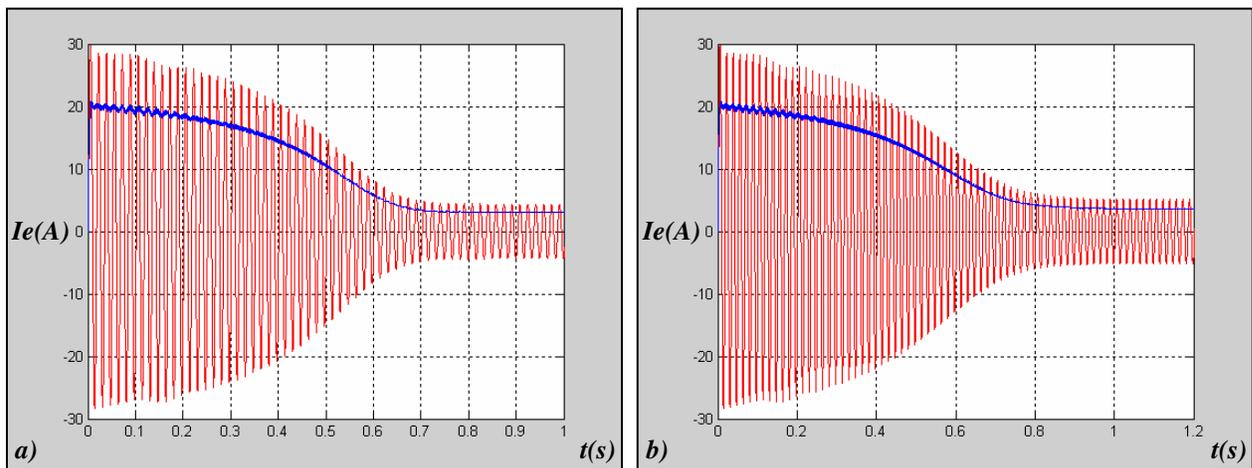


Figura 3.20 Resposta de Sensor de Corrente na Partida

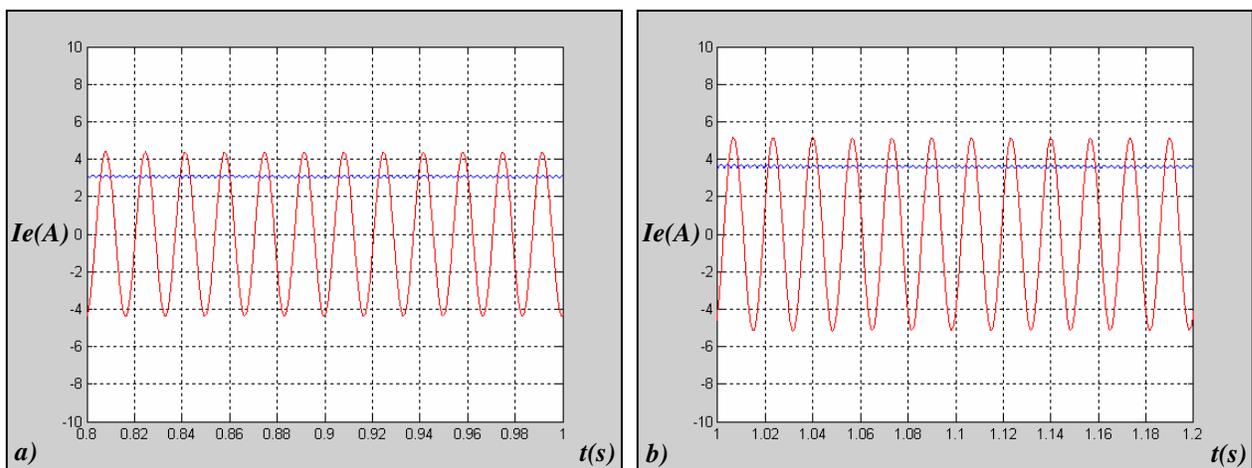
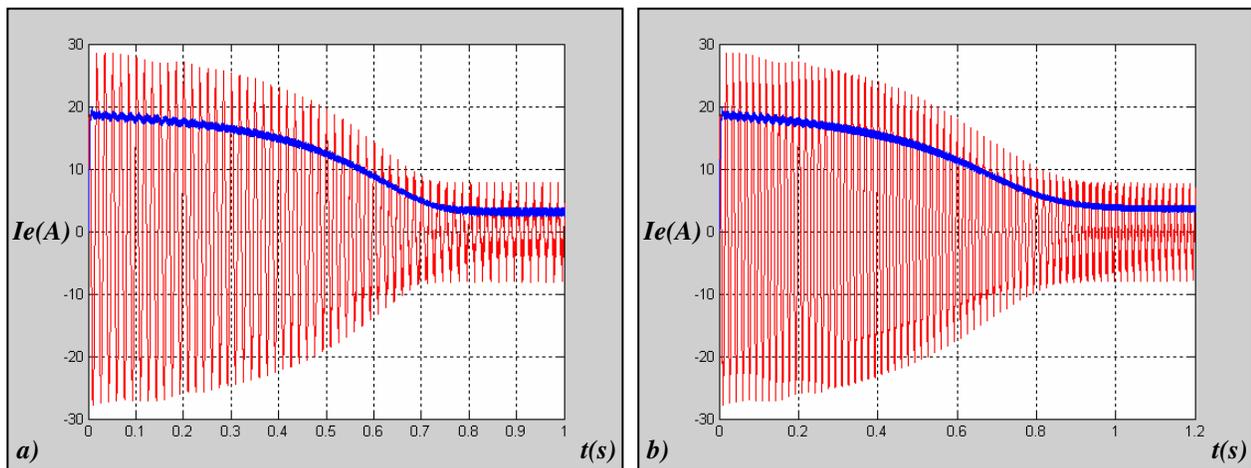


Figura 3.21 Resposta de Sensor de Corrente em Regime

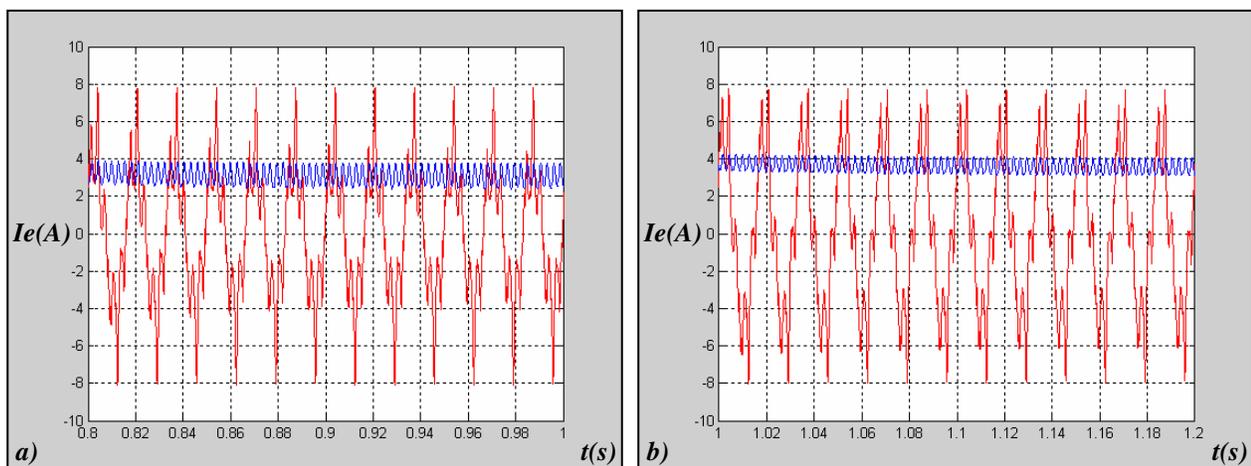
- a) gerador síncrono operando *sem* corrente de excitação no circuito de campo.
- b) gerador síncrono operando *com* corrente de excitação no circuito de campo.

### 3.5.4 Simulação do Sensor de Corrente com Inversor

O tempo de resposta do sensor de corrente é aproximadamente  $20ms$ . Este valor garante que o sistema de controle operando em malha fechada atue como um circuito de proteção capaz de limitar a corrente de entrada  $I_e$  do motor de indução e impedir o desligamento desnecessário do sistema de controle, tendo como referência, valores múltiplos da corrente nominal  $I_n$  do motor de indução. Apresentam-se a seguir alguns resultados da simulação computacional do sensor de corrente, considerando o motor de indução acionado pelo inversor de tensão  $PWM$ .



**Figura 3.22** Resposta de Sensor de Corrente na Partida com Inversor



**Figura 3.23** Resposta de Sensor de Corrente em Regime com Inversor

- a)** gerador síncrono operando *sem* corrente de excitação no circuito de campo.
- b)** gerador síncrono operando *com* corrente de excitação no circuito de campo.

### 3.5.5 Motor de Indução com Perfil de Frequência Senoidal

Para poder avaliar o comportamento dinâmico do motor de indução acoplado ao gerador síncrono acionado pelo inversor de tensão, foi aplicada uma tensão de alimentação com perfil de frequência do tipo senoidal. A partir dos resultados obtidos através de simulação computacional pode-se verificar a validade dos modelos matemáticos utilizados neste trabalho. Apresentam-se a seguir as curvas de frequência do estator  $F_e$  (campo girante), frequência do rotor  $F_r$ , frequência de escorregamento  $F_s = F_e - F_r$ , tensão de entrada  $V_e$  e corrente de entrada  $I_e$ .

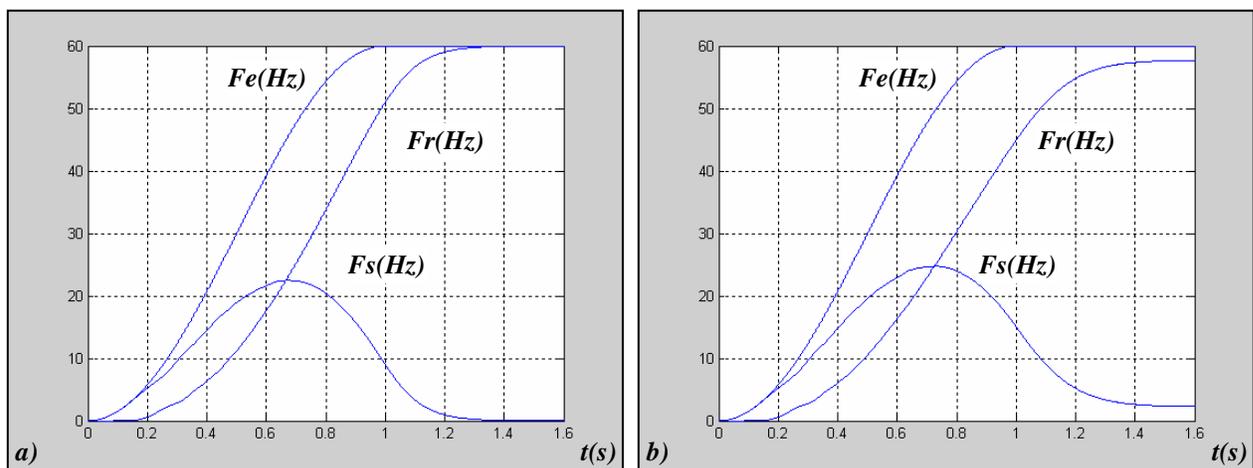


Figura 3.24 Perfil de Frequência Senoidal com Inversor

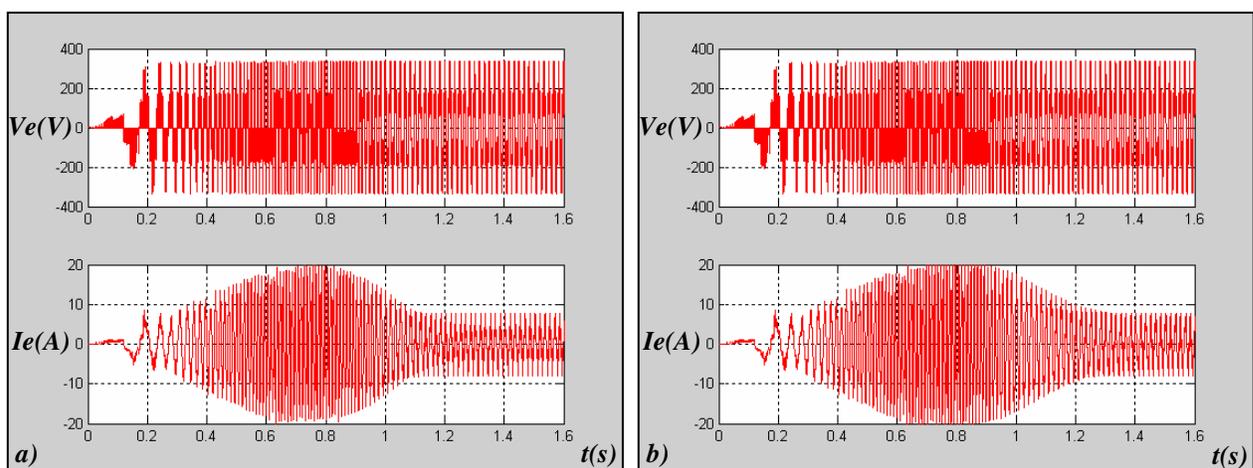
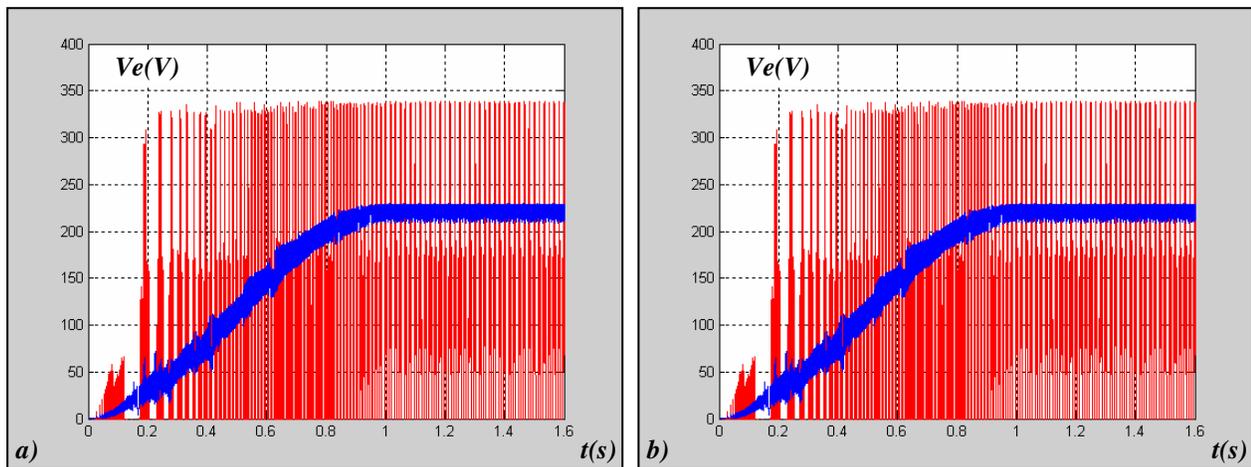


Figura 3.25 Tensão e Corrente do Motor de Indução com Inversor

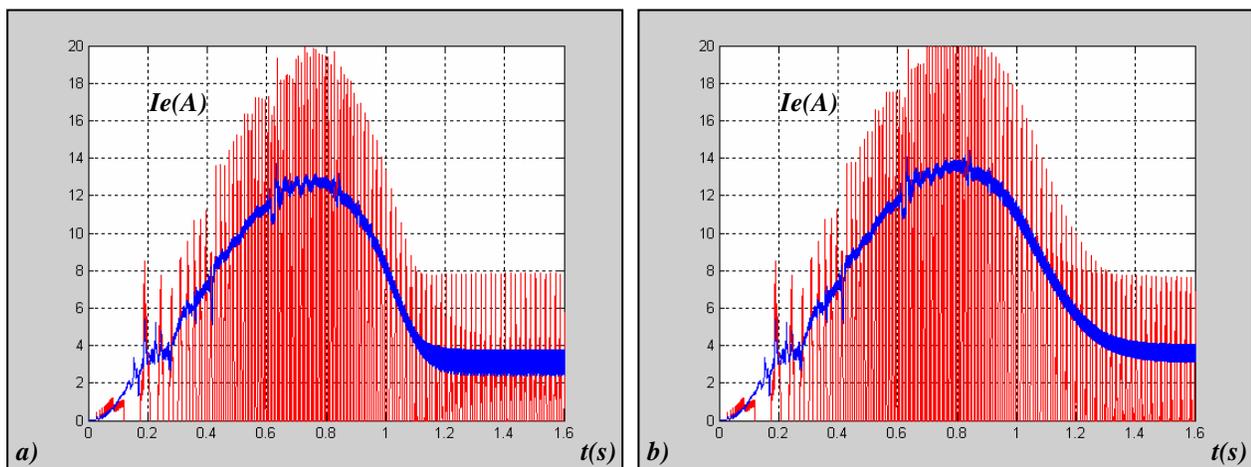
- a) gerador síncrono operando *sem* corrente de excitação no circuito de campo.
- b) gerador síncrono operando *com* corrente de excitação no circuito de campo.

### 3.5.6 Sensor de Corrente com Perfil de Frequência Senoidal

O circuito eletrônico que faz parte do sensor de corrente utilizado neste trabalho, opera como um detector de envoltória do sinal de corrente, afetado por um ganho operacional ajustável e devidamente calibrado, para fornecer um sinal analógico equivalente ao valor eficaz da corrente de entrada do motor de indução trifásico acionado por inversor de tensão *PWM*. Apresentam-se a seguir as formas de onda de tensão e corrente de entrada com seus respectivos sinais analógicos equivalentes aos valores eficazes obtidos através de simulação computacional.



**Figura 3.26** Tensão de Entrada do Motor de Indução com Inversor



**Figura 3.27** Corrente de Entrada do Motor de Indução com Inversor

- a)** gerador síncrono operando *sem* corrente de excitação no circuito de campo.
- b)** gerador síncrono operando *com* corrente de excitação no circuito de campo.

## Capítulo 4

### Implementação do Sistema de Controle

#### 4.1 Introdução

Devido à necessidade de avaliar a estratégia de controle proporcional integral derivativo e as novas estratégias de controle nebuloso e neuro-nebuloso, foi implementado em laboratório um sistema de controle digital para acionamento de um motor de indução acoplado a um gerador síncrono, utilizando um inversor de tensão do tipo *PWM*.

Este sistema de controle e acionamento permitiu a realização de ensaios operacionais de partida e regime com conjugado de carga ajustável. A aquisição de dados em tempo real garante a possibilidade de efetuar a aquisição e o armazenamento da corrente de entrada, do conjugado de carga e da frequência de rotação do conjunto motor-gerador.

Apresenta-se na **Figura 4.1** o diagrama em blocos do sistema de controle e acionamento implementado em laboratório detalhando todos os elementos necessários para testar, analisar e avaliar as estratégias de controle apresentadas no **Capítulo 3**.

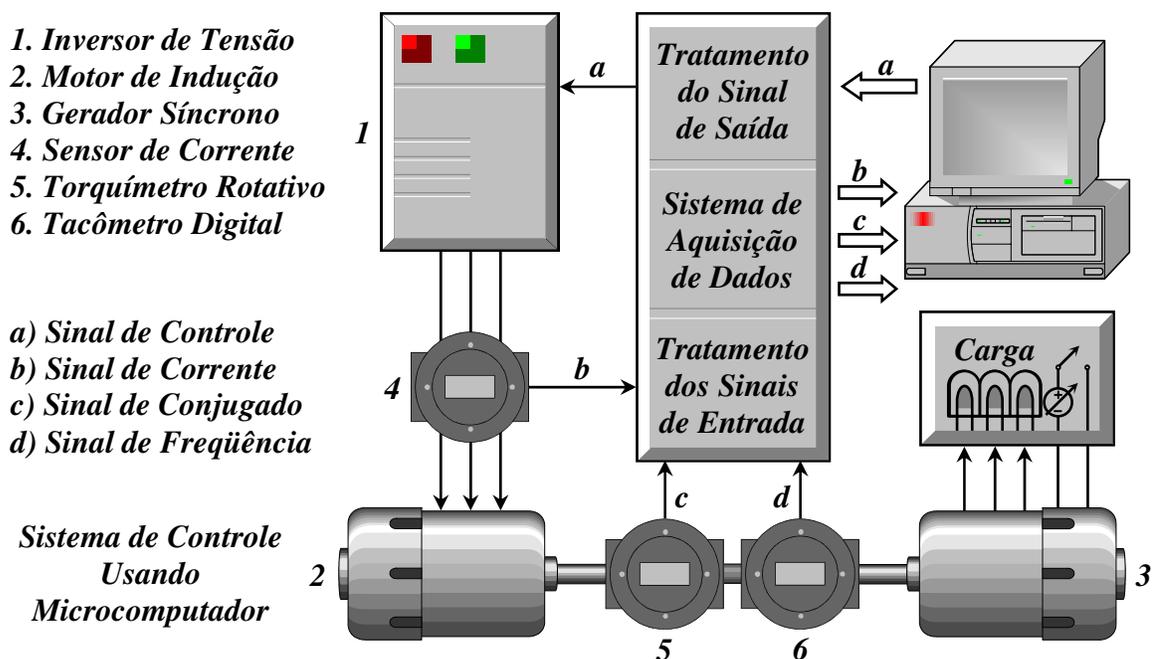


Figura 4.1 Diagrama em Blocos do Sistema de Controle

## 4.2 *Descrição do Sistema de Controle*

O sistema de controle implementado em laboratório é composto por um inversor de tensão, duas máquinas elétricas trifásicas (o motor de indução e o gerador síncrono), um sensor de corrente, um torquímetro rotativo e um tacômetro digital. Ainda fazem parte do sistema os circuitos para tratamento de sinais, aquisição de dados e execução de algoritmos de controle. Apresenta-se a seguir a descrição dos elementos que integram o sistema de controle.

### 4.2.1 *Inversor de Tensão*

No laboratório foi utilizado um inversor de tensão do tipo *PWM*, fabricado pela empresa *Telemecanique*, modelo *ALTIVAR*, com uma tensão nominal de *220V*, potência nominal *3.0kW*, frequência nominal *60Hz* e frequência de saída variável de *7Hz* até *85Hz*. Este inversor de tensão é capaz de suportar uma corrente de partida de aproximadamente *16A*.

### 4.2.2 *Motor de Indução*

Um motor de indução trifásico fabricado pela empresa *Weg*, modelo *80-1190* de *4* pólos, foi usado nos ensaios realizados em laboratório. Os dados do motor de indução são os seguintes: potência nominal  $P_n=1.0Cv$ , tensão nominal  $V_n=220V$ , conjugado nominal  $T_n=4.0Nm$ , corrente nominal  $I_n=3.8A$ , frequência nominal  $F_n=60Hz$  e momento de inércia  $J_o=0.0015kgm^2$ .

### 4.2.3 *Gerador Síncrono*

Um alternador trifásico fabricado pela empresa *Kohlbach*, modelo *FK-600* com *4* pólos, foi utilizado como carga acoplada ao eixo do motor de indução. Os dados do gerador síncrono são os seguintes: tensão nominal de *220V*, corrente nominal *10.5A*, potência nominal de *4.0kVA*, frequência nominal de *60Hz*, fator de potência *0.85* e rendimento de *85%*.

Este alternador trifásico possui momento de inércia  $J_c=0.045kgm^2$  e gera um conjugado de carga  $T_c$  proporcional à frequência de rotação  $F_r$ , controlado pela corrente de excitação  $I_c$ , a qual pode ser ajustada para gerar um conjugado de carga  $T_c$  equivalente ao conjugado nominal do motor de indução, operando com frequência nominal de *60Hz*.

#### 4.2.4 *Sensor de Corrente*

O sensor de corrente usado neste trabalho foi fabricado pela empresa *Nana Electronics*, modelo *NNC-20CTA*. Trata-se de um sensor de corrente de efeito *Hall* com circuito amplificador alimentado por tensão simétrica de *15V*, capaz de medir uma corrente instantânea de até *100A* e fornecer um sinal de saída de até *4.0V*, com tempo de resposta de *15μs* e precisão de *1%*.

#### 4.2.5 *Torquímetro Rotativo*

Trata-se de um equipamento fabricado pela empresa *Sodmex*, modelo *MT-103*, capaz de medir o torque com a deformação de dispositivos *Strain Gage* ligados em ponte de *Wheatstone*. Principais características: escala nominal de *50Nm*, tensão de alimentação de *10V*, consumo de corrente máximo de *30mA*, tensão de saída nominal de *20mV* e rotação máxima de *2000rpm*.

#### 4.2.6 *Tacômetro Digital*

O tacômetro digital usado para medir frequência de rotação, foi fabricado pela *Hohner*, modelo *BC* tipo incremental, capaz de gerar *1000* pulsos ópticos por rotação, com uma tensão de alimentação de *5V*, consumo de corrente máximo de *90mA*, rotação máxima de *6000rpm* e um sinal de saída do tipo onda quadrada padrão *TTL*, com frequência máxima de *100kHz*.

### 4.3 *Circuitos Condicionadores de Sinais*

Para operação normal do sistema de controle é necessário o tratamento dos sinais de entrada provenientes dos elementos sensores e o tratamento do sinal de saída aplicado na entrada de controle do inversor de tensão. Estes sinais devem ser devidamente amplificados e equalizados para compatibilizar as características de entrada e saída dos elementos do sistema.

Um circuito dedicado efetua o condicionamento dos sinais de entrada para que o sistema de aquisição de dados possa converter os sinais analógicos em sinais digitais equivalentes. De modo análogo, o sistema de aquisição converte o sinal digital de saída em um sinal analógico que devidamente condicionado atua como variável de controle do inversor de tensão. Apresentam-se a seguir as características dos sinais de entrada e saída do sistema de controle.

### 4.3.1 Sinal de Controle

A execução de um algoritmo de controle específico provoca uma atualização periódica da variável de controle, a qual deve ser representada internamente com pelo menos **16** bits para garantir a precisão do algoritmo de controle. Trata-se de um sinal digital que deve ser convertido para um sinal analógico através de um conversor *D/A* de **12** bits, antes de ser usado como um sinal de controle do circuito inversor de tensão trifásico.

Devido às características desfavoráveis da entrada de controle do circuito inversor de tensão trifásico que possui baixa impedância de entrada, foi necessário implementar um circuito para tratamento do sinal de controle com resposta totalmente linear, capaz fornecer uma corrente de até **500mA** e isolar ruídos de chaveamento com picos de até **50V**.

Assim, o sinal de controle foi isolado galvanicamente através do acoplador óptico **4N-25** e reconstituído usando o amplificador operacional **TCA-0372**. Para garantir a linearidade do sinal de controle, o sinal analógico aplicado na entrada do acoplador óptico é recuperado na base do foto-transistor e amplificado para um valor proporcional à frequência da tensão sintetizada pelo circuito inversor na razão de **8Hz** ou **240rpm** por **Volt**.

### 4.3.2 Sinal de Conjugado

Para medir o torque no eixo do motor de indução foi utilizado um torquímetro rotativo que fornece um sinal analógico de **20mV** equivalente a um conjugado nominal de **50Nm**. Devido ao baixo valor de tensão foi necessário implementar um circuito amplificador diferencial para tratamento do sinal analógico usando o amplificador operacional **LF-353**.

O sinal de conjugado não foi usado no sistema de controle, pois o motor de indução apresenta um conjugado nominal de **4.0Nm** e um com conjugado máximo de partida de **12Nm**, enquanto que o torquímetro rotativo possui um fundo de escala de **50Nm**. Trata-se de um sinal de baixa qualidade proporcional ao conjugado instantâneo na razão de **200mV** por **Nm**.

Portanto, o sinal de conjugado foi usado apenas para avaliar o comportamento do motor de indução trifásico e comparar com os resultados obtidos em simulação computacional, usando os modelos matemáticos desenvolvidos neste trabalho, considerando o motor de indução ligado na rede de alimentação ou acionado pelo inversor de tensão.

### 4.3.3 Sinal de Corrente

O sinal de corrente obtido através do sensor de efeito *Hall* é diretamente proporcional à ao valor instantâneo da corrente de fase motor de indução. Este sinal analógico passa por um circuito condicionador composto de um amplificador-retificador de precisão, implementado com o amplificador operacional *LF-353* e em seguida passa por um simples filtro capacitivo de  $47\mu F$ , para diminuir o ruído provocado pelo chaveamento do circuito inversor de tensão.

Este sinal de corrente é fundamental para operação do sistema de controle digital que possui uma malha de controle de corrente atuando como um circuito de proteção ativo, limitando a corrente de entrada do motor de indução e evitando o desligamento desnecessário do sistema de acionamento que tem como função principal o controle da frequência de rotação.

Portanto, com tempo de resposta de aproximadamente  $20ms$ , temos na saída do circuito condicionador um sinal de tensão que corresponde à corrente de fase do motor de indução na razão de  $200mV$  por *Ampère*. Trata-se de um sinal analógico que deve ser convertido para um sinal digital equivalente, usando o conversor *A/D* de  $12$  bits da placa de aquisição.

### 4.3.4 Sinal de Frequência

Com a finalidade de medir e controlar a frequência de rotação *Fr* do motor de indução, foi utilizado um tacômetro digital com sinal de saída do tipo onda quadrada padrão *TTL*. Para traçar as curvas apresentadas nos resultados práticos foi implementado um circuito conversor de frequência para tensão, usando o circuito integrado *VFC-032* fabricado pela *Burr-Brown*.

O circuito conversor é capaz de fornecer um sinal de tensão proporcional à frequência de rotação, na razão de  $62.5mV$  por *Hz*. Este sinal de tensão é usado como sinal de realimentação na malha de controle de frequência, trata-se de um sinal analógico que deve ser convertido para um sinal digital equivalente, usando um conversor *A/D* de  $12$  bits da placa de aquisição.

Os sinais de corrente e frequência são utilizados em um sistema de controle digital que opera em malha fechada, monitorando em tempo real a corrente de entrada e a frequência de rotação, procurando anular as diferenças em relação à frequência desejada e limitar o valor da corrente de entrada. As estratégias de controle apresentadas neste trabalho não utilizam modelos matemáticos para determinar as condições de operação do motor de indução.

## 4.4 Sistema de Aquisição de Dados

No sistema de controle implementado em laboratório, utiliza-se um microcomputador equipado com placa de aquisição de dados para executar os algoritmos de controle, no entanto, o sistema de controle foi concebido para ser implementado usando um sistema dedicado baseado em microcontroladores, equipado com circuitos condicionadores e conversores de sinais.

Inicialmente, foi implementado em laboratório o protótipo de um sistema de controle dedicado utilizando os seguintes circuitos integrados: microcontrolador *MCS-80C31*, memórias semicondutoras *28C256* e *62C256*, decodificador *74LS373*, conversores *ADC-0804*, *DAC-7801* e *VFC-032*, amplificadores operacionais *LF-353* e *TCA-0372* e acoplador óptico *4N-25*.

O primeiro algoritmo testado no protótipo foi o algoritmo com estratégia de controle do tipo *PID*, que não apresentou resultados satisfatórios devido a dificuldade de alterar e avaliar sistematicamente o algoritmo de controle. Portanto, para tornar o sistema de controle digital mais flexível, a solução foi utilizar um microcomputador equipado com placa de aquisição de dados e elaborar os algoritmos de controle emulando as funções básicas do circuito protótipo.

O microcomputador utilizado neste trabalho estava equipado com um microprocessador *Pentium III* de *550MHz*, sistema operacional *Windows 98-SE* e uma placa de aquisição de dados *Advantech* modelo *PCL-711*. A linguagem de programação usada na elaboração dos algoritmos de controle foi a linguagem *C++* que permite trabalhar diretamente com as variáveis na forma binária, emulando as operações lógicas e aritméticas do microcontrolador *MCS-80C31*.

### 4.4.1 Placa de Aquisição de Dados

A placa de aquisição de dados que equipa o microcomputador utilizado neste trabalho, foi fabricada pela *Advantech*, modelo *PCL-711*, com barramento de comunicação *ISA* de 8 bits, que funciona a partir da leitura e escrita em endereços específicos de entrada e saída. Portanto, com algumas linhas de programação é possível acessar todos os recursos da placa.

Esta placa de aquisição de dados utiliza 16 endereços consecutivos de entrada e saída, que permitem acessar todos os recursos da placa de aquisição, com endereço base configurado a partir de pequenas chaves lógicas. Apresentam-se na *Tabela 4.1* os recursos disponíveis na placa de aquisição de dados e as características básicas das suas entradas e saídas.

**Tabela 4.1 Características da Placa de Aquisição de Dados**

<i>PCL-711</i>	<i>Entradas Analógicas</i>	<i>Saída Analógica</i>
<i>Número de Canais</i>	<i>08</i>	<i>01</i>
<i>Número de Bits</i>	<i>12</i>	<i>12</i>
<i>Tipo de Conversor</i>	<i>AD-574</i>	<i>AD-7541</i>
<i>Tempo de Conversão</i>	<i>25<math>\mu</math>s</i>	<i>30<math>\mu</math>s</i>
<i>Faixa de Operação</i>	<i>-5V a 5V</i>	<i>0 a 10V</i>
<i>PCL-711</i>	<i>Entradas Digitais</i>	<i>Saídas Digitais</i>
<i>Número de Canais</i>	<i>16</i>	<i>16</i>
<i>Padrão de Sinal</i>	<i>TTL</i>	<i>TTL</i>

Obs.: Nas entradas analógicas são considerados apenas os 8 bits mais significativos.

#### 4.4.2 Sinais de Entrada e Saída

Os sinais analógicos de entrada da placa de aquisição de dados são: conjugado de carga, corrente de entrada e frequência de rotação. Já o sinal analógico de saída corresponde à variável de controle do circuito inversor de tensão. Estes sinais são devidamente isolados, amplificados e equalizados, através de circuitos condicionadores que definem a razão operacional e o fundo de escala, adequados para operação de controle em malha fechada. Apresentam-se na **Tabela 4.2** as características básicas dos sinais de entrada e saída da placa de aquisição de dados.

**Tabela 4.2 Características dos Sinais de Entrada e Saída**

<i>Sinal Analógico</i>	<i>Razão Operacional</i>	<i>Fundo de Escala</i>	
<i>Conjugado</i>	<i>200mV por Newton metro</i>	<i>5V</i>	<i>25Nm</i>
<i>Corrente</i>	<i>200mV por Ampère</i>	<i>5V</i>	<i>25A</i>
<i>Frequência</i>	<i>62.5mV por Hertz</i>	<i>5V</i>	<i>80Hz</i>
<i>Controle</i>	<i>8Hz ou 240rpm por Volt</i>	<i>10V</i>	<i>2400rpm</i>

Obs.: O sinal de conjugado não é utilizado para operação em malha fechada.

## 4.5 Apresentação de Resultados

Nesta seção, apresentam-se os resultados obtidos nas simulações computacionais e nos ensaios práticos realizados em laboratório. Estes resultados permitem avaliar o comportamento do motor de indução trifásico, operando com diferentes situações de carga e comparar a resposta do sistema utilizando diferentes estratégias e parâmetros operacionais de controle.

O conjunto utilizado nos ensaios práticos realizados em laboratório é composto de um motor de indução trifásico, ligado direto na rede ou acionado por um inversor de tensão, sempre acoplado a um gerador síncrono conectado a uma carga resistiva representada por um conjunto de três lâmpadas incandescentes ligadas em triângulo, lâmpadas de **250W** com tensão de **220V**. Esta carga resistiva é capaz de dissipar até **750W** de potência, considerando que a tensão de saída do gerador síncrono trifásico determina a potência dissipada nas lâmpadas.

A tensão de saída do gerador é ajustada através da corrente de excitação do circuito de campo para proporcionar um conjugado de carga  **$T_c$**  equivalente ao conjugado nominal do motor de indução, operando com frequência nominal de **60Hz**. A corrente de excitação  **$I_c$**  é controlada através de uma fonte de tensão, podendo ser interrompida com auxílio de um circuito interruptor para que o motor de indução possa operar com carga nominal ou em vazio.

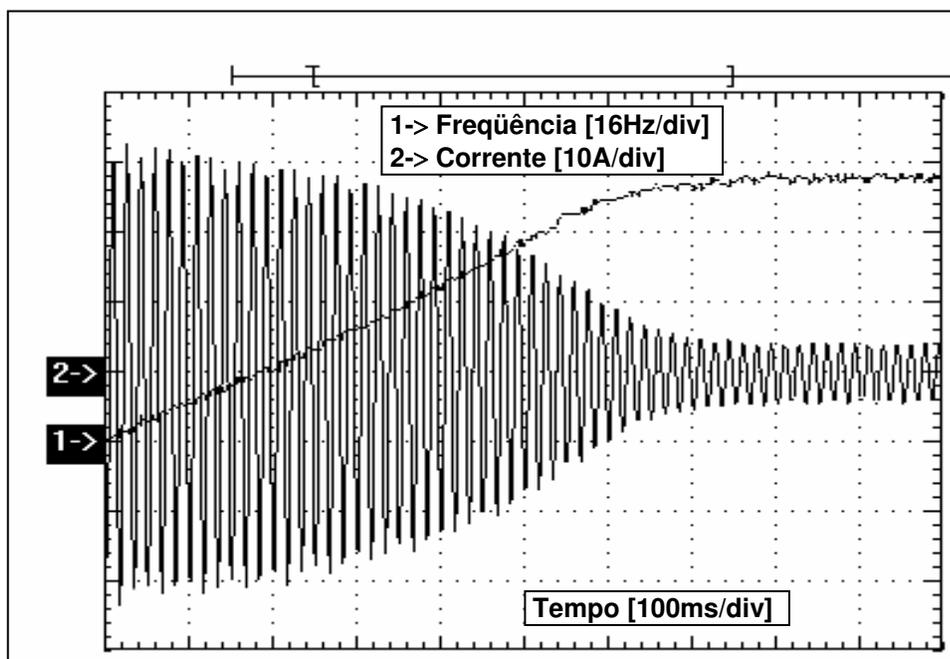
### 4.5.1 Motor de Indução e Gerador Síncrono

Apresenta-se na **Tabela 4.3** um resumo dos ensaios práticos realizados em laboratório, para verificar a resposta do sensor de corrente e o comportamento do motor de indução ligado direto na rede e acoplado ao gerador síncrono. Nos ensaios a seguir, considera-se o circuito de campo do gerador síncrono operando com ou sem corrente de excitação.

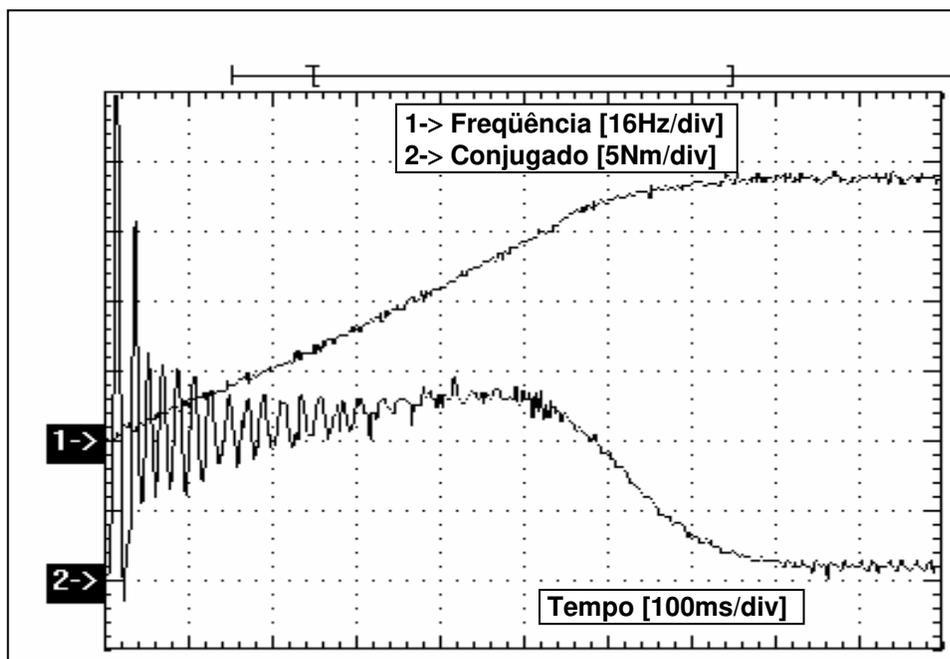
**Tabela 4.3 Ensaios com Motor de Indução e Gerador Síncrono**

<b>Gerador Síncrono</b>	<b>Resultados Práticos</b>		
<b>Corrente de Excitação</b>	<b>Característica do Ensaio</b>	<b>Código do Ensaio</b>	<b>Figuras</b>
<b>Ausente</b>	<b>Partida Direta</b>	<b>PD</b>	<b>4.2 e 4.3</b>
<b>Ausente</b>	<b>Corrente Eficaz</b>	<b>CE</b>	<b>4.4 e 4.5</b>
<b>Presente</b>	<b>Partida Direta</b>	<b>PDex</b>	<b>4.6 e 4.7</b>
<b>Presente</b>	<b>Corrente Eficaz</b>	<b>CEex</b>	<b>4.8 e 4.9</b>

**Ensaio PD - Partida Direta**  
**Motor de Indução Ligado Direto na Rede**  
**Gerador Síncrono sem Corrente de Excitação**

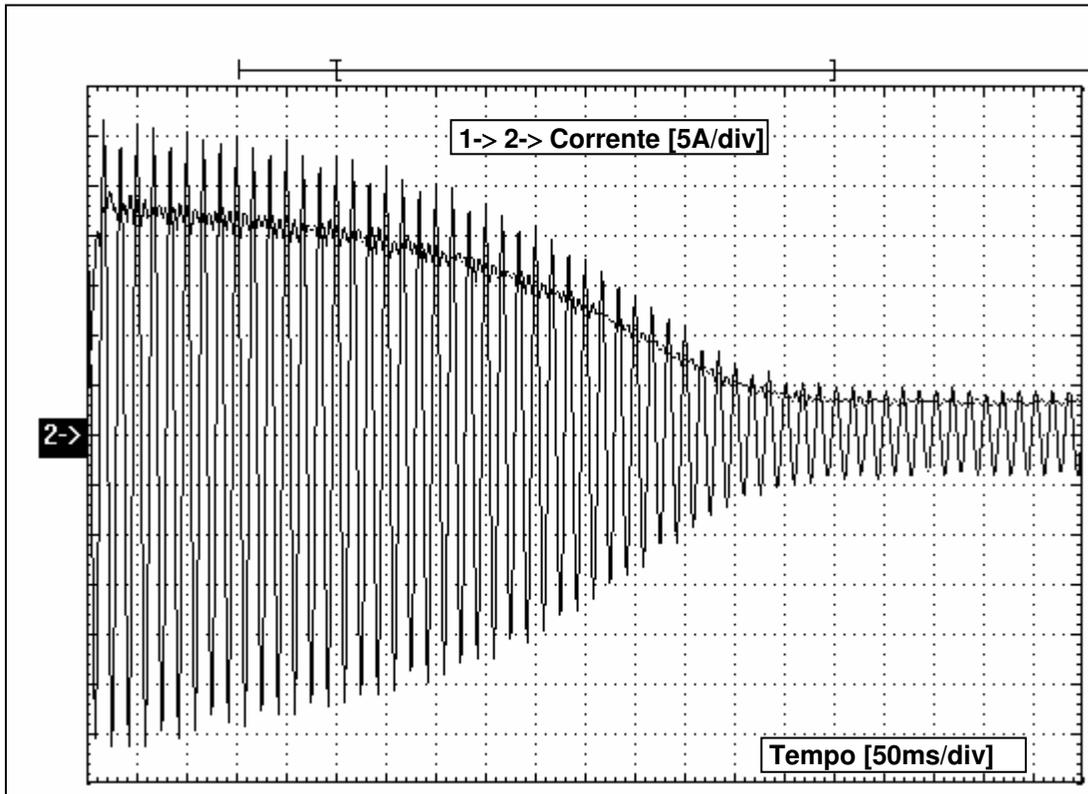


**Figura 4.2 Corrente de Partida e Frequência de Rotação (Ensaio PD)**

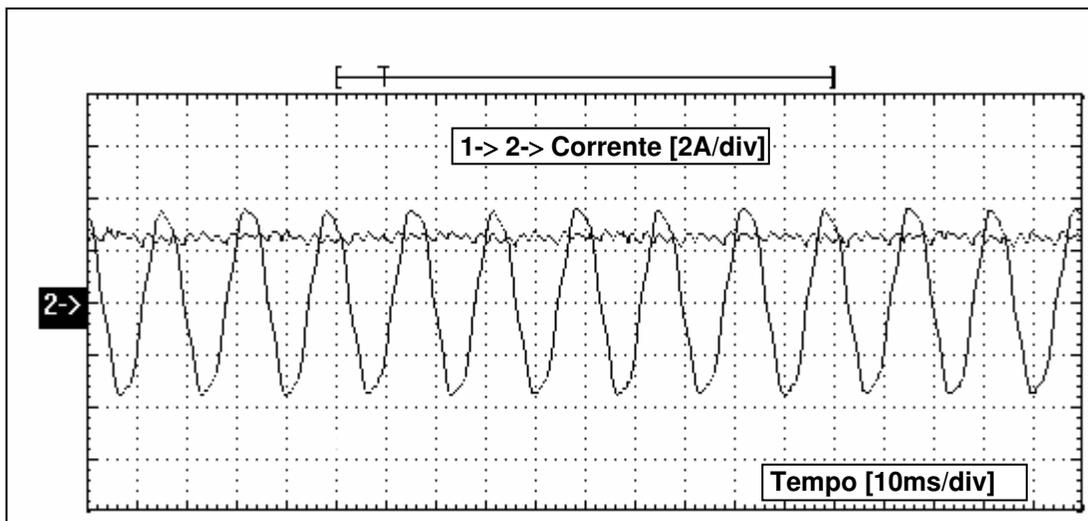


**Figura 4.3 Conjugado de Carga e Frequência de Rotação (Ensaio PD)**

*Ensaio CE - Corrente Eficaz*  
*Motor de Indução Ligado Direto na Rede*  
*Gerador Síncrono sem Corrente de Excitação*

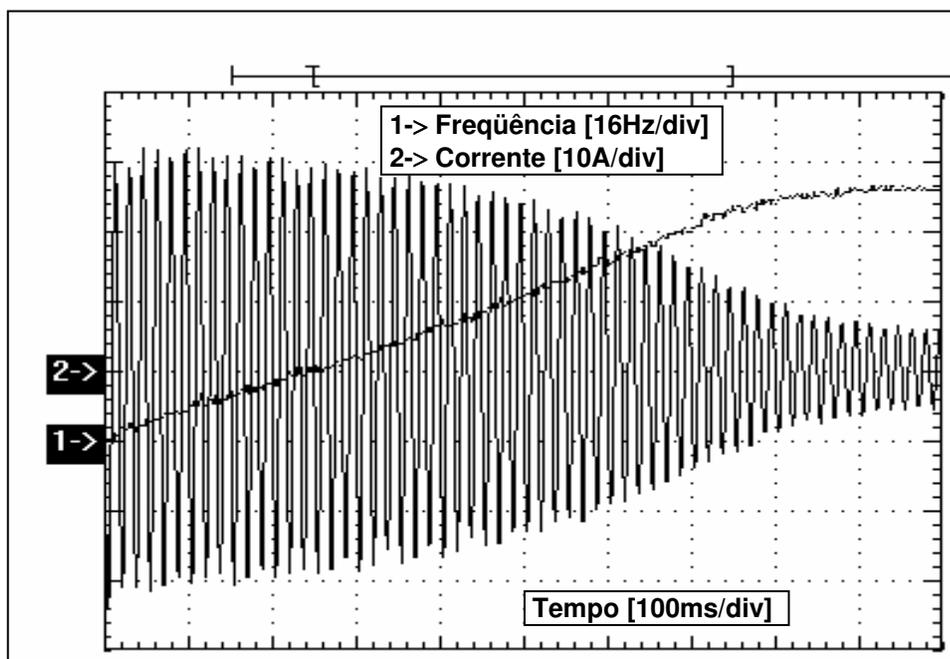


*Figura 4.4 Corrente de Partida: Valor Instantâneo e Valor Eficaz (Ensaio CE)*

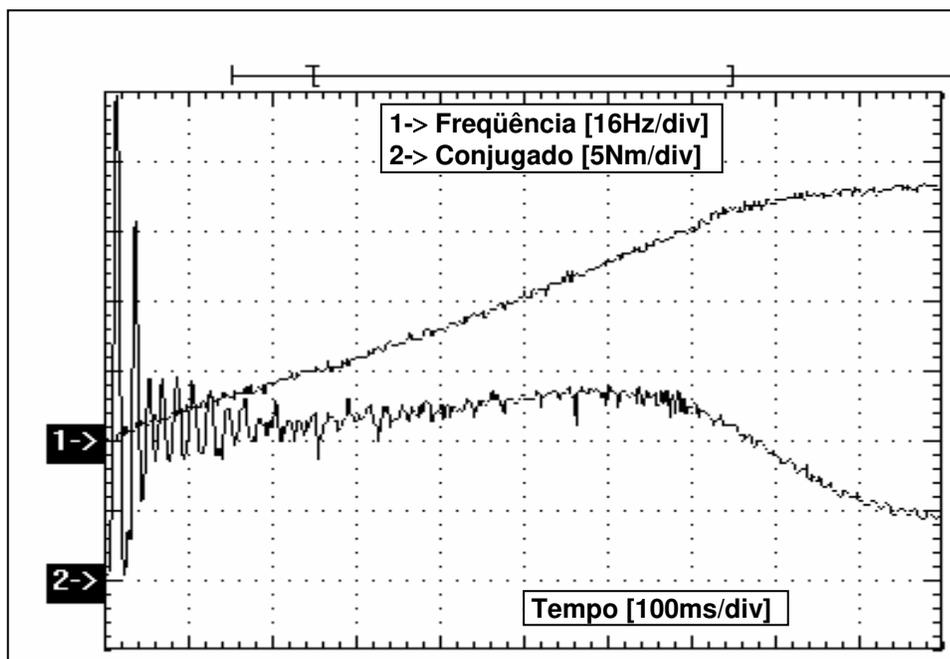


*Figura 4.5 Corrente de Regime: Valor Instantâneo e Valor Eficaz (Ensaio CE)*

**Ensaio PDex - Partida Direta**  
**Motor de Indução Ligado Direto na Rede**  
**Gerador Síncrono com Corrente de Excitação**

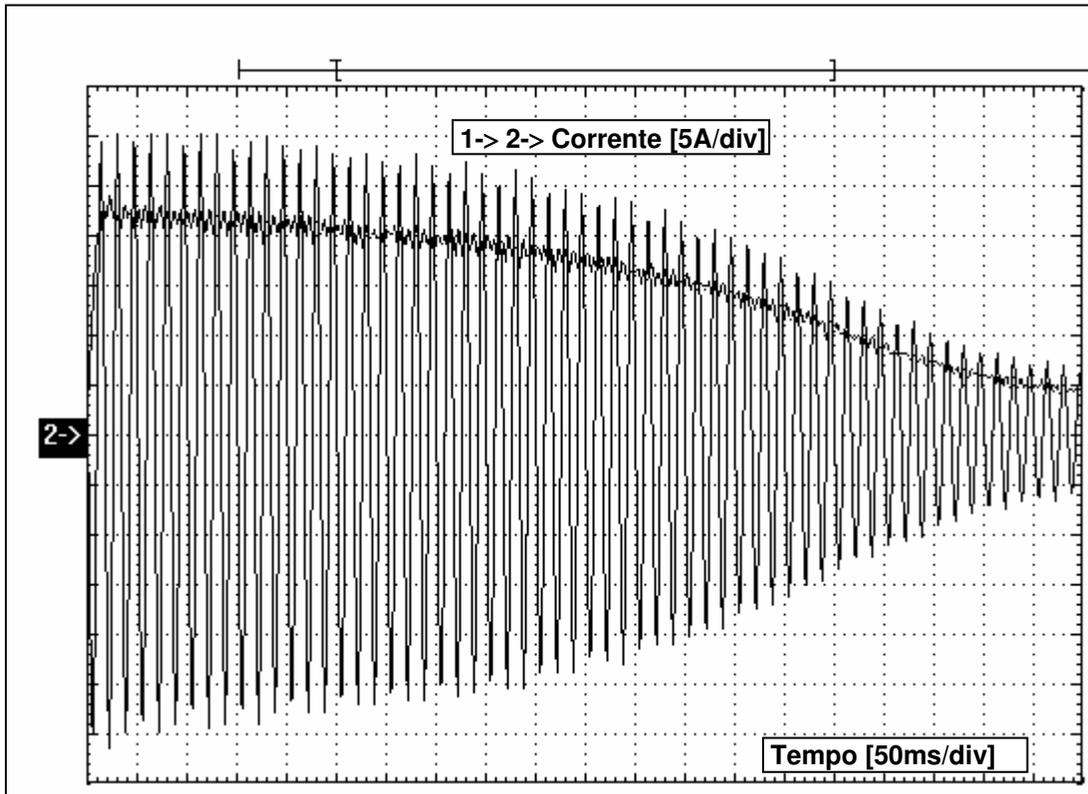


**Figura 4.6** Corrente de Partida e Frequência de Rotação (Ensaio PDex)

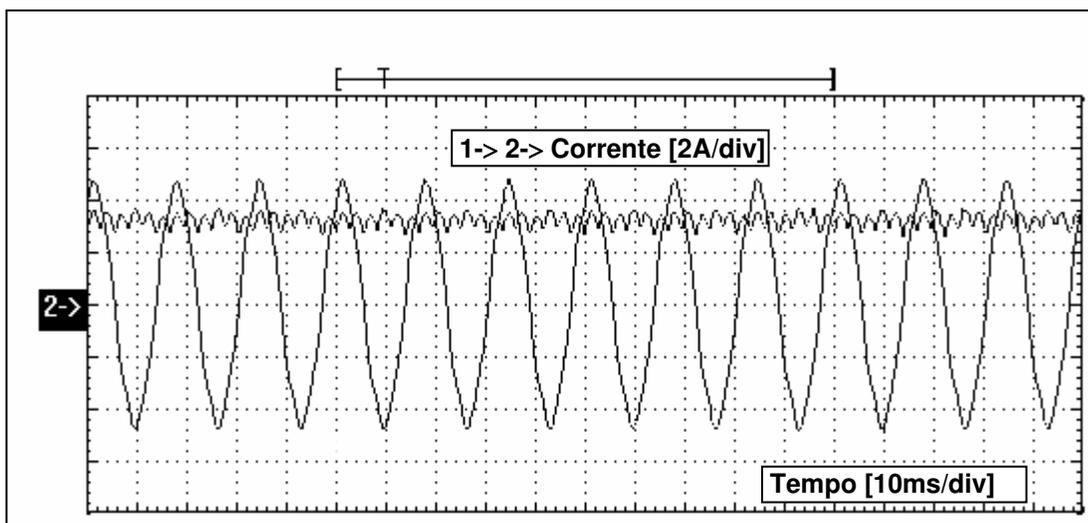


**Figura 4.7** Conjugado de Carga e Frequência de Rotação (Ensaio PDex)

*Ensaio CEex - Corrente Eficaz*  
*Motor de Indução Ligado Direto na Rede*  
*Gerador Síncrono com Corrente de Excitação*



*Figura 4.8 Corrente de Partida: Valor Instantâneo e Valor Eficaz (Ensaio CEex)*



*Figura 4.9 Corrente de Regime: Valor Instantâneo e Valor Eficaz (Ensaio CEex)*



Nas figuras a seguir apresentam-se resultados práticos e simulados usando estratégia de controle proporcional integral derivativa para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.11** até **Figura 4.16** considera-se o gerador síncrono trifásico operando sem corrente de excitação no circuito de campo, portanto, o motor de indução trabalha em vazio e o tempo de resposta em vazio é menor que o tempo de resposta com carga.

A **Figura 4.11** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.12** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada  **$I_e$** .

A **Figura 4.13** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.14** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada  **$I_e$** .

A **Figura 4.15** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.16** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada  **$I_e$** .

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento  **$F_s$**  representada pela diferença entre a frequência do estator  **$F_e$**  e a frequência do rotor  **$F_r$** .

A corrente de entrada  **$I_e$**  apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento  **$F_s$** , portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

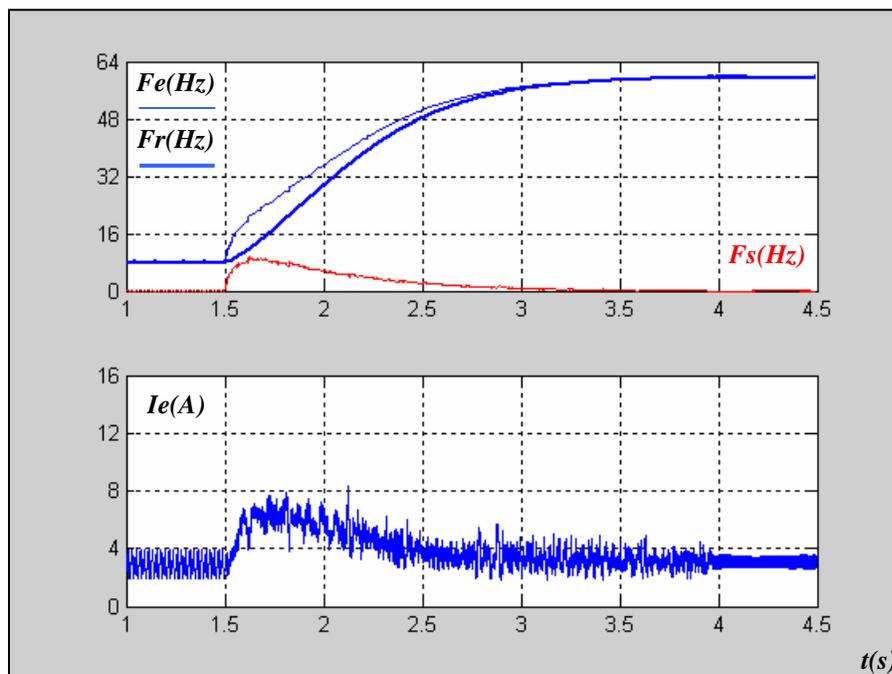
A estratégia de controle proporcional integral derivativa considera o gradiente do erro de frequência  **$\delta E_r$** , portanto, nos resultados obtidos pode-se observar uma ausência de sobre-sinal na frequência de rotação  **$F_r$**  e uma diminuição no tempo de resposta do motor de indução, quando comparado com as estratégias de controle do tipo proporcional integral.

O tempo de resposta do motor de indução é inversamente proporcional ao valor limite da corrente de entrada  **$I_e$** , assim, o tempo de resposta para um valor limite de **8A** é aproximadamente duas vezes maior que o tempo de resposta para um valor limite de **16A**.

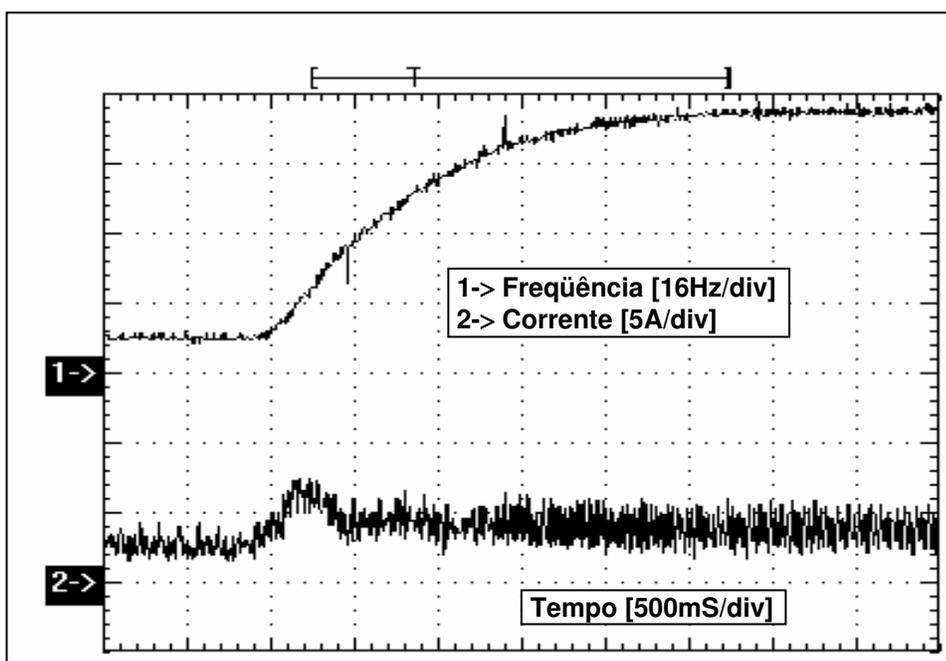
Nas figuras a seguir pode-se observar uma pequena diferença no tempo de resposta do motor de indução, na comparação entre os resultados práticos e simulados, devido aos modelos matemáticos, principalmente, o modelo do inversor de tensão **PWM**.

---

**Ensaio PID08 - Controle PID**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**



**Figura 4.11** Resultado de Simulação com Matlab (Ensaio PID08)



**Figura 4.12** Resultado Prático obtido em Laboratório (Ensaio PID08)

**Ensaio PID12 - Controle PID**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

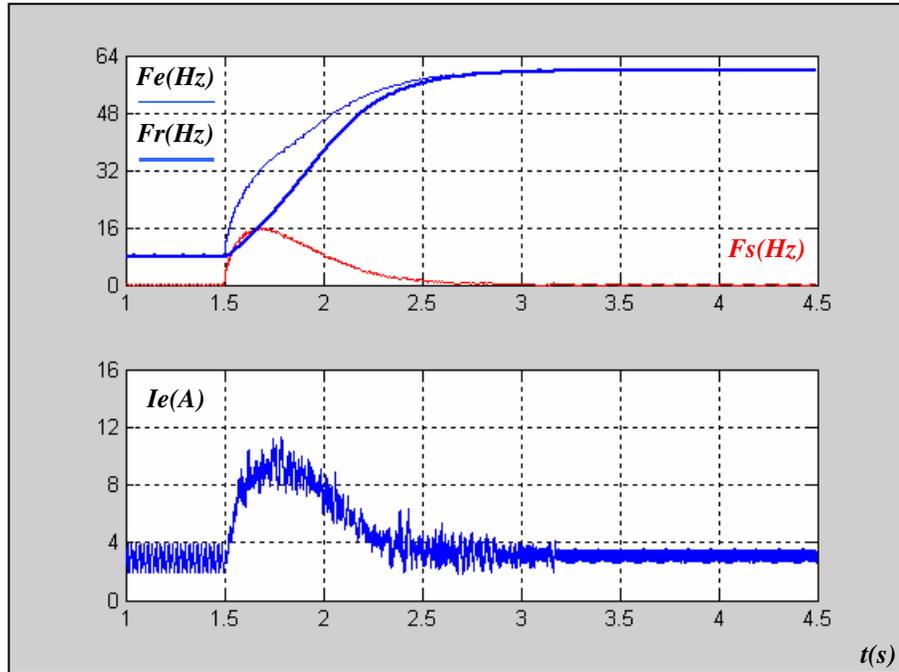


Figura 4.13 Resultado de Simulação com Matlab (Ensaio PID12)

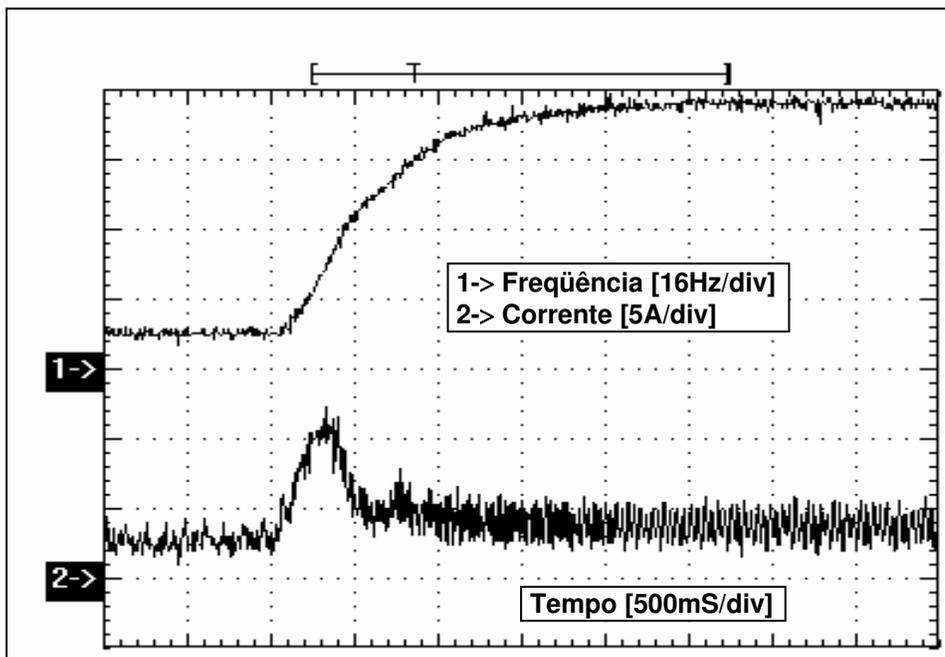


Figura 4.14 Resultado Prático obtido em Laboratório (Ensaio PID12)

**Ensaio PID16 - Controle PID**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

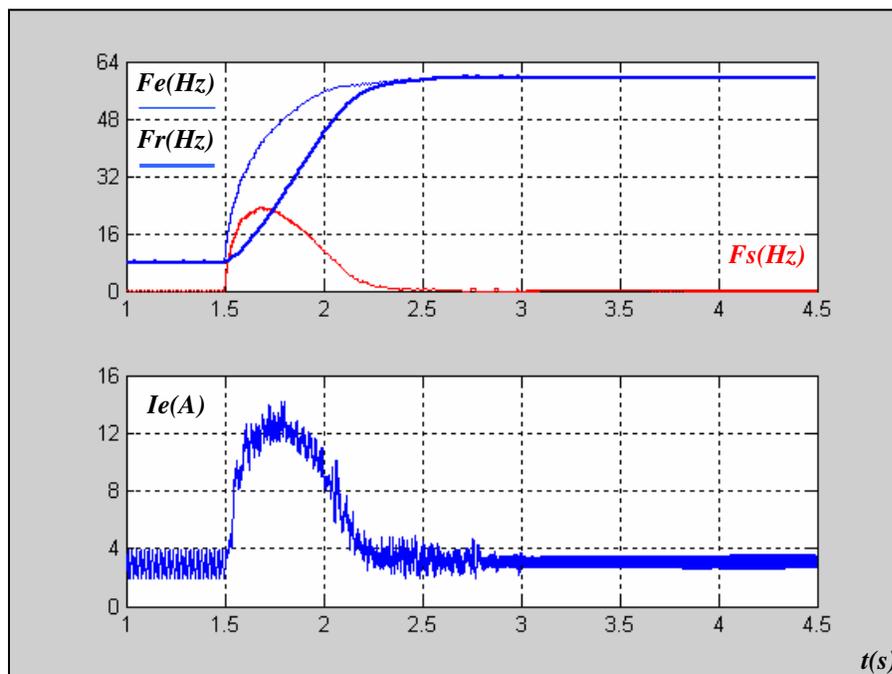


Figura 4.15 Resultado de Simulação com Matlab (Ensaio PID16)

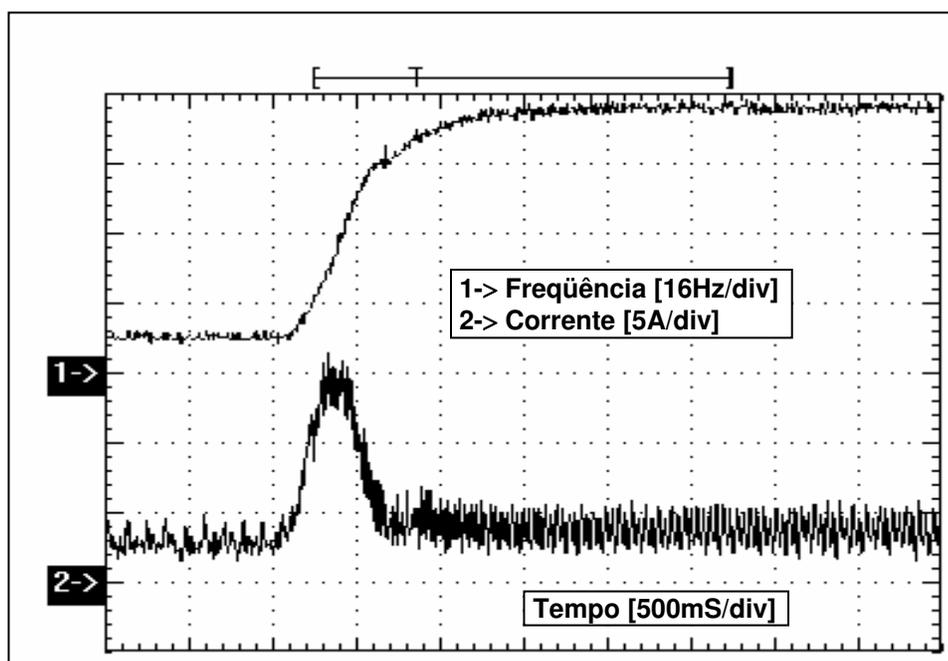


Figura 4.16 Resultado Prático obtido em Laboratório (Ensaio PID16)

Nas figuras a seguir apresentam-se os resultados práticos e simulados usando estratégia de controle proporcional integral derivativa para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.17** até **Figura 4.22** considera-se o gerador síncrono trifásico operando com corrente de excitação no circuito de campo, portanto, o motor de indução trabalha com carga e o tempo de resposta com carga é maior que o tempo de resposta em vazio.

A **Figura 4.17** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.18** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada  **$I_e$** .

A **Figura 4.19** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.20** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada  **$I_e$** .

A **Figura 4.21** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.22** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada  **$I_e$** .

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento  **$F_s$**  representada pela diferença entre a frequência do estator  **$F_e$**  e a frequência do rotor  **$F_r$** .

A corrente de entrada  **$I_e$**  apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento  **$F_s$** , portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

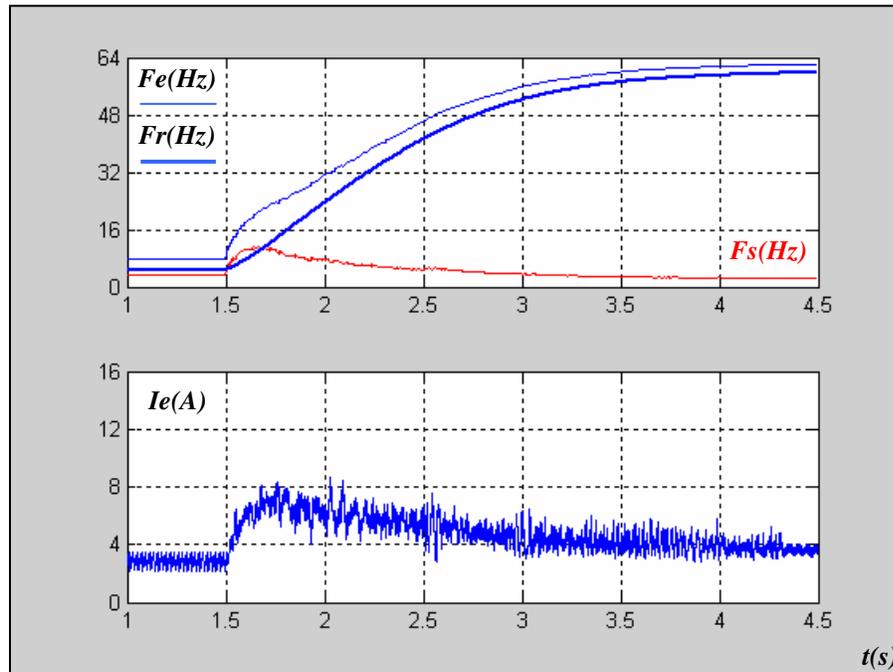
A estratégia de controle proporcional integral derivativa considera o gradiente do erro de frequência  **$\delta E_r$** , portanto, nos resultados obtidos pode-se observar uma ausência de sobre-sinal na frequência de rotação  **$F_r$**  e uma diminuição no tempo de resposta do motor de indução, quando comparado com as estratégias de controle do tipo proporcional integral.

O tempo de resposta do motor de indução é inversamente proporcional ao valor limite da corrente de entrada  **$I_e$** , assim, o tempo de resposta para um valor limite de **8A** é aproximadamente duas vezes maior que o tempo de resposta para um valor limite de **16A**.

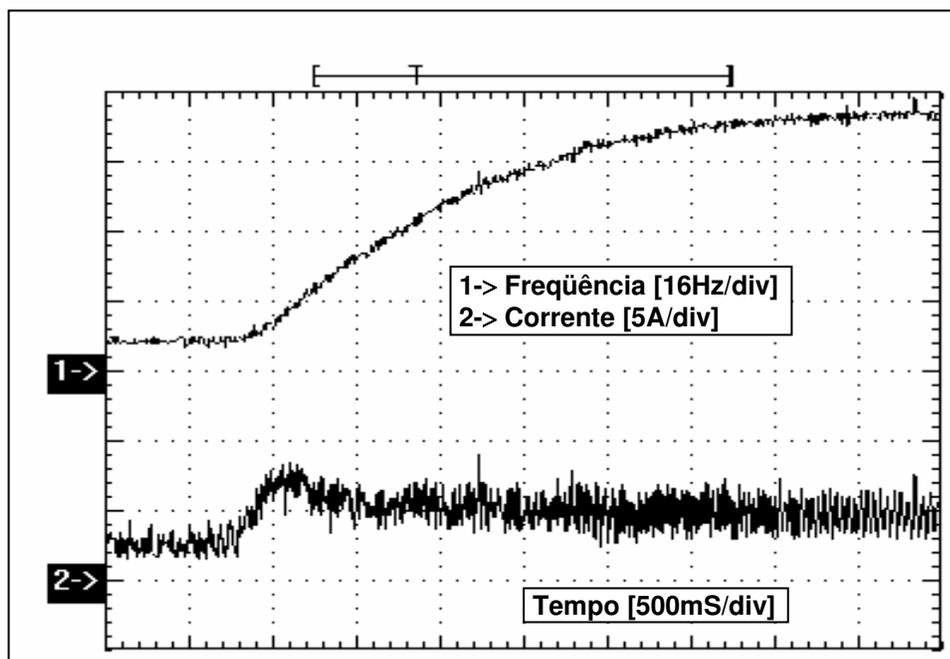
Nas figuras a seguir pode-se observar uma pequena diferença no tempo de resposta do motor de indução, na comparação entre os resultados práticos e simulados, devido aos modelos matemáticos, principalmente, o modelo do inversor de tensão **PWM**.

---

**Ensaio PID08ex - Controle PID**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**



**Figura 4.17** Resultado de Simulação com Matlab (Ensaio PID08ex)



**Figura 4.18** Resultado Prático obtido em Laboratório (Ensaio PID08ex)

**Ensaio PID12ex - Controle PID**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

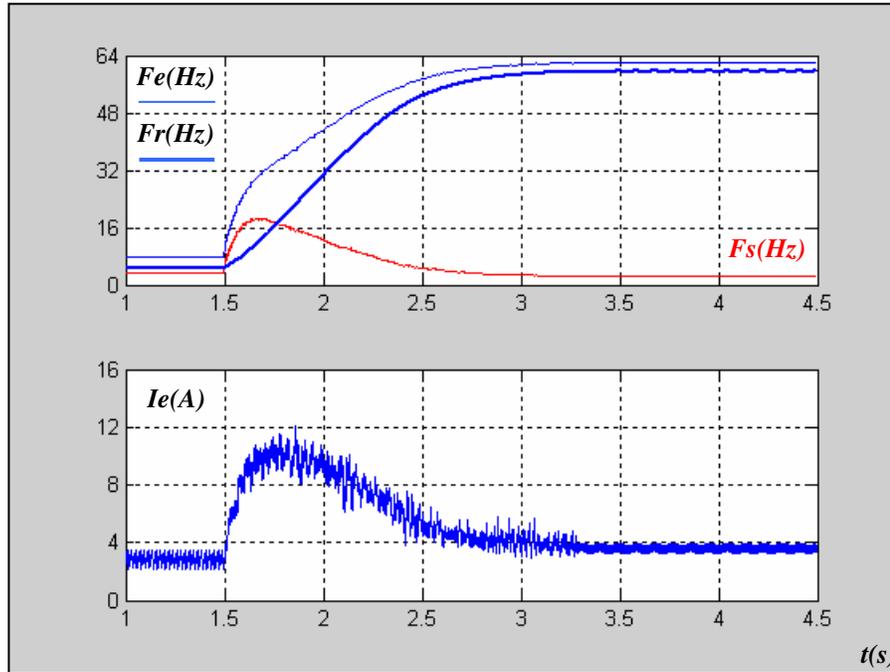


Figura 4.19 Resultado de Simulação com Matlab (Ensaio PID12ex)

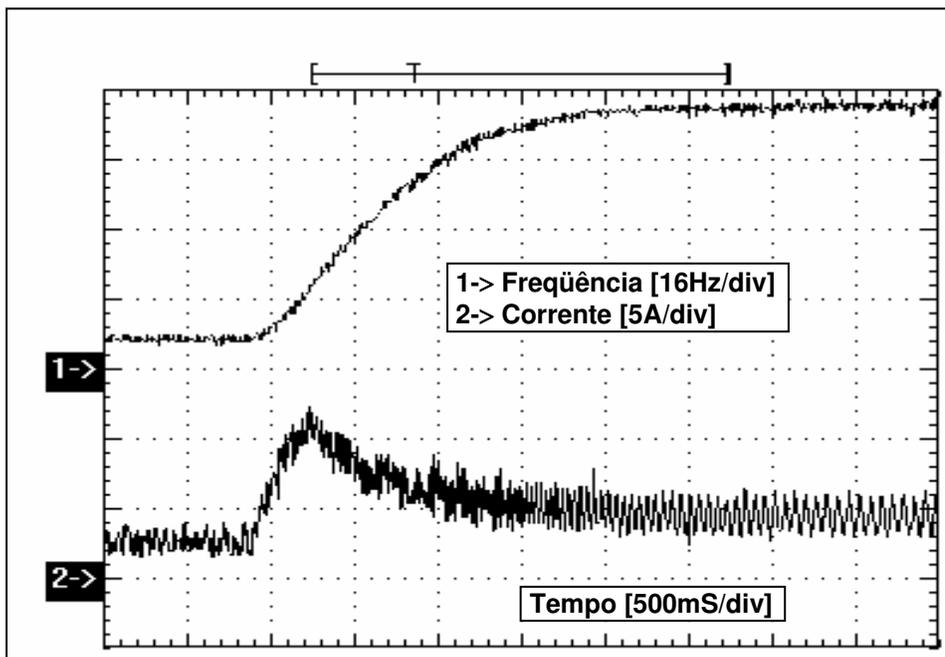


Figura 4.20 Resultado Prático obtido em Laboratório (Ensaio PID12ex)

**Ensaio PID16ex - Controle PID**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

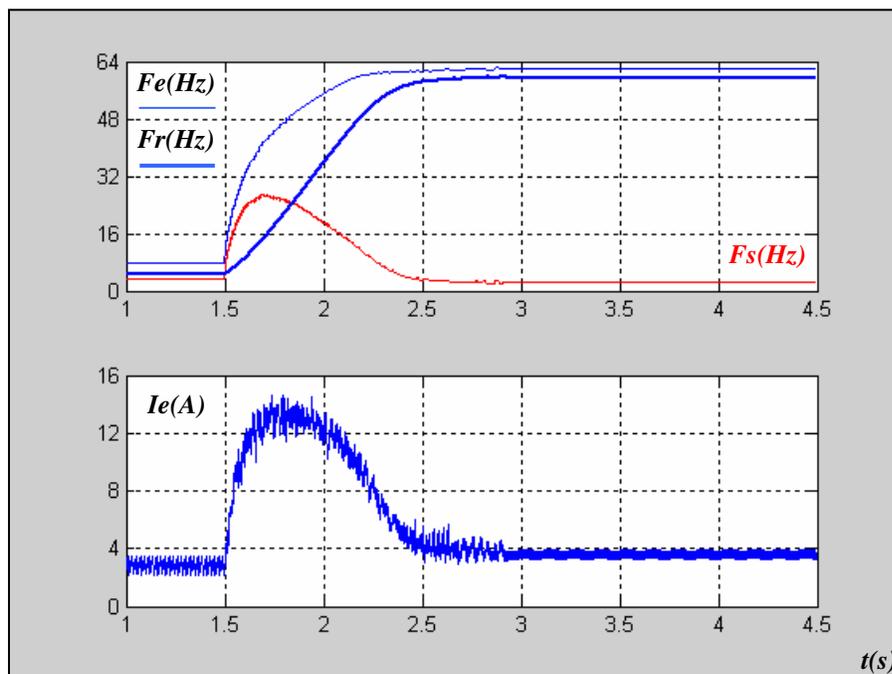


Figura 4.21 Resultado de Simulação com Matlab (Ensaio PID16ex)

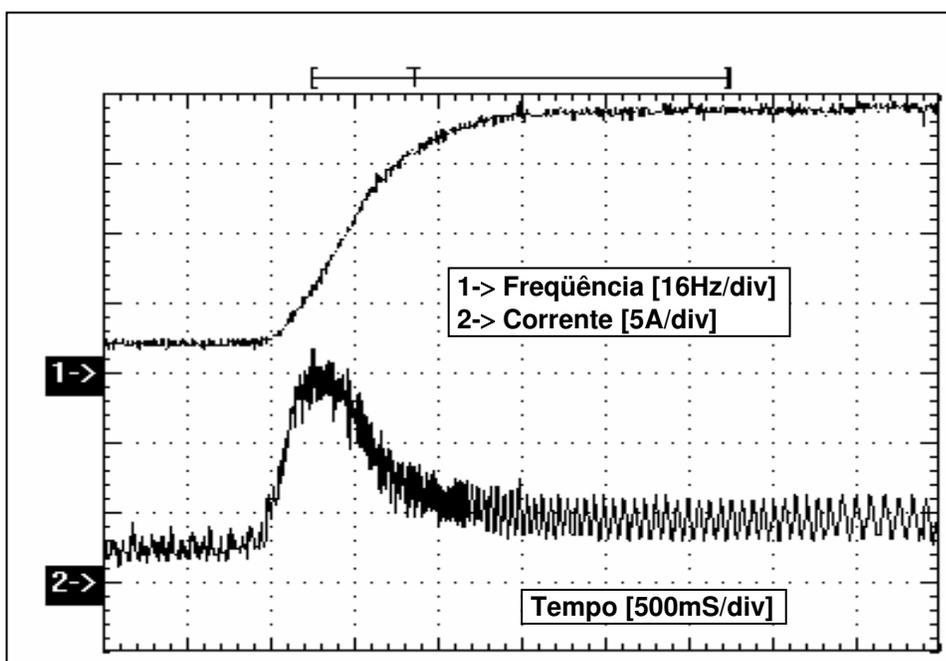
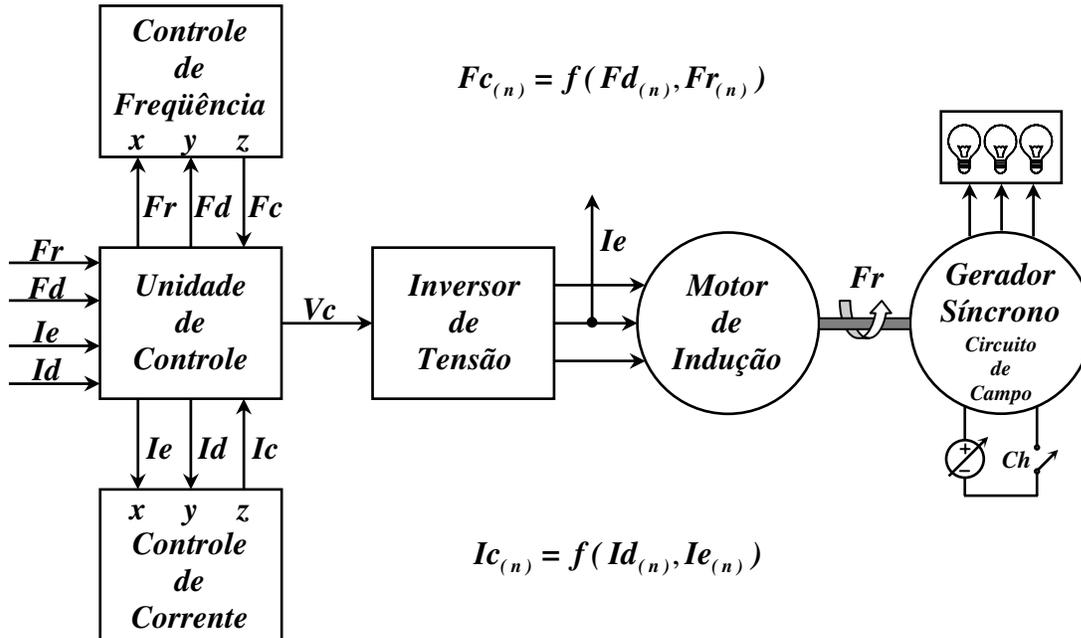


Figura 4.22 Resultado Prático obtido em Laboratório (Ensaio PID16ex)

### 4.5.3 Controle Nebuloso com Inferência Direta

Apresenta-se na **Figura 4.23** o sistema de controle, com motor de indução acionado por inversor de tensão **PWM**, acoplado ao gerador síncrono, usando estratégia de controle **NiD**.



**Figura 4.23** Sistema de Controle Nebuloso com Inferência Direta

$$Vc_{(n)} = Vc_{(n-1)} + (Fc_{(n)} \times Ic_{(n)})Kp \quad (4.2)$$

Na **Tabela 4.5** apresentam-se as características básicas das simulações computacionais e dos ensaios práticos realizados em laboratório. Nos ensaios e simulações, considera-se o circuito de campo do gerador síncrono trifásico, operando com ou sem corrente de excitação.

**Tabela 4.5** Ensaios com Controle Nebuloso com Inferência Direta

Gerador Síncrono	Motor de Indução	Resultados Práticos e Simulados	
Corrente de Excitação	Limite de Corrente	Código do Ensaio	Figuras
Ausente	$(2 \times I_n) \Rightarrow 08A$	NiD08	4.24 e 4.25
Ausente	$(3 \times I_n) \Rightarrow 12A$	NiD12	4.26 e 4.27
Ausente	$(4 \times I_n) \Rightarrow 16A$	NiD16	4.28 e 4.29
Presente	$(2 \times I_n) \Rightarrow 08A$	NiD08ex	4.30 e 4.31
Presente	$(3 \times I_n) \Rightarrow 12A$	NiD12ex	4.32 e 4.33
Presente	$(4 \times I_n) \Rightarrow 16A$	NiD16ex	4.34 e 4.35

Nas figuras a seguir apresentam-se resultados práticos e simulados usando estratégia de controle nebulosa com inferência direta para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.24** até **Figura 4.29** considera-se o gerador síncrono trifásico operando sem corrente de excitação no circuito de campo, portanto, o motor de indução trabalha em vazio e o tempo de resposta em vazio é menor que o tempo de resposta com carga.

A **Figura 4.24** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.25** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada **Ie**.

A **Figura 4.26** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.27** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada **Ie**.

A **Figura 4.28** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.29** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada **Ie**.

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento **Fs** representada pela diferença entre a frequência do estator **Fe** e a frequência do rotor **Fr**.

A corrente de entrada **Ie** apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento **Fs**, portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

A estratégia de controle nebulosa com inferência direta não trata o gradiente do erro de frequência  $\delta Er$ , portanto, nos resultados obtidos pode-se observar a ocorrência de sobre-sinal na frequência de rotação **Fr**. O sobre-sinal ocorre para uma corrente de entrada **Ie** com valor limite maior que **12A** ou para uma frequência de escorregamento **Fs** maior que **20Hz**.

Na **Figura 4.28** pode-se observar um instante de desaceleração ( $Fs < 0$ ), provocado pela estratégia de controle nebulosa com inferência direta, na tentativa de reverter o sobre-sinal que ocorre na frequência de rotação do motor de indução, durante o transitório de partida.

Na **Figura 4.29** pode-se observar uma diferença significativa presente na forma de onda da corrente de entrada **Ie**, durante o instante de desaceleração ( $Fe < Fr$ ), comparando resultados práticos e simulados, devido a simplificação dos modelos matemáticos.

**Ensaio NiD08 - Controle NiD**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

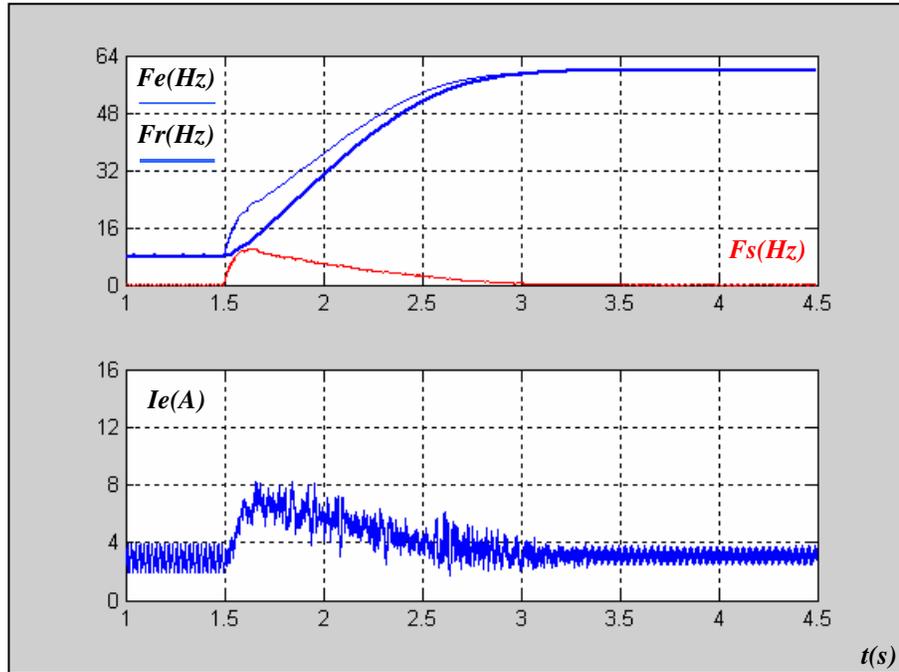


Figura 4.24 Resultado de Simulação com Matlab (Ensaio NiD08)

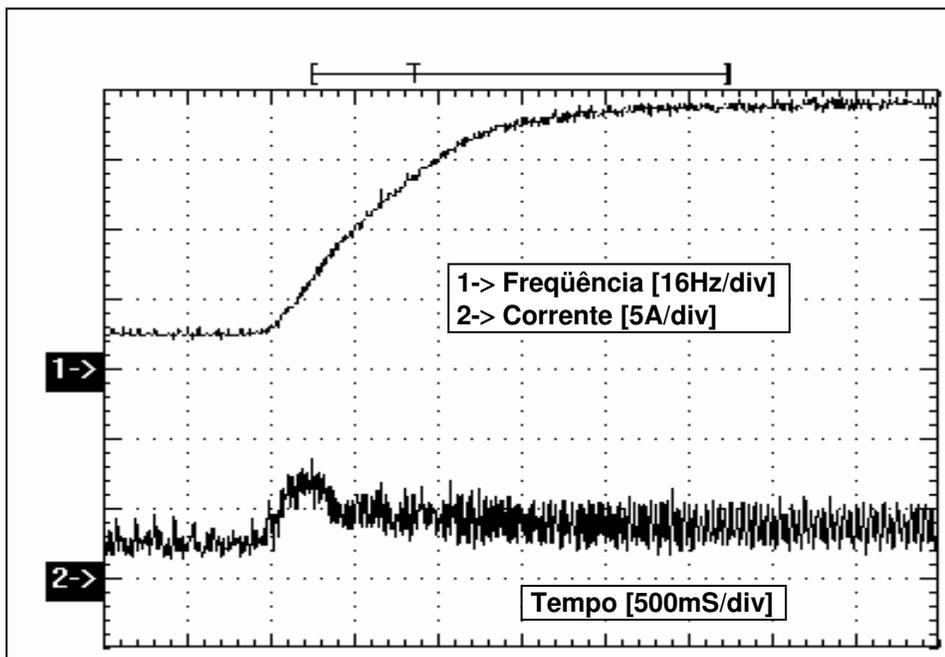


Figura 4.25 Resultado Prático obtido em Laboratório (Ensaio NiD08)

**Ensaio NiD12 - Controle NiD**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

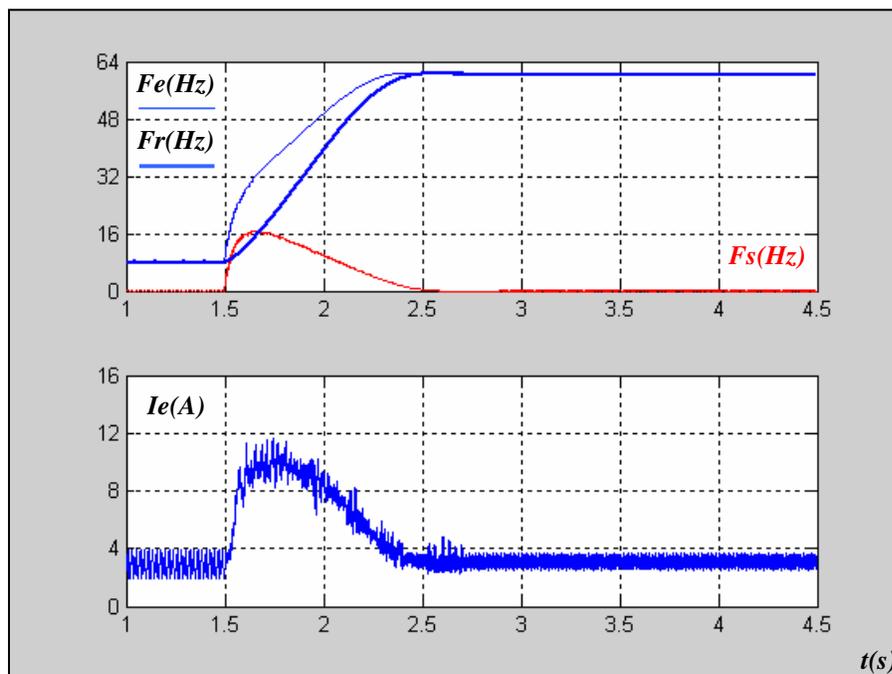


Figura 4.26 Resultado de Simulação com Matlab (Ensaio NiD12)

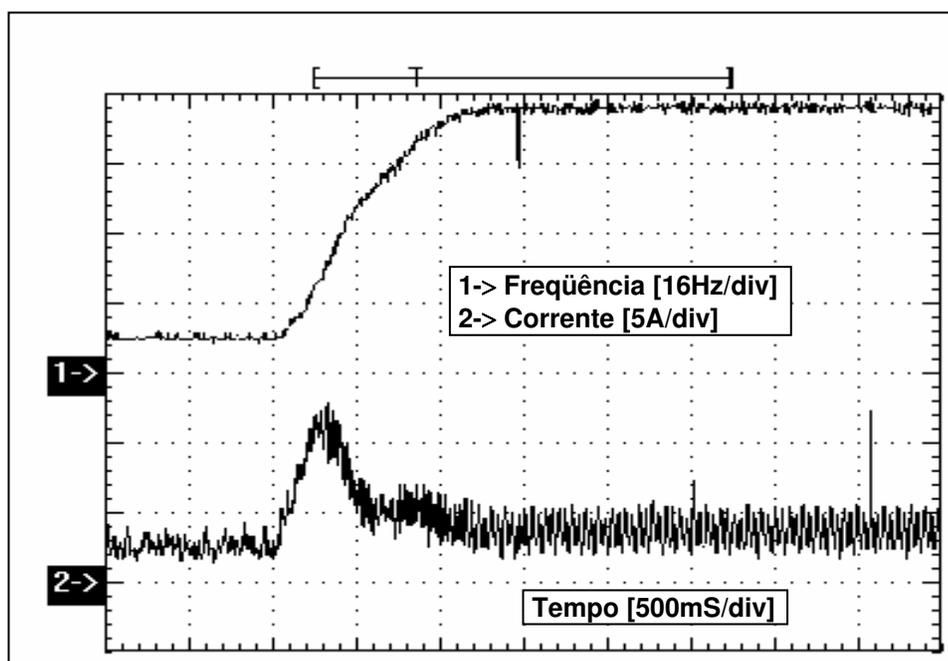


Figura 4.27 Resultado Prático obtido em Laboratório (Ensaio NiD12)

**Ensaio NiD16 - Controle NiD**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

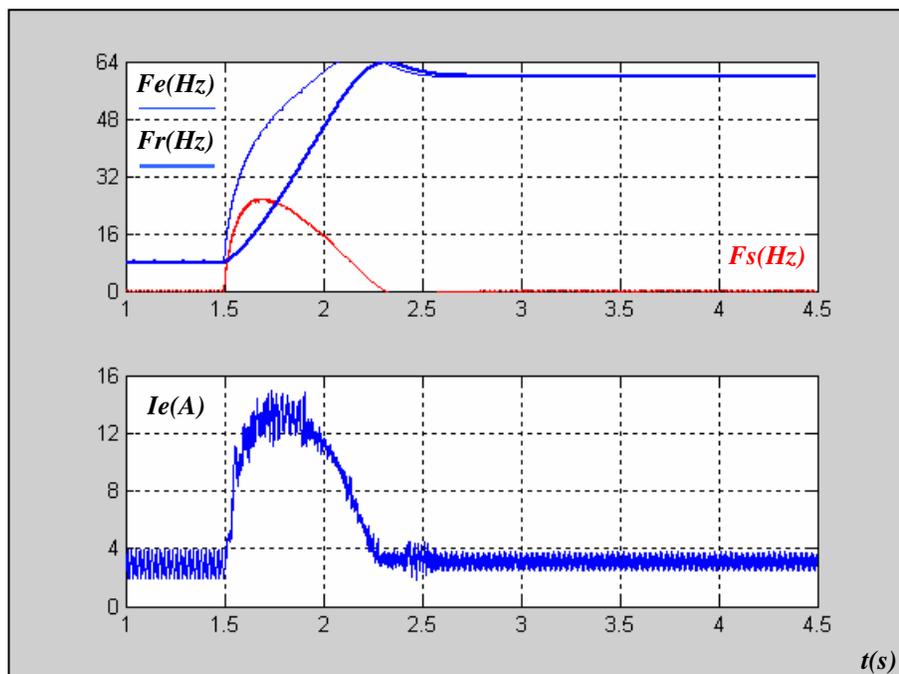


Figura 4.28 Resultado de Simulação com Matlab (Ensaio NiD16)

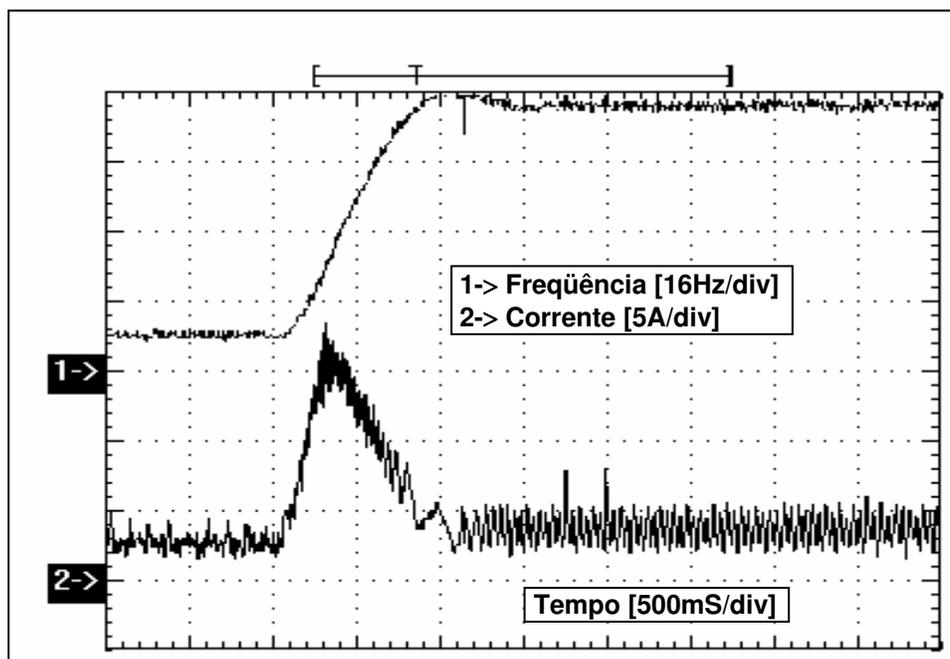


Figura 4.29 Resultado Prático obtido em Laboratório (Ensaio NiD16)

Nas figuras a seguir apresentam-se resultados práticos e simulados usando estratégia de controle nebulosa com inferência direta para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.30** até **Figura 4.35** considera-se o gerador síncrono trifásico operando com corrente de excitação no circuito de campo, portanto, o motor de indução trabalha com carga e o tempo de resposta com carga é maior que o tempo de resposta em vazio.

A **Figura 4.30** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.31** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada **Ie**.

A **Figura 4.32** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.33** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada **Ie**.

A **Figura 4.34** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.35** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada **Ie**.

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento **Fs** representada pela diferença entre a frequência do estator **Fe** e a frequência do rotor **Fr**.

A corrente de entrada **Ie** apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento **Fs**, portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

A estratégia de controle nebulosa com inferência direta não trata o gradiente do erro de frequência  $\delta Er$ , portanto, nos resultados obtidos pode-se observar a ocorrência de sobre-sinal na frequência de rotação **Fr**. O sobre-sinal ocorre para uma corrente de entrada **Ie** com valor limite maior que **12A** ou para uma frequência de escorregamento **Fs** maior que **20Hz**.

Na **Figura 4.34** pode-se observar um instante de operação em vazio (**Fs=0**), gerado pela estratégia de controle nebulosa com inferência direta, na tentativa de reverter o sobre-sinal que ocorre na frequência de rotação do motor de indução, durante o transitório de partida.

Na **Figura 4.35** pode-se observar uma diferença significativa presente na forma de onda da corrente de entrada, durante o instante de operação em vazio (**Fe=Fr**), comparando resultados práticos e simulados, devido a simplificação dos modelos matemáticos.

---

**Ensaio NiD08ex - Controle NiD**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

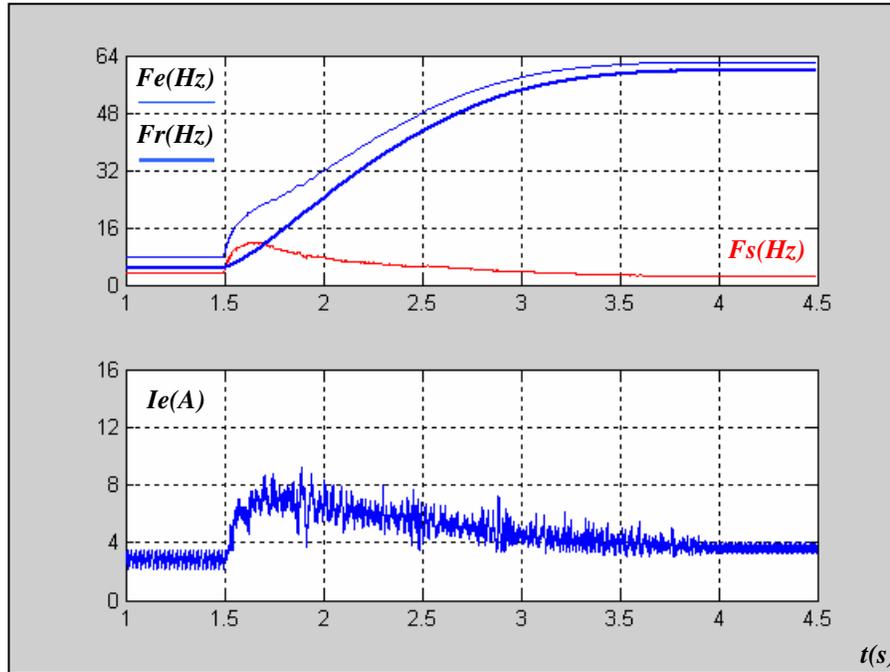


Figura 4.30 Resultado de Simulação com Matlab (Ensaio NiD08ex)

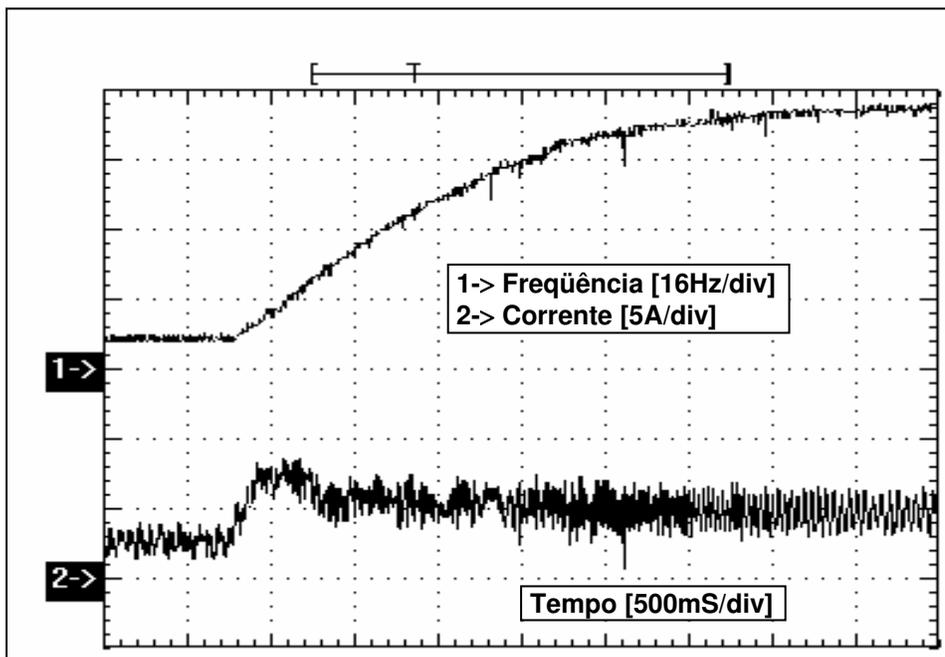
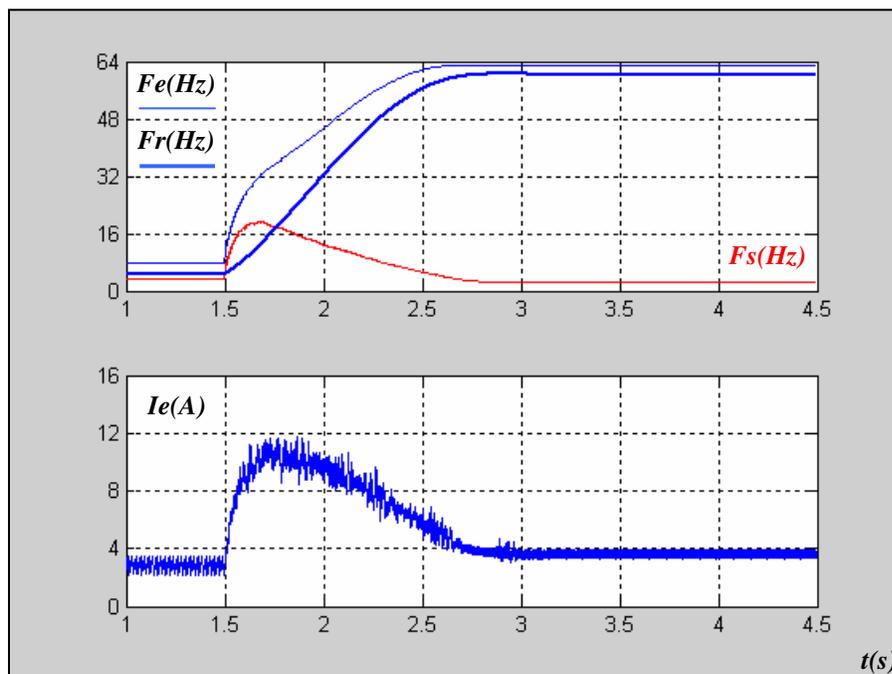
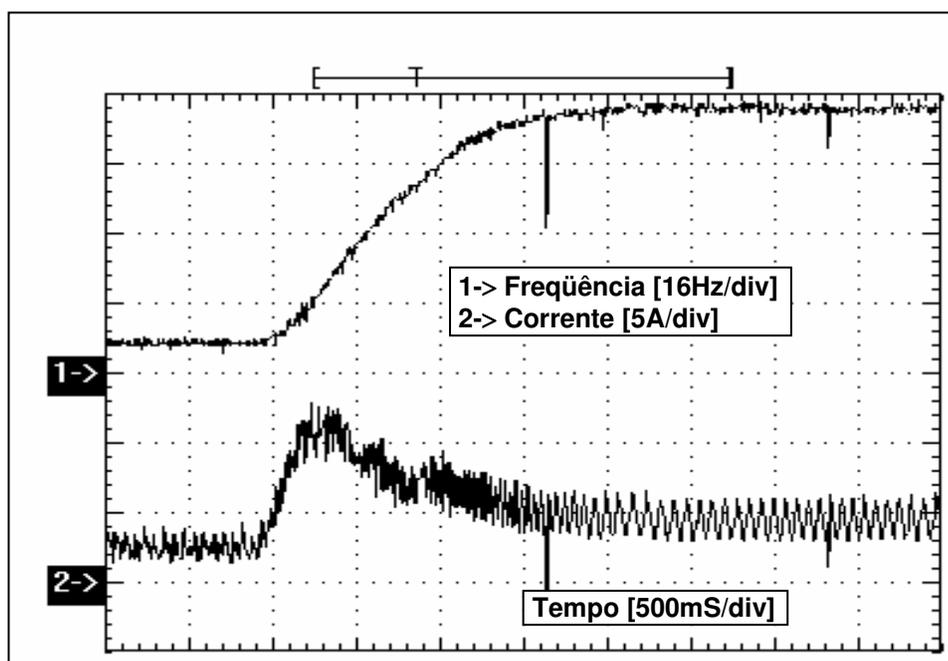


Figura 4.31 Resultado Prático obtido em Laboratório (Ensaio NiD08ex)

**Ensaio NiD12ex - Controle NiD**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**



**Figura 4.32** Resultado de Simulação com Matlab (Ensaio NiD12ex)



**Figura 4.33** Resultado Prático obtido em Laboratório (Ensaio NiD12ex)

**Ensaio NiD16ex - Controle NiD**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

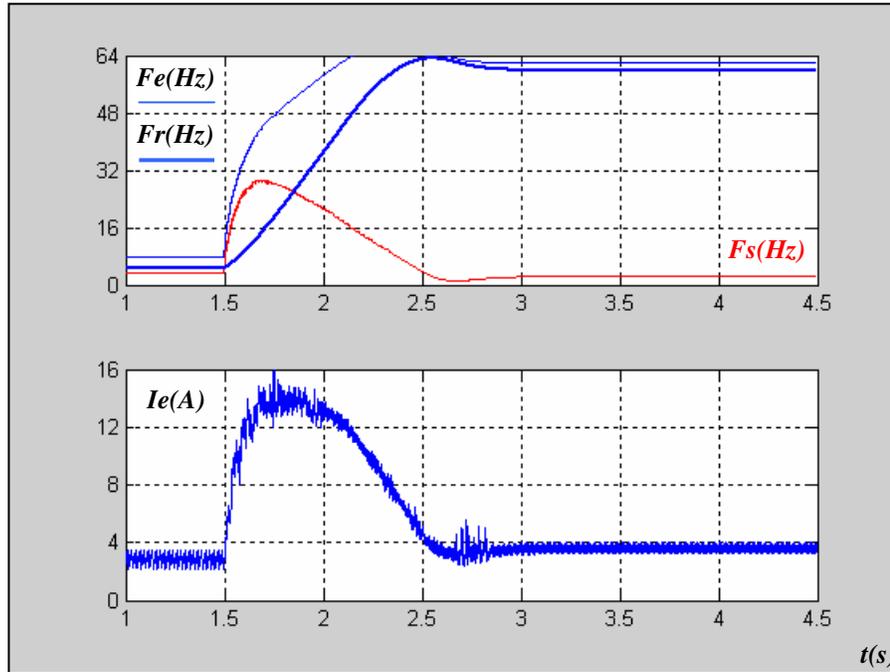


Figura 4.34 Resultado de Simulação com Matlab (Ensaio NiD16ex)

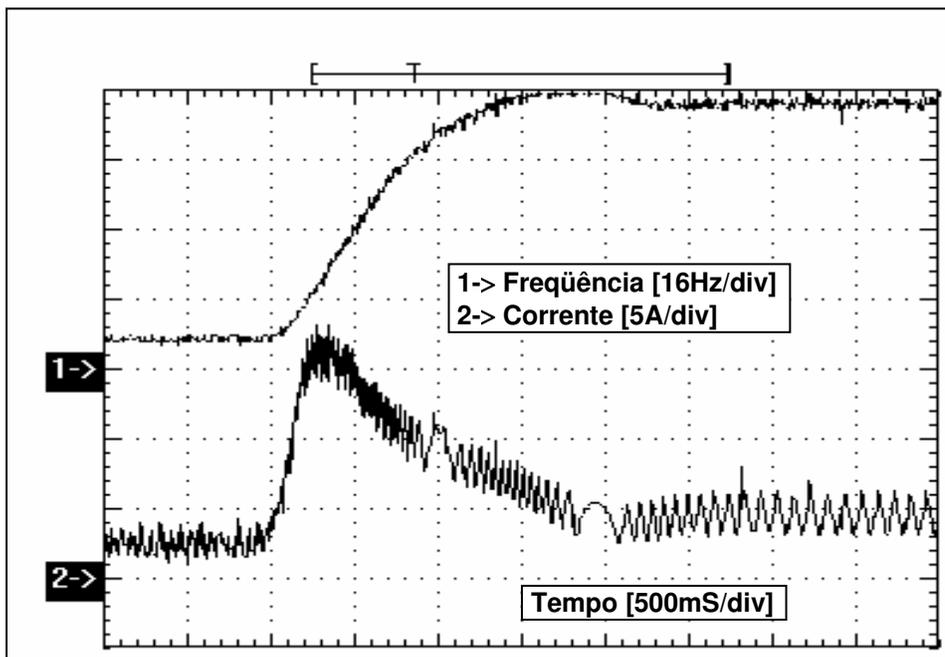
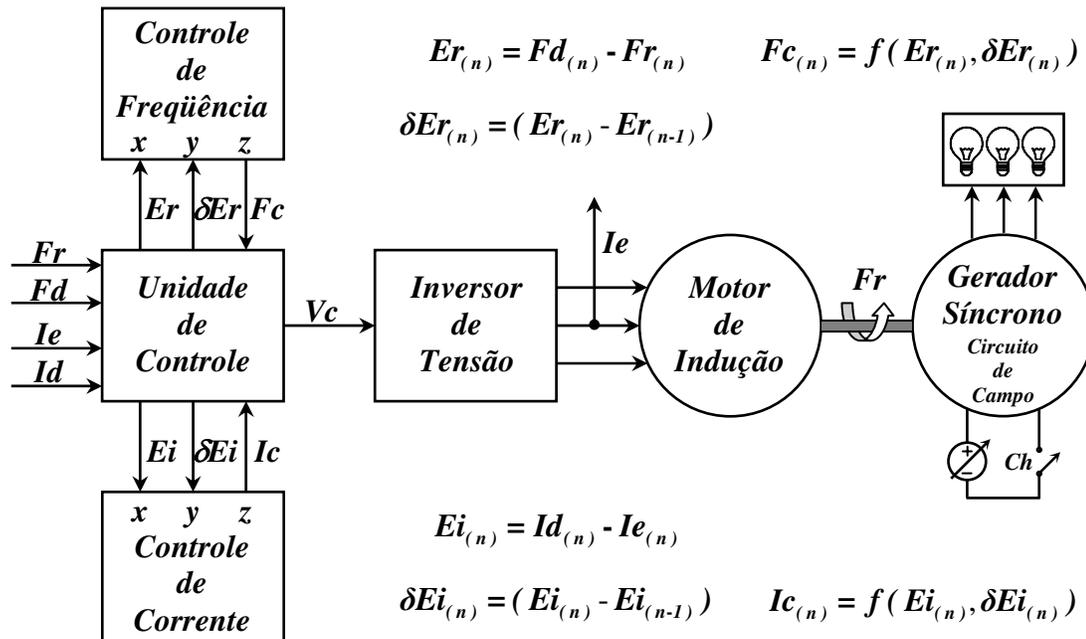


Figura 4.35 Resultado Prático obtido em Laboratório (Ensaio NiD16ex)

#### 4.5.4 Controle Nebuloso com Inferência Indireta

Apresenta-se na **Figura 4.36** o sistema de controle, com motor de indução acionado por inversor de tensão *PWM*, acoplado ao gerador síncrono, usando estratégia de controle *NiI*.



**Figura 4.36** Sistema de Controle Nebuloso com Inferência Indireta

$$Vc_{(n)} = Vc_{(n-1)} + (Fc_{(n)} \times Ic_{(n)})Kp \quad (4.3)$$

Apresenta-se na **Tabela 4.6** os resultados obtidos nas simulações computacionais e nos ensaios práticos realizados em laboratório. Nos ensaios e simulações, considera-se o circuito de campo do gerador síncrono trifásico, operando com ou sem corrente de excitação.

**Tabela 4.6** Ensaios com Controle Nebuloso com Inferência Indireta

Gerador Síncrono	Motor de Indução	Resultados Práticos e Simulados	
Corrente de Excitação	Limite de Corrente	Código do Ensaio	Figuras
Ausente	$(2 \times I_n) \Rightarrow 08A$	NiI08	4.37 e 4.38
Ausente	$(3 \times I_n) \Rightarrow 12A$	NiI12	4.39 e 4.40
Ausente	$(4 \times I_n) \Rightarrow 16A$	NiI16	4.41 e 4.42
Presente	$(2 \times I_n) \Rightarrow 08A$	NiI08ex	4.43 e 4.44
Presente	$(3 \times I_n) \Rightarrow 12A$	NiI12ex	4.45 e 4.46
Presente	$(4 \times I_n) \Rightarrow 16A$	NiI16ex	4.47 e 4.48

Nas figuras a seguir apresentam-se resultados práticos e simulados usando estratégia de controle nebulosa com inferência indireta para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.37** até **Figura 4.42** considera-se o gerador síncrono trifásico operando sem corrente de excitação no circuito de campo, portanto, o motor de indução trabalha em vazio e o tempo de resposta em vazio é menor que o tempo de resposta com carga.

A **Figura 4.37** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.38** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada  **$I_e$** .

A **Figura 4.39** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.40** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada  **$I_e$** .

A **Figura 4.41** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.42** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada  **$I_e$** .

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento  **$F_s$**  representada pela diferença entre a frequência do estator  **$F_e$**  e a frequência do rotor  **$F_r$** .

A corrente de entrada  **$I_e$**  apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento  **$F_s$** , portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

A estratégia de controle nebulosa com inferência indireta considera o gradiente do erro de frequência  **$\delta E_r$** , portanto, nos resultados obtidos pode-se observar uma ausência de sobre-sinal na frequência de rotação  **$F_r$**  e uma diminuição no tempo de resposta do motor de indução, quando comparado com as estratégias de controle do tipo proporcional integral.

O tempo de resposta do motor de indução é inversamente proporcional ao valor limite da corrente de entrada  **$I_e$** , assim, o tempo de resposta para um valor limite de **8A** é aproximadamente duas vezes maior que o tempo de resposta para um valor limite de **16A**.

Nas figuras a seguir pode-se observar uma pequena diferença no tempo de resposta do motor de indução, na comparação entre os resultados práticos e simulados, devido aos modelos matemáticos, principalmente, o modelo do inversor de tensão **PWM**.

---

**Ensaio NiI08 - Controle NiI**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

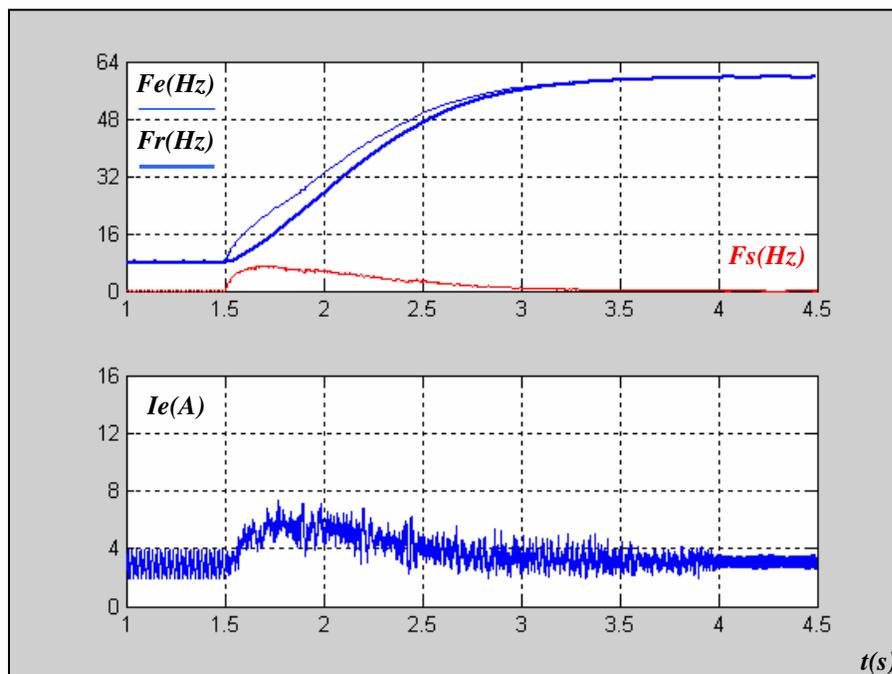


Figura 4.37 Resultado de Simulação com Matlab (Ensaio NiI08)

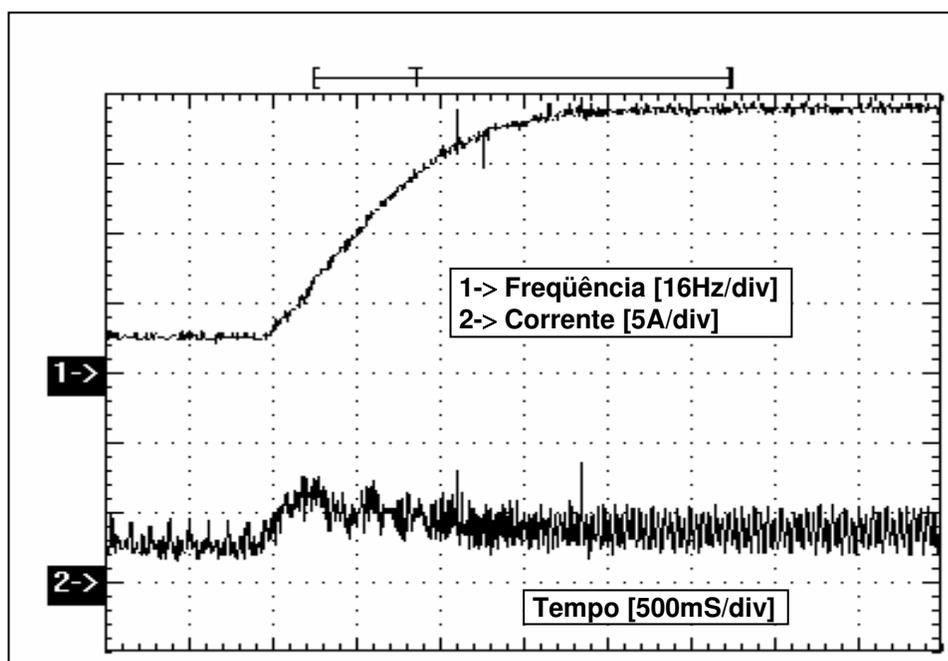


Figura 4.38 Resultado Prático obtido em Laboratório (Ensaio NiI08)

**Ensaio Ni12 - Controle Ni1**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

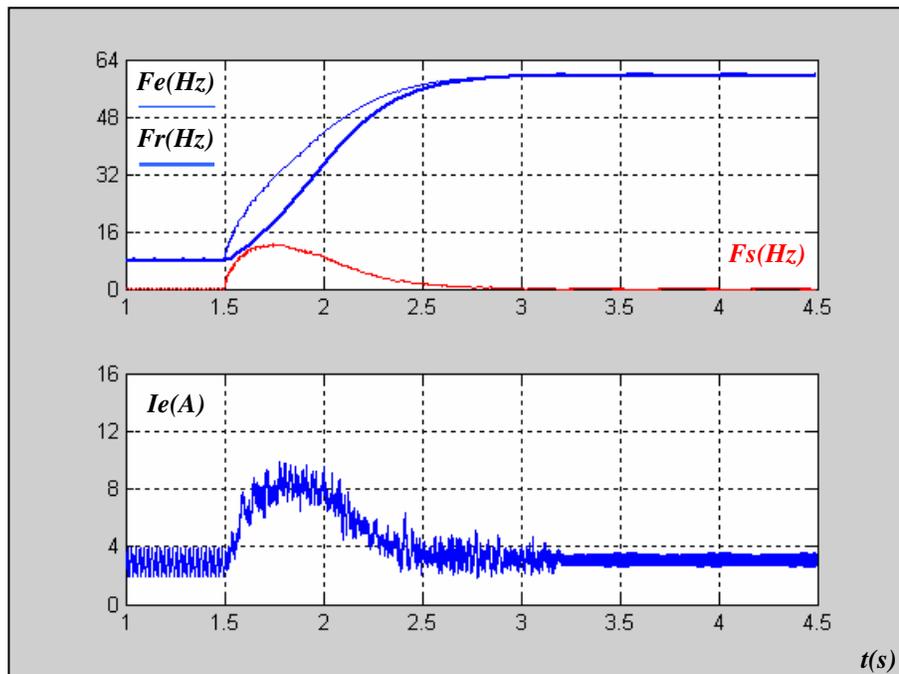


Figura 4.39 Resultado de Simulação com Matlab (Ensaio Ni12)

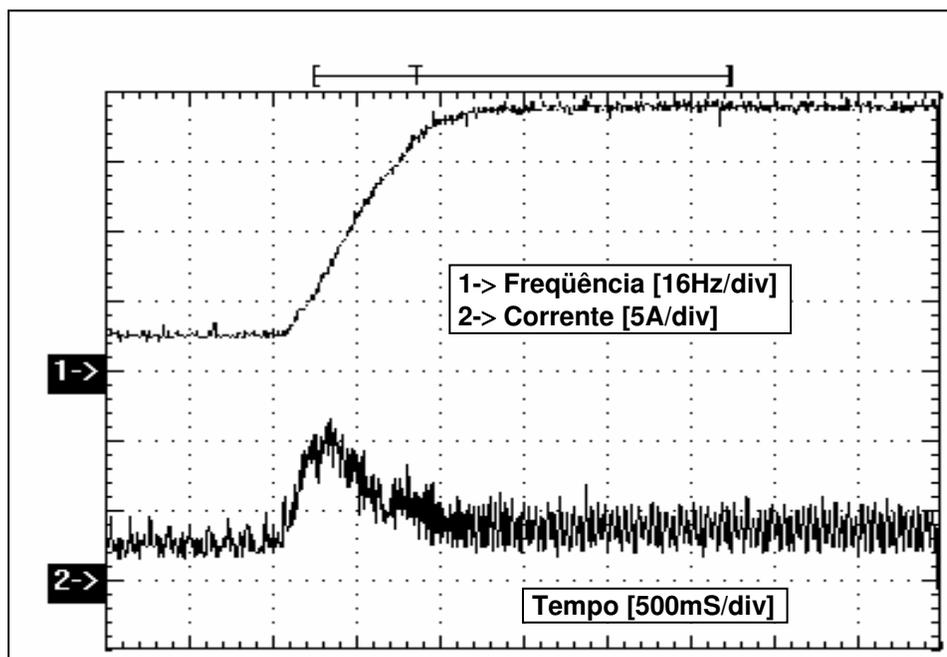


Figura 4.40 Resultado Prático obtido em Laboratório (Ensaio Ni12)

**Ensaio NiI16 - Controle NiI**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

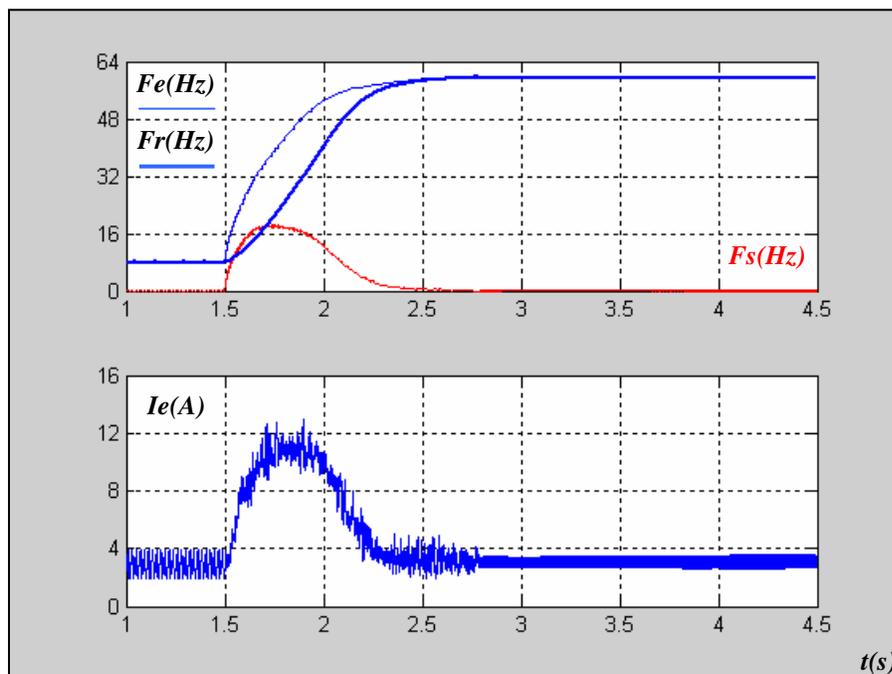


Figura 4.41 Resultado de Simulação com Matlab (Ensaio NiI16)

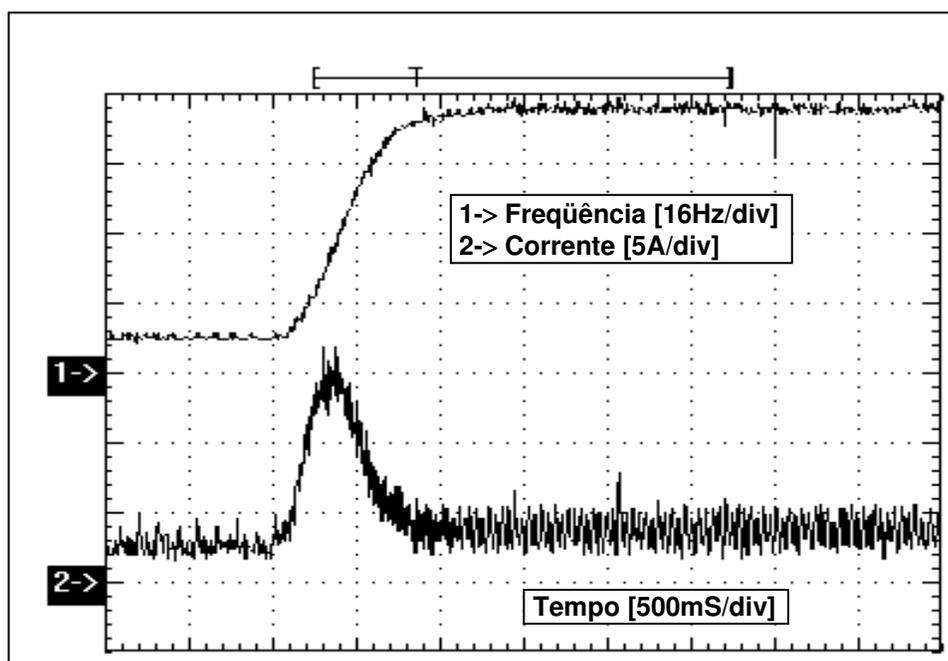


Figura 4.42 Resultado Prático obtido em Laboratório (Ensaio NiI16)

Nas figuras a seguir apresentam-se os resultados práticos e simulados usando estratégia de controle nebulosa com inferência indireta para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.43** até **Figura 4.48** considera-se o gerador síncrono trifásico operando com corrente de excitação no circuito de campo, portanto, o motor de indução trabalha com carga e o tempo de resposta com carga é maior que o tempo de resposta em vazio.

A **Figura 4.43** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.44** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada  **$I_e$** .

A **Figura 4.45** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.46** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada  **$I_e$** .

A **Figura 4.47** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.48** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada  **$I_e$** .

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento  **$F_s$**  representada pela diferença entre a frequência do estator  **$F_e$**  e a frequência do rotor  **$F_r$** .

A corrente de entrada  **$I_e$**  apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento  **$F_s$** , portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

A estratégia de controle nebulosa com inferência indireta considera o gradiente do erro de frequência  **$\delta E_r$** , portanto, nos resultados obtidos pode-se observar uma ausência de sobre-sinal na frequência de rotação  **$F_r$**  e uma diminuição no tempo de resposta do motor de indução, quando comparado com as estratégias de controle do tipo proporcional integral.

O tempo de resposta do motor de indução é inversamente proporcional ao valor limite da corrente de entrada  **$I_e$** , assim, o tempo de resposta para um valor limite de **8A** é aproximadamente duas vezes maior que o tempo de resposta para um valor limite de **16A**.

Nas figuras a seguir pode-se observar uma pequena diferença no tempo de resposta do motor de indução, na comparação entre os resultados práticos e simulados, devido aos modelos matemáticos, principalmente, o modelo do inversor de tensão **PWM**.

---

**Ensaio NiI08ex - Controle NiI**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

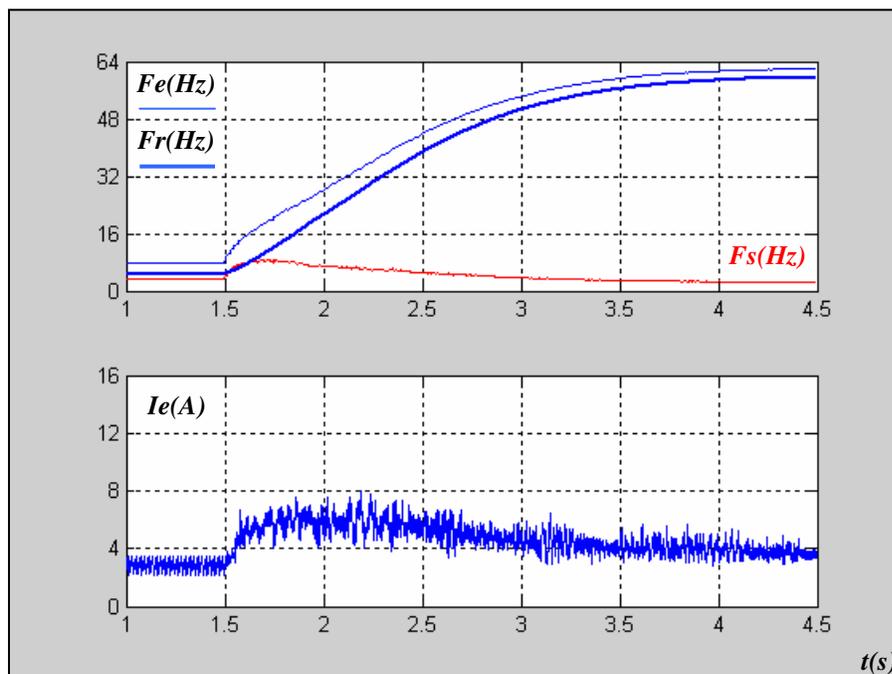


Figura 4.43 Resultado de Simulação com Matlab (Ensaio NiI08ex)

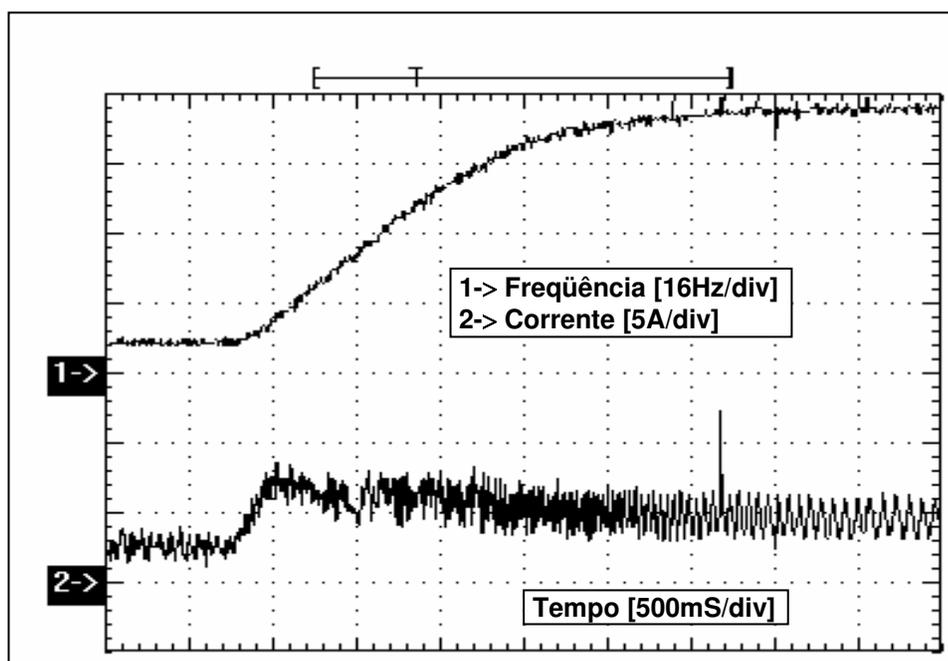


Figura 4.44 Resultado Prático obtido em Laboratório (Ensaio NiI08ex)

**Ensaio NiI12ex - Controle NiI**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

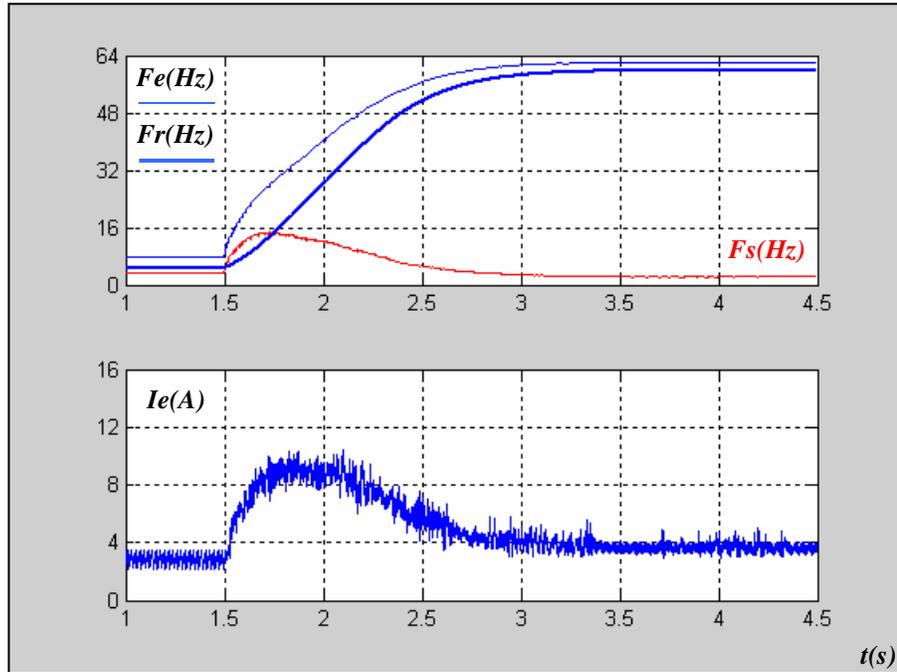


Figura 4.45 Resultado de Simulação com Matlab (Ensaio NiI12ex)

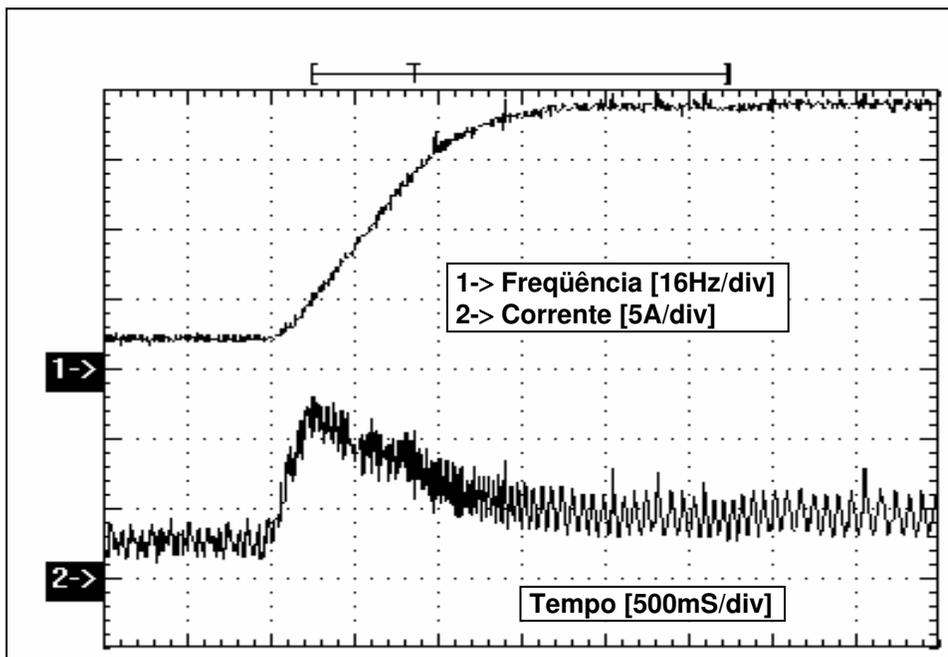


Figura 4.46 Resultado Prático obtido em Laboratório (Ensaio NiI12ex)

**Ensaio NiI16ex - Controle NiI**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

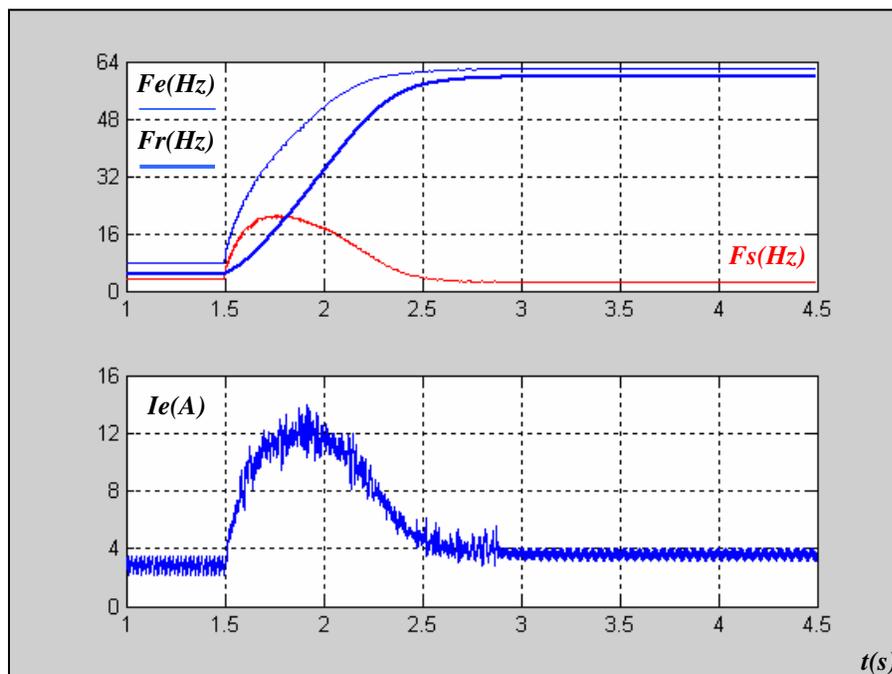


Figura 4.47 Resultado de Simulação com Matlab (Ensaio NiI16ex)

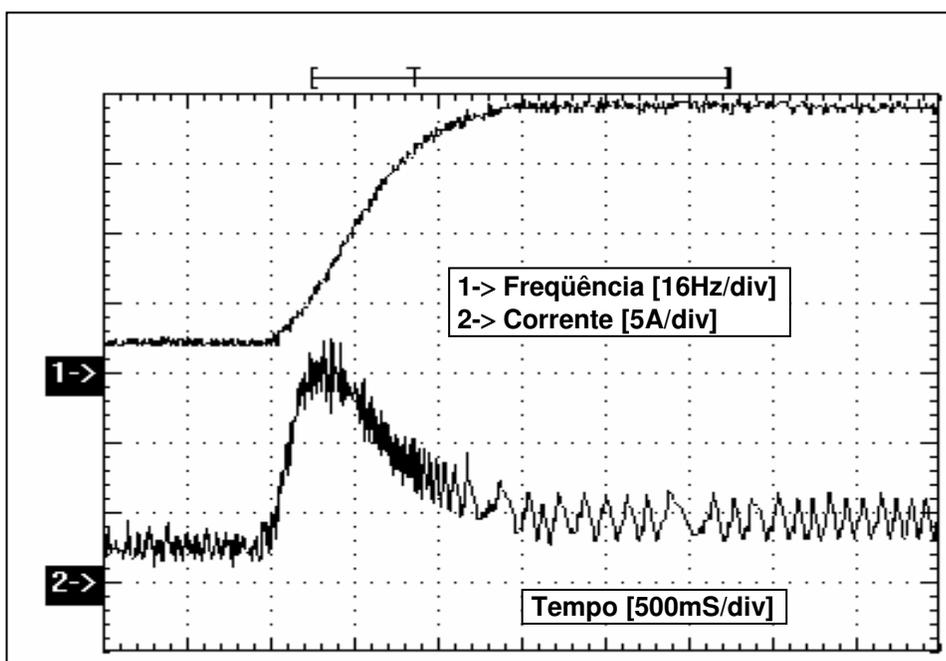
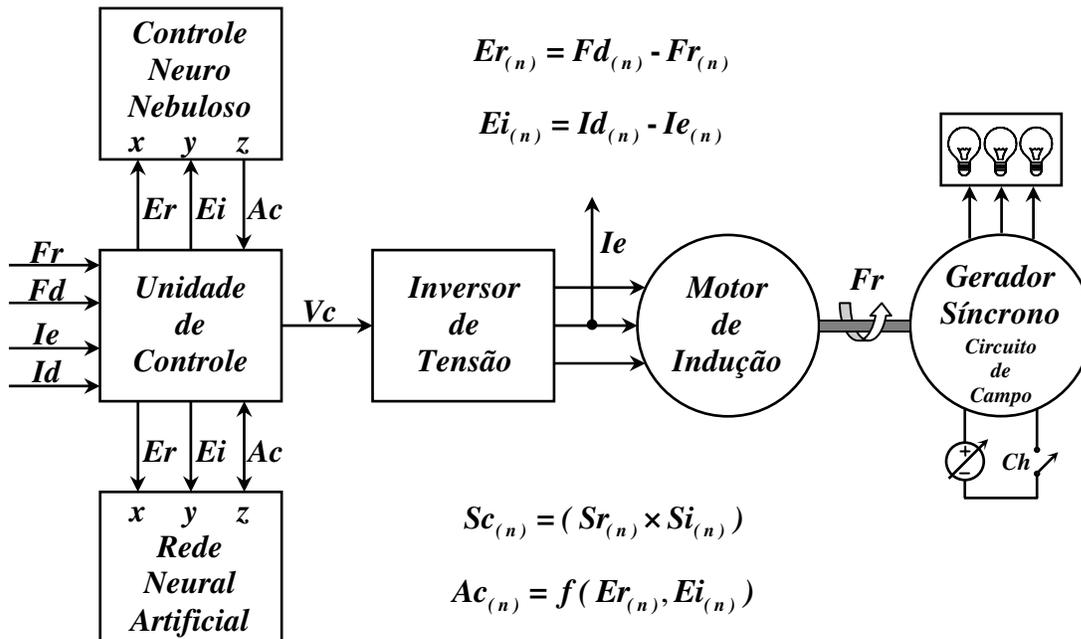


Figura 4.48 Resultado Prático obtido em Laboratório (Ensaio NiI16ex)

#### 4.5.5 Estratégia de Controle Neuro-Nebulosa

Apresenta-se na **Figura 4.49** o sistema de controle, com motor de indução acionado por inversor de tensão **PWM**, acoplado ao gerador síncrono, usando estratégia de controle **NeN**.



**Figura 4.49** Sistema com Estratégia de Controle Neuro-Nebulosa

$$Vc_{(n)} = Vc_{(n-1)} + (Sc_{(n)} \times Ac_{(n)})Kp \quad (4.4)$$

Apresenta-se na **Tabela 4.7** os resultados obtidos nas simulações computacionais e nos ensaios práticos realizados em laboratório. Nos ensaios e simulações, considera-se o circuito de campo do gerador síncrono trifásico, operando com ou sem corrente de excitação.

**Tabela 4.7** Ensaios com Estratégia de Controle Neuro-Nebulosa

Gerador Síncrono	Motor de Indução	Resultados Práticos e Simulados	
Corrente de Excitação	Limite de Corrente	Código do Ensaio	Figuras
Ausente	$(2 \times In) \Rightarrow 08A$	NeN08	4.50 e 4.51
Ausente	$(3 \times In) \Rightarrow 12A$	NeN12	4.52 e 4.53
Ausente	$(4 \times In) \Rightarrow 16A$	NeN16	4.54 e 4.55
Presente	$(2 \times In) \Rightarrow 08A$	NeN08ex	4.56 e 4.57
Presente	$(3 \times In) \Rightarrow 12A$	NeN12ex	4.58 e 4.59
Presente	$(4 \times In) \Rightarrow 16A$	NeN16ex	4.60 e 4.61

Nas figuras a seguir apresentam-se os resultados práticos e simulados obtidos utilizando estratégia de controle digital neuro-nebulosa para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.50** até **Figura 4.55** considera-se o gerador síncrono trifásico operando sem corrente de excitação no circuito de campo, portanto, o motor de indução trabalha em vazio e o tempo de resposta em vazio é menor que o tempo de resposta com carga.

A **Figura 4.50** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.51** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada  **$I_e$** .

A **Figura 4.52** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.53** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada  **$I_e$** .

A **Figura 4.54** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.55** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada  **$I_e$** .

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento  **$F_s$**  representada pela diferença entre a frequência do estator  **$F_e$**  e a frequência do rotor  **$F_r$** .

A corrente de entrada  **$I_e$**  apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento  **$F_s$** , portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

A estratégia de controle neuro-nebulosa não trata o gradiente do erro de frequência  **$\delta E_r$** , portanto, nos resultados obtidos pode-se observar a ocorrência de um sobre-sinal na frequência de rotação do motor  **$F_r$** . O sobre-sinal ocorre para uma corrente de entrada  **$I_e$**  com valor limite maior que **12A** ou para uma frequência de escorregamento  **$F_s$**  maior que **20Hz**.

Na **Figura 4.54** pode-se observar um instante de desaceleração ( **$F_s < 0$** ), provocado pela estratégia de controle digital neuro-nebulosa, na tentativa de reverter o sobre-sinal que ocorre na frequência de rotação do motor de indução, durante o transitório de partida.

Na **Figura 4.55** pode-se observar uma diferença significativa presente na forma de onda da corrente de entrada  **$I_e$** , durante o instante de desaceleração ( **$F_e < F_r$** ), comparando resultados práticos e simulados, devido a simplificação dos modelos matemáticos.

**Ensaio NeN08 - Controle NeN**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

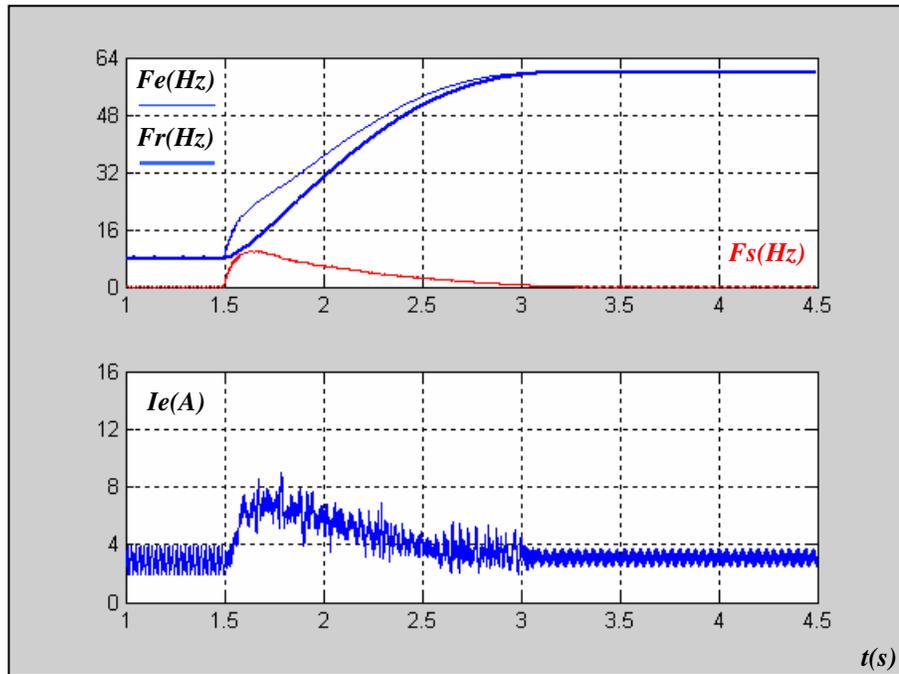


Figura 4.50 Resultado de Simulação com Matlab (Ensaio NeN08)

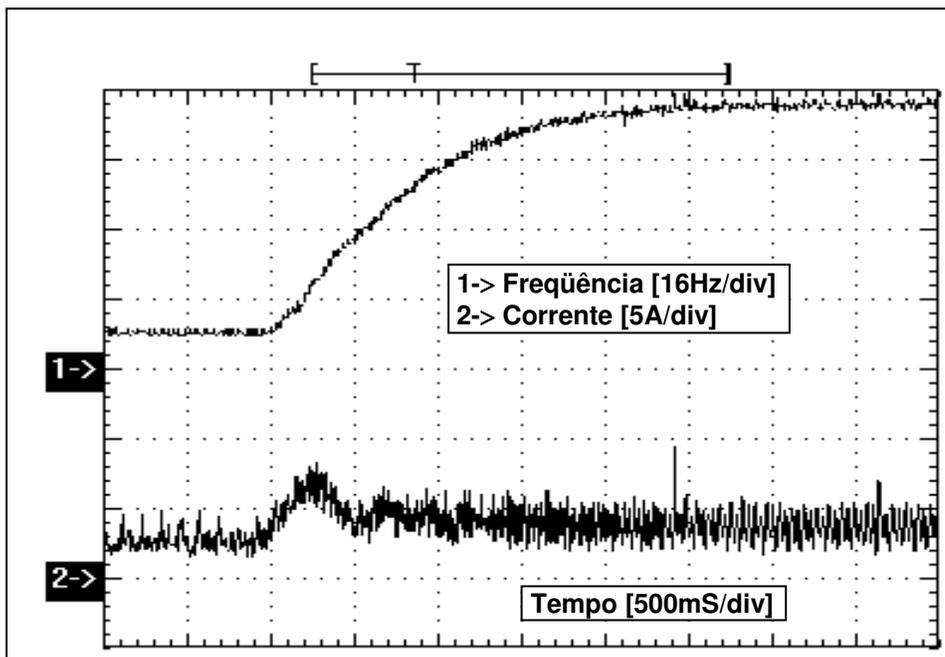
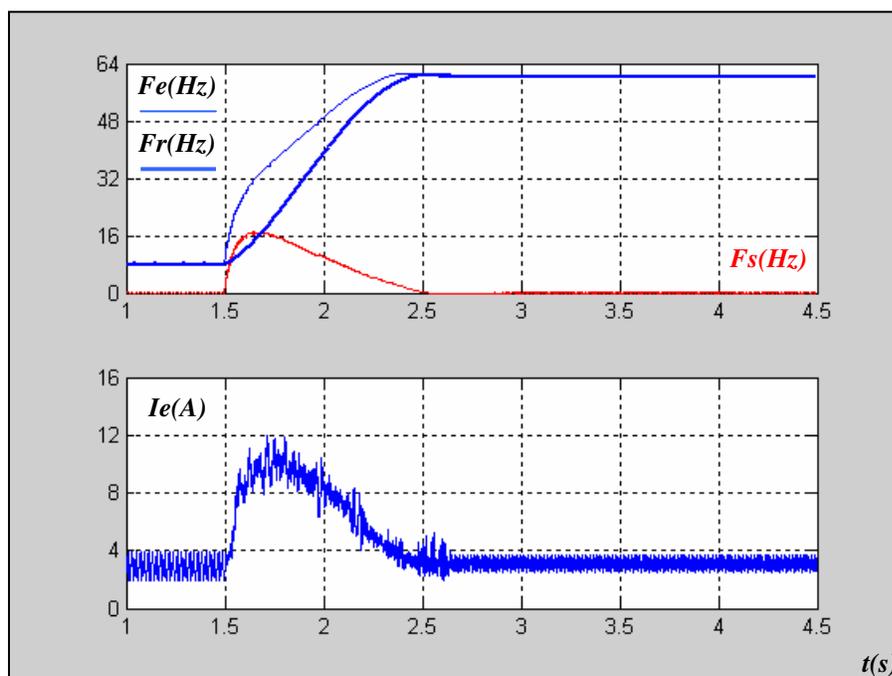
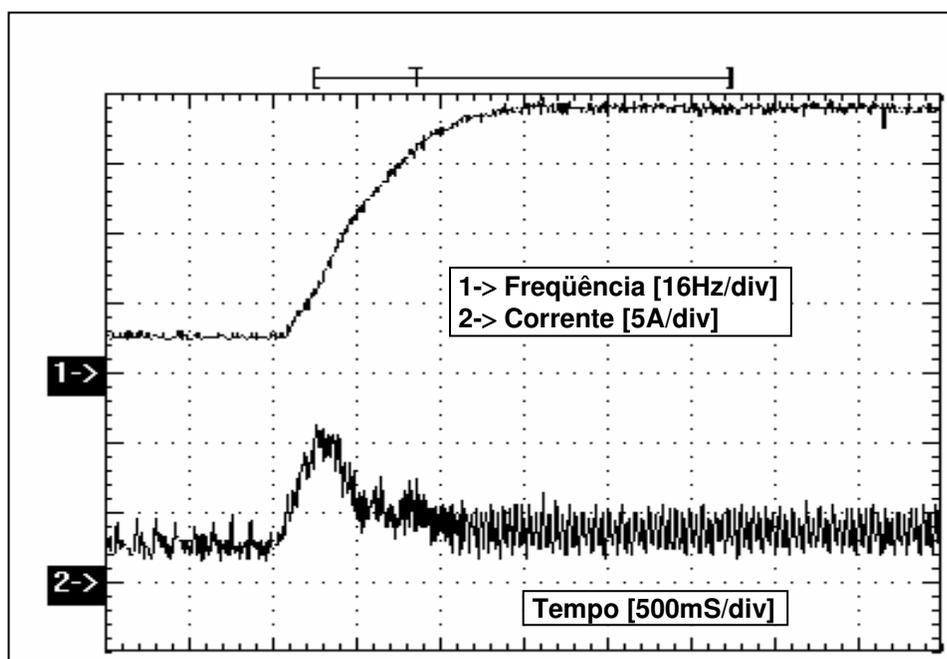


Figura 4.51 Resultado Prático obtido em Laboratório (Ensaio NeN08)

**Ensaio NeN12 - Controle NeN**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**

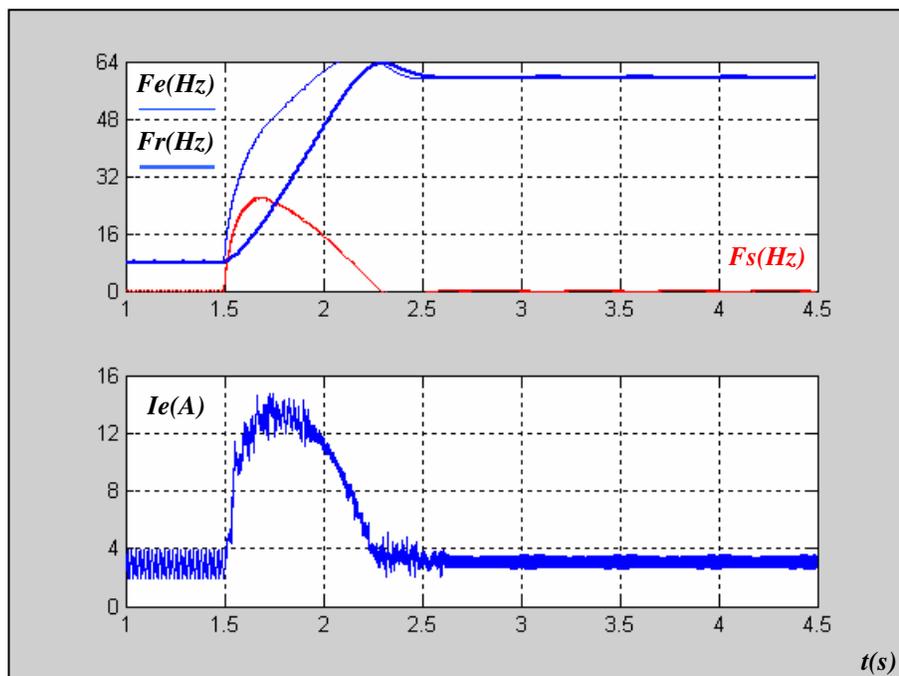


**Figura 4.52** Resultado de Simulação com Matlab (Ensaio NeN12)

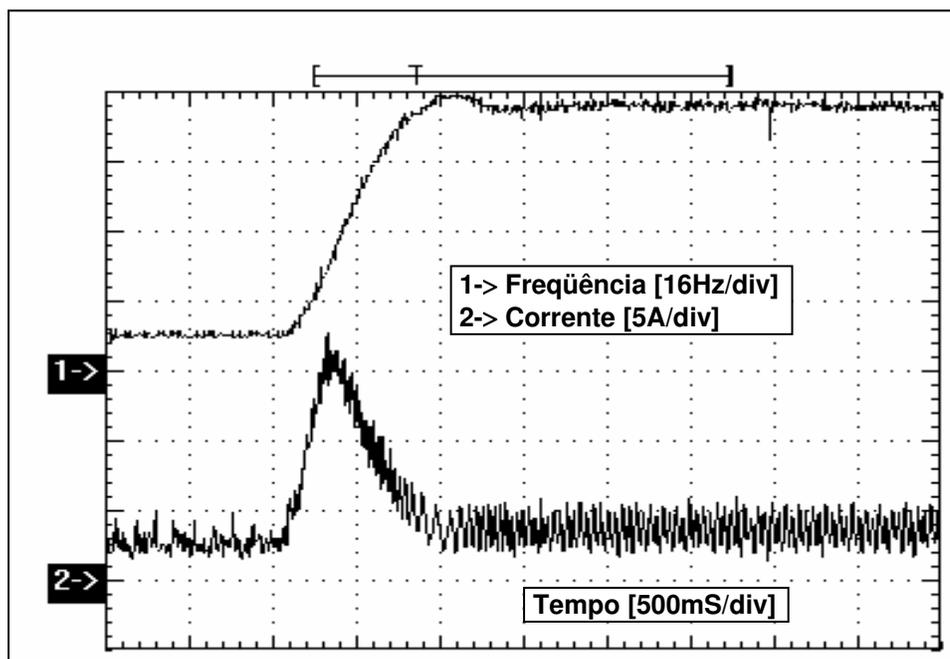


**Figura 4.53** Resultado Prático obtido em Laboratório (Ensaio NeN12)

**Ensaio NeN16 - Controle NeN**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono sem Corrente de Excitação**



**Figura 4.54** Resultado de Simulação com Matlab (Ensaio NeN16)



**Figura 4.55** Resultado Prático obtido em Laboratório (Ensaio NeN16)

Nas figuras a seguir apresentam-se os resultados práticos e simulados obtidos utilizando estratégia de controle digital neuro-nebulosa para acionamento de um motor de indução acoplado a um gerador síncrono trifásico e alimentado através de um inversor de tensão.

Da **Figura 4.56** até **Figura 4.61** considera-se o gerador síncrono trifásico operando com corrente de excitação no circuito de campo, portanto, o motor de indução trabalha com carga e o tempo de resposta com carga é maior que o tempo de resposta em vazio.

A **Figura 4.56** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.57** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **8A** para corrente de entrada **Ie**.

A **Figura 4.58** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.59** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **12A** para corrente de entrada **Ie**.

A **Figura 4.60** apresenta o resultado da simulação computacional utilizando o programa **MATLAB™** enquanto que a **Figura 4.61** apresenta o resultado prático correspondente obtido em laboratório, considerando uma limitação de **16A** para corrente de entrada **Ie**.

Nas figuras correspondentes aos resultados obtidos através de simulação computacional pode-se observar o comportamento dinâmico da frequência de escorregamento **F<sub>s</sub>** representada pela diferença entre a frequência do estator **F<sub>e</sub>** e a frequência do rotor **F<sub>r</sub>**.

A corrente de entrada **Ie** apresenta um comportamento dinâmico muito semelhante ao da frequência de escorregamento **F<sub>s</sub>**, portanto, pode-se concluir que durante o transitório de partida, a corrente de entrada do motor de indução depende da frequência de escorregamento.

A estratégia de controle neuro-nebulosa não trata o gradiente do erro de frequência **δEr**, portanto, nos resultados obtidos pode-se observar a ocorrência de um sobre-sinal na frequência de rotação do motor **Fr**. O sobre-sinal ocorre para uma corrente de entrada **Ie** com valor limite maior que **12A** ou para uma frequência de escorregamento **F<sub>s</sub>** maior que **20Hz**.

Na **Figura 4.60** pode-se observar um instante de operação em vazio (**F<sub>s</sub>=0**), gerado pela estratégia de controle digital neuro-nebulosa, na tentativa de reverter o sobre-sinal que ocorre na frequência de rotação do motor de indução, durante o transitório de partida.

Na **Figura 4.61** pode-se observar uma diferença significativa presente na forma de onda da corrente de entrada, durante o instante de operação em vazio (**F<sub>e</sub>=F<sub>r</sub>**), comparando resultados práticos e simulados, devido a simplificação dos modelos matemáticos.

**Ensaio NeN08ex - Controle NeN**  
**Limite de Corrente - 08 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

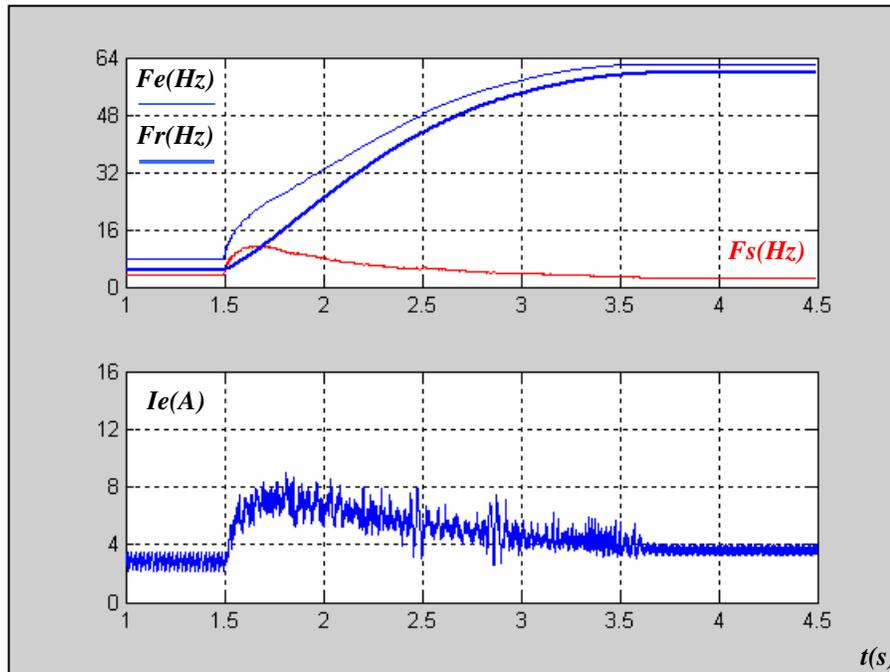


Figura 4.56 Resultado de Simulação com Matlab (Ensaio NeN08ex)

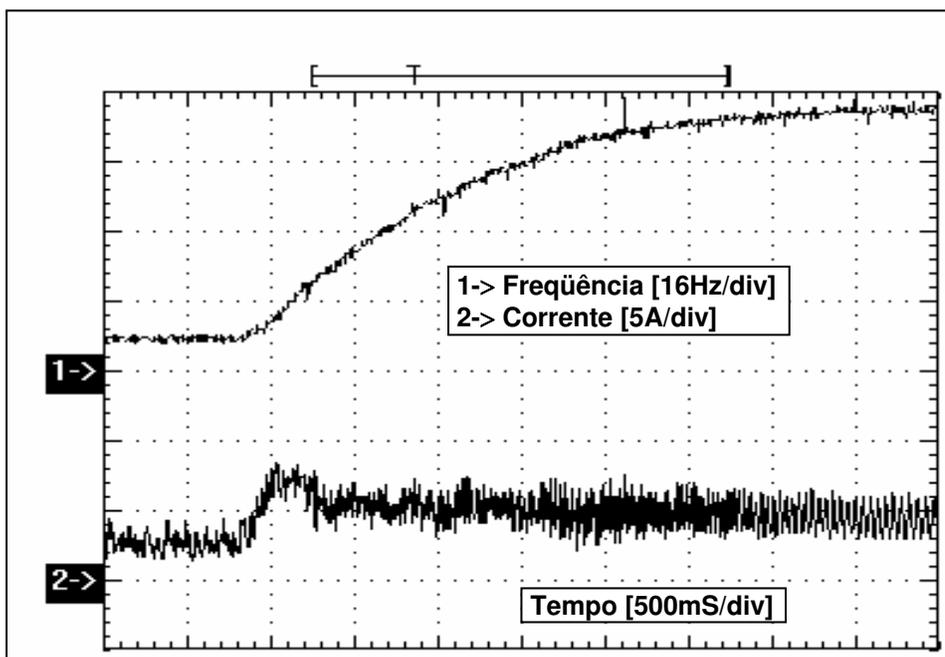
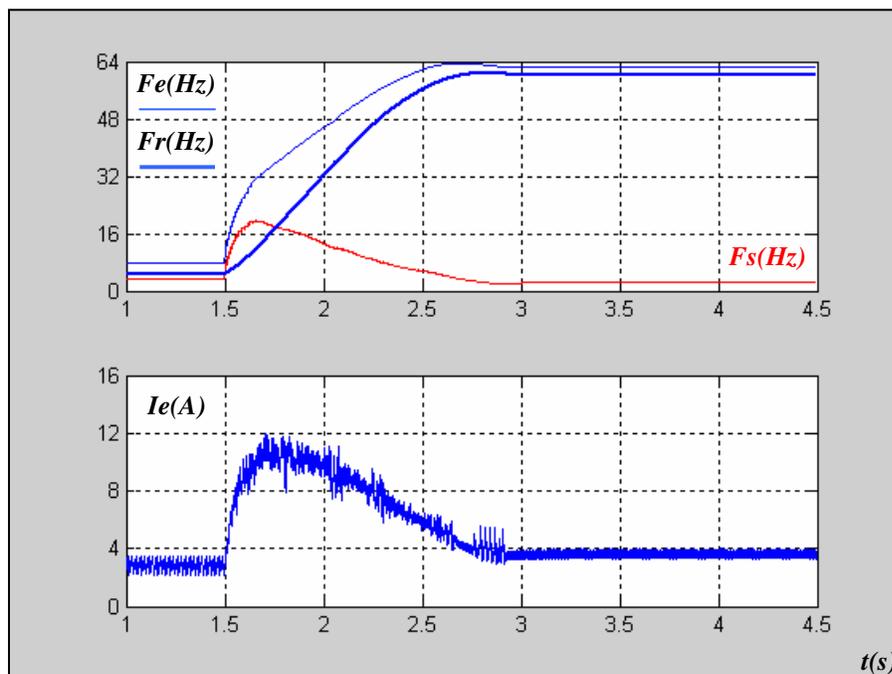
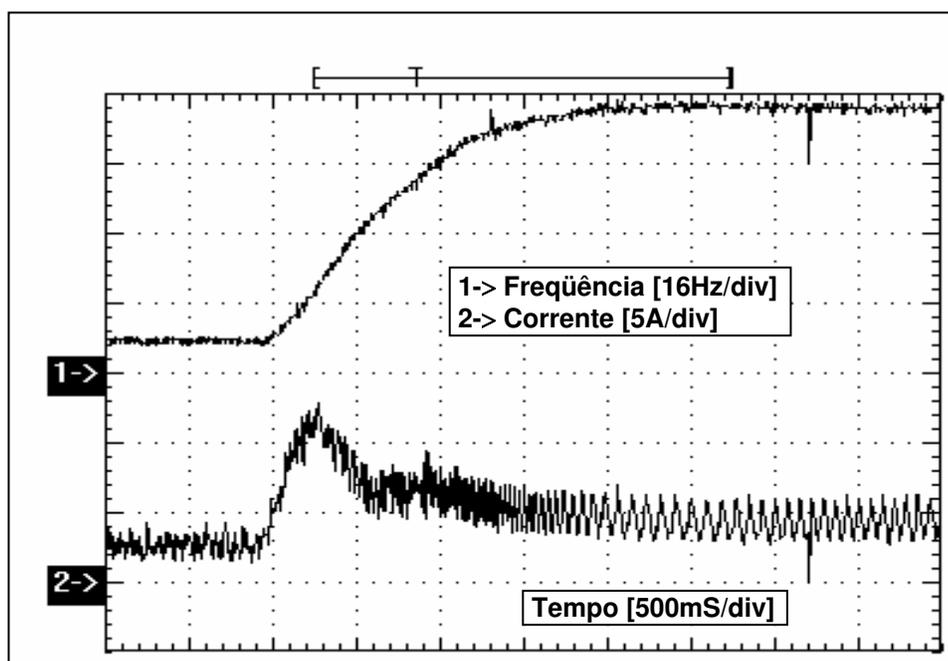


Figura 4.57 Resultado Prático obtido em Laboratório (Ensaio NeN08ex)

**Ensaio NeN12ex - Controle NeN**  
**Limite de Corrente - 12 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**



**Figura 4.58** Resultado de Simulação com Matlab (Ensaio NeN12ex)



**Figura 4.59** Resultado Prático obtido em Laboratório (Ensaio NeN12ex)

**Ensaio NeN16ex - Controle NeN**  
**Limite de Corrente - 16 Ampères**  
**Motor de Indução Acionado por Inversor**  
**Gerador Síncrono com Corrente de Excitação**

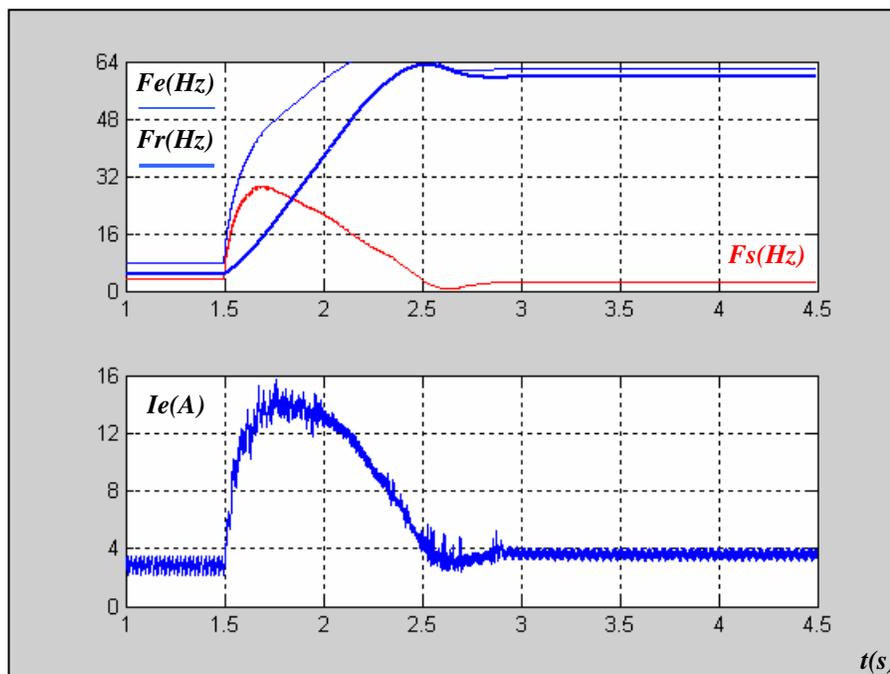


Figura 4.60 Resultado de Simulação com Matlab (Ensaio NeN16ex)

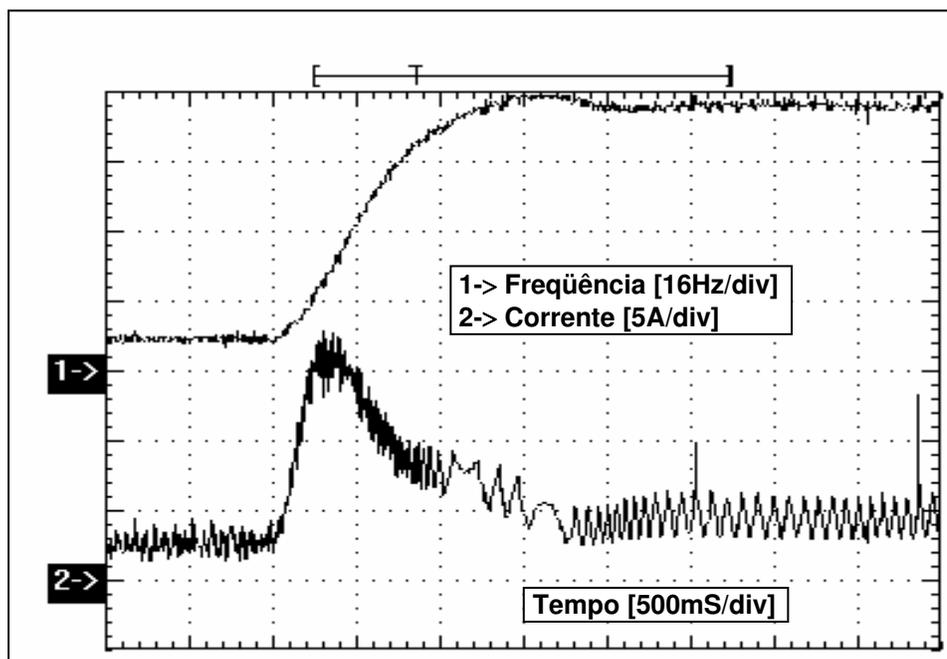


Figura 4.61 Resultado Prático obtido em Laboratório (Ensaio NeN16ex)

## Capítulo 5

### Análise de Resultados e Conclusão

#### 5.1 Introdução

Apresenta-se neste capítulo a análise dos resultados obtidos em laboratório utilizando um motor de indução trifásico, ligado direto na rede de alimentação ou acionado por um inversor de tensão, acoplado a um gerador síncrono trifásico, conectado a uma carga resistiva capaz de dissipar até **750W** de potência ativa. A tensão de saída do gerador síncrono foi ajustada através da corrente de excitação aplicada no circuito de campo, para proporcionar carga nominal ao motor de indução, operando com frequência de **60Hz**. A corrente de excitação do circuito de campo do gerador foi controlada através do ajuste de uma fonte de tensão e podia ser interrompida para o motor de indução operar em vazio, sempre acoplado ao gerador síncrono.

Na implementação do sistema de controle foi utilizado um microcomputador equipado com placa de aquisição de dados e circuitos condicionadores de sinais analógicos. Na elaboração dos algoritmos de controle, considerou-se um sistema digital concebido para ser implementado com o microcontrolador **MCS-80C31** da família **Intel-8051**. Com a placa de aquisição de dados **PCL-711** foi possível estabelecer a precisão adequada para converter as variáveis analógicas de entrada e saída em variáveis digitais equivalentes, conforme indicado na **Tabela 5.1**.

**Tabela 5.1 Resolução da Placa de Aquisição de Dados**

<i>Placa de Aquisição de Dados</i>	<i>Entradas</i>		<i>Saída</i>
	<i>Fr [Hz]</i>	<i>Ie [A]</i>	<i>Vc [Hz]</i>
<i>Variáveis</i>	<i>Fr [Hz]</i>	<i>Ie [A]</i>	<i>Vc [Hz]</i>
<i>Precisão de Cálculo</i>	$\frac{80}{1024}$	$\frac{25}{1024}$	$\frac{80}{2048}$
<i>Resolução</i>	<i>12 Bits</i>	<i>12 Bits</i>	<i>12 Bits</i>
<i>Sinal</i>	$\pm$	$\pm$	$+$

A resolução proporcionada pela placa de aquisição de dados não é a mesma adotada nas estratégias de controle, tendo em vista a arquitetura simplificada do controlador digital, baseada em microcontroladores de 8 bits, na qual foram implementados os algoritmos de controle.

Apresenta-se na **Tabela 5.2** a resolução adotada nas diferentes estratégias de controle e a precisão de cálculo decorrente da representação binária das variáveis de entrada do controlador.

**Tabela 5.2 Representação das Variáveis de Entrada do Controlador**

<b>Estratégias de Controle</b>	<b>Proporcional Integral Derivativa</b>		<b>Nebulosa de Inferência Direta</b>		<b>Nebulosa de Inferência Indireta</b>		<b>Neuro Nebulosa</b>	
	<b><math>E_r, \delta E_r</math> [Hz]</b>	<b><math>E_i, \delta E_i</math> [A]</b>	<b><math>F_r, F_d</math> [Hz]</b>	<b><math>I_e, I_d</math> [A]</b>	<b><math>E_r, \delta E_r</math> [Hz]</b>	<b><math>E_i, \delta E_i</math> [A]</b>	<b><math>Abs(E_r)</math> [Hz]</b>	<b><math>Abs(E_i)</math> [A]</b>
<b>Precisão de Cálculo</b>	$\frac{64}{128}$	$\frac{16}{128}$	$\frac{64}{256}$	$\frac{16}{256}$	$\frac{64}{128}$	$\frac{16}{128}$	$\frac{64}{256}$	$\frac{16}{256}$
<b>Resolução</b>	8 Bits	8 Bits	8 Bits	8 Bits	8 Bits	8 Bits	8+1 Bits	8+1 Bits
<b>Sinal</b>	±	±	+	+	±	±	±	±

Apresenta-se na **Tabela 5.3** a resolução adotada nas diferentes estratégias de controle e a precisão de cálculo decorrente da representação binária das variáveis de saída do controlador.

**Tabela 5.3 Representação das Variáveis de Saída do Controlador**

<b>Estratégia de Controle</b>	<b>Proporcional Integral Derivativa</b>		<b>Nebulosa de Inferência Direta</b>		<b>Nebulosa de Inferência Indireta</b>		<b>Neuro Nebulosa</b>
	<b><math>F_c</math></b>	<b><math>I_c</math></b>	<b><math>F_c</math></b>	<b><math>I_c</math></b>	<b><math>F_c</math></b>	<b><math>I_c</math></b>	<b><math>Abs(A_c)</math></b>
<b>Precisão de Cálculo</b>	$\frac{4}{128}$	$\frac{4}{128}$	$\frac{4}{128}$	$\frac{4}{128}$	$\frac{4}{128}$	$\frac{4}{128}$	$\frac{16}{256}$
<b>Resolução</b>	8 Bits	8 Bits	8 Bits	8 Bits	8 Bits	8 Bits	8+1 Bits
<b>Sinal</b>	±	±	±	±	±	±	±

A estratégia de controle nebulosa de inferência direta apresenta a melhor precisão de cálculo, pois opera somente com valores positivos nas entradas do controlador digital, enquanto que as demais estratégias de controle operam com valores positivos e negativos, incluindo o bit de sinal na representação binária das variáveis de entrada do controlador.

Na estratégia de controle neuro-nebulosa, o sinal da variável de saída deve ser tratado separadamente, devido à necessidade de manter os 8 bits significativos da representação binária, para minimizar os erros de interpolação gerados pela rede neural artificial.

A precisão de cálculo utilizada na obtenção das variáveis de saída do controlador, não afeta a precisão global do sistema de controle digital, que depende apenas da resolução adotada na representação binária das variáveis de entrada do controlador digital.

## 5.2 *Análise de Resultados*

Para avaliar as estratégias de controle foram realizados vários ensaios práticos utilizando diferentes algoritmos de controle, considerando o motor indução sempre acoplado ao um gerador síncrono, com seu circuito de campo operando com uma corrente de excitação ajustada para proporcionar carga nominal ao motor de indução. Os ensaios práticos foram realizados utilizando controle proporcional integral derivativo, controle nebuloso e controle neuro-nebuloso.

Para diferentes estratégias de controle, valores múltiplos da corrente nominal do motor de indução foram aplicados na entrada de corrente desejada  $I_d$ . As curvas apresentadas no final do *Capítulo 4*, mostram a resposta do motor de indução para um degrau de  $8\text{Hz}$  a  $60\text{Hz}$ , aplicado na entrada de frequência desejada  $F_d$ , com limitação de corrente de  $8A$ ,  $12A$  ou  $16A$ .

O inversor de tensão *PWM* utilizado nos ensaios práticos é bastante antigo e o ruído gerado pelo chaveamento dos transistores muito elevado, como pode ser observado nas curvas de corrente eficaz, no entanto, o sistema de controle foi capaz de ajustar a frequência de rotação e sustentar a limitação da corrente de entrada do motor de indução trifásico.

Os resultados dos ensaios práticos do acionamento do motor de indução foram obtidos em laboratório, considerando diferentes condições de partida, utilizando como carga um gerador síncrono trifásico, cujo conjugado resistente varia linearmente com a frequência de rotação do motor. Portanto, quando o motor de indução, acoplado ao gerador síncrono, atinge a frequência nominal  $60\text{Hz}$ , o conjugado de carga também atinge o valor nominal  $4.0\text{Nm}$ .

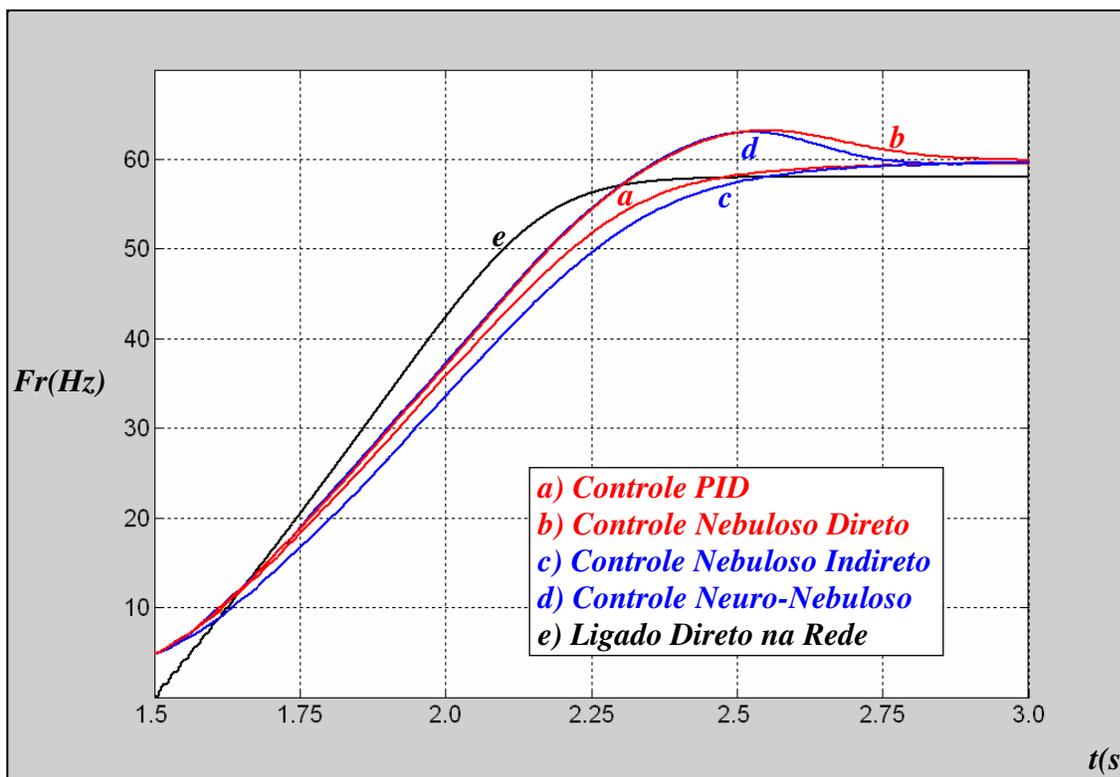
### 5.2.1 *Tempo de Resposta do Motor de Indução*

O tempo de resposta do motor de indução depende, principalmente, do valor da corrente de partida que o circuito de acionamento é capaz de suprir, assim, quando o motor de indução está ligado direto na rede elétrica de alimentação o tempo de resposta é o menor possível, por outro lado, o acionamento do motor de indução com frequência variável não é possível.

Com a utilização de um inversor de tensão é possível controlar a frequência de rotação do motor de indução, no entanto, o tempo de resposta depende da capacidade de corrente do inversor de tensão. Neste trabalho, foi utilizado um inversor de tensão capaz de fornecer uma corrente de partida de até quatro vezes o valor eficaz da corrente nominal do motor de indução.

Deste modo, foi possível avaliar o tempo de resposta do motor de indução, acionado por um inversor de tensão, com auxílio de um controlador digital operando em malha fechada, capaz de controlar a frequência de rotação e a corrente fornecida pelo inversor de tensão ao motor de indução, utilizando diferentes estratégias de controle. Apresentam-se na **Figura 5.1** os resultados obtidos através de simulação computacional, mostrando o tempo de resposta do motor de indução para diferentes condições de operação e diferentes estratégias de controle.

O tempo de resposta do motor de indução ligado direto na rede elétrica de alimentação, acoplado ao gerador síncrono com conjugado resistente equivalente ao conjugado nominal do motor de indução em **60Hz**, sem nenhum tipo de estratégia de controle, é aproximadamente igual a **1.0** segundo, para uma corrente de partida maior que **20A** eficazes.



**Figura 5.1** Tempo de Resposta do Motor de Indução com Inversor de Tensão

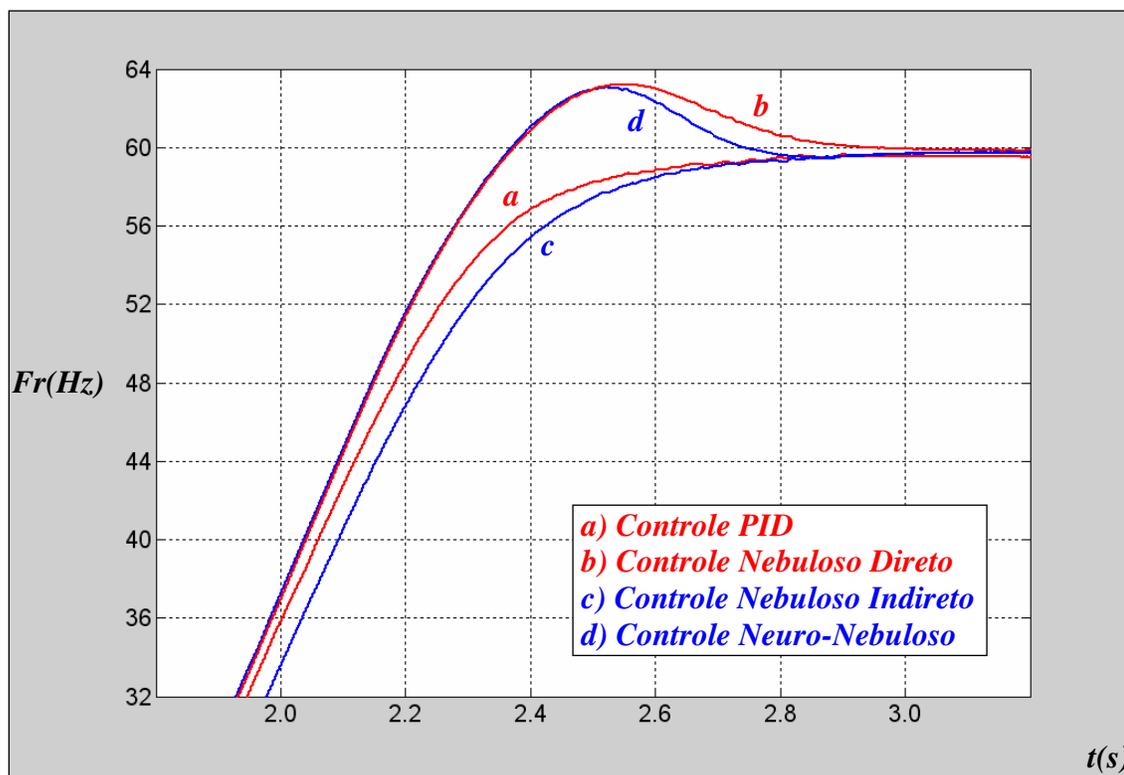
O tempo de resposta do motor de indução acionado por um inversor de tensão **PWM**, acoplado ao gerador síncrono com conjugado resistente equivalente ao conjugado nominal do motor de indução, utilizando as estratégias de controle proporcional integral derivativo, nebulosa de inferência direta, nebulosa de inferência indireta e neuro-nebulosa, é aproximadamente igual a **1.5** segundos, para uma corrente de partida limitada em **16A** eficazes.

### 5.2.2 Comportamento do Sistema de Controle

O comportamento do sistema de controle depende, principalmente, da estratégia de controle utilizada no acionamento do motor de indução, portanto, quando o sistema de controle permite uma corrente de partida igual a **16A**, é possível avaliar os parâmetros de especificação da curva característica de um sistema de segunda ordem, tais como: constante de tempo, tempo de subida, tempo de pico, tempo de acomodação, valor de pico e sobre-sinal.

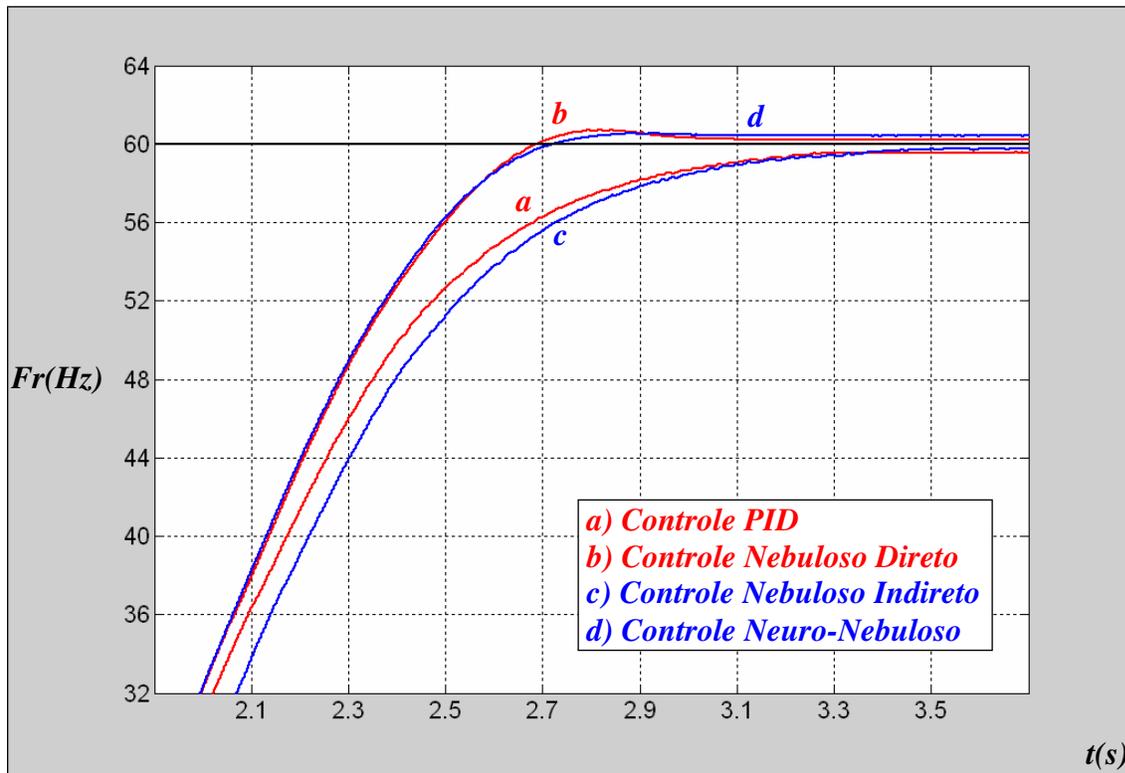
Para valores de corrente de partida em torno de **8A**, a curva característica do sistema não permite avaliar todos os parâmetros de especificação de um sistema de segunda ordem, assim, apresenta-se a seguir, as curvas características do sistema de controle, para valores de corrente de partida de **12A** e **16A**, considerando as diferentes estratégias de controle.

A **Figura 5.2** mostra as curvas características do sistema de controle para acionamento do motor de indução, com corrente de partida limitada em **16A**, obtidas através de simulação computacional, considerando diferentes estratégias de controle, para um degrau de **8Hz** a **60Hz**, aplicado na entrada de referência do controlador digital, no instante de **1.5** segundos.



**Figura 5.2** Comportamento do Sistema de Controle com Corrente de Partida 16A

A **Figura 5.3** mostra as curvas características do sistema de controle para acionamento do motor de indução, com corrente de partida limitada em **12A**, obtidas através de simulação computacional, considerando diferentes estratégias de controle, para um degrau de **8Hz** a **60Hz**, aplicado na entrada de referência do controlador digital, no instante de **1.5** segundos.

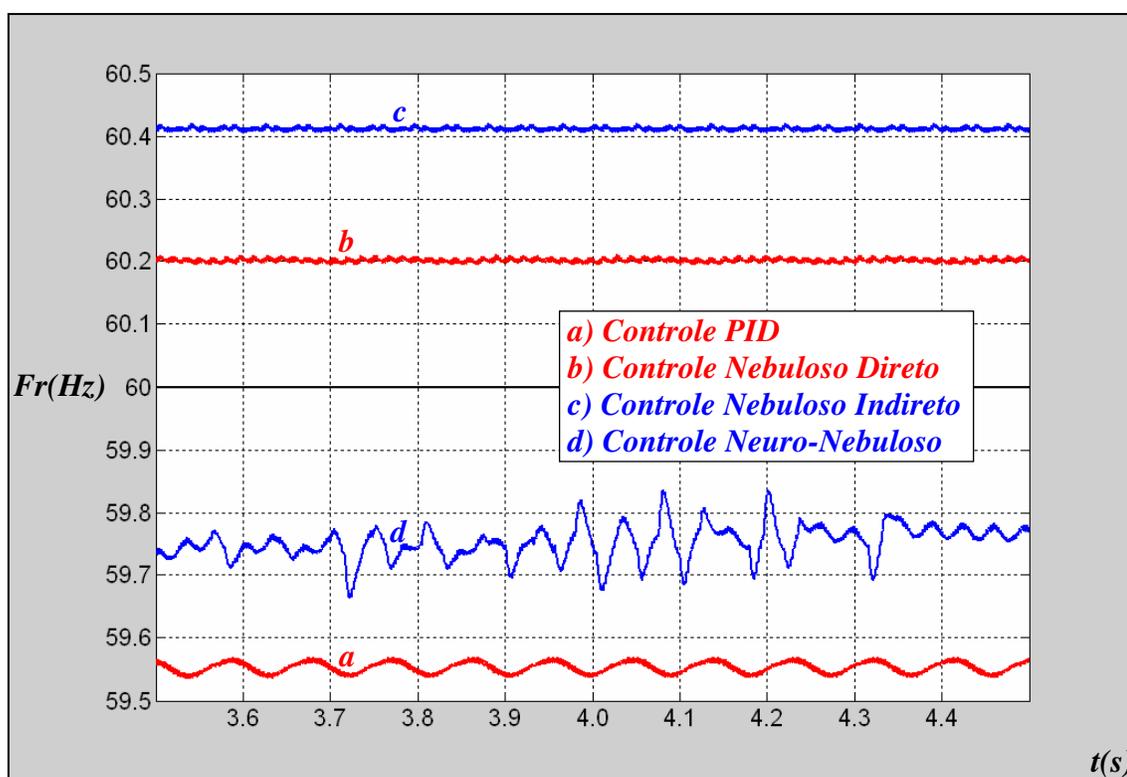


**Figura 5.3** Comportamento do Sistema de Controle com Corrente de Partida 12A

Considerando o comportamento do sistema controle digital, para uma corrente de partida limitada em **16A**, fica evidente a ação de controle derivativa, presente nas estratégias de controle proporcional integral derivativa e nebulosa de inferência indireta, evitando a ocorrência de sobre-sinal sem comprometer o tempo de resposta, conforme indicado na **Figura 5.2**. Por outro lado, para uma corrente de partida limitada em **12A**, o sobre-sinal presente nas estratégias de controle nebulosa de inferência direta e neuro-nebulosa não tem valor significativo, portanto, o tempo de resposta do motor de indução melhora sensivelmente, conforme indicado na **Figura 5.3**.

Outro fator que interfere no tempo de resposta do motor de indução e no comportamento do sistema de controle é o tempo de amostragem usado no controlador digital operando em malha fechada. Nos ensaios realizados em laboratório o tempo de amostragem usado foi **T=1.0ms**, que corresponde ao valor mínimo possível, limitado pelo ambiente de desenvolvimento.

A **Figura 5.4** mostra as curvas de regime das estratégias de controle para acionamento do motor de indução, com corrente de partida limitada em **12A**, obtidas através de simulação computacional, considerando as estratégias de controle proporcional integral derivativa, nebulosa de inferência direta, nebulosa de inferência indireta e neuro-nebulosa, para um sinal degrau de **8Hz** a **60Hz** aplicado na entrada de referência do controlador digital, no instante de **1.5** segundos.



**Figura 5.4** Precisão do Sistema para Diferentes Estratégias de Controle

A precisão do sistema de controle digital depende da estratégia de controle adotada para acionamento do motor de indução. A estratégia de controle nebulosa de inferência direta é mais precisa, sendo capaz de atuar com a precisão de **0.25Hz** manipulando apenas valores positivos de frequência de rotação  $F_r$  e corrente de entrada  $I_e$ . As estratégias de controle proporcional integral derivativa, nebulosa de inferência indireta e neuro-nebulosa, atuam com uma precisão de **0.50Hz** manipulando valores positivos e negativos de erro de frequência  $E_r$  e erro de corrente  $E_i$ .

Como a estratégia de controle neuro-nebulosa utiliza apenas uma malha de controle, está sujeita aos ruídos presentes no sinal de corrente, provocados pelo chaveamento dos dispositivos semicondutores do inversor de tensão, conforme indicado na **Figura 5.4**. Os erros de interpolação gerados pela rede neural artificial também afetam a estratégia de controle neuro-nebulosa.

### 5.3 Conclusão

Considerando as técnicas de controle vetorial para acionamento de máquinas elétricas de indução, mais eficientes e precisas que as técnicas de controle escalar, o objetivo principal deste trabalho foi o desenvolvimento de um controlador digital simples e eficiente, utilizando conceitos de lógica nebulosa, aplicado no acionamento de máquinas elétricas industriais, sem a necessidade de recorrer aos modelos matemáticos das máquinas.

Trata-se de um trabalho de pesquisa aplicada, que apresenta o projeto de um controlador digital de arquitetura simplificada, que permite reduzir o tempo de processamento necessário para a parametrização de um sistema de controle nebuloso, utilizando uma estrutura padronizada para representação das funções de pertinência e ponderação dos termos lingüísticos, relacionados com as variáveis de entrada e saída de controlador nebuloso.

Para avaliar o desempenho deste controlador, foi implementado um sistema digital para acionamento de motor de indução, com controle da frequência de rotação e limitação da corrente de entrada, capaz de gerar em tempo real, um perfil de frequência adequado, sempre que um novo valor da frequência de operação seja estabelecido, impedindo que a corrente de entrada ultrapasse o limite estipulado, evitando o desligamento desnecessário do sistema.

Nesta aplicação prática destacam-se as novas estratégias de controle desenvolvidas e as contribuições oferecidas aos usuários de máquinas elétricas de indução. Pelos resultados práticos obtidos, pode-se verificar a versatilidade e a confiabilidade de um sistema de controle digital de baixo custo, baseado em microcontroladores de 8 bits da família *Intel-8051*. O microcontrolador *MCS-80C31* é capaz de acessar vários blocos de *64 kbytes* de memória de dados que armazenam todas as ações de controle. Os algoritmos de controle são armazenados em *64 kbytes* de memória específicos para programas escritos na linguagem do microcontrolador.

O uso de microcontroladores mais poderosos, com arquitetura de *16* ou *32* bits, não pode ser descartada, tendo em vista as aplicações industriais que exigem sistemas de controle mais eficientes e precisos. Considerando microcontroladores com arquitetura de *16* bits, o controlador nebuloso desenvolvido neste trabalho teria maior capacidade para armazenar dados e executar operações lógicas e aritméticas com maior rapidez e precisão, no entanto, a implementação do controlador nebuloso considerando microcontroladores com arquitetura de *32* bits, implicaria na utilização de memórias com grande capacidade de armazenamento.

### 5.3.1 Estratégias de Controle

Inicialmente, foram desenvolvidas as estratégias de controle digital do tipo nebulosa de inferência direta e nebulosa de inferência indireta, que foram comparadas com uma estratégia de controle do tipo proporcional integral derivativa, para demonstrar a possibilidade de se utilizar as novas estratégias, sem comprometer o tempo de resposta e a precisão do sistema.

O cálculo de todas as ações de controle é feito através um único operador de agregação, representado por uma simples expressão algébrica, que atua como um mecanismo de interpolação linear, capaz de mapear apenas superfícies planas. Com auxílio de uma rede neural artificial do tipo *CMAC*, que atua como um mecanismo de interpolação não-linear, pode-se implementar um controlador neuro-nebuloso capaz de mapear superfícies não-planas.

Considerando que estas estratégias de controle possuem duas malhas de realimentação, uma para controle da frequência de rotação e outra para controle da corrente de entrada, uma nova estratégia de controle do tipo neuro-nebulosa foi desenvolvida.

Usando uma única malha de controle a partir das variáveis de entrada erro de frequência *Er* e erro de corrente *Ei*, esta nova estratégia de controle apresentou ótimos resultados como pode ser observado na *Figura 5.2* que mostra a resposta do sistema de controle para uma corrente de entrada limitada em *16A* e na *Figura 5.4* que mostra a precisão das quatro estratégias de controle digital desenvolvidas neste trabalho de engenharia aplicada.

Todas as estratégias de controle podem ser utilizadas para determinar o menor tempo de aceleração, respeitando os limites de operação do motor de indução e principalmente, respeitando a capacidade de corrente do inversor de tensão. Por outro lado, o procedimento de desaceleração do motor de indução depende do momento de inércia da carga.

Como o momento de inércia do gerador síncrono é **30** vezes maior que o momento de inércia do motor de indução, o procedimento de desaceleração torna-se muito complexo para as estratégias de controle desenvolvidas neste trabalho, sendo necessário estimar o valor da corrente de entrada em função do valor da frequência de escorregamento.

Apresenta-se a seguir proposta de trabalhos futuros considerando a importância de se estimar a corrente de entrada *Ie* do motor de indução a partir da frequência de escorregamento, sendo possível também estimar a frequência de escorregamento *Fs* a partir da corrente de entrada do motor de indução, evitando o uso de sensores mecânicos.

### 5.3.2 *Trabalhos Futuros*

A utilização de redes neurais artificiais de estrutura simplificada, como é o caso **CMAC**, pode auxiliar na estimação da corrente de entrada, a partir da frequência de escorregamento, ou ainda, na estimação da frequência de escorregamento, a partir da corrente de entrada. Trata-se do desenvolvimento de um controlador adaptativo, capaz de simplificar o sistema de aquisição de dados, descartando uma das variáveis de entrada, durante o acionamento do motor de indução.

A estrutura da rede neural baseada no **CMAC** implementada em linguagem de máquina, fica residente na memória de programa do microcontrolador, usando a memória de dados para armazenar informações durante o um processo de treinamento, considerando diversas condições de operação do motor de indução, visando obter um conjunto de informações consistente.

A primeira possibilidade seria armazenar a corrente de entrada **Ie** em função da variável de controle **Vc** e da frequência de escorregamento **Fs**, durante o processo de treinamento, para obter  $Ie=f(Vc,Fs)$  durante o procedimento de estimação, descartando a aquisição de corrente.

A segunda possibilidade seria armazenar a frequência de escorregamento **Fs** em função da variável de controle **Vc** e da corrente de entrada **Ie**, durante o processo de treinamento, para obter  $Fs=f(Vc,Ie)$  durante o procedimento de estimação, descartando a aquisição de frequência.

Alguns estudos foram realizados no sentido de desenvolver um controlador adaptativo, mas existe a dificuldade de estabelecer os ensaios necessários para obter um conjunto de dados consistente, considerando todas as possibilidades de acionamento de um motor de indução. Um exemplo de utilização do controlador adaptativo seria durante o procedimento de desaceleração, onde é preciso estimar a corrente de entrada a partir da frequência de escorregamento.

Considerando a primeira possibilidade, para obter o valor de  $Ie=f(Vc,Fs)$ , o processo de treinamento ocorre durante o procedimento de aceleração, deste modo, a informação armazenada pode ser utilizada durante o procedimento de desaceleração, assumindo que a corrente de entrada varia da mesma forma, permitindo que todas estratégias de controle desenvolvidas neste trabalho sejam utilizadas durante os procedimentos de aceleração e desaceleração.

## Apêndice 1

### Lógica Nebulosa

#### 1.1 Introdução

A lógica nebulosa é a lógica baseada na teoria dos conjuntos nebulosos que difere dos sistemas lógicos tradicionais em suas características e seus detalhes. Nesta lógica, o raciocínio exato corresponde a um caso limite do raciocínio aproximado, interpretado como um processo de composição de relações nebulosas. O valor verdade de uma proposição pode ser um subconjunto nebuloso de qualquer conjunto parcialmente ordenado, ao contrário dos sistemas lógicos binários, onde o valor verdade só pode assumir dois valores: verdadeiro  $I$  ou falso  $0$ .

Nos sistemas lógicos multivalores o valor verdade de uma proposição pode ser um elemento de um conjunto finito ou uma relação booleana. Na lógica nebulosa, os valores verdade são expressos lingüisticamente, (*pouco verdade, verdade, muito verdade,...*), onde cada termo lingüístico é interpretado como um subconjunto nebuloso do intervalo unitário.

Nos sistemas lógicos clássicos, os predicados são exatos e o modificador mais utilizado é a negação enquanto que na lógica nebulosa os predicados são nebulosos (*alto, médio, baixo,...*) e uma variedade de modificadores de predicados são possíveis (*muito, mais ou menos,...*).

Estes modificadores são essenciais para geração de termos lingüísticos consistentes (*muito alto, mais ou menos perto,...*). Nos sistemas lógicos clássicos existem somente os quantificadores existenciais e universais. A lógica nebulosa admite, em adição, uma ampla variedade de quantificadores (*pouco, vários, usualmente,...*).

Na lógica nebulosa existe a opção adicional de se utilizar probabilidades lingüísticas (*provável, altamente provável, improvável, etc...*), interpretadas como números nebulosos e manipuladas através da aritmética nebulosa. Em contraste com a lógica clássica, o conceito de possibilidade é interpretado a partir de subconjuntos nebulosos (ZADEH, 1988).

Nas teorias de controle clássica e moderna o primeiro passo para implementar o controle de um processo, é derivar o modelo matemático que descreve o processo. Este procedimento, requer que se conheça detalhadamente o processo a ser controlado, o que nem sempre é factível se o processo é muito complicado (ZADEH, 1973).

As teorias de controle existentes se aplicam a uma grande variedade de sistemas, onde o processo é bem definido. Várias técnicas de controle, tais como: controle linear multivariável (DOYLE & SKIN, 1981), sistemas lineares estocásticos (BERTSEKAS, 1976), estimação de estado a partir de medidas ruidosas e classes de problemas não-lineares determinísticos, foram criadas e aplicadas com sucesso em um grande número de problemas bem postulados.

Entretanto, todas estas técnicas de controle não são capazes de resolver problemas reais cuja modelagem matemática é impraticável. Por exemplo, em diversas situações um volume considerável de informações essenciais só é conhecido a priori de forma qualitativa. Do mesmo modo, os critérios para avaliação de desempenho de sistemas só estão disponíveis na forma de termos lingüísticos. Assim, este panorama leva a imprecisões e falta de exatidão que inviabilizam a maioria das teorias de controle utilizadas até agora.

A teoria de modelagem e o controle nebuloso (LEE, 1990) são técnicas para se manusear informações qualitativas de uma maneira rigorosa. Tais técnicas consideram o modo como a falta de exatidão e a incerteza são descritas e, fazendo isso, tornam-se suficientemente poderosos para manipular de maneira conveniente os conhecimentos adquiridos.

A teoria de controle nebuloso trata do relacionamento entre entradas e saídas, agregando vários parâmetros de processo e de controle. Portanto, esta teoria permite a consideração de processos mais complexos, de modo que os sistemas de controle resultantes proporcionam um resultado mais acurado, além de um desempenho estável e robusto.

A grande simplicidade de implementação de sistemas de controle baseados em lógica nebulosa pode reduzir a complexidade de um projeto a um ponto em que problemas anteriormente intratáveis tornaram-se solucionáveis. A utilização da lógica nebulosa em sistemas de controle de processos em tempo real através de computadores ou microcomputadores é das mais convenientes pois não envolve nenhum problema computacional sério.

## ***1.2 Fundamentos da Lógica Nebulosa***

Nesta seção serão apresentadas idéias básicas sobre a teoria de conjuntos e a lógica nebulosa visando à modelagem e o desenvolvimento de sistemas de controle. Apesar de exigir uma complexa base formal sustentando sua utilização na modelagem e controle de sistemas e processos, será evidenciado aqui somente o necessário para o entendimento da teoria básica de controle nebuloso (PEDRYCS & GOMIDE, 1998).

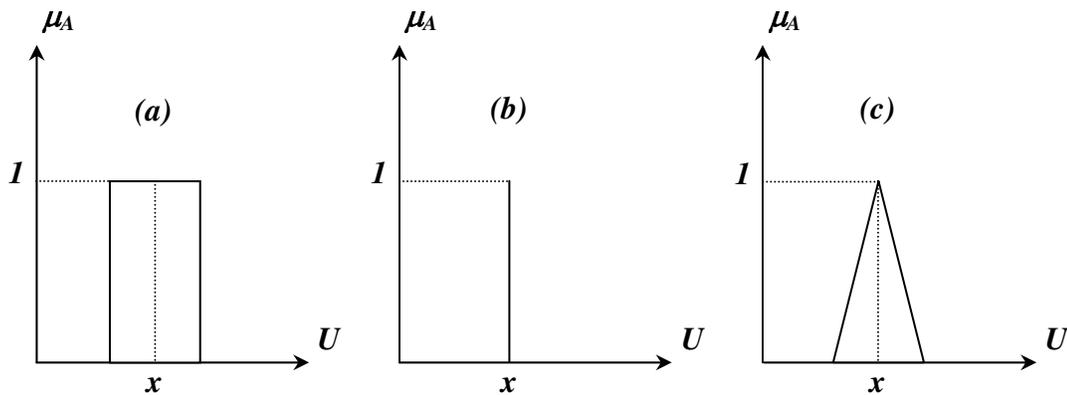
Na teoria clássica de conjuntos, um elemento ou pertence a um conjunto ou não pertence a um conjunto. Portanto, dado um universo de discurso  $U$  e um elemento particular  $x \in U$ , o índice de pertinência  $\mu_A(x)$  com respeito a um conjunto  $A$  é dado por:

$$\begin{cases} \mu_A(x) = 1 & \text{se } x \in A \\ \mu_A(x) = 0 & \text{se } x \notin A \end{cases} \quad (1.1)$$

A função  $\mu_A(x)$  é chamada de função característica na teoria clássica de conjuntos.

$$\mu_A(x) : U \rightarrow \{0, 1\} \quad (1.2)$$

Freqüentemente, uma generalização desta idéia é usada, por exemplo, para manipulação de dados com erros limitados. Todos os números dentro de um erro percentual terão índices de pertinência iguais a  $1$  e todos os demais índices de pertinência iguais a  $0$ . Para o caso preciso, o índice de pertinência é  $1$  somente no número exato e  $0$  para todos os demais casos.



**Figura 1.1 Funções de Pertinência**

Como proposto por (ZADEH, 1965), uma caracterização mais ampla, na medida em que sugere que alguns elementos são mais membros de um conjunto do que outros, o índice de pertinência pode então assumir qualquer valor real entre  $0$  e  $1$ , sendo que o valor  $0$  indica uma completa exclusão e o valor  $1$  representa completa pertinência. Esta generalização aumenta o poder de expressão da função característica.

Assim, para expressar a idéia de que uma temperatura tem seu valor por volta de  $25^\circ\text{C}$ , pode-se utilizar uma função de pertinência triangular, com pico em  $x$  igual a  $25$ , para sugerir a idéia de que quanto mais perto de  $25$ , mais se identifica o conceito representado.

Finalmente, seja  $U$  um conjunto de objetos denominados genericamente por  $x$ . O conjunto  $U$  é chamado de universo de discurso, podendo ser contínuo ou discreto. Um conjunto nebuloso  $A$  em um universo de discurso  $U$  é definido pela função de pertinência  $\mu_A(x)$  que assume valores dentro do intervalo  $[0, I]$ .

$$\mu_A(x) : U \rightarrow [0, I] \quad (1.3)$$

O conjunto suporte de um conjunto nebuloso  $A$  é o subconjunto dos pontos  $x$  de  $U$  tal que  $\mu_A(x) > 0$ . Um conjunto nebuloso cujo conjunto suporte é um único ponto de  $U$  com  $\mu_A(x)$  igual  $I$  é chamado de um conjunto nebuloso unitário.

Sejam  $A$  e  $B$  dois conjuntos nebulosos em  $U$  com funções de pertinência  $\mu_A(x)$  e  $\mu_B(x)$ , respectivamente. As operações de conjuntos tais como a união  $A \cup B$  e a intersecção  $A \cap B$  são definidas pelo sistema de equações (1.4) e o complemento  $\bar{A}$  é dado pela equação .

$$\begin{cases} \mu_{A \cup B}(x) = \mu_A(x) |s| \mu_B(x) \\ \mu_{A \cap B}(x) = \mu_A(x) |t| \mu_B(x) \end{cases} \quad (1.4)$$

Onde:  $|t|$  - é uma norma triangular (norma-t) e  
 $|s|$  - é uma co-norma triangular (norma-s).

$$\mu_{\bar{A}}(x) = I - \mu_A(x) \quad (1.5)$$

Uma norma triangular é uma função para qualquer  $(x, y, z, w) \in [0, I]$ , tal que:

$$|t| : [0, I] \times [0, I] \rightarrow [0, I]$$

Satisfazendo as seguintes propriedades:

- (i)  $(x |t| w) \leq (y |t| z)$  se  $(x \leq y)$  e  $(w \leq z)$
- (ii)  $(x |t| y) = (y |t| x)$
- (iii)  $(x |t| y) |t| z = x |t| (y |t| z)$
- (iv)  $(x |t| 0) = 0$  e  $(x |t| I) = x$

Uma co-norma triangular é uma função, para qualquer  $(x,y,z,w) \in [0,1]$ , tal que:

$$|s| : [0,1] \times [0,1] \rightarrow [0,1]$$

Satisfazendo as seguintes propriedades:

$$(i) \quad (x|s|w) \leq (y|s|z) \text{ se } (x \leq y) \text{ e } (w \leq z)$$

$$(ii) \quad (x|s|y) = (y|s|x)$$

$$(iii) \quad (x|s|y)|s|z = x|s|(y|s|z)$$

$$(iv) \quad (x|s|0) = x \text{ e } (x|s|1) = 1$$

Exemplos de normas-t podem incluir o mínimo  $\wedge$  e o produto algébrico  $\otimes$  e exemplos de normas-s podem incluir o máximo  $\vee$  e a soma limitada  $\oplus$ . Assim, se  $A_1, A_2, \dots, A_n$  são conjuntos nebulosos em  $U_1, U_2, \dots, U_n$  então uma relação nebulosa é representada por um conjunto nebuloso em  $U_1 \times U_2 \times \dots \times U_n$  expresso da seguinte maneira, para  $(u_1, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n$ .

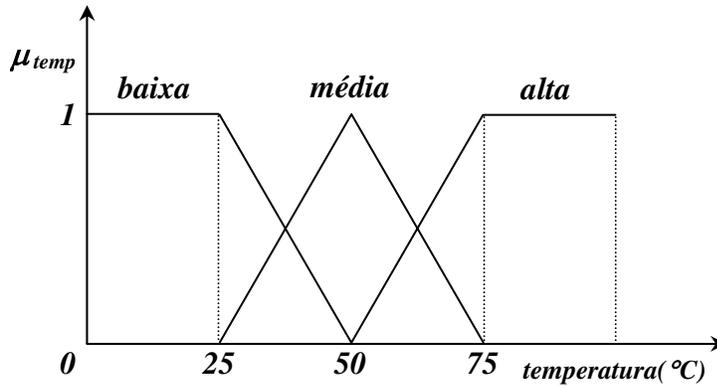
$$R = [(u_1, \dots, u_n), \mu_R(u_1, \dots, u_n)] \quad (1.6)$$

Se  $R$  e  $P$  são relações segundo a lógica nebulosa em  $U \times V$  e  $V \times W$  respectivamente, então a composição de  $R$  e  $P$  é uma relação denotada por  $R \circ P$  definida para  $u \in U, v \in V$  e  $w \in W$ .

$$R \circ P = [(u, w), \sup_v (\mu_R(u, v) | t | \mu_P(v, w))] \quad (1.7)$$

Para expressar os mais diversos conceitos é muito comum a utilização de elementos qualitativos ao invés de valores quantitativos. Elementos típicos incluem (*mais ou menos, alto, não muitos, médio,...*). Estas idéias são capturadas pela definição de variável lingüística. Uma variável lingüística tem por característica assumir valores de um conjunto de termos lingüísticos, ou seja, palavras ou frases.

Portanto, ao invés de assumir instâncias numéricas, a variável lingüística *temperatura* poderá assumir como valor um dos membros do conjunto (*baixa, média, alta*). Para atribuir um significado aos termos lingüísticos, associa-se cada um deles a um conjunto nebuloso definido sobre um universo de discurso comum.



**Figura 1.2 Variável Linguística Temperatura**

A forma mais comum de expressar o conhecimento é por meio de regras lógicas do tipo (*condição-ação*). Nestas regras, um conjunto de condições descrevendo uma parcela observável das saídas do processo é associado com uma ação de controle que deverá manter ou levar o processo às condições de operação desejadas. Tipicamente, uma condição é uma proposição, envolvendo variáveis linguísticas, sobre o valor de alguma das variáveis de entrada, como por exemplo (*o erro é grande e positivo*). De modo análogo, uma típica ação de controle é uma descrição que envolve variáveis linguísticas, como por exemplo (*aumente um pouco a vazão*).

A idéia geral é representar o conhecimento por meio de um conjunto de regras nas quais as condições são dadas a partir de um conjunto de termos linguísticos associados às variáveis de entrada/saída do processo que são variáveis de entrada/saída do controlador/modelo.

As ações de controle ou as saídas do modelo são expressas de modo similar para cada variável de controle ou saída. Regras do tipo (*se-então*) são chamadas declarações condicionais ou simplesmente regras nebulosas. Dependendo do propósito ser controle ou modelagem, podem ser chamadas ainda de regras de controle ou regras de modelagem.

Uma regra nebulosa é interpretada como uma relação nebulosa  $R_i$ , definida como:

*Se (x é  $A_i$ ) e (y é  $B_i$ ) então (z é  $C_i$ )*

$$\mu_{R_i}(u, v, w) = \mu_{A_i \text{ and } B_i \rightarrow C_i}(u, v, w) = \mu_{A_i}(u) \text{ and } \mu_{B_i}(v) \rightarrow \mu_{C_i}(w) \quad (1.8)$$

Onde:  $A_i \text{ and } B_i$  é um conjunto nebuloso  $A_i \times B_i$  em  $U \times V$ ;  
 $A_i \text{ and } B_i \rightarrow C_i$  é uma relação nebulosa em  $U \times V \times W$  e  
 $(\rightarrow)$  denota o operador nebuloso de implicação.

Considerando regras nebulosas do tipo  $A \rightarrow B$ , alguns exemplos de operadores nebulosos de implicação incluem segundo (LEE, 1990):

$$\begin{aligned}\mu_{R_1}(x,y) &= \mu_A(x) \wedge \mu_B(y) \\ \mu_{R_2}(x,y) &= \mu_A(x) \vee \mu_B(y)\end{aligned}\tag{1.9}$$

Em um sistema nebuloso, representando um modelo ou um controlador, cada regra é expressa por uma relação nebulosa. Assim, o comportamento do sistema é caracterizado pelo conjunto das relações nebulosas associadas às regras, e o sistema como um todo será então representado por uma única relação nebulosa, que representa uma combinação de todas as relações nebulosas provenientes das diversas regras:

$$R = \text{agreg}(R_1, R_2, \dots, R_i, \dots, R_n)\tag{1.10}$$

Usualmente, o operador de agregação é interpretado como um operador de união, usando a operação *max*, embora exista uma ampla classe de operadores de agregação. Em lógica nebulosa uma importante regra de inferência é decorrente do *modus ponens* generalizado. Uma regra nebulosa é interpretada pela lei de inferência composicional, sugerida por (ZADEH, 1973).

$$\begin{aligned}\text{Fato:} & \quad (x \text{ é } A') \\ \text{Regra:} & \quad \text{Se } (x \text{ é } A) \text{ então } (y \text{ é } B) \\ \text{Conseqüência:} & \quad (y \text{ é } B')\end{aligned}$$

Nesta abordagem, uma regra nebulosa *se*  $(x \text{ é } A)$  *então*  $(y \text{ é } B)$ , descrita como  $A \rightarrow B$ , é transformada em uma relação nebulosa  $R_{A \rightarrow B}$ . Assim, para  $u \in U$  e  $v \in V$ , temos:

$$\mu_{R_{A \rightarrow B}}(u,v) = \min(\mu_A(u), \mu_B(v))\tag{1.11}$$

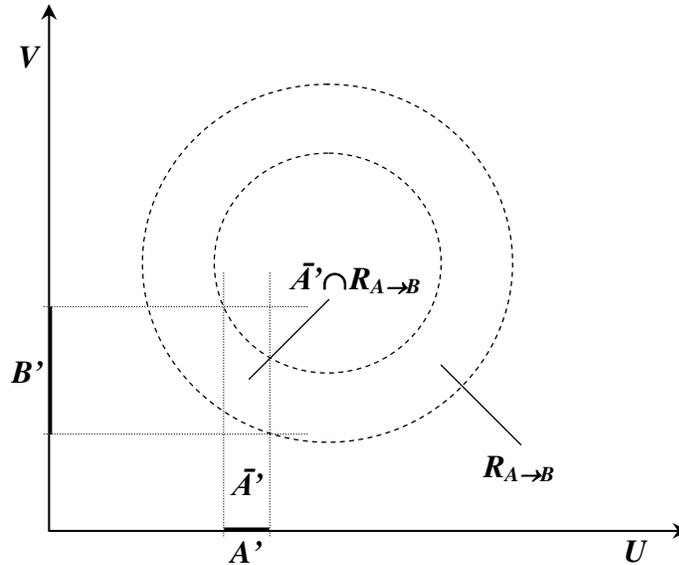
Onde: *min* é o operador de implicação.

Dado um fato  $(x \text{ é } A')$  ou simplesmente  $A'$  e uma regra nebulosa descrita como  $A \rightarrow B$ , a lei de inferência composicional de *Zadeh* diz que  $B'$  é igual a  $A'$  o  $R_{A \rightarrow B}$ .

$$\mu_{B'}(v) = \max_u \left[ \min(\mu_{A'}(u), \mu_{R_{A \rightarrow B}}(u,v)) \right]\tag{1.12}$$

Esta é a lei de inferência *min-max*, cuja interpretação gráfica é apresentada a seguir na *Figura 1.3*, onde a intersecção é dada pela norma-t *min* e a projeção em  $V$  pelo operador *max*. Esta lei é constituída de duas etapas: a intersecção das extensões cilíndricas de  $A'$ ,  $\bar{A}'$  com  $R_{A \rightarrow B}$ , e a projeção desta intersecção em  $V$ . Em geral, a regra da composição é expressa por:

$$\mu_{B'}(v) = \sup_u [\mu_{A'}(u) \wedge \mu_{R_{A \rightarrow B}}(u, v)] \quad (1.13)$$



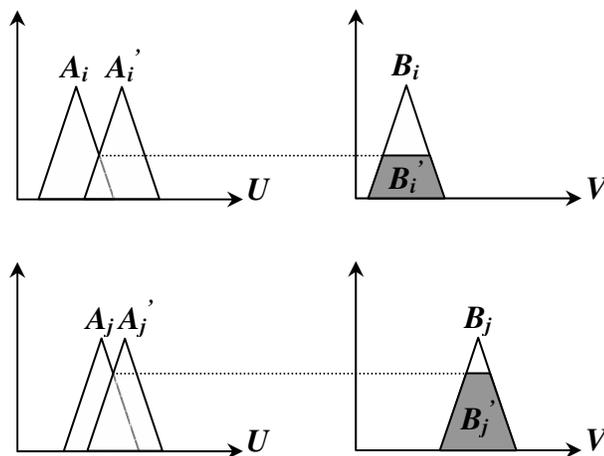
*Figura 1.3 Regra de Inferência Composicional*

Quando mais de uma regra é acionada, as contribuições das diversas regras após a inferência são combinadas pelo operador de agregação. Por exemplo, supondo-se que  $B_1', \dots, B_n'$  são os resultados derivados das diversas regras acionadas, todos relacionados com a mesma variável lingüística, então o resultado combinado  $B'$  é dado pela seguinte expressão:

$$B' = \bigcup_i^n B_i \quad (1.14)$$

Onde:  $\bigcup$  representa o operador de agregação do tipo união.

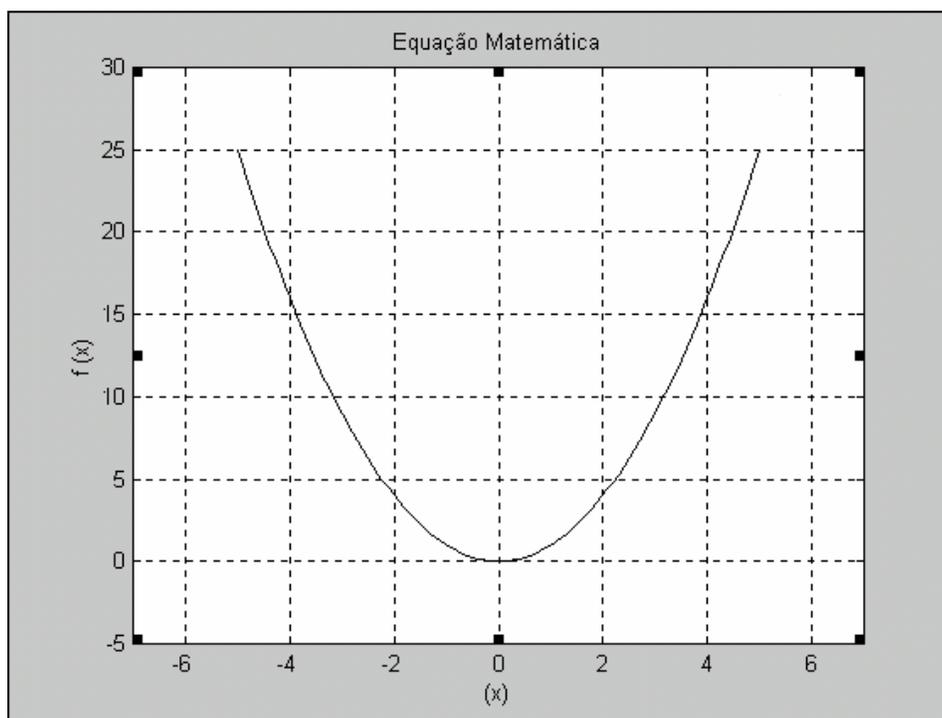
Apresenta-se na *Figura 1.4* o processo de inferência do tipo *min-max* quando existem duas regras  $A_i \rightarrow B_i$  e  $A_j \rightarrow B_j$ . O fato de entrada é representado pelo conjunto nebuloso  $A'$ .



*Figura 1.4 Mecanismo de Inferência*

### 1.3 Descrição de Sistemas e Processos

Para modelar um sistema é necessário descrever o comportamento do mesmo para análise de desempenho, simulação computacional ou projeto de controladores. Desta forma, podem ser consideradas três categorias de representação de sistemas: equações matemáticas, regras nebulosas (lingüísticas) e redes neurais artificiais (YAMAKAWA, 1993).

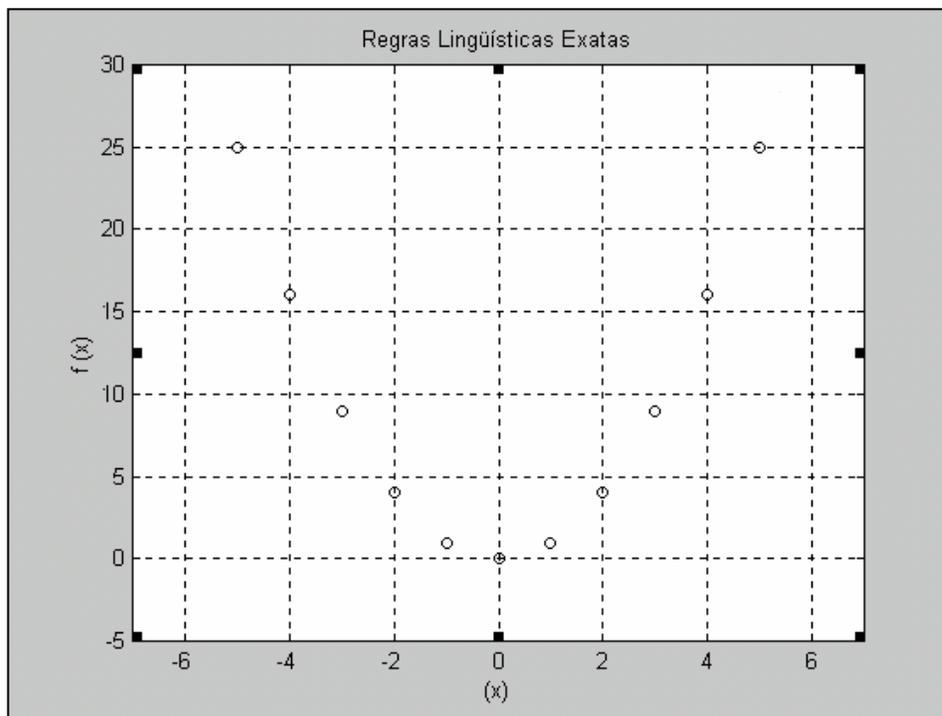


*Figura 1.5 Função Descrita por Equação Matemática*

Equações relacionais ou diferenciais descrevem a dinâmica e/ou a cinética de sistemas (ou o conhecimento sobre o sistema) em uma forma muito conveniente. Se a relação entre a entrada  $x$  e a saída  $f(x)$  do sistema ou a relação entre a causa  $x$  e a resposta  $f(x)$  é obtida como mostrado na **Figura 1.5**, então  $f(x)$  é descrita por  $f(x)=(x)^2$ .

A descrição de um sistema a partir de uma equação matemática é de grande importância, no entanto, é muito difícil identificar a equação que descreve exatamente o comportamento do sistema, especialmente em casos de sistemas complexos e com múltiplas variáveis. Em adição, é também muito difícil reescrever a equação quando a relação entre  $x$  e  $f(x)$  varia aleatoriamente.

Portanto, esta descrição não é apropriada para a maioria dos sistemas complexos tais como: sistemas não-lineares e sistemas variantes no tempo. O Princípio de Incompatibilidade de **Zadeh**, diz que: à medida que a complexidade do sistema aumenta, a possibilidade de descrever um sistema com equações matemáticas diminui.



**Figura 1.6** Função Descrita por Regras Linguísticas Exatas

Uma outra abordagem apresentada na **Figura 1.6**, consiste em descrever a relação entre a entrada  $x$  e a saída  $f(x)$ , através de regras do tipo  $R_i$ , para  $(i=1, \dots, N)$ .

$$\text{Regra } R_i \quad \text{Se } x \text{ é } A_i \text{ então } f(x) \text{ é } B_i$$

Onde  $x$  representa a variável independente e  $f(x)$  representa a variável dependente, sendo  $A_i$  e  $B_i$  constantes lingüísticas e  $N$  o número de dados experimentais que descrevem a função. Quando  $A_i$  e  $B_i$  são constantes lingüísticas com valores numéricos exatos, conforme os sistemas clássicos da inteligência artificial, teríamos o seguinte:

<b>Regra <math>R_{01}</math></b>	<b>Se <math>x</math> é -5 então <math>f(x)</math> é 25</b>
<b>Regra <math>R_{02}</math></b>	<b>Se <math>x</math> é -4 então <math>f(x)</math> é 16</b>
<b>Regra <math>R_{03}</math></b>	<b>Se <math>x</math> é -3 então <math>f(x)</math> é 09</b>
<b>Regra <math>R_{04}</math></b>	<b>Se <math>x</math> é -2 então <math>f(x)</math> é 04</b>
<b>Regra <math>R_{05}</math></b>	<b>Se <math>x</math> é -1 então <math>f(x)</math> é 01</b>
<b>Regra <math>R_{06}</math></b>	<b>Se <math>x</math> é 0 então <math>f(x)</math> é 00</b>
<b>Regra <math>R_{07}</math></b>	<b>Se <math>x</math> é 1 então <math>f(x)</math> é 01</b>
<b>Regra <math>R_{08}</math></b>	<b>Se <math>x</math> é 2 então <math>f(x)</math> é 04</b>
<b>Regra <math>R_{09}</math></b>	<b>Se <math>x</math> é 3 então <math>f(x)</math> é 09</b>
<b>Regra <math>R_{10}</math></b>	<b>Se <math>x</math> é 4 então <math>f(x)</math> é 16</b>
<b>Regra <math>R_{11}</math></b>	<b>Se <math>x</math> é 5 então <math>f(x)</math> é 25</b>

A vantagem deste tipo de abordagem é a facilidade de mudar a descrição do sistema. Por exemplo, quando apenas uma variação local é verificada, basta modificar os valores das regras correspondentes, pois as regras são independentes umas das outras. Assim, a descrição do sistema através de regras é apropriada para sistemas com aprendizagem e sistemas adaptativos.

Por outro lado, existem também algumas desvantagens, quando é dado que  $x$  é igual a **4** a conclusão obtida é  $f(x)$  igual a **16** a partir do casamento entre o dado  $x$  igual a **4** e o antecedente da regra  **$R_{10}$** . Este é, essencialmente, o procedimento básico de inferência em sistemas baseados em regras da inteligência artificial clássica.

Contudo, se é dado que  $x$  é igual a **4.5** nada pode ser inferido a partir do conjunto de regras mencionado pois, não existe nenhuma regra com antecedente  $x$  igual a **4.5**. Isto mostra que os sistemas clássicos binários são pouco eficientes em relação ao conhecimento impreciso, com ruído, ou com variação em dados de entrada, e que é necessário uma enorme quantidade de regras (base de conhecimento) para se obter um resultado ou desempenho significativo.

Conseqüentemente, este tipo clássico de inferência demanda tempo devido a necessidade de se verificar o casamento entre um dado de entrada e uma base de regras muito grande.

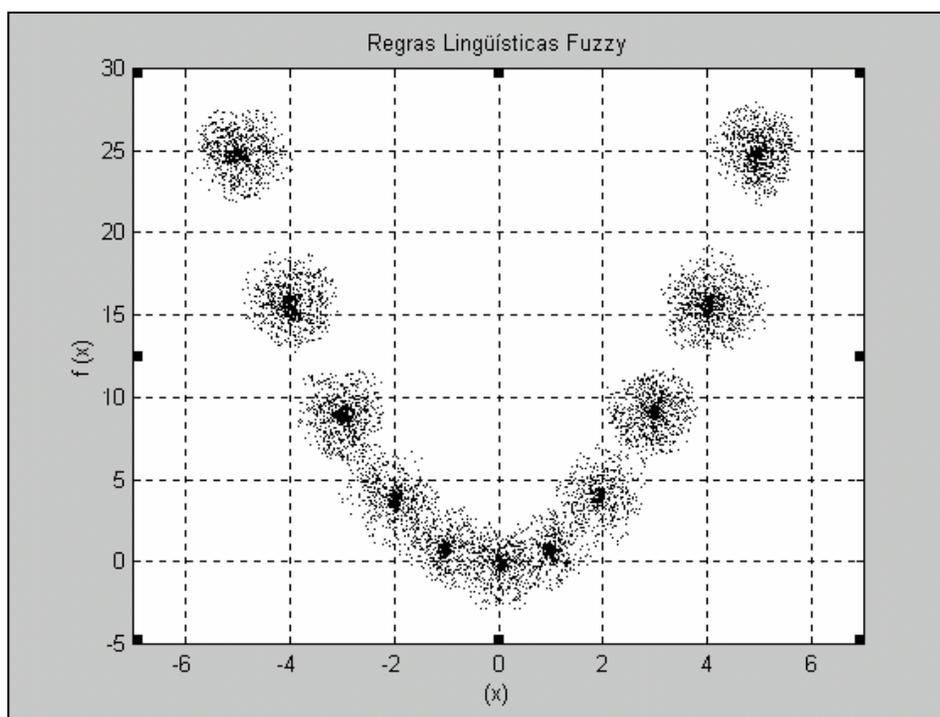
Este mecanismo de inferência não é apropriado para manusear regras contraditórias pois, poderia produzir conclusões também contraditórias a partir de um mesmo dado de entrada. Estas desvantagens também estão presentes nos sistemas clássicos de inteligência artificial, baseados no processamento simbólico e não no processamento do significado dos termos lingüísticos.

Alternativamente, poderíamos utilizar estas mesmas regras, mas interpretando-as como regras nebulosas. Neste caso,  $A_i$  e  $B_i$  seriam termos lingüísticos associados à variável  $x$ , cada um destes termos associados a um conjunto nebuloso a fim de se estabelecer seu significado. Deste modo, poderíamos descrever a relação entre a entrada  $x$  e a saída  $f(x)$  por:

<b>Regra <math>R_{01}</math></b>	<b><i>Se <math>x</math> é entorno de -5 então <math>f(x)</math> é entorno de 25</i></b>
<b>Regra <math>R_{02}</math></b>	<b><i>Se <math>x</math> é entorno de -4 então <math>f(x)</math> é entorno de 16</i></b>
<b>Regra <math>R_{03}</math></b>	<b><i>Se <math>x</math> é entorno de -3 então <math>f(x)</math> é entorno de 09</i></b>
<b>Regra <math>R_{04}</math></b>	<b><i>Se <math>x</math> é entorno de -2 então <math>f(x)</math> é entorno de 04</i></b>
<b>Regra <math>R_{05}</math></b>	<b><i>Se <math>x</math> é entorno de -1 então <math>f(x)</math> é entorno de 01</i></b>
<b>Regra <math>R_{06}</math></b>	<b><i>Se <math>x</math> é entorno de 0 então <math>f(x)</math> é entorno de 00</i></b>
<b>Regra <math>R_{07}</math></b>	<b><i>Se <math>x</math> é entorno de 1 então <math>f(x)</math> é entorno de 01</i></b>
<b>Regra <math>R_{08}</math></b>	<b><i>Se <math>x</math> é entorno de 2 então <math>f(x)</math> é entorno de 04</i></b>
<b>Regra <math>R_{09}</math></b>	<b><i>Se <math>x</math> é entorno de 3 então <math>f(x)</math> é entorno de 09</i></b>
<b>Regra <math>R_{10}</math></b>	<b><i>Se <math>x</math> é entorno de 4 então <math>f(x)</math> é entorno de 16</i></b>
<b>Regra <math>R_{11}</math></b>	<b><i>Se <math>x</math> é entorno de 5 então <math>f(x)</math> é entorno de 25</i></b>

Neste caso, a relação precisa entre a entrada  $x$  e a saída  $f(x)$ , apresentada na **Figura 1.6** é **fuzzificada**, passando a ser tratada como uma relação contínua. A **Figura 1.7** mostra esta relação nebulosa, que fornece valores razoáveis para qualquer dado no universo de discurso, através de mecanismos de inferência nebulosa e **defuzzificação**. Os mecanismos de inferência nebulosa e o procedimento de **defuzzificação**, proporcionam uma forma razoável e simples de interpolação, com muito menos dados e dados não exatos.

A inferência nebulosa apresenta um comportamento similar ao descrito por funções matemáticas. Ainda mais, é muito mais fácil elaborar novas regras nebulosas do que equações matemáticas, quando as características do sistema ou processo são variantes no tempo. Em geral, somente umas poucas regras são adicionadas ou revisadas independentemente das outras regras, no caso de equações matemáticas, vários parâmetros devem ser avaliados simultaneamente.



**Figura 1.7 Função Descrita por Regras Lingüísticas Nebulosas**

As principais características dos modelos nebulosos de sistemas e processos são:

- a) apropriada para descrever sistemas complexos com uma razoável quantidade de conhecimento;
- b) facilita a seleção das variáveis lingüísticas e os correspondentes valores a serem usados nas regras nebulosas;
- c) possibilita a memorização o conhecimento utilizando uma categoria relativamente pequena de palavras;
- d) permite a comunicação dos modelos com projetistas e analistas devido ao uso da linguagem natural.

Deve-se notar que, enquanto a inferência baseada na regra de inferência *modus ponens* nos sistemas clássicos de inteligência artificial é baseada somente em processamento simbólico, a inferência nebulosa, usando raciocínio aproximado, é baseada tanto no processamento simbólico como no processamento do significado.

As regras lingüísticas nebulosas podem representar um conhecimento explicitamente e este conhecimento deve ser utilizado, explicitamente, para inferência. As redes neurais artificiais, ao contrário, representam tanto o conhecimento como a inferência, implicitamente.

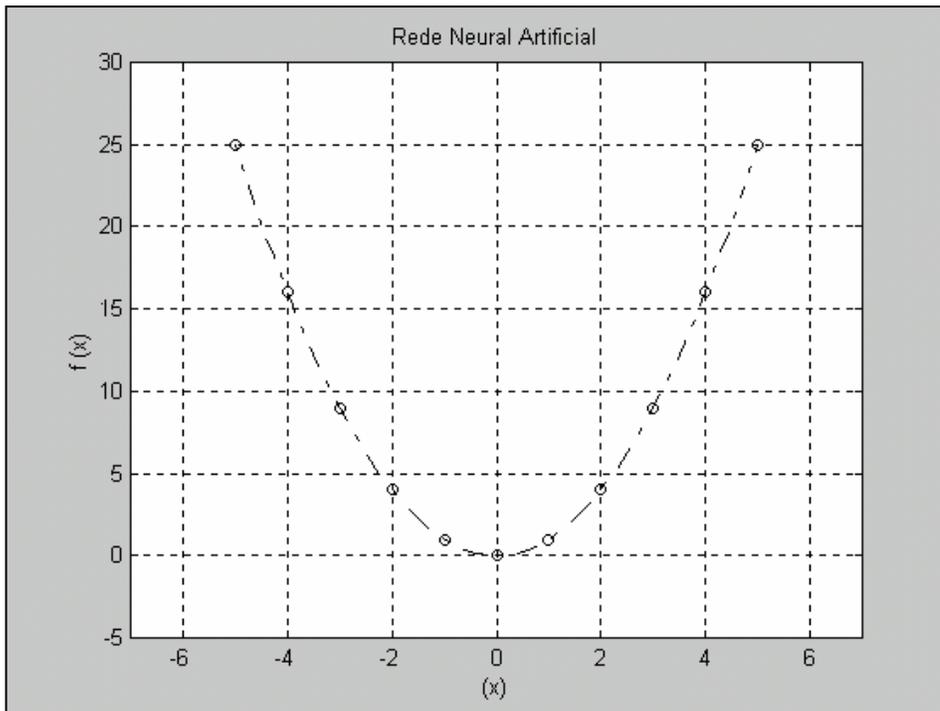
Portanto, as regras lingüísticas nebulosas representam o conhecimento estruturado enquanto as redes neurais artificiais representam, em geral, o conhecimento não estruturado. Existem, no entanto, classes de redes neurais artificiais nebulosas que permitem a detecção do conhecimento e sua estrutura (FIGUEIREDO et al., 1993).

Finalmente, uma alternativa para modelar e controlar um sistema complexo, consiste em descrevê-lo através de um conjunto de parâmetros, cujo exemplo típico são as redes neurais artificiais. Uma rede neural artificial é composta por um número finito de elementos simples interconectados, cada elemento realizando uma operação de agregação, conforme o modelo físico de um neurônio. Assim, quando  $W_{ij}$  é o peso atribuído ao sinal de uma entrada  $e_i$  para o  $j$ -ésimo neurônio,  $\theta_j$  e  $s_j$  são respectivamente, o limiar e o sinal de saída do  $j$ -ésimo neurônio e o sinal de saída  $s_j$  é tipicamente dado pela seguinte expressão:

$$s_j = h \left( \sum_i^n w_{ij} e_i - \theta_j \right) \quad (1.15)$$

Onde:  $h$  é em geral uma função de agregação sigmoideal.

$$h(x) = \frac{1}{1 + \exp(-x)} \quad (1.16)$$



**Figura 1.8 Função Descrita por Rede Neural Artificial**

Usando este modelo de neurônio, a relação existente entre  $x$  e  $f(x)$  pode ser caracterizada por uma rede neural artificial. Tanto os pesos  $W_{ij}$  como os limiares  $\theta_j$  podem ser obtidos através inúmeras iterações de aprendizado utilizando algoritmos computacionais clássicos de propagação retroativa, como o método dos mínimos quadrados entre outros.

Uma rede neural artificial é capaz de realizar interpolações de forma similar ao caso da inferência nebulosa. Contudo, em geral, é muito difícil de entender, apenas pela observação do conjunto de parâmetros, como a rede neural se comporta. Além disso, existem problemas para determinação da topologia mais apropriada para o problema em questão, bem como problemas de convergência durante o processo de aprendizagem.

Se o sistema está sujeito a variações de comportamento, uma nova aprendizagem deve ser executada para se obter um novo conjunto de pesos e limiares correspondentes e, muitas vezes, os resultados não são razoáveis. Isto é, existe uma maior dificuldade em projetar sistemas para modelo e controle com redes neurais artificiais do que com sistemas nebulosos.

Esta é uma das razões pela qual os sistemas nebulosos têm sido mais populares na solução de problemas práticos do que as redes neurais artificiais. No entanto, pesquisa recente vem gerando classes de sistemas neuro-nebulosos, onde as vantagens e desvantagens de ambas as abordagens são tratadas de forma a criar sistemas mais eficientes tanto do ponto de vista da solução de problemas, quanto da análise e projeto dos mesmos (GOMIDE & ROCHA, 1992).

## 1.4 *Sistemas de Controle Nebulosos*

A idéia básica em controle nebuloso, é modelar as ações a partir de conhecimento especialista, ao invés de, necessariamente, modelar o processo em si. Este procedimento nos leva a uma abordagem diferente dos métodos convencionais de controle de processos, desenvolvidos através de modelos matemáticos dos processos, de modo a derivar as ações de controle como uma função dependente do estado do processo. A motivação para esta nova abordagem veio de casos onde o conhecimento especialista de controle era disponível, seja por meio de operadores ou projetistas, e os modelos matemáticos envolvidos muito complexos.

A *Figura 1.9* apresenta a estrutura de um processo com controlador nebuloso, indicando seus componentes básicos para operação em malha fechada: a interface de *fuzzificação*, a base de conhecimento, a base de dados, o procedimento de inferência e a interface de *defuzzificação*.

---

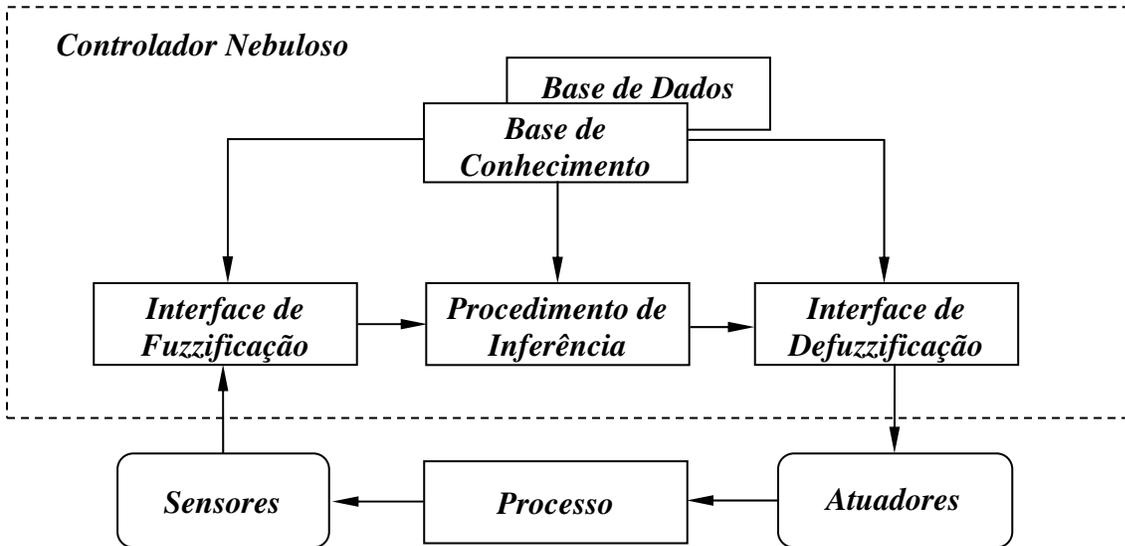


Figura 1.9 Estrutura Básica de um Controlador Nebuloso

### 1.4.1 Interface de Fuzzificação

A interface de *fuzzificação* toma os valores analógicos das variáveis de entrada, gerados por circuitos sensores, e faz um escalonamento para condicionar os valores numéricos de entrada a universos de discurso normalizados. Portanto, o procedimento de *fuzzificação* consiste em transformar os valores numéricos de entrada em conjuntos nebulosos, de modo que possam se tornar instâncias de variáveis linguísticas.

### 1.4.2 Base de Conhecimento

A base de conhecimento consiste de um conjunto de regras, caracterizando a estratégia de controle e suas metas, modelando as ações de controle a partir de conhecimento especialista, obtido através da experiência prática dos engenheiros e operadores que observam o processo, adquirindo o conhecimento necessário sobre seu comportamento estático e dinâmico.

### 1.4.3 Base de Dados

A base de dados armazena todas as informações necessárias para as discretizações dos universos de discurso, as partições nebulosas dos espaços de entrada e saída, as definições das funções de pertinência e as operações lógicas e aritméticas executadas pelo controle.

#### **1.4.4 Procedimento de Inferência**

O procedimento de inferência processa os dados nebulosos de entrada, junto com o conjunto de regras nebulosas, de modo a inferir as ações de controle, aplicando o operador de implicação e as regras de inferência da lógica nebulosa, de acordo com a base de conhecimento disponível. Apenas uma base de conhecimento consistente, permite o procedimento de inferência adequado às necessidades do processo, garantindo a eficiência do controlador.

#### **1.4.5 Interface de Defuzzificação**

A interface de *defuzzificação* transforma as ações de controle nebulosas inferidas em ações de controle não-nebulosas, utilizando um operador de agregação adequado. Em seguida, efetua um escalonamento, de modo a compatibilizar os valores normalizados com os valores reais dos universos de discurso das variáveis de saída do controlador nebuloso. As variáveis de saída interferem na ação dos circuitos atuadores, responsáveis pelo controle do processo.

#### **1.4.6 Operador de Agregação**

Após a inferência da ação de controle nebulosa, é necessária a determinação de uma ação de controle não-nebulosa, que melhor represente a decisão nebulosa, para ser efetivamente enviada ao controle do processo através dos circuitos atuadores. Apesar de não haver nenhum procedimento sistemático para escolha da estratégia de *defuzzificação*, as mais comuns incluem: o critério do máximo, que escolhe o ponto onde a função inferida tem seu máximo; a média dos máximos, que representa o valor médio dentre todos os pontos de máximo e o método do centro de área, que retorna o centro de área da função inferida.

### **1.5 Projeto de Controladores Nebulosos**

Durante o projeto de controladores nebulosos, é necessária a definição de parâmetros a partir da experiência do projetista, ou através de observações e experimentos práticos. Dado um processo, alguns dos parâmetros são fixos, dentro das condições normais de operação, sendo que outros precisam ser alterados visando o ajuste do controlador nebuloso.

Os parâmetros fixos são chamados de parâmetros estruturais enquanto que os variáveis são chamados de parâmetros de sintonização.

***Parâmetros Estruturais:***

- a) Variáveis de entrada e saída
- b) Operações lógicas e aritméticas
- c) Variáveis lingüísticas
- d) Funções de pertinência
- e) Intervalos de discretização
- f) Conjunto de regras

***Parâmetros de Sintonização:***

- a) Universo de discurso das variáveis
- b) Parâmetros das funções de pertinência
- c) Ganho e off-set das entradas e saídas

A sintonização é uma das etapas mais custosas do projeto de um controlador nebuloso. A grande flexibilidade que decorre da existência de muitos parâmetros exige um grande esforço por parte do projetista para obter o melhor desempenho do controlador. Esta sintonização é feita por meio de busca, o que caracteriza uma atividade típica de inteligência artificial.

De acordo com o conhecimento que se tem sobre o processo a ser controlado, a técnica de projeto do controlador nebuloso deverá ser mais ou menos elaborada. Quando se tem pouco conhecimento sobre o processo, deve-se adotar uma metodologia de projeto mais conservadora, de modo a evitar um mau comportamento do controlador nebuloso.

É necessário que alguns cuidados sejam observados pelo projetista. Em primeiro lugar, deve-se verificar se é realmente necessário o uso de um controlador nebuloso. Caso o processo possa ser controlado por um controlador convencional, sem grandes transtornos, este poderá constituir uma boa solução. Se os controladores convencionais não são apropriados devido a complexidade do processo, deve-se considerar o uso dos controladores nebulosos. Mesmo com pouco conhecimento sobre o processo, pode-se considerar o uso de um controlador nebuloso.

## Apêndice 2

### Redes Neurais Artificiais

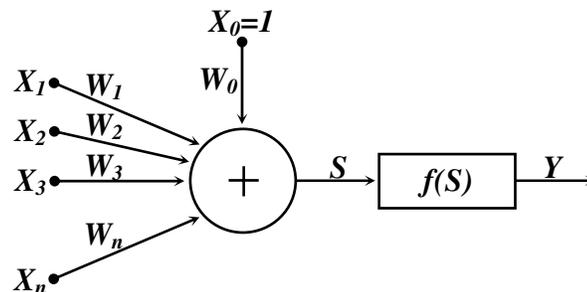
#### 2.1 Introdução

Uma *RNA* (Rede Neural Artificial) consiste basicamente de uma arquitetura específica para processamento de informações estruturadas segundo as redes neurais biológicas que possui habilidades de identificação de funções e classificação de padrões. Conforme as redes neurais biológicas as redes neurais artificiais assimilam informações pela técnica de tentativa e erro.

Todos os tipos de *RNA* possuem capacidade de generalização, ou seja, podem apresentar soluções satisfatórias nas situações para as quais não houve aprendizado. Outra característica importante das *RNA* é a de não perder a informação já armazenada caso ocorra uma destruição total de uma parte da rede. Isto pode ser verificado no funcionamento normal do sistema nervoso humano se alguns milhões de neurônios sejam anulados (HOPFIELD, 1982).

#### 2.2 Elemento Linear Adaptativo

O elemento básico de uma *RNA* é o neurônio artificial inspirado no modelo do neurônio biológico idealizado por (MCCULLOCH & PITTS, 1943). A *Figura 2.1* mostra o neurônio artificial chamado *ADALINE* (Adaptive Linear Element) inspirado no neurônio biológico.

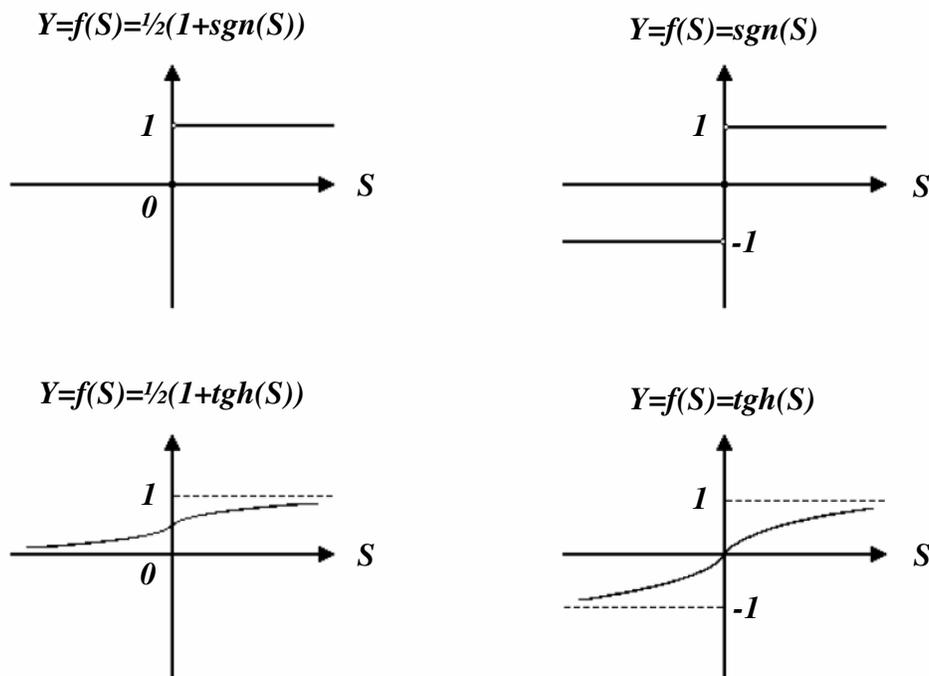


*Figura 2.1 Elemento Linear Adaptativo Padrão*

Os elementos do vetor de entrada  $X=[X_1X_2...X_N]^T$  podem pertencer a conjuntos com valores contínuos ou discretos. O peso  $W_0$  determina a saída do neurônio quando a entrada  $X_0=1$ .

A função  $f(S)$  de saída do neurônio é do tipo não decrescente com domínio real. A saída  $Y$  pertence ao intervalo  $[0,1]$  quando se deseja que o neurônio responda com sinais de reforço numa maior ou menor intensidade.

A saída  $Y$  pertence ao intervalo  $[-1,1]$  quando o neurônio tiver que assumir caráter de reforço ou de inibição numa forma mais ou menos intensa. A **Figura 2.2** mostra alguns exemplos de funções de saída de **ADALINE**.



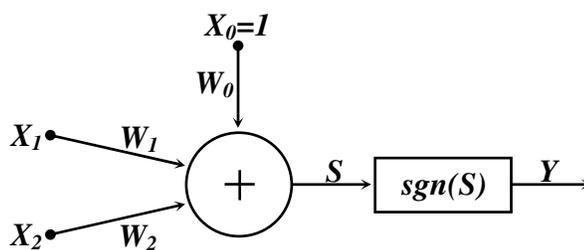
**Figura 2.2** Funções Típicas de Ativação de Saída

A saída do neurônio é computada através da equação (2.1) apresentada a seguir:

$$S = W_0 + \sum_{i=1}^N (W_i X_i) \quad (2.1)$$

A equação (2.1) representa um hiperplano no  $R^N$ , onde  $W=[W_1 W_2 \dots W_N]^T$  define um vetor ortogonal. Portanto, o peso  $W_0$  corresponde ao termo independente da equação geral do hiperplano separador e proporciona seu deslocamento em relação à origem, sendo importante para sistemas dedicados a classificação de padrões.

A **Figura 2.3** apresenta um **ADALINE** com duas entradas e três pesos classificando os pontos de treinamento (padrões) de acordo com as saídas desejadas.



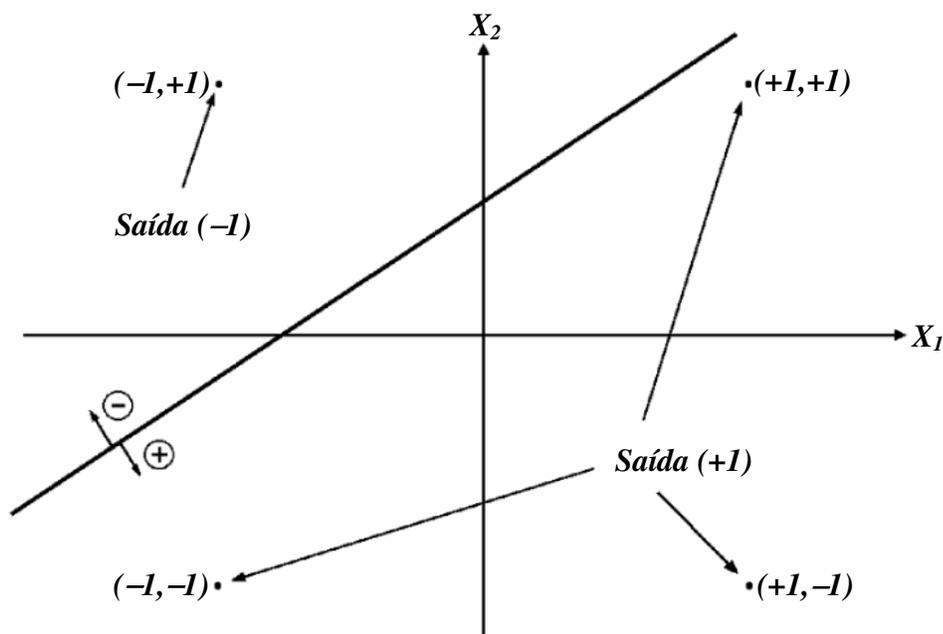
**Figura 2.3 Exemplo de Elemento Linear Adaptativo**

A **Tabela 2.1** define a função a ser reproduzida pelo **ADALINE** da **Figura 2.3**.

**Tabela 2.1 Função Reproduzida**

Ponto de Entrada ( $X_1, X_2$ )	Saída Desejada ( $Y$ )
(+1,+1)	(+1)
(+1,-1)	(+1)
(-1,-1)	(+1)
(-1,+1)	(-1)

A **Figura 2.4** apresenta um hiperplano separador para os valores desejados da saída  $Y$ .



**Figura 2.4 Hiperplano Separador**

O hiperplano separador da **Figura 2.4** pode ser representado pela seguinte equação:

$$W_2 X_2 + W_1 X_1 + W_0 = 0 \quad (2.2)$$

Note que as saídas descritas na **Tabela 2.1** definem uma função linearmente separável. Segundo **Widrow** e **Lehr** qualquer função linearmente separável, com  $N$  variáveis, pode ser bem reproduzida por um **ADALINE** com  $N$  entradas (WIDROW & LEHR, 1990).

Em geral, um único **ADALINE** não resolve os problemas mais complexos. Uma solução viável consiste em organizar vários **ADALINES** de modo a formar uma camada ou ainda, várias camadas, construindo estruturas chamadas **MADALINE**. Portanto, outras arquiteturas de **RNA** devem ser utilizadas para resolver problemas mais complexos.

Uma **RNA** tipo **FeedForward** é aquela onde a informação segue um único sentido desde a entrada até a saída. Uma **RNA** tipo **Hopfield** é uma rede recorrente construída por conexões entre **ADALINES** em uma mesma camada. Outra forma alternativa é conectar a saída de uma **RNA** do tipo **FeedForward** às entradas de um ou mais neurônios desta mesma **RNA**.

### 2.3 Cerebellar Model Articulation Controller

**Albus** desenvolveu uma arquitetura de rede neural artificial baseada em estudos neuro-fisiológicos de cerebelos de mamíferos com um sistema motor aprimorado, tais como o homem. O **CMAC** (Cerebellar Model Articulation Controller) é capaz de reproduzir funções não-lineares, de múltiplas entradas e múltiplas saídas, definidas num conjunto compacto, isto é, num conjunto fechado e limitado (ALBUS, 1975).

Os comandos vindos de centros de informação de níveis mais altos do cérebro e as informações sobre membros e juntas do sistema motor entram no cerebelo através das fibras musgosas. As células granulares transformam os vetores de entrada em subconjuntos de fibras paralelas ativas. Os pesos das sinapses ajustáveis correspondentes às fibras paralelas ativas produzirão a saída do cerebelo.

As células de **Purkinje** somarão os estímulos positivos diretamente enquanto que os estímulos negativos serão somados através das células em cesto e das células estreladas. Durante o processo de treinamento as sinapses ajustáveis são controladas pelas fibras trepadeiras trazendo informações de aprendizado. A **Figura 2.5** representa o modelo neuro-fisiológico do cerebelo de um mamífero com sistema motor aprimorado.

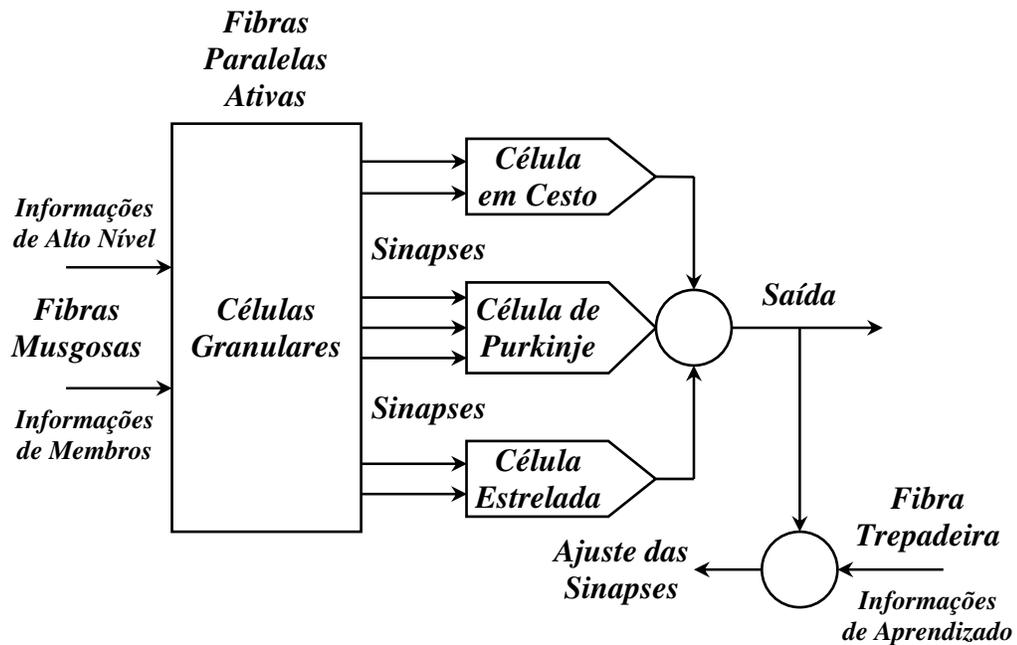


Figura 2.5 Modelo Simplificado do Cerebelo

## 2.4 Transformações Presentes no CMAC

Cada vetor de entrada do *CMAC* possui o seguinte formato  $S=[S_1S_2...S_N]^T$ . A saída é um escalar, porém,  $P$  estruturas operando sobre o mesmo conjunto de entrada devem gerar o vetor de valores estimados  $Y^e$ . Portanto, o *CMAC* fornece uma saída estimada  $Y^e=h(S)$  sendo a função  $h(S)$  decomposta em três transformações distintas descritas a seguir:

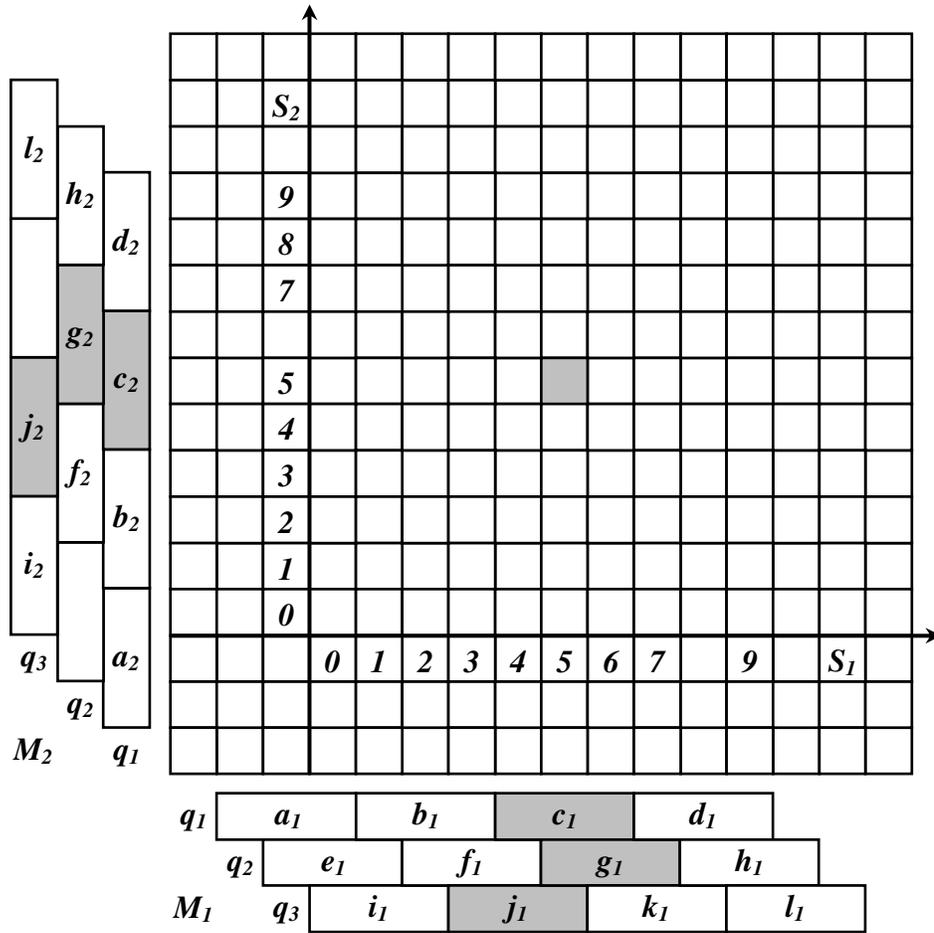
$$S \rightarrow M \rightarrow A \rightarrow Y^e$$

Sendo:  $S$  o vetor de entrada da rede,  $M$  a variável vetorial intermediária,  $A$  o vetor de células de associação e  $Y^e$  o vetor de saída estimada.

### 2.4.1 Transformação $S \rightarrow M$

Cada vetor de entrada  $S=[S_1S_2...S_N]^T$  é composto por  $N$  variáveis que podem assumir valores contínuos ou discretos. A faixa de variação de cada  $S_i$ , com  $i \in \{1, 2, \dots, N\}$ , é quantizada através de  $C$  funções de quantização  $\{q_1, q_2, \dots, q_C\}$  em  $D$  intervalos, sendo que há sobreposição de uma parte do intervalo de quantização entre duas funções adjacentes.

A **Figura 2.6** mostra um exemplo para  $N=2$  e  $D=4$  onde  $S_i$  é representado com resolução unitária no intervalo de 0 a 9, sendo cada variável discretizada por  $C=3$  funções de quantização.



**Figura 2.6** Funções de Quantização

A cada variável  $S_i$  está associado um vetor  $M_i$  com  $C$  elementos discretos que são os campos de recepção dados por  $\{q_1, q_2, \dots, q_C\}$ . Para todo valor inteiro de  $S_i$  existe um único vetor  $M_i$ , tal que,  $M_i = [q_1(S_i) q_2(S_i) \dots q_C(S_i)]^T$ . Para  $S = [S_1 S_2]^T$ , com  $S_1 = S_2 = 5$ , teremos:

$$\begin{aligned}
 q_1(S_1) &= q_1(5) = c_1 \\
 q_2(S_1) &= q_2(5) = g_1 \\
 q_3(S_1) &= q_3(5) = j_1 \\
 \end{aligned}
 \qquad
 M_1 = [c_1 \ g_1 \ j_1]^T$$
  

$$\begin{aligned}
 q_1(S_2) &= q_1(5) = c_2 \\
 q_2(S_2) &= q_2(5) = g_2 \\
 q_3(S_2) &= q_3(5) = j_2 \\
 \end{aligned}
 \qquad
 M_2 = [c_2 \ g_2 \ j_2]^T$$

Com a transformação  $S \rightarrow M$  a informação de um único vetor  $S$  é transmitida para  $N$  vezes  $C$  canais de dados discretos através das funções  $\{q_1, q_2, \dots, q_C\}$  aplicadas em  $S$ . Portanto, pequenas variações em  $S$  podem produzir alterações em  $M$  mas não em todos os seus elementos.

Com isso, a dois vetores de entrada próximos em  $S$ , corresponderão canais coincidentes, mas não a todos. Isto confere ao **CMAC** a capacidade de generalização que é importante para armazenamento e recuperação de informações, quando duas situações apresentam semelhanças que podem ser não coincidentes.

### 2.4.2 Transformação $M \rightarrow A$

No modelo proposto na **Figura 2.5**, a seleção de fibras paralelas ativas, feita pelas células granulares, é simulada no **CMAC** pela concatenação dos vetores discretos  $M_i$  a fim de se obter o vetor de células de associação. No exemplo ilustrado pela **Figura 2.6**, cada um dos  $C=3$  elementos dos vetores discretos  $M_1$  e  $M_2$  podem assumir  $D=4$  valores diferentes resultantes das funções de quantização  $\{q_1, q_2, q_3\}$ . Deste modo, a formação de um par ordenado devido à aplicação de  $q_1$  em  $S_1$  e  $S_2$ , onde o primeiro elemento pertence a  $\{a_1, b_1, c_1, d_1\}$  e o segundo elemento pertence a  $\{a_2, b_2, c_2, d_2\}$ , oferece na prática **16** diferentes possibilidades de arranjo.

A concatenação das primeiras posições de  $M_1$  e  $M_2$  pode gerar **16** resultados diferentes, isto é, deve haver **16** células de associação e, por conseguinte, **16** pesos associados, sendo que somente uma é selecionada. Analogamente, teremos **16** pesos associados à aplicação de  $q_2$  e  $q_3$  em  $S_1$  e  $S_2$ , o que resulta em um número total de pesos dado por  $Z=48$ .

Portanto, a concatenação de  $N$  vetores discretos  $M_i$ , produz um vetor  $A=[A_1 A_2 \dots A_Z]^T$  de células de associação com  $C$  elementos selecionados aos quais se atribui valor **1** enquanto os demais elementos são zerados. Os elementos selecionados em  $A$  formam o conjunto chamado  $A_i$ .

Dois vetores de entrada  $S_1$  e  $S_2$  muito próximos, a ponto de pertencerem aos mesmos intervalos de quantização, terão iguais conjuntos  $A_i$ , isto é,  $A_1=A_2$  e  $h(S_1)=h(S_2)$ .

Quando  $S_1$  e  $S_2$  forem próximos sem pertencerem aos mesmos intervalos de quantização, então, a intersecção de  $A_1$  e  $A_2$  terá menos de  $C$  elementos e  $h(S_1)$  será diferente de  $h(S_2)$  devido às células de associação ativas que não são comuns. Quando  $S_1$  e  $S_2$  forem distantes, os conjuntos  $A_1$  e  $A_2$  serão disjuntos e  $h(S_1)$  será independente de  $h(S_2)$ .

### 2.4.3 Transformação $A \rightarrow Y^e$

As sinapses entre as células granulares e a célula de *Purkinje* contribuem de forma a estimular a ação característica desta, enquanto que as sinapses entre as células granulares e as células em cesto e estreladas inibem a ação característica da célula de *Purkinje*.

O algoritmo do *CMAC* gera a saída, somando os elementos do vetor  $W=[W_1 W_2 \dots W_Z]^T$  escolhidos pelas células não nulas do vetor  $A$ . Portanto, o  $j$ -ésimo elemento do vetor de saída  $Y^e$ , para  $j \in \{1, 2, \dots, P\}$  é calculado através da equação (2.3).

$$Y_j^e = \sum_{k=1}^Z (A_k W_{jk}) \quad (2.3)$$

A transformação  $S \rightarrow A$  é usada no cálculo de  $P$  elementos  $Y_j^e$  do vetor  $Y^e$ , cada um deles resultante de uma transformação  $A \rightarrow Y^e$ . Logo, existe um vetor  $W_j$  para cada saída do *CMAC*.

## 2.5 Treinamento do *CMAC*

Numa tentativa de imitar o mecanismo de adaptação das sinapses no cerebelo, utiliza-se o algoritmo dos mínimos quadrados para atualizar os pesos correspondentes a uma saída errônea.

Suponha que o *CMAC* deva reproduzir a função  $Y_j = f(S_i)$  com  $S_i \in S$ . Deste modo, o *CMAC* deve fornecer uma função  $Y_j^e = h(S_i)$  tão próximo de  $Y_j = f(S_i)$  quanto possível.

Sendo  $\rho$  a precisão requerida para cada par  $Y_j$  e  $Y_j^e$ , quando  $(Y_j - Y_j^e) \leq \rho$ , não é necessário modificar nenhum peso, pois a informação já está armazenada. Porém, quando  $(Y_j - Y_j^e) > \rho$ , é necessário somar a cada peso que contribuiu para  $Y_j^e$  a quantidade:

$$\frac{1}{C}(Y_j - Y_j^e) \quad (2.4)$$

Vetorialmente, esta lei de treinamento recursivo pode ser descrita pela equação a seguir:

$$W_{j(t+1)} = W_{j(t)} + \frac{1}{C}(Y_j - Y_j^e)_{(t)} \quad (2.5)$$

Devido à sua capacidade de generalização, torna-se desnecessário treinar o *CMAC* em pontos muito próximos em  $S$ , o que permite usar um conjunto reduzido de pontos de treinamento para assimilar o comportamento da função  $Y = f(S)$ .

Obviamente, o processo de adaptação deve ser repetido várias vezes para ajustar os pesos eventualmente compartilhados pelos pontos de treinamento. No *CMAC*, a habilidade de obter  $Y_1^e \neq Y_2^e$  para  $S_1 \neq S_2$  depende de quantos pesos são compartilhados pelas duas saídas e de quanto os dois conjuntos de pesos diferem entre si.

Até então, não havia sido publicada uma comprovação formal da convergência deste processo de treinamento. Segundo (PARKS & MILITZER, 1992) o algoritmo pode convergir para um único ponto, um ciclo limite de pontos ou para uma mínima zona de captura. A demonstração é baseada na interpretação geométrica do algoritmo proposto por *Albus*.

## 2.6 Requisitos de Memória

O *CMAC* apresenta a desvantagem de requerer uma grande quantidade de memória quando se trabalha com muitas entradas e saídas. O tamanho da memória requerida pelo *CMAC* é dado pelo número de saídas  $P$  e pelo número de células de associação  $Z$ . Do mesmo modo,  $Z$  depende do número de funções de quantização  $C$  e do número de intervalos de quantização  $D$ , relativos a cada uma das  $N$  variáveis de entrada.

A memória física associada ao exemplo apresentado na *Figura 2.6* deve armazenar um vetor de pesos  $W$  com  $P$  vezes  $Z$  igual a 48 elementos. Aumentando uma entrada e uma saída, mantendo  $C=3$  e  $D=4$  o número de elementos do vetor de pesos  $W$  deve ser  $P$  vezes  $Z$  igual 384.

Uma forma de se reduzir a memória física do *CMAC* emprega o código *Hash*. Este procedimento baseia-se em computar os endereços dos elementos de um pequeno vetor de pesos através de um gerador de números pseudoaleatórios que usa o endereço de uma célula de associação como argumento (WONG & SIDERIS, 1992).

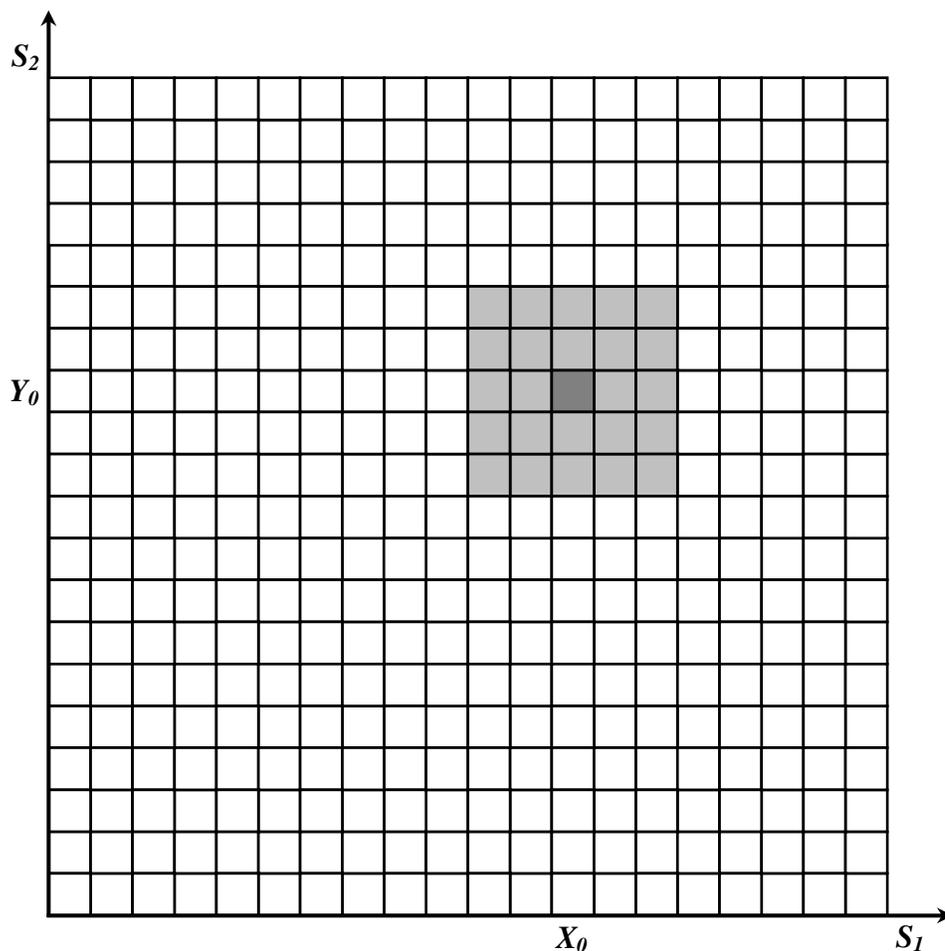
Embora o código *Hash* resolva o problema do tamanho da memória física necessária, podem ocorrer colisões e o mesmo peso pode ser endereçado para duas células de associação distintas. Este peso será somado duas vezes, diminuindo a precisão da representação numérica.

Por outro lado, para  $S_1$  e  $S_2$  afastados, ocorre uma generalização indesejada o que interfere na convergência do processo de treinamento, ou seja, o sistema pode ficar instável conforme a entrada aplicada. Esses dois efeitos indesejáveis podem ser minimizados aumentando a quantidade de intervalos de quantização e a quantidade de pesos, pois a sobreposição será menor devido ao maior número de possibilidades de endereçamento.

## 2.7 CMAC Alternativo

Uma estrutura modificada para o **CMAC** aplicada na solução de problemas em que não é necessária uma grande quantidade memória foi proposta por (KRAFT & CAMPAGNA, 1990).

Considerando a função  $Y=f(S)$ , divide-se o vetor de entrada  $S$  em  $Z$  intervalos conforme a **Figura 2.7**. A cada célula associa-se um elemento numérico que corresponde ao peso e o vetor de entrada  $S$  pertence a um único intervalo  $A_k$ , para  $k \in \{1, 2, \dots, Z\}$ .



**Figura 2.7** Conjunto  $S$  e Região Ativa para  $S=(X_0, Y_0)$

Para um dado vetor de entrada  $S$ , a estrutura modificada fornece uma saída estimada  $Y^e=h(S)$  obtida pela média aritmética dos pesos contidos numa região centrada na célula  $A_k$ , que será chamada de região ativa. A região ativa é determinada pelo vetor de entrada  $S$  e por um número natural  $B$  que representa a quantidade de intervalos tomados no sentido crescente e decrescente do eixo de cada variável de entrada e centrados na célula  $A_k$ .

O conceito de região ativa no *CMAC* proposto por *Kraft* e *Campagna* é muito similar ao conceito de conjunto de células de associação selecionadas no *CMAC* proposto por *Albus*. Por analogia, representaremos a quantidade de células da região ativa através da equação (2.6), onde o número  $N$  representa a quantidade de elementos do vetor de entrada.

Assim, na *Figura 2.7*,  $B=2$ ,  $N=2$  e a região ativa possui 25 células.

$$C = (2B + 1)^N \quad (2.6)$$

Dessa forma, se tomarmos valores de entrada  $S_1$  e  $S_2$  próximos, as respectivas regiões ativas podem ter até  $C$  células em comum e  $Y_1^e$  é idêntico a  $Y_2^e$ . À medida que  $S_1$  se afasta de  $S_2$  as regiões ativas têm cada vez menos células em comum e  $Y_1^e$  é diferente de  $Y_2^e$ .

Do mesmo modo do *CMAC* original, a taxa de generalização depende do parâmetro  $C$ . Genericamente, a saída dessa nova configuração é dada pela equação (2.7), onde  $W_{Ra}$  representa um peso associado a qualquer célula pertencente à região ativa.

$$Y_{Ra}^e = \frac{1}{C} \sum_{Ra} W_{Ra} \quad (2.7)$$

O treinamento consiste em modificar os pesos de forma que o valor  $Y^e$  se aproxime de  $Y$ , quando  $S$  é representado por um conjunto finito de pontos. A regra de treinamento é baseada no algoritmo dos mínimos quadrados e pode ser representada pela equação (2.8).

$$W_{Ra_{(t+1)}} = W_{Ra_{(t)}} + \beta (Y_{Ra} - Y_{Ra}^e)_{(t)} \quad (2.8)$$

Onde o termo  $W_{Ra}$  é um peso da região ativa e o coeficiente  $\beta$  é a taxa de aprendizado, geralmente  $0 < \beta \leq 1$ . Este procedimento deve ser repetido para todos os pontos de treinamento.

Esta estrutura pode ser implementada facilmente quando não se requer muitos intervalos na partição feita em  $S$ . Sua maior desvantagem é a não utilização de uma técnica para redução do número de pesos. Observando a complexidade do problema a ser abordado e a quantidade de intervalos necessários, esta estrutura pode ser generalizada para mais de duas variáveis de entrada. Se desejarmos reproduzir  $P$  saídas, podemos utilizar  $P$  estruturas em paralelo.

## 2.8 CMAC Pesos Discretos

De forma geral, o controle digital envolve a conversão analógico-digital de dados, com isso, as variáveis amostradas e as ações de controle ficam restritas a conjuntos finitos de valores discretos, obtidos a partir de um conversor analógico digital.

**Kraft e Campagna** abordaram a representação numérica dos pesos em ponto fixo com a finalidade de redução da memória de dados e aumento da velocidade de processamento, em detrimento da precisão. Porém, cada peso era armazenado utilizando sempre dois bytes e a memória física era reduzida devido ao uso do código **Hash**, principalmente.

Para restringir os pesos e a saída do **CMAC** a conjuntos arbitrários de valores discretos, uma nova proposta foi apresentada por (BORDON & TEIXEIRA, 1995). Na representação numérica dos pesos e dos valores de saída pode ser usado um número de bits previamente especificado dentro da faixa de **1** a **16** bits. Neste caso, não é necessário discretizar as variáveis de entrada pois a escolha dos campos de recepção é feita por funções de saída discretas.

Supondo que um peso  $W \in [W_{-m}, W_m]$  seja aproximado para um dos seus  $(2m+1)$  valores discretos então, o menor incremento  $\Delta w$  será calculado através da equação (2.9):

$$\Delta w = \frac{W_m - W_{-m}}{2m} \quad (2.9)$$

Portanto, para cada peso  $W \in [W_{-m}, W_m]$  será associado um número inteiro indicado por  $v \in \{-m, -m+1, \dots, 0, \dots, m-1, m\}$ , ao qual corresponde o valor do peso discreto  $W_v$ , mais próximo do valor amostrado, conforme a equação (2.10) apresentada a seguir:

$$W_v = W_o + v \Delta w \quad (2.10)$$

Escolhendo  $b$  bits para representar os  $(2m+1)$  valores inteiros de  $v$ , deve-se obedecer a relação apresentada pela equação (2.11) para que  $m$  seja corretamente especificado.

$$2m = 2^b - 1 \quad (2.11)$$

De maneira análoga, supondo que a saída desejada para o **CMAC**, representada por  $Y \in [Y_{-n}, Y_n]$ , seja aproximada para um dos seus  $(2n+1)$  valores discretos, o menor incremento  $\Delta y$  correspondente será calculado através da seguinte equação:

$$\Delta y = \frac{Y_n - Y_{-n}}{2n} \quad (2.12)$$

Portanto, para cada valor  $Y \in [Y_{-n}, Y_n]$  será associado um número inteiro indicado por  $u \in \{-n, -n+1, \dots, 0, \dots, n-1, n\}$ , ao qual corresponde o valor da saída discreta  $Y_u$  mais próximo do valor amostrado, conforme a equação apresentada a seguir:

$$Y_u = Y_o + u \Delta y \quad (2.13)$$

Escolhendo  $b$  bits para representar os  $(2n+1)$  valores inteiros de  $u$ , deve-se obedecer a relação apresentada pela equação (2.14) para que  $n$  seja corretamente especificado.

$$2n = 2^b - 1 \quad (2.14)$$

O próximo passo é descrever a saída do *CMAC* com pesos discretos, através de um algoritmo de treinamento, a partir dos números inteiros  $v$  e  $u$  que definem os pesos e as saídas, respectivamente, executando as operações com números de ponto fixo.

$$W_m - W_{-m} = a (Y_n - Y_{-n}) \quad (2.15)$$

A equação (2.15), válida para  $a \geq 1$ , garante que os pesos tenham maior liberdade durante o processo de treinamento. Além disso, para  $W_o = Y_o = 0$  pode-se afirmar que:

$$m \Delta w = a (n \Delta y) \quad (2.16)$$

Se este algoritmo de treinamento for aplicado ao *CMAC* de *Kraft* e *Campagna*, a saída  $Y^e$  será dada pela soma dos pesos da região ativa e se este algoritmo de treinamento for aplicado ao *CMAC* de *Albus*, a saída  $Y^e$  será dada pela soma dos pesos das posições não nulas de um vetor de células de associação. Portanto, a saída do *CMAC* com pesos discretos, para a versão de *Kraft* e *Campagna* ou para a versão de *Albus*, é dada pelas equações (2.17) ou (2.18), respectivamente.

$$Y^e = \sum_{k=1}^c (W_{v_k}) = \sum_{k=1}^c (W_o + v_k \Delta w) \quad (2.17)$$

$$Y^e = C W_o + \Delta w \sum_{k=1}^c (v_k) \quad (2.18)$$

Para uma taxa de generalização  $C$ , através da equação (2.18) pode-se definir o valor de  $S_o$  como a soma dos  $C$  números inteiros  $v_k$ , para  $k \in \{1, 2, \dots, C\}$ , dada pela seguinte equação:

$$S_o = \sum_{k=1}^C (v_k) \quad (2.19)$$

Assim, a saída do **CMAC** com pesos discretos pode ser representada a equação a seguir:

$$Y^e = C W_o + \Delta w S_o \quad (2.20)$$

A partir dos inteiros  $(v_1, v_2, \dots, v_C)$  para os quais a saída  $Y^e$  do **CMAC** discreto coincide com a melhor aproximação discreta da saída desejada para o **CMAC**, segue que:

$$Y^e = C W_o + \Delta w S_o = Y_o + u \Delta y \quad (2.21)$$

$$(a n) u \Delta y = (a n) \Delta w S_o$$

$$(m u) \Delta w = (a n) \Delta w S_o$$

$$(m u) = (a n) S_o$$

$$S_o = \frac{m u}{a n} \quad (2.22)$$

A equação (2.22) mostra o valor da soma de números inteiros  $S_o$  quando a convergência para um ponto de treinamento ocorre. Assim, o erro inteiro correspondente ao do  $j$ -ésimo ponto de treinamento, durante a  $t$ -ésima iteração do processo de treinamento, é dado pela equação:

$$\varepsilon_{j(t)} = \frac{m}{a n} (u_j)_{(t)} - (S_o)_j)_{(t)} \quad (2.23)$$

Por analogia a regra de **Widrow-Hoff**, pode-se escrever o algoritmo de atualização dos pesos discretos do **CMAC**, onde  $(0 < \beta \leq 1)$  é a constante de aprendizado.

$$v_{k(t+1)} = v_{k(t)} + \frac{\beta}{C} (\varepsilon_j)_{(t)} \quad (2.24)$$

Portanto, a memória requerida para o armazenamento de informações no **CMAC** com pesos discretos é reduzida devido ao fato de que, armazenar um número inteiro  $v_k$  utilizando representação em ponto fixo com **16** bits, ou menos, requer uma memória menor do que um número representado em ponto flutuante que necessita de, pelo menos, **32** bits.

## ***Apêndice 3***

### ***Máquinas Elétricas de Indução***

#### ***3.1 Introdução***

As máquinas de corrente contínua têm sido largamente usadas em aplicações de controle de velocidade enquanto que as máquinas de corrente alternada têm sido tradicionalmente usadas em aplicações de velocidade constante, onde as características de estado estacionário são as mais importantes. O comportamento dinâmico de uma máquina de corrente alternada é complexo, por envolver não somente o controle direto de magnitudes de corrente e tensão, mas também da frequência elétrica e da posição relativa entre os fluxos magnéticos.

No entanto, apesar das máquinas de corrente contínua apresentarem um comportamento dinâmico mais simples, quando comparadas com alguns tipos de máquinas de corrente alternada, são inferiores em vários aspectos físicos e operacionais, tais como: confiabilidade, custo, peso, manutenção, potência e velocidade. Portanto, dentre as máquinas de corrente alternada destacam-se as máquinas elétricas de indução por serem mais robustas e confiáveis.

#### ***3.2 Motores de Indução Trifásicos***

Nos motores de indução trifásicos uma corrente elétrica trifásica balanceada é fornecida diretamente aos enrolamentos do estator e por ação de um transformador, outra corrente elétrica trifásica é induzida nos enrolamentos do rotor. Portanto, existe um acoplamento entre as fases do estator e do rotor que varia de acordo com a posição do rotor, implicando em um aumento de complexidade na descrição do comportamento dinâmico do motor de indução, que passa a ser baseado em equações diferenciais com coeficientes variantes no tempo.

Sempre que se possa garantir um suprimento balanceado de potência elétrica trifásica, o que ocorre quando o motor é alimentado via conversores de tensão ou corrente, pode ser descrito pela teoria dos eixos  $d$ - $q$ . Neste caso, as variáveis passam a ser representadas em um sistema bifásico composto por eixos ortogonais mutuamente desacoplados, um denominado eixo direto  $d$  e outro denominado eixo em quadratura  $q$ , constituindo um sistema de referência  $d$ - $q$ .

A adoção de um novo sistema de eixos de referência  $d$ - $q$  girando à velocidade síncrona, permite representar parâmetros que variam senoidalmente em função do tempo, como grandezas constantes, conduzindo à obtenção de modelos matemáticos mais simples para o motor indução, ainda que baseados em equações diferenciais com coeficientes variantes no tempo.

### 3.3 Modelo Matemático do Motor de Indução

Um motor de indução é representado por meio de um sistema de equações diferenciais não-lineares, devido ao efeito de acoplamento entre as fases do estator e do rotor, que depende da posição angular do rotor em relação ao estator. Este efeito de acoplamento torna os coeficientes das equações diferenciais variáveis. A fim de evitar a resolução de um sistema de equações diferenciais com coeficientes variáveis, utiliza-se uma transformação de coordenadas com um sistema de referência adequado, que torna os coeficientes das equações diferenciais constantes.

O método para fazer esta transformação é o das equações de *Park*, que utiliza a teoria dos eixos  $d$ - $q$ , os quais podem ser definidos para um sistema de referência girante ou estacionário. Com um sistema de referência estacionário os eixos  $d^s$  e  $q^s$  são fixos no estator enquanto que os eixos  $d^e$  e  $q^e$  podem girar com a velocidade do rotor ou com a velocidade síncrona, sendo o mais recomendado para analisar o comportamento de um motor de indução trifásico alimentado por um circuito conversor de frequência (KRAUSE & THOMAS, 1965).

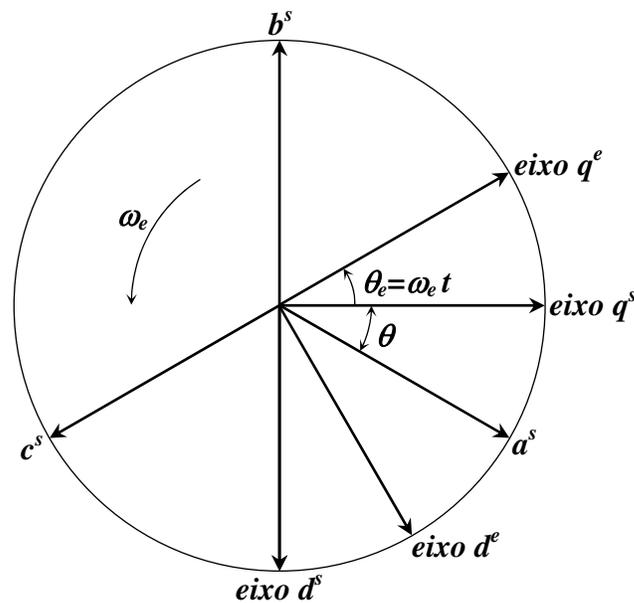


Figura 3.1 Diagrama Fasorial

A transformação das tensões  $v_{as}$ ,  $v_{bs}$ ,  $v_{cs}$  de um sistema trifásico equilibrado em tensões  $v_{ds}^s$ ,  $v_{qs}^s$ ,  $v_{os}^s$  de um sistema de referência estacionário, é dada pelo sistema de equações (3.1) e sua transformação inversa pelo sistema de equações (3.2).

$$\begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \\ \cos(\theta - 120^\circ) & \sin(\theta - 120^\circ) & 1 \\ \cos(\theta + 120^\circ) & \sin(\theta + 120^\circ) & 1 \end{bmatrix} \begin{bmatrix} v_{qs}^s \\ v_{ds}^s \\ v_{os}^s \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} v_{qs}^s \\ v_{ds}^s \\ v_{os}^s \end{bmatrix} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - 120^\circ) & \cos(\theta + 120^\circ) \\ \sin(\theta) & \sin(\theta - 120^\circ) & \sin(\theta + 120^\circ) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} v_{as} \\ v_{bs} \\ v_{cs} \end{bmatrix} \quad (3.2)$$

Nos sistemas de equações (3.1) e (3.2) para  $\theta$  igual a  $\theta$ , o eixo  $q^s$  coincide com o eixo  $a^s$ . Portanto, o sistema de equações (3.3) apresenta as tensões do sistema trifásico, enquanto que o sistema de equações (3.4) mostra as tensões referidas aos eixos estacionários  $d^s$  e  $q^s$ .

$$\begin{cases} v_{as} = v_{qs}^s \\ v_{bs} = -\frac{1}{2}v_{qs}^s - \frac{\sqrt{3}}{2}v_{ds}^s \\ v_{cs} = -\frac{1}{2}v_{qs}^s + \frac{\sqrt{3}}{2}v_{ds}^s \end{cases} \quad (3.3)$$

$$\begin{cases} v_{qs}^s = \frac{2}{3}v_{as} - \frac{1}{3}v_{bs} - \frac{1}{3}v_{cs} = v_{as} \\ v_{ds}^s = -\frac{1}{\sqrt{3}}v_{bs} + \frac{1}{\sqrt{3}}v_{cs} \end{cases} \quad (3.4)$$

As tensões referidas aos eixos estacionários  $d^s$  e  $q^s$  são transformadas para os eixos  $d$ - $q$  que giram na velocidade síncrona, através do sistema de equações (3.5). A transformação inversa das tensões referidas aos eixos estacionários  $d^s$  e  $q^s$  é efetuada pelo sistema de equações (3.6).

$$\begin{cases} v_{qs} = v_{qs}^s \cos(\omega_e t) - v_{ds}^s \sin(\omega_e t) \\ v_{ds} = v_{qs}^s \sin(\omega_e t) + v_{ds}^s \cos(\omega_e t) \end{cases} \quad (3.5)$$

$$\begin{cases} v_{qs}^s = v_{qs} \cos(\omega_e t) + v_{ds} \sin(\omega_e t) \\ v_{ds}^s = v_{ds} \cos(\omega_e t) - v_{qs} \sin(\omega_e t) \end{cases} \quad (3.6)$$

As tensões do estator e rotor da máquina de indução operando como motor, referidas ao sistema de referência que gira na velocidade síncrona são dadas pelas equações (3.7) e (3.8).

$$\begin{cases} v_{qs} = R_s i_{qs} + \frac{d}{dt} \lambda_{qs} + \omega_e \lambda_{ds} \\ v_{ds} = R_s i_{ds} + \frac{d}{dt} \lambda_{ds} - \omega_e \lambda_{qs} \end{cases} \quad (3.7)$$

$$\begin{cases} v_{qr} = R_r i_{qr} + \frac{d}{dt} \lambda_{qr} + (\omega_e - \omega_r) \lambda_{dr} \\ v_{dr} = R_r i_{dr} + \frac{d}{dt} \lambda_{dr} - (\omega_e - \omega_r) \lambda_{qr} \end{cases} \quad (3.8)$$

Considerando-se as indutâncias próprias e mútuas do rotor e do estator obtidas através do sistema de equações (3.9), as expressões dos fluxos concatenados quando referidas ao sistema de referência utilizado, são dadas pelo sistema de equações (3.10).

$$\begin{cases} L_s = L_{ls} + L_m \\ L_r = L_{lr} + L_m \end{cases} \quad (3.9)$$

$$\begin{cases} \lambda_{qs} = L_{ls} i_{qs} + L_m (i_{qs} + i_{qr}) \\ \lambda_{qr} = L_{lr} i_{qr} + L_m (i_{qs} + i_{qr}) \\ \lambda_{ds} = L_{ls} i_{ds} + L_m (i_{ds} + i_{dr}) \\ \lambda_{dr} = L_{lr} i_{dr} + L_m (i_{ds} + i_{dr}) \end{cases} \quad (3.10)$$

O conjugado eletromagnético  $T_e$  desenvolvido pelo motor de indução é determinado pela equação (3.11). Com o motor de indução operando com conjugado de carga  $T_c$  o comportamento dinâmico do sistema é descrito pela equação (3.12), onde  $J$  representa o momento de inércia do rotor e da carga e  $p$  o número de pólos do motor de indução.

$$T_e = \frac{3}{4} (\lambda_{dr} i_{qs} - \lambda_{qr} i_{ds}) p \quad (3.11)$$

$$T_e - T_c = J \frac{2}{p} \frac{d}{dt} \omega_r \quad (3.12)$$

### 3.4 *Conversor de Freqüência*

O método mais adequado para controlar a velocidade dos motores de indução, é o que utiliza os conversores de freqüência, uma vez que através dos mesmos é possível alimentar os motores de indução, com tensão e freqüência variáveis, de modo que seja mantida constante a relação tensão e freqüência. Mantendo-se constante a relação tensão e freqüência, ao mesmo tempo em que se evita a saturação do fluxo, mantém-se inalterado o conjugado máximo do motor de indução trifásico. Desta forma, é possível obter a velocidade de operação desejada, impondo-se o perfil de aceleração mais adequado, de modo o propiciar partidas mais suaves.

Quando o conversor de freqüência é alimentado por fonte de tensão fixa, configura-se um inversor de tensão. Neste tipo de inversor os dispositivos semicondutores de potência sempre permanecem polarizados diretamente devido à tensão de alimentação, portanto, é necessário o uso de comutação forçada quando se emprega tiristores como elementos de chaveamento. Como alternativa é possível o uso de comutação natural quando se empregam transistores de potência.

Uma das técnicas mais usadas para geração dos pulsos de comutação dos dispositivos semicondutores, é a modulação por largura de pulsos *PWM* (Pulse Width Modulation) com eliminação de harmônicas. Neste método, são estabelecidos os pulsos de controle cuja forma de onda é caracterizada pelo uso de ângulos de comutação predeterminados, de tal forma que o conteúdo harmônico possa ser controlado, eliminando-se os elementos indesejáveis.

### 3.5 *Modelo Matemático do Conversor*

A operação do motor de indução alimentada por um inversor de tensão do tipo *PWM*, pode ser analisada a partir da modelagem do inversor de onda quadrada. A tensão de fase na saída do inversor de onda quadrada pode ser representada de forma aproximada por uma série de *Fourier*. Para operação balanceada, a tensão da fase *a* é representada pela equação (3.13), onde  $V_d$  representa a tensão contínua fornecida pela fonte de tensão (KRAUSE & LIPO, 1969).

Em operação normal as tensões de fase na saída do inversor podem ser consideradas como um sistema trifásico e simétrico de tensões, formado pelas componentes fundamentais e suas harmônicas. Analogamente, as tensões das fases *b* e *c* são obtidas a partir da equação (3.13), substituindo-se  $(\omega_e t)$ , por  $(\omega_e t - (2/3)\pi)$  e  $(\omega_e t + (2/3)\pi)$ , respectivamente.

$$v_{as} = \frac{2}{\pi} \left[ \cos(\omega_e t) + \frac{1}{5} \cos(5\omega_e t) + \frac{1}{7} \cos(7\omega_e t) + \dots \right] V_d \quad (3.13)$$

A amplitude da componente fundamental e suas harmônicas podem ser determinadas a partir do valor instantâneo da tensão de alimentação do inversor  $V_d$ . Transformando-se estas tensões segundo o sistema de referência  $d-q$ , que gira na velocidade síncrona, através do sistema de equações (3.5) e (3.6), obtém-se o novo sistema de equações (3.14) apresentado a seguir.

$$\begin{cases} v_{qs} = \frac{2}{\pi} \left[ 1 + \frac{2}{35} \cos(6\omega_e t) - \frac{2}{143} \cos(12\omega_e t) + \dots \right] V_d \\ v_{ds} = \frac{2}{\pi} \left[ \frac{12}{35} \text{sen}(6\omega_e t) - \frac{24}{143} \text{sen}(12\omega_e t) + \dots \right] V_d \end{cases} \quad (3.14)$$

$$\begin{cases} g_{qs} = \left[ 1 + \frac{2}{35} \cos(6\omega_e t) - \frac{2}{143} \cos(12\omega_e t) + \dots \right] \\ g_{ds} = \left[ \frac{12}{35} \text{sen}(6\omega_e t) - \frac{24}{143} \text{sen}(12\omega_e t) + \dots \right] \end{cases} \quad (3.15)$$

A velocidade de rotação do sistema de referência  $d-q$  síncrono, depende da velocidade angular da componente fundamental da tensão aplicada ao estator do motor de indução, que por sua vez, está relacionada com a frequência de comutação do circuito inversor.

Os termos da série de *Fourier* do sistema de equações (3.14) podem ser definidos como duas variáveis através do sistema de equações (3.15). Assim, as tensões de alimentação do motor de indução, quando referidas ao sistema de referência  $d-q$ , passam a ser representadas através do sistema de equações (3.16) a partir das funções de chaveamento  $g_{qs}$  e  $g_{ds}$ .

$$\begin{cases} v_{qs} = \frac{2}{\pi} g_{qs} V_d \\ v_{ds} = \frac{2}{\pi} g_{ds} V_d \end{cases} \quad (3.16)$$

Para a operação do inversor em modo *PWM*, as tensões de fase do motor de indução são representadas pelo produto das tensões com forma de onda quadrada e um trem de pulsos de amplitude unitária, modulados em largura de pulso (ADAMS & FOX, 1972).

Uma forma de simulação deste sistema é apresentada a seguir, sendo que algumas modificações foram introduzidas de forma a tornar possível a análise do comportamento do sistema motor mais inversor empregando-se taxa de pulsos variável (HOLTZ, 1994).

Neste modelo foi utilizado um trem de pulsos distribuídos simetricamente em relação a componente fundamental da tensão sintetizada, cujo período  $T_e$  é dado pela equação (3.17), onde  $\omega_e$  é igual à frequência angular instantânea de operação do inversor e  $N_p$  representa o número de pulsos contidos dentro de um período da componente fundamental da tensão sintetizada.

$$T_e = \frac{2\pi}{\omega_e N_p} \quad (3.17)$$

A largura de pulsos nominal  $T_b$  é obtida a partir do número de pulsos  $N_p$  e do período da frequência angular nominal  $\omega_b$ , conforme apresentado na equação (3.18).

$$T_b = \frac{2\pi}{\omega_b N_p} \quad (3.18)$$

Pode-se estabelecer uma relação entre a frequência angular instantânea  $\omega_e$  de operação do motor de indução e a frequência angular nominal  $\omega_b$ , através da equação (3.19).

$$k = \frac{\omega_e}{\omega_b} \quad (3.19)$$

Verifica-se facilmente que a relação entre a largura do pulso  $T_b$  e seu período  $T$  mantém a mesma relação  $k$ . Desta forma, à medida que varia a frequência angular da tensão de saída do inversor, varia na mesma proporção, a relação entre a largura do pulso e seu período.

Considerando que o valor médio das tensões de fase na saída do inversor depende desta relação, torna-se possível assim manter constante o fluxo no entreferro do motor, de modo que quando o inversor estiver operando na frequência angular nominal do motor de indução, atinge-se o modo de chaveamento onde predomina a forma de onda quadrada. O trem de pulsos pode ser representado matematicamente pela série de *Fourier*, através da equação (3.20).

$$P_m = k + \sum_{n=1}^{\infty} \frac{(-1)^n}{n\pi} \left\{ \text{sen} \left[ n k \omega_b N_p (t) \right] - \text{sen} \left[ n k \omega_b N_p \left( t - \frac{2\pi}{\omega_b N_p} \right) \right] \right\} \quad (3.20)$$

Incorporando-se  $P_m$ , como um fator de modulação às expressões das tensões  $v_{qs}$  e  $v_{ds}$  do inversor de onda quadrada, representadas pelo sistema de equações (3.16), obtém-se o sistema de equações (3.21) que representa as tensões referidas no sistema *d-q* para o inversor *PWM*.

$$\begin{cases} v_{qs} = \frac{2}{\pi} g_{qs} V_d P_m \\ v_{ds} = \frac{2}{\pi} g_{ds} V_d P_m \end{cases} \quad (3.21)$$



## Referências Bibliográficas

- [01] ADAMS, R. D.; FOX, R. S. “Several Modulation Techniques for a Pulse Width Modulated Inverter”. **IEEE Transactions on Industry Applications**, Vol. 8, 1972.
- [02] ALBUS, J. S. “A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller”. **Journal of Dynamic Systems, Measurement and Control, American Society of Mechanical Engineers**, Vol. 97, September 1975.
- [03] ATHANI, V. V.; DESHPANDE, S. M. “Microprocessor Control of a Three-Phase Inverter in Motor Speed Control System”. **IEEE Transactions on Industrial Electronics and Control Instruments**, Vol. 27, 1980.
- [04] BERTSEKAS, D. P. “Dynamic Programming and Stochastic Control”. **Mathematics in Science and Engineering**, Vol. 125, Academic Press, 1976.
- [05] BORDON, M. E.; SILVA, I. N.; AVOLIO, E. “Design of a Neurofuzzy Controller with Simplified Architecture”. **International Journal of Neural Systems**, Vol. 11, pp. 229-235, 2001.
- [06] BORDON, M. J.; TEIXEIRA, M. C. “Improving Memory Requirement Using Cerebellar Model Articulation Controller with Discrete Weights”. **Proceedings of International Fuzzy Association World Congress**, pp.599-602, July 1995.
- [07] BORDON, M. J.; TEIXEIRA, M. C. **Sistemas de Controle Automático Baseados no Cerebellar Model Articulation Controller**, Dissertação de Mestrado, Faculdade de Engenharia de Ilha Solteira, Unesp, 121 p., 1995.
- [08] DOYLE, J. C.; SKIN, G. “Multivariable Feedback Design Concept for Classical Modern Synthesis”. **IEEE Transactions on Automatic Control**, Vol. 26, 1981.
- [09] FIGUEIREDO, M.; GOMIDE, F.; PEDRYCZ, W. “Fuzzy Neural Network: Structure and Learning”. **Proceedings of 5th IFSA World Congress**, Seul, Corea, 1993.
- [10] FONTES, I. R.; SZAJNER, J. **Controle de Velocidade de Motor de Indução Trifásico Alimentado por Inversor PWM através de Microcomputador**, Dissertação de Mestrado, Faculdade de Engenharia Elétrica, Unicamp, 137 p., 1993.
- [11] GOMIDE, F. ; GUDWIN, R. “Modelagem, Controle, Sistemas e Lógica Fuzzy”. **SBA Controle & Automação**, Vol. 5, pp. 97-115, 1994.
- [12] GOMIDE, F.; ROCHA, A. “A Neurofuzzy Controller”. **Proceedings of International Conference on Fuzzy Logic and Neural Network**, Furokawa, Japan, 1992.
- [13] GUILLAUME, S. “Designing Fuzzy Inference Systems from Data: An Interpretability Review”. **IEEE Transactions on Fuzzy Systems**, Vol. 9, 2001.
- [03] HOLTZ, J. “Pulse Width Modulation for Electronic Power Conversion”. **Proceedings of IEEE Special Issue on Power Electronics and Motion Control**, Vol. 82, 1994.
- [14] HOPFIELD, J. “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”. **Proceedings of National Academy Sciences**, Vol. 79, 1982.

- 
- [15] JOETTEN, R.; MAEDER, G. "Control Methods for Good Dynamic Performance Induction Motor Drives Based on Current and Voltages as Measured Quantities". **IEEE Transactions on Industry Applications**, Vol. 19, pp. 356-363, 1983.
- [16] KALMAN, R. E. "Design of a Self Optimizing Control System". **Transactions on American Society of Mechanical Engineers**, Vol. 80, pp. 468-478, 1958.
- [17] KRAFT, L. G.; CAMPAGNA, D. P. "A Comparison between CMAC Neural Network Control and Traditional Adaptive Control Systems", **IEEE Control System Magazine**, pp.36-43, April 1990.
- [18] KRAUSE, P. C.; LIPO, T. A. "Analysis and Simplified Representations of a Rectifier-Inverter Induction Motor Drive". **IEEE Transactions on Power Apparatus and Systems**, Vol. 88, pp. 588-596, 1969.
- [19] KRAUSE, P. C.; THOMAS, C. H. "Simulation of Symmetrical Induction Machinery". **IEEE Transactions on Power Apparatus and Systems**, Vol. 84, 1965.
- [20] KRIKELIS, N. J.; FASSOIS, S. D. "Microprocessor Implementation of PID Controllers and Lead-Lag Compensators". **IEEE Transactions on Industrial Electronics**, Vol. 31, 1984.
- [21] KRISHNAN, T.; RAMASWAMI, B. "A Fast Response DC Motor Speed Control System". **IEEE Transactions on Industry Applications**, Vol. 10, pp. 643-651, 1974.
- [22] LEE, C. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I and Part II". **IEEE Transactions on Systems, Man and Cybernetics**, Vol. 20, pp. 404-435, 1990.
- [23] LIPO, T. A. "The Analysis of Induction Motor with Voltage Control by Symmetrical Triggered Thyristors". **IEEE Transactions on Power Apparatus and Systems**, Vol. 90, pp. 515-525, 1971.
- [24] MANDANI, E. H.; ASSILIAN, S. "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller". **International Journal of Man-Machine Studies**, Vol. 7, pp. 1-13, 1975.
- [25] MATSUO, T.; LIPO, T. A. "A Rotor Parameter Identification Scheme for Vector-Controlled Induction Motor Drives". **IEEE Transactions on Industry Applications**, Vol. 21, pp. 624-632, 1985.
- [26] MCCULLOCH, W. S.; PITTS, W. "A logical Calculus of the Ideas Immanent in the Nervous Activity". **Bulletin of Mathematical Biophysics**, Vol. 5, pp. 115-133, 1943.
- [27] NELSON, R. H.; LIPO, T. A.; KRAUSE, P. C. "Stability Analysis of Symmetrical Induction Machinery". **IEEE Transactions on Power Apparatus and Systems**, Vol. 88, pp. 553-562, 1969.
- [28] OLIVEIRA, M.; FIGUEIREDO, M.; GOMIDE, F. A. C. "A Neurofuzzy Approach for Autonomous Control". **Proceedings of International Conference on Fuzzy Logic, Neural Nets and Soft Computing**, Fukuoka, Japan, 1994.
- [29] PARKS, C.; MILITZER, J. "A Comparison of Five Algorithms for the Training of CMAC Memories for Learning Control Systems", **Automatica Journal of International Federation of Automatic Control**, Vol. 28, N°. 05, 1992.
-

- [30] PEDRYCZ, W.; GOMIDE, F. **An Introduction to Fuzzy Sets Analysis and Design**, Massachusetts Institute Technology Press, Massachusetts, USA, 1998.
- [31] SMITH, H. W.; DAVISON, E. J. "Design of Industrial Regulators". **Proceedings of IEE**, Vol. 119, pp. 1210-1216, 1972.
- [32] STARR, B. G.; LOON, J. C. "LSI Circuit for AC Motor Speed Control". **Electronic Components and Applications**, Vol. 2, August 1980.
- [33] TAKAGI, T.; SUGENO, M. "Fuzzy Identification of System and its Application to Modeling and Control". **IEEE Transactions on Systems, Man and Cybernetics**, Vol. 15, pp. 116-132, 1985.
- [34] WIDROW, B.; LEHR, M. A. "30 Years of Adaptive Neural Networks: Perceptron, Madaline and Backpropagation". **Proceedings of IEEE**, Vol. 78, September 1990.
- [35] WONG, F. W.; SIDERIS, A. "Learning Convergence in the Cerebellar Model Articulation Controller". **IEEE Transactions on Neural Networks**, Vol. 03, 1992.
- [36] YAMAKAWA, T. "A Fuzzy Inference Engine in Nonlinear Analog Mode and Its Application to a Fuzzy Logic Control". **IEEE Transactions on Neural Networks**, Vol. 4, pp. 496-522, 1993.
- [37] ZADEH, L. A. "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes". **IEEE Transactions on Systems, Man and Cybernetics**, Vol. 3, pp. 28-44, 1973.
- [38] ZADEH, L. A. "Fuzzy Sets". **Information and Control**, Vol. 8, pp. 338-353, 1965.
- [39] ZADEH, L. A. "Fuzzy Logic". **IEEE Computer**, Vol. 21, pp. 83-92, April 1988.



## ***Bibliografia Complementar***

- [01] ADVANTECH COMPANY, **PCL 711 PC MultiLab Card User's Manual**, Advantech Company, 51 p., September, 1990.
- [02] INTEL CORPORATION, **MCS 80C31 CHMOS Single Chip 8 Bit Microcontroller**, Intel Corporation Literature Sales, 16 p., January, 1995.
- [03] INTEL CORPORATION, **MCS<sup>®</sup> 51 Microcontroller Family User's Manual**, Intel Corporation Literature Sales, 334 p., February, 1994.
- [04] KRISHNAN, R. **Electric Motor Drives Modeling Analysis and Control**, New Jersey: Prentice-Hall, 2001, 626 p. ISBN 0-13-091014-7.
- [05] KUO, B. C. **Automatic Control Systems**, New Jersey: Prentice-Hall, 1995, 897 p. ISBN 0-13-0304759-8.
- [06] MOTOROLA INCORPORATED, **4N 25 Optoisolator Transistor Output in Motorola Optoelectronics Device Data**, Motorola Literature Distribution, 6 p., 1995.
- [07] MOTOROLA INCORPORATED, **AN 571A Isolation Techniques Using Optical Couplers in Motorola Optoelectronics Device Data**, Motorola Literature Distribution, 10 p., 1995.
- [08] MOTOROLA INCORPORATED, **TCA 0372 Dual Power Operational Amplifier in Motorola Analog IC Device Data**, Motorola Literature Distribution, 8 p., 1999.
- [09] NANA ELECTRONICS, **NNC 20CTA Hall Current Sensor with Amplifier**, Nana Electronics Company, 35 p., 1990.
- [10] NATIONAL SEMICONDUCTOR, **ADC 0804 CMOS 8 Bit Microprocessor Compatible Analog to Digital Converter**, National Semiconductor Corporation, 16 p., August, 1997.
- [11] NATIONAL SEMICONDUCTOR, **LF 353 Wide Bandwidth Dual JFET Input Operational Amplifier**, National Semiconductor Customer Support Center, 14 p., December, 2003.
- [12] TEXAS INSTRUMENTS, **DAC 7801 CMOS 12 Bit Microprocessor Compatible Digital to Analog Converter**, Texas Instruments Incorporated, 19 p., February, 2004.
- [13] TEXAS INSTRUMENTS, **VFC 032 Voltage to Frequency and Frequency to Voltage Converter**, Texas Instruments Incorporated, 9 p., October, 2003.
- [14] ZADEH, L. A. "Fuzzy Logic". **IEEE Computer**, Vol. 21, pp. 83-92, April 1988.