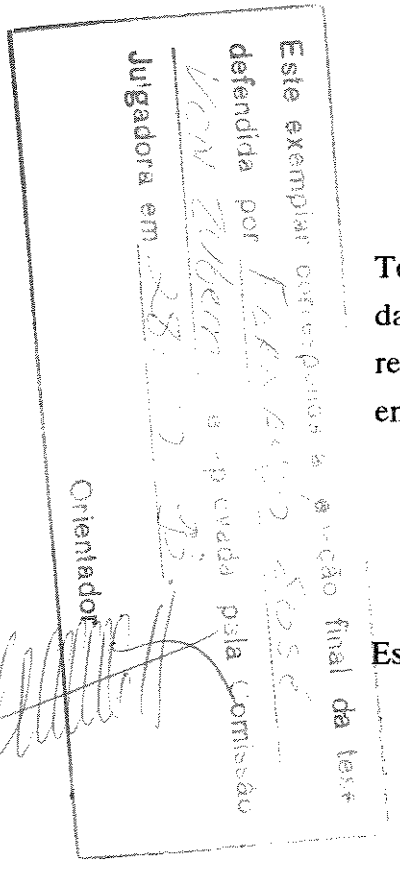
Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica
Departamento de Engenharia da Computação e Automação Industrial

REDES NEURAIS APLICADAS AO CONTROLE DE MÁQUINA DE INDUÇÃO

FERNANDO JOSÉ VON ZUBEN

Orientador: PROF. DR. MÁRCIO LUIZ DE ANDRADE NETTO
(DCA-FEE-UNICAMP)

Co-orientador: PROF. DR. EDSON BIM
(DSCE-FEE-UNICAMP)



Tese apresentada à Faculdade de Engenharia Elétrica da Unicamp (FEE-UNICAMP) como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica.

CAMPINAS
Estado de São Paulo - Brasil
Outubro de 1993

UNICAMP
BIBLIOTECA CENTRAL

Dedico a meus pais, Lorisse e Claudio
e irmãos, Paula e Claudio José

Agradecimentos

Agradeço ...

Ao Prof. Dr. Márcio Luiz de Andrade Netto pela orientação, incentivo e indispensáveis ensinamentos durante o desenvolvimento deste estudo.

Ao Prof. Dr. Edson Bim pela co-orientação, incentivo e pronta disponibilidade na discussão de inúmeras dúvidas.

Ao Prof. Dr. Jaime Szajner pela contribuição na fase de delineamento do projeto.

Ao Prof. Edson Pinheiro de Lima por sugestões importantes com relação ao esquema de controle vetorial da máquina de indução.

Aos colegas e amigos Luiz Eduardo Roncato Cordeiro e Myriam Regattieri de Biase da Silva que por diversos motivos ajudaram-me no desenvolvimento deste trabalho.

Aos demais colegas pelo apoio e companheirismo.

Aos professores que chefiaram o Departamento de Computação e Automação Industrial (DCA) durante o período de desenvolvimento deste trabalho por permitirem o acesso às instalações.

À Faculdade de Engenharia Elétrica pela oportunidade de realizar o curso de Mestrado.

À FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo suporte financeiro (Proc. nº 91/1511-8).

Índice

Lista de Símbolos vii

Resumo ix

Abstract xi

1. **Introdução** 1
    1.1. Motivação para a aplicação de redes neurais em contextos dinâmicos 1
    1.2. O controle da máquina de indução utilizando redes neurais 1
    1.3. Descrição do problema de controle 2
    1.4. Objetivo 3
    1.5. Organização do texto 3

2. **Redes neurais artificiais** 5
    2.1. Redes neurais: um tema multidisciplinar 5
    2.2. Redes neurais: um histórico 5
    2.3. Redes neurais no contexto de engenharia e inteligência artificial 9
    2.4. Especificação das principais características de uma rede neural artificial 11
        2.4.1. Unidades processadoras 12
        2.4.2. Conexões entre as unidades 13
        2.4.3. Regra de propagação 13
        2.4.4. Função de Ativação 14
        2.4.5. Função de saída 17
        2.4.6. Unidade processadora resultante 17
        2.4.7. Topologias básicas 18
        2.4.8. Critérios de processamento 18
    2.5. A representação do conhecimento em redes neurais artificiais 19
    2.6. Aprendizagem em redes neurais 20
        2.6.1. Estrutura do modelo 20
        2.6.2. Critério de desempenho 21
        2.6.3. O escopo do processo de aprendizagem 21
        2.6.4. Paradigmas de aprendizado 21
        2.6.5. Aprendizagem por ajuste da intensidade das conexões 22
    2.7. O estudo de redes neurais como mapeadores não-lineares 23
        2.7.1. O problema do OU-exclusivo 23
        2.7.2. Um problema mais geral de mapeamento não-linear 26
        2.7.3. A capacidade de aproximação universal de redes neurais artificiais 28

2.8. Treinamento de redes neurais artificiais 29
2.8.1. Análise do processo de treinamento supervisionado 29
2.8.2. Representação dos dados para treinamento supervisionado 30
2.8.3. O algoritmo de treinamento 32
2.8.3.1. Algoritmos de 1ª ordem 33
2.8.3.2. As limitações do algoritmo de treinamento de 1ª ordem baseado no método do gradiente 41
2.8.3.3. O algoritmo de treinamento de 2ª ordem 42
2.8.3.3.1. Nomenclatura e conceitos a serem utilizados no desenvolvimento do algoritmo 42
2.8.3.3.2. Estratégia de otimização 44
2.8.3.3.3. A aplicação do método do gradiente conjugado no desenvolvimento do algoritmo de treinamento de 2ª ordem 44
2.8.3.4. Comparação entre os algoritmos de 1ª e 2ª ordem 49

3. **Teoria de Sistemas de Controle** 53
3.1. Sistemas Dinâmicos 54
3.2. Sistemas não-lineares 55
3.2.1. Diferenças entre sistemas lineares e não-lineares 55
3.2.2. Atratores 57
3.3. Estabilidade de sistemas dinâmicos 57
3.4. Sistemas dinâmicos não-estacionários 59
3.5. Controle de sistemas físicos 60
3.5.1. O conceito de realimentação em teoria de controle 60
3.5.2. Controle clássico 61
3.5.3. Controle Moderno 62
3.5.4. Controle Adaptativo 63
3.5.4.1. Um breve histórico 64
3.5.4.2. Métodos de Controle Adaptativo 65
3.5.4.2.1. Planejamento de ganho 66
3.5.4.2.2. Controle adaptativo por modelo de referência 67
3.5.4.2.3. Regulador auto-ajustável 69
3.5.4.3. Resultados recentes da teoria de controle adaptativo 70
3.6. Necessidades atuais e perpectivas futuras na área de sistemas de controle 72

4. **Redes neurais aplicadas à identificação e controle de sistemas dinâmicos** 75
4.1. Configurações baseadas na aplicação de algoritmos de treinamento supervisionado 75
4.1.1. Emulação de controladores existentes 77
4.1.2. Preditor adaptativo 77
4.1.3. Identificação de sistemas dinâmicos 78
4.1.4. Identificação da dinâmica inversa de um sistema não-linear 79
4.1.5. A obtenção de um modelo diferenciável para o processo 80
4.1.6. A rede neural como um filtro não-linear adaptativo 81
4.2. Análise de critérios de implementação 83
4.2.1. As etapas envolvidas no processo de implementação 84
4.2.2. Um exemplo de implementação 85
4.2.2.1. Hipóteses iniciais 86
4.2.2.2. Escopo do conhecimento 86
4.2.2.3. Representação do conhecimento 87
4.2.2.4. Natureza dos padrões de treinamento 87
4.2.2.5. Disponibilidade dos padrões de treinamento 87
4.2.2.6. Pré-processamento dos padrões 87
4.2.2.7. Apresentação dos padrões na fase de treinamento 89
4.2.2.8. Representação do aprendizado 90
4.2.2.9. Avaliação do treinamento 90
4.2.2.10. Escopo do processo de treinamento 91
4.2.2.11. Implementação do treinamento 91
4.2.2.12. Arbitramento de parâmetros 92
4.2.2.13. Definição do processamento 92
4.2.2.14. Definição das principais características da unidade processadora 92
4.2.2.15. Definição da topologia ou arquitetura da rede neural 92
4.2.2.16. Definição do critério de processamento 92
4.2.2.17. Implementação numérica das principais características da unidade processadora 93
4.2.2.18. Dimensionamento da rede neural 93
4.2.2.19. Especificações numéricas do processamento 94
4.2.2.20. Resultado do processo de treinamento 95
4.3. Considerações a respeito de arquiteturas de redes neurais utilizadas na representação de sistemas dinâmicos 97
4.3.1. Comparação entre as duas arquiteturas de redes neurais na representação de comportamentos dinâmicos 97

4.3.2. Redes neurais recorrentes 99
4.3.2.1. Aplicação da rede de Elman linear em contextos dinâmicos .. 100
4.3.2.2. Simulações computacionais - Rede de Elman linear 102
4.3.2.3. Extensão para o caso não-linear 106
4.3.2.4. Um exemplo de simulação - identificação de um sistema não-linear 108
4.3.3. Análise do processo de treinamento de redes neurais recorrentes 108
4.3.3.1. O fenômeno de bifurcação 109
4.3.3.2. Procedimentos de minimização dos efeitos de instabilidade no treinamento 113

5. **Máquina de Indução - Modelagem e Técnicas de Controle** 115
5.1. Máquina de Indução - Conceitos básicos 116
5.1.1. O Modelo d-q da Máquina de Indução 117
5.1.1.1. A Representação por Espaço de Estados 119
5.1.1.2. Obtenção do modelo discreto 122
5.2. Inversores - Conceitos Básicos 124
5.3. Técnicas de controle para operação em velocidade variável 125
5.3.1. A obtenção do desacoplamento entre as variáveis de controle 126
5.3.2. O Princípio da Orientação de Fluxo 127
5.3.2.1. Orientação baseada no fluxo de rotor 128
5.3.2.2. Métodos de determinação da posição do fluxo 130
5.3.2.2.1. O método direto 130
5.3.2.2.2. O método indireto 131
5.3.2.2.3. O método por realimentação de estado 134
5.3.2.2.3.1. Observador de fluxo utilizando redes neurais recorrentes 136
5.3.3. Resultados de simulação do processo de controle 142
5.3.3.1. Simulações utilizando o método indireto 143
5.3.3.2. Simulações utilizando o método por realimentação de estado 147

6. **Análise e solução do problema de sensibilidade à variação de parâmetros** 151
6.1. Estudo da variação de parâmetros 152
6.1.1. Modelo dinâmico do efeito da variação de parâmetros 153
6.1.2. Modelo estático do efeito da variação de parâmetros 154
6.1.3. Comparação entre os modelos estático e dinâmico 156
6.2. Análise do desempenho do método indireto de controle vetorial quando sujeito à variação de parâmetros 159

6.3. Análise do desempenho do método por realimentação de estado quando sujeito à variação de parâmetros 164
6.3.1. O observador expandido de fluxo 168

6.4. Estimação não-linear de parâmetros utilizando redes neurais 172
6.4.1. O procedimento de solução do problema 173
6.4.2. O esquema de ajuste de parâmetros 176
6.4.3. Implementação da solução proposta 177
6.4.4. A evolução do treinamento da rede neural 179
6.5. A introdução do estimador de parâmetros baseado em redes neurais na malha de controle vetorial 181
6.5.1. Resultados de simulação para o método indireto 182
6.5.2. Resultados de simulação para o método por realimentação de estado 187
6.6. Utilização de um observador de fluxo na obtenção do conjugado efetivamente produzido na máquina 192
6.6.1. Resultados de simulação para o método indireto 195
6.6.2. Resultados de simulação para o método por realimentação de estado 199

7. **Conclusões** 205

A. **A capacidade de aproximação universal de redes neurais artificiais** 207
A.1. Motivação para o emprego de redes neurais em tarefas de aproximação 207
A.2. Análise do potencial de aproximação de redes neurais 209
A.3. Comparação com outros métodos de aproximação 210

B. **Análise de Sistemas Dinâmicos por Espaço de Estados** 213
B.1. Representação de Sistemas Lineares de Tempo Contínuo por Espaço de Estados 214
B.1.1. Sistemas descritos por equações diferenciais lineares de ordem n 214
B.1.2. A não-unicidade do conjunto de variáveis de estado 216
B.1.3. Equação de estado linear invariante no tempo 217
B.1.3.1. Solução da equação de estado homogênea 217
B.1.3.2. Solução da equação de estado não-homogênea 218
B.1.3.3. Matriz de Transição de Estados 219
B.1.3.4. Matriz de Transferência 220
B.1.4. Equação de estado linear variante no tempo 220
B.2. Representação de Sistemas Lineares de Tempo Discreto por Espaço de Estados 221
B.2.1. Discretização de Equações de Estado de Tempo Contínuo 221

B.2.2. Solução da Equação de Estado, Controlabilidade e Observabilidade 223
B.2.3. Obtenção da função de entrada-saída: realizabilidade 225
B.3. Representação de sistemas não-lineares por espaço de estados 226
B.3.1. Observabilidade no caso não-linear 228

C. **Conjugado Eletromagnético em Máquinas Trifásicas de Corrente Alternada** .. 229

D. **Relações de transformação entre sistemas de referência trifásicos e bifásicos** ... 233
D.1. Transformação de um sistema trifásico em um sistema bifásico equivalente 233
D.2. Transformação de um sistema bifásico estacionário em um sistema bifásico girante 234

E. **Parâmetros e condições nominais da máquina de indução** 235

F. **Parâmetros e resultados do processo de treinamento de redes neurais artificiais** 237
F.1. Aproximação de um período de senóide (Capítulo 4) 237
F.2. Aproximação de uma representação não-linear por espaço de estados (Capítulo 4) 238
F.3. Observador de fluxo de rotor (Capítulo 5) 238
F.4. Observador expandido de fluxo de rotor (Capítulo 6) 239
F.5. Estimador de constante de tempo de rotor (Capítulo 6) 239

Bibliografia 241

Lista de Símbolos

1) Subíndices

s - variáveis e parâmetros de estator

r - variáveis e parâmetros de rotor

q - variáveis de eixo em quadratura

d - variáveis de eixo direto

2) Superíndices

s - variáveis na referência estacionária

e - variáveis na referência síncrona

* - variáveis de referência e parâmetros nominais

^ - variáveis e parâmetros estimados

3) Variáveis

i_a^{s*} - corrente de referência da fase "a"

i_b^{s*} - corrente de referência da fase "b"

i_c^{s*} - corrente de referência da fase "c"

i_a^s - corrente efetiva da fase "a"

i_b^s - corrente efetiva da fase "b"

i_c^s - corrente efetiva da fase "c"

$\bar{I}_s^s$ - vetor corrente de estator (referência estacionária)

i_{qs}^s - corrente de estator de eixo em quadratura (referência estacionária)

i_{ds}^s - corrente de estator de eixo direto (referência estacionária)

$\bar{I}_s^e$ - vetor corrente de estator (referência síncrona)

i_{qs}^e - corrente de estator de eixo em quadratura (referência síncrona)

i_{ds}^e - corrente de estator de eixo direto (referência síncrona)

$\bar{I}_r^s$ - vetor corrente de rotor (referência estacionária)

i_{qr}^s - corrente de rotor de eixo em quadratura (referência estacionária)

i_{dr}^s - corrente de rotor de eixo direto (referência estacionária)

$\bar{I}_r^e$ - vetor corrente de rotor (referência síncrona)

i_{qr}^e - corrente de rotor de eixo em quadratura (referência síncrona)

i_{dr}^e - corrente de rotor de eixo direto (referência síncrona)

$\bar{V}_s^e$ - vetor tensão de estator

$\bar{V}_r^e$ - vetor tensão de rotor

$\overline{\psi}_r^e$ - vetor fluxo de rotor

$\overline{\psi}_r^{e*}$ - vetor fluxo de rotor de referência

ψ_{qr}^{e*} - fluxo de rotor de referência de eixo em quadratura

ψ_{dr}^{e*} - fluxo de rotor de referência de eixo direto

ψ_{qr}^{e} - fluxo de rotor de eixo em quadratura

ψ_{dr}^{e} - fluxo de rotor de eixo direto

w_e - velocidade angular síncrona

w_r - velocidade angular do rotor (referência no estator)

w_{slip} - velocidade angular de escorregamento

T_e^* - conjugado eletromagnético de referência

T_e - conjugado eletromagnético efetivamente produzido na máquina

T_e^{mec} - estimativa do conjugado eletromagnético efetivamente produzido na máquina utilizando variáveis de origem mecânica

T_e^{elet} - estimativa do conjugado eletromagnético efetivamente produzido na máquina utilizando variáveis de origem eletromagnética

T_L - conjugado de carga

θ_e - posição espacial do vetor fluxo de rotor

T - período de amostragem

WSnom - velocidade angular nominal de escorregamento

TEnom - conjugado eletromagnético nominal

Fnom - valor nominal para o vetor fluxo de rotor

4) Parâmetros

R_s - resistência de estator por fase

R_r - resistência de rotor por fase (referida ao estator)

L_{ls} - indutância de dispersão do estator

L_{lr} - indutância de dispersão do rotor (referida ao estator)

L_m - indutância mútua de magnetização (estator/rotor)

L_s - indutância própria do estator

L_r - indutância própria do rotor (referida ao estator)

P - número de pólos

K - coeficiente de atrito viscoso

J - momento de inércia do eixo da máquina

τ_r - constante de tempo de rotor

α - razão entre a constante de tempo (τ_r) real e a nominal

β - razão entre a indutância mútua (L_m) real e a nominal

γ - razão entre a resistência de rotor (R_r) real e a nominal

Resumo

Esta dissertação apresenta técnicas não-lineares aplicadas ao problema de controle de velocidade da máquina de indução trifásica com rotor tipo gaiola de esquilo. Redes neurais artificiais são utilizadas para desenvolver estratégias de controle adaptativo baseadas em identificação não-linear de sistemas dinâmicos e estimação não-linear de parâmetros. Os resultados principais incluem arquiteturas de redes neurais e algoritmos de treinamento desenvolvidos especialmente para aplicação a sistemas dinâmicos, um observador não-linear de fluxo de rotor baseado em redes neurais recorrentes e um estimador não-linear da constante de tempo de rotor baseado em redes neurais não-recorrentes. Resultados de simulação computacional são obtidos e o desempenho da estrutura de controle resultante é analisado.

Abstract

This dissertation presents nonlinear techniques applied to the speed control problem of the three-phase squirrel-cage induction machine. Artificial neural networks are used to develop adaptive control strategies based on nonlinear dynamic system identification and nonlinear parameter estimation. The principal results include neural network architectures and training algorithms specially developed for dynamic systems application, a nonlinear rotor flux observer based on recurrent neural networks, and a nonlinear rotor time constant estimator based on feed forward neural networks. Computer simulation results are obtained and the performance of the resulting control structure is analyzed.

Capítulo 1

Introdução

1.1. Motivação para a aplicação de redes neurais em contextos dinâmicos

Os métodos de processamento de sistemas inteligentes são alvo de estudo desde a época de filósofos como Platão (427-347 a.C.) e Aristóteles (384-322 a.C.). Apesar disso, pode-se afirmar que os fundamentos dos sistemas inteligentes artificiais são um resultado dos últimos 50 anos de pesquisa. Particularmente, o que hoje é conhecido como rede neural artificial começou a ser desenvolvido em meados deste século. Mas foi no início dos anos 80 que os resultados mais importantes do ponto de vista de aplicação foram obtidos. Com isso, o campo de atuação se ampliou muito, inclusive gerando perspectivas de aplicação na área de controle de sistemas dinâmicos, a qual será enfatizada no desenvolvimento deste trabalho.

O potencial apresentado pelas redes neurais artificiais na aproximação de mapeamentos não-lineares estáticos e dinâmicos representa a principal motivação para sua aplicação a problemas de identificação e controle de sistemas dinâmicos. Para este tipo de aplicação, redes neurais multi-camadas recorrentes e não-recorrentes e algoritmos de treinamento supervisionado são geralmente utilizados.

Sistemas de controle baseados em técnicas lineares e aplicados a problemas de controle adaptativo impõem fortes restrições quanto à natureza do sistema a ser controlado, além de considerarem uma dependência linear com relação às variáveis básicas do sistema e seus respectivos retardos no tempo. Métodos conexionistas de identificação, por sua vez, operam em sistemas de diferentes graus de complexidade, através da utilização de representações de dimensão maior que a dimensão intrínseca do sistema e da utilização de um grande número de operadores não-lineares similares entre si. Este tipo de flexibilidade de representação, muitas vezes atribuída à analogia com propriedades presentes em sistemas neurais biológicos, é uma das principais características dos sistemas conexionistas.

Portanto, a introdução de redes neurais em arquiteturas de controle permite a extensão para o caso não-linear de técnicas de identificação e estimação originalmente desenvolvidas para o caso linear.

1.2. O controle da máquina de indução utilizando redes neurais

O controle da máquina de indução com rotor tipo gaiola de esquilo para operação em velocidade variável exige o conhecimento preciso de variáveis e parâmetros da máquina de forma a garantir respostas dinâmicas rápidas e um bom desempenho em regime estacionário.

Como a máquina pode apresentar parâmetros variantes no tempo, a manutenção do nível de eficácia do processo de controle está vinculado ao emprego de técnicas de identificação do comportamento dinâmico e estimação de parâmetros.

O comportamento dinâmico não-linear e não-estacionário da máquina de indução representa uma forte motivação para o emprego de redes neurais no processo de implementação destas técnicas.

1.3. Descrição do problema de controle

A simplificação do processo de representação do comportamento dinâmico da máquina de indução obtida com a adoção de um referencial d-q síncrono representa uma forte motivação para o emprego de técnicas de controle vetorial. Estas técnicas exigem o conhecimento da posição instantânea do vetor de fluxo para definir a orientação do sistema de referência de forma a garantir o desacoplamento entre as variáveis de controle, condição necessária para obtenção de um bom desempenho dinâmico.

Com a manutenção da estrutura original da máquina de indução com rotor tipo gaiola de esquilo, a orientação do sistema de referência baseada na determinação da posição espacial do vetor de fluxo é prejudicada pela dificuldade de acesso a parâmetros e variáveis do rotor.

Neste trabalho, duas técnicas de controle vetorial são utilizadas, ambas baseadas no conhecimento da dinâmica inversa da máquina e na geração dos comandos de alimentação por intermédio de inversores. Essas técnicas diferem no método de determinação da posição do fluxo:

- o método indireto: procura estimar a posição espacial do vetor de fluxo assumindo a condição de desacoplamento entre as variáveis de controle e utilizando parâmetros da máquina. Se os parâmetros da máquina variarem no tempo, é necessário o emprego de métodos de estimação para promover a atualização dos correspondentes parâmetros do controlador.
- método por realimentação de estado: utiliza um modelo dinâmico da máquina para estimar os valores das componentes d-q de fluxo a serem realimentadas para o controlador. Os sinais realimentados são, então, comparados com os comandos de referência e participam do processo de geração dos comandos de controle. Se os parâmetros da máquina variarem no tempo, é necessário o emprego de métodos de estimação para promover a atualização dos correspondentes parâmetros do modelo dinâmico.

Os métodos de estimação de parâmetros devem operar em paralelo com o controlador, e sempre que forem detectadas variações de parâmetros da máquina, os correspondentes parâmetros do controlador devem ser imediatamente atualizados de forma

a garantir a manutenção do desacoplamento entre as variáveis de controle, condição necessária para a obtenção de respostas dinâmicas rápidas.

1.4. Objetivo

O objetivo deste trabalho é o desenvolvimento de técnicas não-lineares de identificação de sistemas dinâmicos e estimação de parâmetros utilizando redes neurais artificiais e posterior aplicação ao problema de controle da máquina de indução para operação em velocidade variável.

No desenvolvimento de técnicas não-lineares de identificação e estimação, é necessário definir-se arquiteturas de redes neurais e algoritmos de treinamento mais adequados para cada aplicação.

Na aplicação ao problema de controle da máquina de indução, é necessário definir-se estratégias de identificação e estimação que permitam o emprego das técnicas não-lineares desenvolvidas.

As estruturas de controle resultantes devem ser implementadas a nível computacional e resultados de simulação devem ser obtidos de forma a demonstrar a eficiência do processo de controle para condições normais de operação da máquina de indução.

1.5. Organização do texto

A dissertação está estruturada em sete capítulos e seis anexos, sendo que os capítulos apresentam o seguinte conteúdo:

Capítulo 1

É discutido o problema de controle e apresentada a motivação para seu tratamento utilizando redes neurais artificiais. O objetivo da dissertação é definido e o conteúdo da dissertação é detalhado.

Capítulo 2

É apresentada uma discussão introdutória a respeito do tema redes neurais artificiais, procurando localizá-lo dentro do contexto em que está inserido. A seguir, são especificadas as principais características de uma rede neural artificial, resultando na definição da unidade processadora básica a ser utilizada na construção das diversas arquiteturas de redes neurais. Na seqüência, é estudada a capacidade de representação apresentada pelas redes neurais artificiais. Finalmente, algoritmos de treinamento supervisionado de 1ª e 2ª ordens para redes neurais multi-camadas recorrentes e não-recorrentes são desenvolvidos.

Capítulo 3

São apresentados conceitos básicos relacionados à teoria de sistemas de controle, a serem utilizados no desenvolvimento do trabalho. A seguir, são apresentados métodos de controle clássico e moderno, enfatizando métodos de controle adaptativo. Finalmente, as necessidades atuais e as perspectivas futuras na área de controle são analisadas.

Capítulo 4

São apresentadas arquiteturas básicas envolvendo redes neurais e sistemas dinâmicos em diversas configurações. A seguir, os critérios básicos de implementação são discutidos e utilizados no desenvolvimento de um exemplo de aplicação. Finalmente são analisadas arquiteturas especializadas para aplicação a sistemas dinâmicos, enfatizando as arquiteturas e propriedades de redes neurais recorrentes.

Capítulo 5

São apresentadas técnicas de modelagem da máquina de indução utilizando o sistema de referência d-q síncrono e representações por espaço de estados. A seguir, técnicas de controle para operação em velocidade variável são desenvolvidas. Finalmente, duas técnicas de controle são implementadas a nível computacional e resultados de simulação são apresentados e analisados.

Capítulo 6

São obtidos resultados teóricos e de simulação referentes à sensibilidade a variação de parâmetros apresentada pelas duas técnicas de controle implementadas no Capítulo 5, indicando a necessidade da aplicação de métodos de ajuste de parâmetros. As técnicas de controle são, então, redefinidas a partir de conceitos básicos da teoria de controle adaptativo e um estimador não-linear de parâmetros é desenvolvido utilizando redes neurais artificiais. O estimador é introduzido na malha de controle e resultados de simulação computacional são obtidos no caso das duas técnicas de controle em questão. Como o estimador de parâmetros utiliza o valor do conjugado de carga, um observador de fluxo capaz de eliminar a necessidade de conhecimento do conjugado de carga é proposto e implementado a nível computacional. Novos resultados de simulação são apresentados e analisados.

Capítulo 7

É apresentada uma análise geral dos resultados obtidos durante todo o desenvolvimento do trabalho e são discutidas alternativas de generalização das técnicas já desenvolvidas.

Capítulo 2

Redes neurais artificiais

2.1. Redes neurais: um tema multidisciplinar

Redes neurais artificiais podem ser definidas como estruturas computacionais projetadas com o objetivo de explorar os princípios organizacionais existentes nos sistemas neurais biológicos. Por sua natureza, o estudo de redes neurais é multidisciplinar, envolvendo pesquisadores de diversas áreas, como neurofisiologia, psicologia, física, computação e engenharia.

Neurofisiologistas e psicólogos estão particularmente interessados em compreender o funcionamento do sistema neural humano. As características de resposta a estímulos apresentada por neurônios individuais bem como redes de neurônios são alvo de estudo dos neurofisiologistas, enquanto os psicólogos estudam as funções do cérebro a nível cognitivo e estão interessados na utilização de técnicas baseadas em redes neurais para criar modelos detalhados do comportamento humano.

Cientistas da área de computação têm em vista a construção de computadores dotados de processamento paralelo, buscando superar as limitações impostas pelos computadores atuais, que realizam processamento serial e simbólico.

Inspirados na habilidade apresentada pelos seres humanos e outros animais no desempenho de funções como o processamento de informações sensoriais e a capacidade de interação com ambientes pouco definidos, os engenheiros estão preocupados em desenvolver sistemas artificiais capazes de desempenhar tarefas semelhantes. Habilidades como capacidade de processamento de informações incompletas ou imprecisas e generalização são propriedades desejadas em tais sistemas.

2.2. Redes neurais: um histórico

As redes neurais artificiais passaram por um interessante processo de evolução, marcado por um período de grande atividade seguido por anos de estagnação nas pesquisas e pelo ressurgimento do interesse científico como conseqüência do desenvolvimento de novas tecnologias e fundamentos teóricos. A seguir, é apresentado um breve histórico da pesquisa em redes neurais, sendo enfatizados alguns resultados e conceitos considerados relevantes no desenvolvimento deste trabalho.

Alguns dos mais destacados pesquisadores envolvidos no estudo e aplicação de redes neurais nas últimas três décadas estão relacionados na tabela 2.1. Esta tabela é dividida de forma a ressaltar o período cronológico mais significativo na atividade científica de cada pesquisador na área de redes neurais, e é tomada como base no processo de sequenciamento histórico.

1943	McCulloch e Pitts
1948	Wiener
1949	Hebb
1957	Rosenblatt
1958	Widrow e Hoff
...	...
1969	Minsky e Papert
...	...
1960-1980	Kohonen, Grossberg, Widrow, Anderson, Caianiello, Fukushima,
...	...
1974	Werbos
...	...
1982	Hopfield
1986	Rumelhart e McClelland

Tabela 2.1 - História da pesquisa em Redes Neurais

Em 1943, McCulloch e Pitts [46], dois pesquisadores da área de biologia, propuseram um modelo de um neurônio como uma unidade de processamento binária (veja figura 2.1) e provaram que estas unidades são capazes de executar muitas das operações que podem ser descritas em termos lógicos. Este modelo, apesar de muito simples, trouxe uma grande contribuição para as discussões sobre a construção dos primeiros computadores digitais, permitindo a criação dos primeiros modelos matemáticos de dispositivos artificiais que buscavam analogias biológicas.

Em 1948, N. Wiener [78] criou a palavra cibernética para descrever, de forma unificada, controle e comunicação nos organismos vivos e nas máquinas.

Em 1949, D. O. Hebb [29] apresentou uma hipótese a respeito da maneira com que a força das sinapses no cérebro se altera em resposta à experiência. Em particular ele sugeriu que as conexões entre células que são ativadas ao mesmo tempo tendem a se fortalecer, enquanto que as outras conexões tendem a se enfraquecer. Esta hipótese passou a influir decisivamente na evolução da teoria de aprendizagem em redes neurais artificiais.

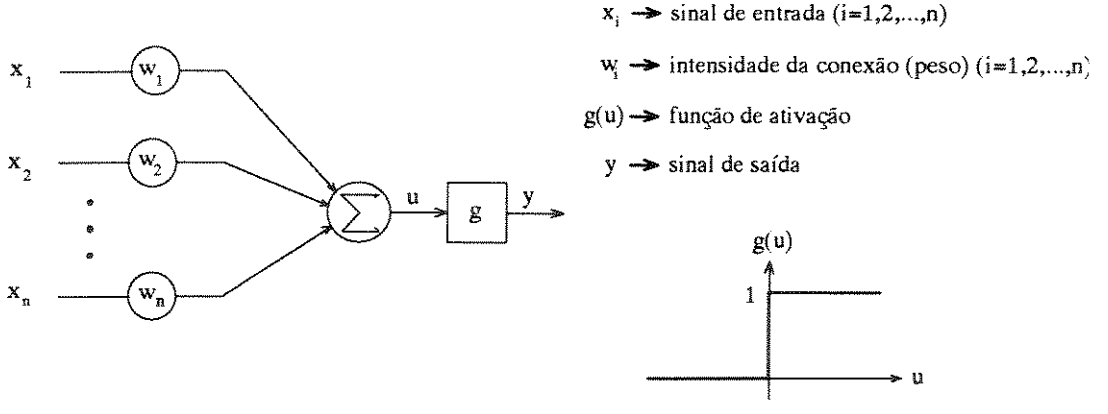


Figura 2.1 - Representação funcional de um neurônio

Em 1957, F. Rosenblatt [60] introduziu uma nova abordagem para o problema de reconhecimento de padrões com o desenvolvimento do perceptron, cuja representação diagramática é apresentada na figura 2.2. Rosenblatt também propôs um algoritmo para ajuste de pesos para o perceptron e provou sua convergência quando os padrões são linearmente separáveis. Por volta do mesmo período, B. Widrow [73] e seus colaboradores desenvolveram o adaline ("**Ada**ptive **Lin**ear **E**lement") para uso em processamento de sinais, o qual apresenta uma estrutura similar ao perceptron.

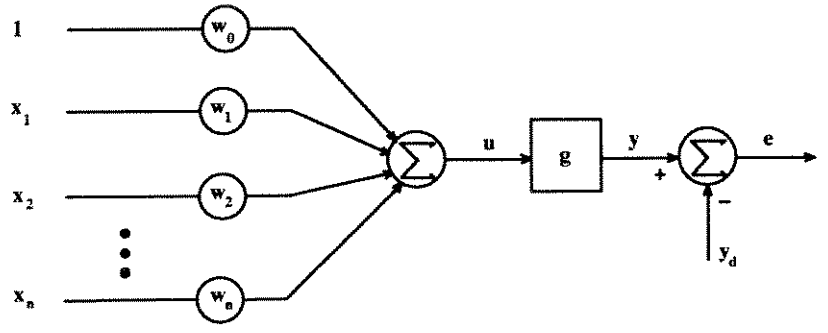


Figura 2.2 - O perceptron

Apesar do sucesso do perceptron e do adaline, a pesquisa em redes neurais passou gradualmente a conviver com dois problemas fundamentais. Devido ao fato de a maior parte das pesquisas até então desenvolvidas ser de natureza heurística, o primeiro problema estava vinculado à carência de resultados teóricos que justificassem a manutenção do interesse científico pela área, o que ocasionou a redução na produção de novas idéias. O segundo problema, e talvez o de maior significado histórico, foi a expectativa exagerada criada pelos próprios pesquisadores desta área, não acompanhada de resultados à altura, o que acelerou a queda de financiamentos para pesquisa. Mas foi após a publicação, em 1969, do livro "Perceptrons" de autoria de M. L. Minsky e S. A. Papert [48], que as pesquisas na área de redes neurais sofreram uma retração significativa. Neste livro, conceitos de matemática moderna como topologia e teoria de grupo são aplicados com o objetivo de analisar as capacidades adaptativas e computacionais de neurônios do tipo apresentado na figura 2.2. Como resultado, os autores demonstraram que o perceptron, apesar de ser capaz de executar as operações booleanas AND e OR, não é capaz de executar outras operações elementares, como o XOR (OU-exclusivo). Além disso, esses autores não acreditavam que uma arquitetura adequada, juntamente com um algoritmo de ajuste de pesos, pudessem ser desenvolvidos de forma a superar esta limitação. Após a publicação destes resultados, a maior parte dos pesquisadores da área de redes neurais passou a buscar alternativas dentro do campo da engenharia e, principalmente, da lógica matemática, que, na época, encontrava-se em franca expansão devido às grandes conquistas realizadas na área de computação.

Apesar deste êxodo generalizado, um número de pesquisadores continuou a trabalhar com redes neurais nos anos 70. Os nomes de T. Kohonen (Finlândia), S. Grossberg, B. Widrow e J. Anderson (Estados Unidos), E. Caianiello (Itália) e K. Fukushima (Japão) estão associados a este período.

Nos anos 80, muitos fatores contribuíram para o ressurgimento definitivo das pesquisas em redes neurais:

- neurofisiologistas foram adquirindo um maior conhecimento acerca do processamento de informação em organismos vivos;
- avanços tecnológicos tornaram disponível um maior potencial computacional a baixo custo, viabilizando ou facilitando simulações e testes com modelos de redes neurais;
- novas teorias para a implementação de algoritmos adaptativos foram desenvolvidas, permitindo a aplicação em sistemas reais.

Em termos práticos, importantes resultados foram obtidos através da aplicação de conceitos conexionistas ao problema de modelagem de materiais paramagnéticos (vidros de spin), realizada por J. Hopfield e publicada em 1982 [31]. Hopfield considerou um conjunto de neurônios dispostos de forma a que suas saídas fossem realimentadas para as entradas. Este modelo, um dos primeiros a introduzir dinâmica em redes neurais, é apresentado na figura 2.3, a seguir, e ajuda a demonstrar como uma rede composta por elementos computacionais simples pode resultar em interessantes comportamentos coletivos. A rede de Hopfield pode ser considerada como um sistema dinâmico com um número finito de estados de equilíbrio, de forma que o sistema invariavelmente irá evoluir para um destes estados a partir de uma condição inicial. É também natural que a localização destes estados de equilíbrio possa ser controlada pela intensidade das conexões (pesos) da rede neural.

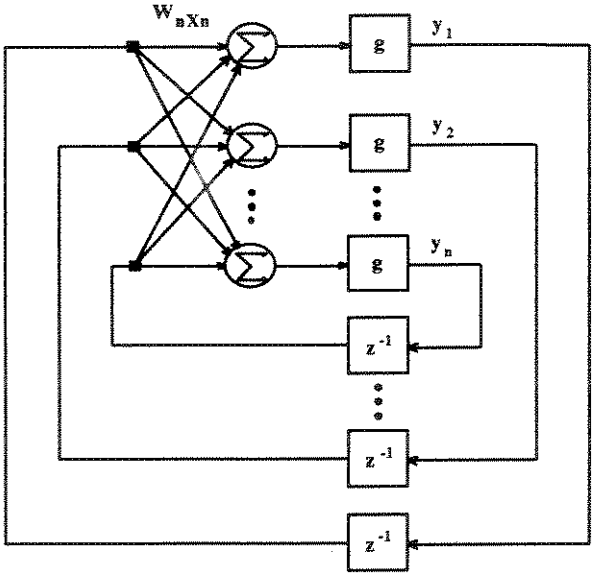


Figura 2.3 - A rede neural de Hopfield

A conclusão interessante adotada por Hopfield foi que tais estados de equilíbrio podem ser utilizados como dispositivos de memória. De uma forma distinta daquela utilizada pelos computadores convencionais, em que o acesso à informação armazenada se dá por meio de um endereço, o acesso ao conteúdo da memória em uma rede de Hopfield se dá permitindo que a rede evolua com o tempo para um de seus estados de equilíbrio. Tais modelos de memória são denominados memórias endereçáveis por conteúdo. O trabalho de Hopfield com este tipo de rede simétrica recorrente atraiu principalmente matemáticos e engenheiros para pesquisa nesta área, e as redes de Hopfield foram estudadas como memórias distribuídas e também utilizadas como ferramentas na solução de problemas de otimização restrita.

No entanto, o fato que efetivamente colocou a área de redes neurais como uma das prioritárias na obtenção de recursos foi o desenvolvimento de um método para ajuste de parâmetros de redes não-recorrentes de múltiplas camadas. Esse método, baseado em um algoritmo denominado retro-propagação ("back-propagation"), tornou-se largamente conhecido após a publicação, em 1986, do livro "Parallel Distributed Processing", editado por J. L. McClelland e D. E. Rumelhart [44,45], fazendo com que pesquisadores das mais diferentes áreas passassem a visualizar interessantes aplicações para redes neurais artificiais. A importância deste método justifica um tratamento mais aprofundado, a ser desenvolvido nas próximas seções.

A consolidação de redes neurais como um novo ramo de pesquisa é evidenciada pela organização de numerosas conferências e fundação de revistas científicas devotadas exclusivamente ao tratamento deste tema, que recebeu a denominação de teoria conexionista.

2.3. Redes neurais no contexto de engenharia e inteligência artificial

A descrição, na seção anterior, da evolução histórica da pesquisa em redes neurais artificiais fornece subsídios para um melhor entendimento do estágio atual de desenvolvimento da pesquisa nesta área, além de ajudar a posicionar a teoria conexionista dentro do universo científico em que está integrada. Uma abordagem bastante apropriada a este propósito foi proposta por A. G. Barto (89)[5] e é apresentada na figura 2.4, onde é estabelecida uma comparação a nível de evolução histórica entre métodos conexionistas, métodos de engenharia e métodos de inteligência artificial. No esquema da figura 2.4, as bifurcações não representam necessariamente momentos históricos bem definidos, mas períodos marcados por uma série de acontecimentos, muitos deles descritos na seção anterior.

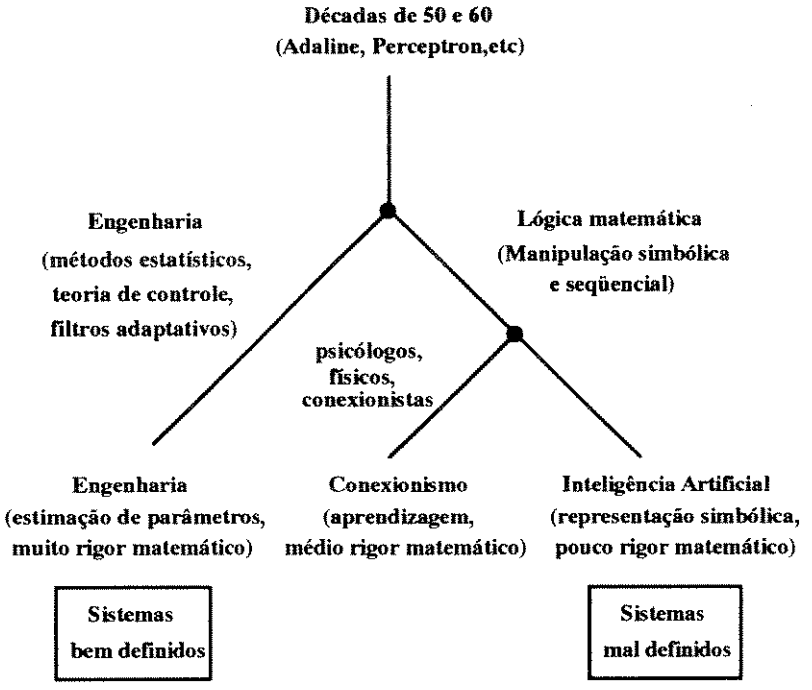


Figura 2.4 - A evolução histórica da pesquisa em redes neurais

A primeira bifurcação evidenciada na figura 2.4 iniciou-se com o lançamento em 1969 do livro "Perceptrons" de autoria de M. L. Minsky e S. A. Papert [48], onde foram apresentadas conclusões desfavoráveis à capacidade de representação de estruturas conexionistas. Estas conclusões, a serem melhor analisadas posteriormente, acabaram por conduzir os pesquisadores da área de redes neurais a buscarem alternativas principalmente no campo da engenharia e da lógica matemática.

A segunda bifurcação ocorreu no início dos anos 80, e é representada pelo surgimento do ramo atual de pesquisa em redes neurais, denominado conexionismo. Um fator fundamental é a sua derivação a partir do ramo de pesquisa baseado na lógica matemática e não do ramo de pesquisa baseado em métodos de engenharia. Esta é uma das razões pela qual, apesar das técnicas conexionistas serem intimamente relacionadas com métodos de engenharia (processamento numérico, gradiente, erro quadrático médio, etc.), a maior parte das pesquisas na área de redes neurais apresenta mais características experimentais e heurísticas do que se supõe existirem em métodos clássicos de engenharia. Uma outra conseqüência desta ramificação é que, na aplicação de redes neurais, representações expansivas são comumente utilizadas, ou seja, a capacidade de representação da rede neural tem sempre uma dimensão maior que a dimensão intrínseca do problema.

Portanto, baseados na representação da figura 2.4, os métodos conexionistas podem ser classificados a meio-termo entre métodos de engenharia e métodos de inteligência artificial, pois se utilizam de técnicas matemáticas aplicadas a problemas de engenharia, ao mesmo tempo em que incorporam representações expansivas e metodologias heurísticas, bem aos moldes de métodos de inteligência artificial.

Apesar da importância da manutenção de alto padrão de rigor matemático, uma abordagem heurística parece ser essencial para o tratamento de problemas que carecem de uma estruturação matemática adequada, ou mesmo para a observação de fenômenos que conduzam à elaboração de conceitos rigorosos do ponto de vista matemático. Dessa forma, a pesquisa com redes neurais apresenta-se como uma via de comunicação entre linhas de pesquisa que se dissociaram há mais de vinte anos. Assim, não deve causar surpresa o fato de que métodos já clássicos para uma das linhas de pesquisa sejam incorporados pela outra, muitas vezes utilizando linguagem e nomenclatura distintas da original.

Quanto à abordagem para problemas de controle, pode-se contar com avançadas metodologias desenvolvidas no campo da engenharia para a solução de variados tipos de problemas. Conseqüentemente, o estudo de métodos conexionistas aplicados a problemas de controle irá preferencialmente se apoiar em tais metodologias, em detrimento de abordagens desenvolvidas no campo da Inteligência Artificial, baseadas em processamento simbólico. Isto não significa que sistemas híbridos, baseados nas duas metodologias, ou mesmo sistemas totalmente baseados em métodos de Inteligência Artificial não estejam sendo aplicados com bons resultados e que não venham a predominar com a evolução das pesquisas na área de controle inteligente.

A concepção de que progressos ainda podem ser obtidos em aplicações de redes neurais a problemas de controle baseados apenas em métodos de engenharia será empregada no desenvolvimento deste trabalho.

2.4. Especificação das principais características de uma rede neural artificial

As redes neurais artificiais a serem utilizadas neste trabalho são baseadas na abordagem conexionista, particularmente nos conceitos de processamento paralelo e distribuído, onde é possível descrever a rede como uma composição de unidades processadoras simples e similares entre si no aspecto funcional (McClelland&Rumelhart-86)[44,45]. É importante salientar que a definição das principais características de uma rede neural artificial está baseada no estudo dos sistemas neurais biológicos, principalmente com relação à composição dos neurônios e à estrutura das interconexões, denominadas sinapses.

Apesar de existirem diversos tipos de neurônios biológicos responsáveis pela execução de uma grande variedade de funções dentro de um mesmo sistema nervoso, é possível distinguir uma estrutura básica para o neurônio, contendo os seguintes componentes:

- uma extensão do corpo celular, denominada axônio, que envia estímulos para outras células nervosas;
- um conjunto de ramificações, denominadas dendritos, que recebem estímulos de outras células nervosas;

- um corpo celular, onde os estímulos recebidos pelos dendritos são processados de forma a produzir um nível de ativação para a célula, a ser propagado na forma de estímulos para outras células por intermédio do axônio.

Muitos modelos de redes neurais artificiais foram desenvolvidos na literatura, diferindo de acordo com as propriedades enfatizadas no processo de modelagem. A abordagem biológica enfatiza as características operacionais dos neurônios e produz os modelos mais complexos. Por outro lado, engenheiros e cientistas da área de computação se preocupam preferencialmente com as capacidades computacionais destas células nervosas. Sendo assim, do ponto de vista computacional, uma rede neural artificial é comumente modelada na forma apresentada na figura 2.5 (Kröse&van der Smagt-93)[37], sendo que os neurônios passam a receber a denominação de unidades processadoras para distinguir da denominação de células nervosas, atribuída aos neurônios biológicos.

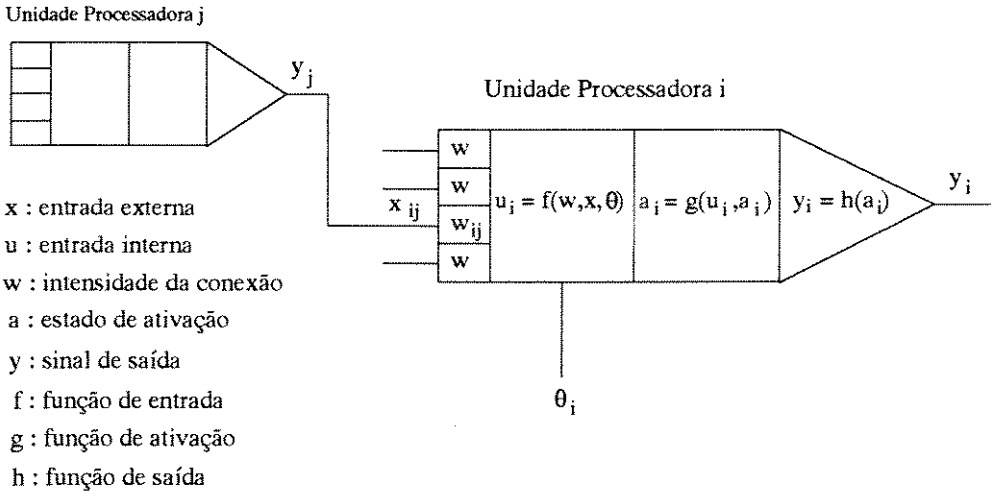


Figura 2.5 - Componentes básicos de uma rede neural artificial

Este é o modelo geral de rede neural artificial a ser adotado no desenvolvimento deste trabalho, sendo que uma discussão detalhada de cada componente, a ser desenvolvida nas próximas seções, permitirá a definição do modelo de rede neural resultante para aplicação.

2.4.1. Unidades processadoras

As unidades processadoras realizam uma seqüência de operações básicas, sendo que será adotado o seguinte critério na sua descrição (veja figura 2.5):

- **i**: índice da unidade processadora em estudo
- **j**: índice das unidades processadoras que enviam sinais (entradas externas) para a unidade processadora **i**

Descrição das operações básicas de uma unidade processadora:

- sinais de entrada x_{ij} são agregados de acordo com uma função de entrada **f**, considerando a intensidade de cada conexão w_{ij} e a entrada externa de polarização θ_i, produzindo a entrada interna u_i na forma:

$$u_i=f(x_{ij},w_{ij},\theta_i)$$

- a entrada interna u_i, juntamente com o estado atual de ativação da unidade $a_{i_{atual}}$, é submetido a uma função de ativação **g** que, por sua vez, define o próximo estado de ativação $a_{i_{prox}}$ na forma:

$$a_{i_{prox}} = g(u_i,a_{i_{atual}})$$

- o sinal de saída y_i é obtido submetendo-se este novo estado de ativação $a_{i_{prox}}$ à função **h** na forma:

$$y_i = h(a_{i_{prox}})$$

Esta saída é, então, propagada às outras unidades através das conexões da rede. É possível observar que o processamento pode ser paralelo, já que todas as unidades podem operar simultaneamente para produzir seus respectivos sinais de saída.

2.4.2. Conexões entre as unidades

A característica essencial das redes neurais artificiais é o alto grau de conectividade entre suas unidades processadoras. A intensidade de conexão é denominada peso, o qual pode assumir valores positivos ou negativos, representando a propriedade de conexão excitadora ou inibidora, respectivamente.

Geralmente cada conexão é definida por um peso w_{ij} que determina o efeito que o sinal de saída da unidade **j** (y_j) exercerá sobre a unidade **i**. Neste caso, por convenção, o sinal de saída y_j é renomeado para x_{ij} e passa a ser denominado entrada externa da unidade **i** (veja figura 2.5).

2.4.3. Regra de propagação

A entrada interna u_i da unidade processadora **i**, é determinada por meio de uma regra de propagação, também denominada função de entrada **f**, que recebe como parâmetros as entradas externas x_{ij}, os pesos das respectivas conexões w_{ij} e o valor de polarização θ_i. Na maioria dos casos, é assumido que a entrada interna é obtida da soma das entradas externas ponderadas pelos pesos das respectivas conexões, além da adição do valor de polarização, de tal forma que:

$$u_i=f(x_{ij},w_{ij},\theta_i)=\sum_{j=1}^{n} w_{ij}x_{ij}+\theta_i$$

onde **n** é o número de unidades processadoras que enviam sinais para à unidade processadora **i**.

Um procedimento comum e totalmente análogo é substituir a entrada externa de polarização θ_i por uma entrada externa constante e de valor unitário ($x_{i0}=1$), onde o peso desta nova conexão assume o valor de θ_i ($w_{i0}=\theta_i$). Com isso, a entrada interna u_i passa a ser expressa na forma:

$$u_i = \sum_{j=0}^{n} w_{ij} x_{ij}$$

2.4.4. Função de Ativação

A função de ativação representa o efeito que a entrada interna e o estado atual de ativação exercem na definição do próximo estado de ativação da unidade. Quando propriedades dinâmicas estão envolvidas na definição do estado de ativação, equações diferenciais (caso contínuo) ou a diferenças (caso discreto) são empregadas. Tendo em vista a simplicidade desejada para as unidades processadoras, geralmente define-se seu estado de ativação como uma função algébrica da entrada interna atual, independente de valores passados do estado de ativação ou mesmo da entrada interna. Geralmente, esta função é monotonicamente não-decrescente e apresenta algum tipo de descontinuidade ou outros tipos especiais de não-linearidade.

O tipo de função de ativação empregado vai determinar se o estado de ativação assume valores discretos ou contínuos. No caso discreto, geralmente se utilizam funções de ativação binárias (por exemplo, função sinal), enquanto que no caso contínuo, além de serem não-lineares e monotonicamente não-decrescentes, as funções de ativação geralmente assumem valores em um intervalo finito do eixo dos números reais. A seguir, são descritos alguns exemplos de função de ativação empregados na literatura (Kosko-92)[35].

Função sinal

É a função de ativação mais empregada no caso discreto, sendo definida na forma[1]:

$$a_i(u_i) = \begin{cases} 0 & \text{se } u_i < 0 \\ [0,1] & \text{se } u_i = 0 \\ 1 & \text{se } u_i > 0 \end{cases}$$

Com relação ao caso $u_i=0$, é conveniente arbitrar o valor de a_i a um dos extremos do intervalo [0,1]. Neste caso, a ativação da unidade processadora é binária.

[1]Esta função já foi utilizada nas configurações de redes neurais apresentadas nas figuras 2.1, 2.2 e 2.3, tendo recebido a denominação de função **g**.

Deve-se observar, também, que a função sinal é linear por partes, e apresenta um descontinuidade em $\mathbf{u}_i$=0, onde sua derivada assume valor infinito.

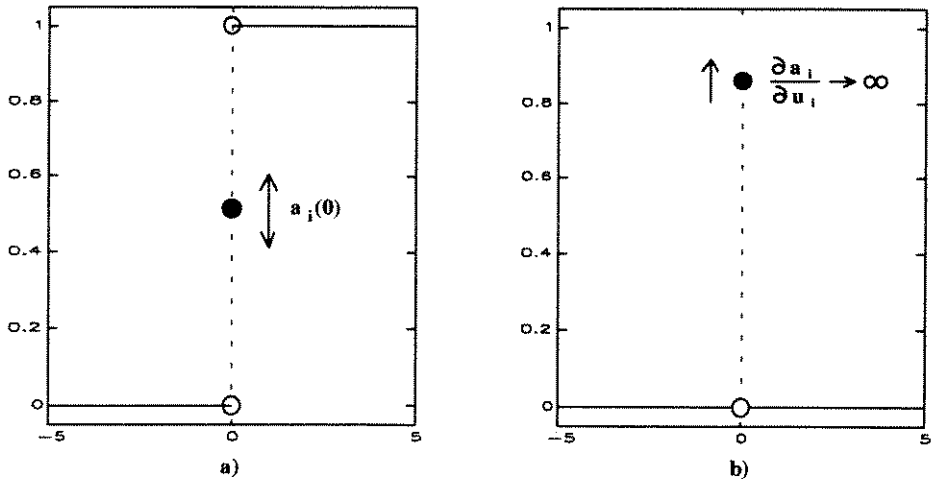


Figura 2.6 - Função sinal (a) e sua derivada em relação à entrada interna (b)

Função semi-linear

A descontinuidade da função sinal é um fator indesejável em aplicações que fazem uso do valor de sua derivada em relação à entrada interna, a qual pode assumir valor infinito. Para limitar a amplitude desta derivada a valores finitos, é comum utilizar-se a função semi-linear, descrita na forma:

$$\mathbf{a}_i(\mathbf{u}_i) = \begin{cases} 1 & \text{se} \quad p\mathbf{u}_i \geq 1 \\ p\mathbf{u}_i & \text{se} \quad 0 < p\mathbf{u}_i < 1 \\ 0 & \text{se} \quad p\mathbf{u}_i \leq 0 \end{cases} \qquad \text{com p constante e positivo.}$$

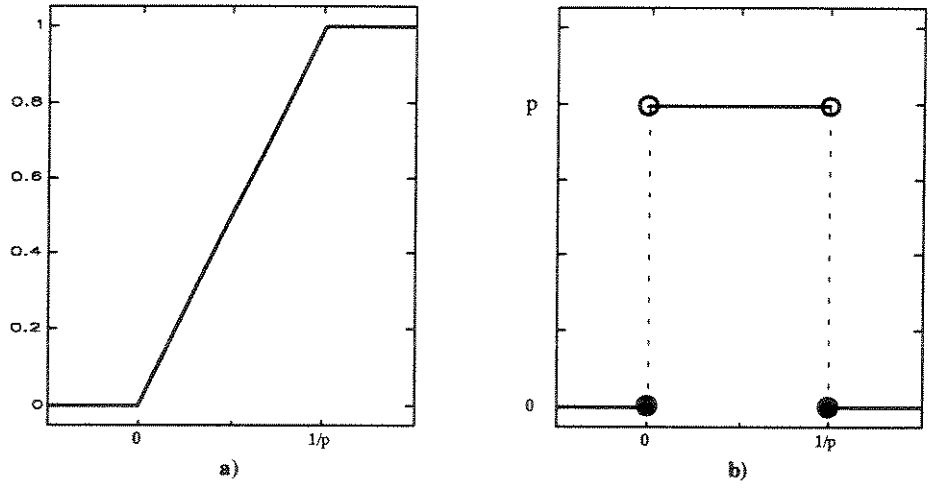


Figura 2.7 - Função semi-linear (a) e sua derivada em relação à entrada interna (b)

Observe que quando $p \to \infty$ a função semi-linear se aproxima da função sinal.

Função Logística

É a função de ativação mais utilizada no caso contínuo, sendo definida na forma:

$$\mathbf{a}_i(\mathbf{u}_i) = \frac{e^{p\mathbf{u}_i}}{e^{p\mathbf{u}_i} + 1} = \frac{1}{1 + e^{-p\mathbf{u}_i}}$$

onde p representa uma constante positiva. Uma informação importante é o valor da derivada parcial em relação à entrada interna, dada na forma:

$$\frac{\partial \mathbf{a}_i}{\partial \mathbf{u}_i} = p\mathbf{a}_i(1 - \mathbf{a}_i) > 0$$

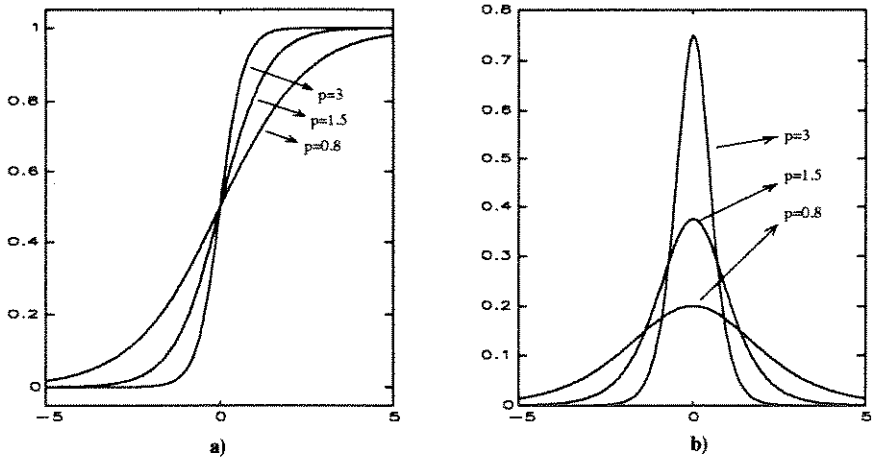


Figura 2.8 - Função logística (a) e sua derivada em relação à entrada interna (b)

Observe que para valores de $p \to \infty$ a função logística se proxima da função sinal. Na realidade, a origem deste tipo de função também está vinculada à preocupação em limitar o intervalo de variação da derivada da função sinal, mas, ao contrário da função semi-linear, esta derivada também é uma função contínua (veja figura 2.8b). Pelo fato de garantir a continuidade da derivada, a função logística é a primeira dentre as funções de ativação já apresentadas a possuir não-linearidades suaves.

Função tangente hiperbólica

Pelo fato da função logística apresentar apenas valores de ativação positivos ($\mathbf{a}_i \in]0,1[$), em muitos casos ela é substituída pela função tangente hiperbólica, que preserva a forma sigmoidal da função logística, mas assume valores positivos e negativos ($\mathbf{a}_i \in]-1,1[$). A função tangente hiperbólica e sua derivada em relação à entrada interna são dadas na forma:

$$\mathbf{a}_i(\mathbf{u}_i) = \tanh(p\mathbf{u}_i) = \frac{e^{p\mathbf{u}_i} - e^{-p\mathbf{u}_i}}{e^{p\mathbf{u}_i} + e^{-p\mathbf{u}_i}} \qquad \text{e} \qquad \frac{\partial \mathbf{a}_i}{\partial \mathbf{u}_i} = p\left(1 - \mathbf{a}_i^2\right) > 0$$

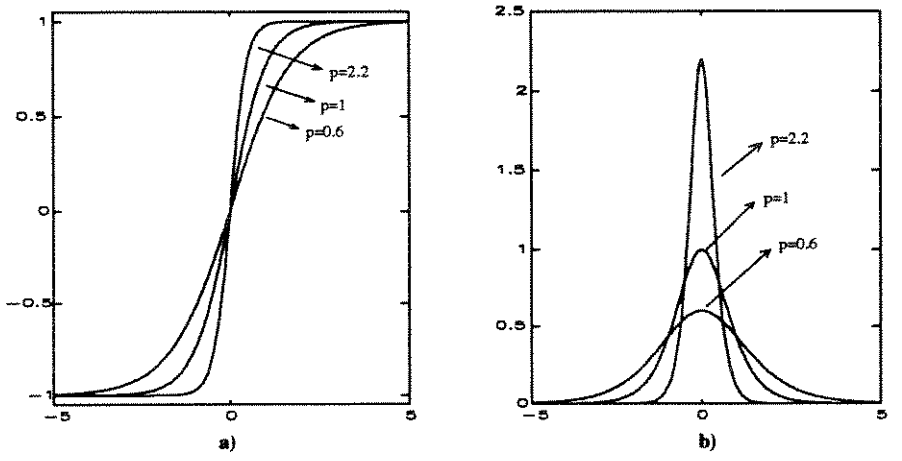


Figura 2.9 - Função tangente hiperbólica (a) e sua derivada em relação à entrada interna (b)

A partir daqui, em todas as redes neurais a serem apresentadas, se a ativação das unidades processadoras for binária, será utilizada a função sinal; e se a ativação for contínua, será utilizada a função tangente hiperbólica com p=1.

2.4.5. Função de saída

Em todas as redes neurais a serem utilizadas no desenvolvimento deste trabalho, o sinal de saída será considerado idêntico ao estado de ativação da unidade, ou seja, $\mathbf{y_i=h(a_i)=a_i}$ com **h** igual à função identidade.

2.4.6. Unidade processadora resultante

A figura 2.10, a seguir, apresenta o resultado das especificações descritas nas subseções precedentes com relação ao esquema geral dos componentes básicos de uma rede neural artificial (figura 2.5). Pode-se observar que a unidade processadora resultante é idêntica ao perceptron (figura 2.2).

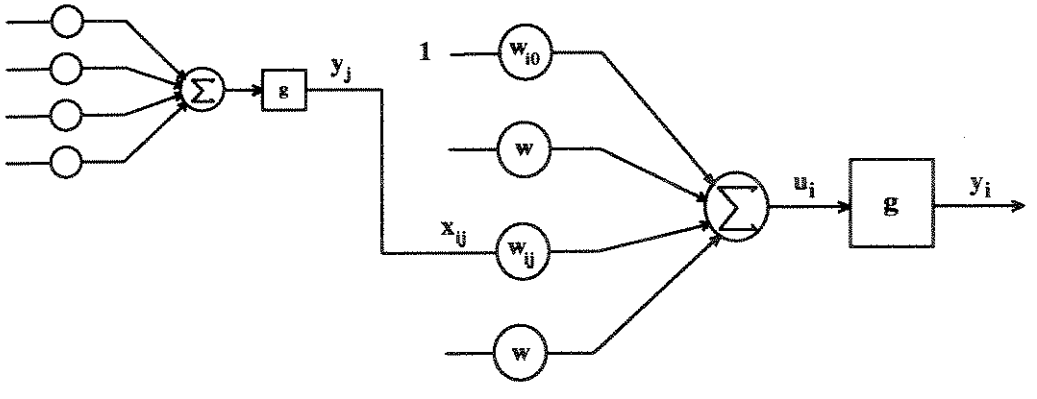


Figura 2.10 - Rede neural artificial - componentes resultantes

2.4.7. Topologias básicas

De acordo com as conexões existentes, é possível identificar três tipos de topologias para as redes neurais artificiais:

1) Redes totalmente interconectadas
2) Redes não-recorrentes em camadas
3) Redes recorrentes em camadas

O primeiro tipo representa a topologia onde cada unidade se conecta com ela própria e com todas as outras unidades da rede. O segundo engloba as topologias onde as unidades estão distribuídas em camadas ou níveis, existindo ligações entre as saídas de unidades de um nível inferior para as entradas de todas as unidades (ou um grupo de unidades) de níveis superiores, e não existindo conexões entre os elementos de um mesmo nível. O terceiro tem a mesma estrutura do segundo tipo, mas com a possibilidade de saídas de unidades de um nível superior estabelecerem conexões com entradas de todas as unidades (ou um grupo de unidades) de um nível inferior àquele, representando uma realimentação interna de informação, geralmente realizada com o auxílio de operadores de atraso de tempo.

Estes três tipos de topologia permitem a construção de redes neurais extremamente complexas, abrangendo a quase totalidade dos sistemas neurais conhecidos e experimentados até hoje. Neste trabalho, serão utilizadas apenas arquiteturas de redes neurais em camadas, sendo importante distinguir três tipos de unidades processadoras:

- **unidades de entrada**: podem receber sinais do mundo exterior
- **unidades de saída**: enviam sinais para o mundo exterior
- **unidades ocultas**: os sinais de entrada e saída permanecem internos à rede.

2.4.8. Critérios de processamento

Pode-se ainda definir redes neurais com processamento contínuo ou discreto no tempo, dependendo do comportamento temporal das funções em cada unidade. Em termos computacionais, é interessante a definição de instantes preferenciais para a execução dessas funções, ou seja, a utilização de redes neurais que operam em instantes discretos no tempo. Neste último caso, pode-se optar por duas políticas básicas para definir os instantes de execução das funções nas células:

- **execução síncrona**: quando todas as unidades entram em operação no mesmo instante de tempo.
- **execução assíncrona**: quando cada unidade tem associada uma função de probabilidade para entrar em operação em cada instante de tempo. Este modo de execução tem maior analogia com os processos biológicos reais, porém acarreta maiores dificuldades nas implementações feitas em computadores digitais.

Neste trabalho, serão adotadas redes neurais com processamento discreto e execução síncrona.

2.5. A representação do conhecimento em redes neurais artificiais

As redes neurais artificiais utilizam a estrutura de suas interconexões para representar o conhecimento. Isto significa que, com base na configuração resultante de rede neural artificial a ser utilizada neste trabalho (figura 2.10), a intensidade das conexões é ajustada com o objetivo de armazenar informação. Como um resultado natural deste processo de ajuste, o conhecimento armazenado fica distribuído pela rede neural. Esta é uma das mais importantes diferenças entre processadores de informação baseados em redes neurais e aqueles baseados nos princípios da máquina de Von Neumann (58)[68], como no caso dos computadores digitais, nos quais o conhecimento é particionado em células de informação (unidade mais simples de conhecimento) que possam ser armazenadas em posições de memória bem definidas. É importante observar que todo conhecimento passível de representação em máquinas de Von Neumann pode também ser adequadamente representado em redes neurais artificiais, por exemplo, considerando-se cada peso da rede como um posição de memória onde as células de informação podem ser armazenadas. Portanto, o tipo de representação de conhecimento empregado no caso das redes neurais artificias fornece a estes sistemas potencialidades de representação ausentes em outros sistemas tradicionais de representação de conhecimento.

Por exemplo, um problema de representação de um sistema linear cuja estrutura paramétrica é conhecida, pode ser resolvido aplicando-se um método convencional de estimação de parâmetros, sendo reservada a cada parâmetro uma posição de memória a ser devidamente preenchida pelo resultado da estimação. Neste caso, existe uma correspondência direta entre as posições de memória e as células de informação a serem armazenadas. No entanto, em problemas de representação mais complexos, as células mínimas de informação não podem ser adequadamente decompostas, exigindo a utilização de estruturas de armazenagem mais poderosas, justificando a utilização de redes neurais artificiais.

A capacidade de representação das redes neurais artificiais geralmente excede a dimensão paramétrica do problema. Com isso, a relação existente entre os pesos da rede neural e as células de informação a serem armazenadas perdem a correspondência direta e ganham a característica de distributividade, onde o conhecimento está distribuído pelas conexões da rede, sendo que cada conexão pode participar na representação de mais de uma célula de informação. Ao contrário do caso de máquinas de Von Neumann, onde a perda ou falha de uma única posição de memória pode acarretar a falha de todo o sistema, em representações baseadas em redes neurais, a sensibilidade à perda ou falha de uma conexão é muito baixa.

2.6. Aprendizagem em redes neurais

A aquisição de conhecimento em redes neurais artificiais se dá pela aplicação de métodos de aprendizagem baseados na modificação do estado da rede através de uma política de ajuste de seus parâmetros e funções. O desenvolvimento de métodos de aprendizagem para a rede neural apresentada na figura 2.10, como também para boa parte das redes neurais que apresentam processamento predominantemente numérico, envolve o tratamento de conceitos muito difundidos na aplicação de métodos de estimação de parâmetros, procedimento fundamental em áreas como classificação de padrões, processamento de sinais e controle adaptativo (Widrow&Stearns-85)[76]; (Ästrom &Wittenmark-89)[3].

Por exemplo, uma maneira de se estruturar um classificador de padrões é assumir que a regra de decisão pertence a uma classe específica de regras ou modelos e que cada regra é selecionada a partir da definição de valores para um conjunto específico de parâmetros. Em controle e processamento de sinais, o que geralmente se procura é um modelo capaz de produzir valores numéricos compatíveis com valores medidos a partir de sistemas físicos. Geralmente se assume que o modelo tem uma estrutura matemática definida e expressa em termos de um conjunto de parâmetros a determinar. Métodos de estimação, neste caso, devem produzir valores adequados para estes parâmetros, a partir de dados na forma de exemplos de como se comporta o sistema a ser modelado.

As técnicas convencionais de estimação impõem fortes restrições quanto à natureza do sistema a ser controlado, além de considerarem uma dependência linear com relação às variáveis básicas do sistema e seus respectivos retardos no tempo. Métodos conexionistas de identificação, por sua vez, operam em sistemas de diferentes graus de complexidade, através da utilização de representações de dimensão maior que a dimensão intrínseca do sistema e da utilização de um grande número de operadores não-lineares similares entre si. Este tipo de flexibilidade de representação, muitas vezes atribuída à analogia com propriedades presentes em sistemas neurais biológicos, é uma das principais características dos sistemas conexionistas a ser explorada no desenvolvimento de métodos de aprendizagem eficientes.

Nas próximas seções, serão analisados os tópicos essenciais vinculados ao problema de aprendizado em redes neurais artificiais.

2.6.1. Estrutura do modelo

A capacidade de representação de redes neurais artificiais varia de acordo com sua estrutura, que, por sua vez, resulta da definição dos componentes básicos da rede neural, já descritos em seções anteriores. Com isso, embora a aplicação de métodos conexionistas envolva predominantemente sistemas do qual pouco conhecimento possa ser

extraído, a definição da estrutura da rede neural a ser empregada representa a introdução de conhecimento prévio junto ao processo de aprendizagem. Geralmente, a imposição de restrições permite uma melhor estruturação e conseqüente redução da complexidade do problema, aumentando a eficiência do processo de aprendizagem e, muitas vezes, viabilizando sua aplicação.

2.6.2. Critério de desempenho

Quando diferentes representações ou então várias estruturas de modelos são aplicadas a um mesmo problema, é necessário definir-se um critério de desempenho a fim de identificar as formulações mais adequadas. O problema principal está na escolha do melhor critério. Geralmente o que se procura é um critério numérico que apresente uma forte correlação com o desempenho global do sistema e, comparado ao processo de aprendizagem, exija uma carga de processamento reduzida.

2.6.3. O escopo do processo de aprendizagem

Quando se estuda a capacidade de aprendizagem de redes neurais é importante qualificar-se o escopo deste processo. Por exemplo, poder-se-ia admitir uma rede que fosse capaz de alterar sua própria topologia, ou ainda de modificar as funções de suas unidades processadoras, ou simplesmente alterar a intensidade de suas interconexões. A grande diversidade de escopos possíveis para o processo de aprendizagem e o evidente crescimento da complexidade para situações mais abrangentes impõem fortes restrições quanto à capacidade da rede em ajustar um conjunto de seus parâmetros e funções. Portanto, não devem ser criadas expectativas exageradas quanto à abrangência do processo de aprendizagem em redes neurais artificiais, ao menos no estágio atual de desenvolvimento.

Neste trabalho, serão aplicados processos de aprendizagem capazes de alterar apenas a intensidade das interconexões (pesos) da rede neural.

2.6.4. Paradigmas de aprendizado

A princípio, é possível distinguir dois grandes paradigmas de aprendizado:

- Aprendizado supervisionado ou associativo;
- Aprendizado não-supervisionado, auto-organizado ou autônomo.

Os qualificativos supervisionado e não-supervisionado são procedentes da teoria de reconhecimento de padrões. A distinção entre os dois paradigmas de aprendizado está baseada na disponibilidade ou não para o algoritmo de aprendizado das respostas corretas às situações de entrada apresentadas. No caso supervisionado, enquadram-se os processos que utilizam tal informação, enquanto que no caso não-supervisionado são incluídos os processos que não a utilizam.

No caso dos algoritmos de aprendizado supervisionado aplicados a problemas de reconhecimento de padrões, a resposta correta pode significar a classe a que o padrão de entrada pertence. Desta forma, é possível detectar erros de classificação que vão comandar o processo de aprendizado. Um procedimento comum é aplicar alguma política de recompensa para as classificações corretas e punição para as classificações incorretas. Em termos matemáticos, os erros de classificação geram sinais de erro numéricos que irão determinar uma direção de ajuste para a intensidade das interconexões na rede neural.

Já no caso dos algoritmos de aprendizado não-supervisionado aplicados a problemas de reconhecimento de padrões, é realizado basicamente um agrupamento dos padrões de forma a estabelecer classes de decisão. Neste caso, não existe a definição de erro, sendo que o processo de aprendizado é conduzido pela definição de políticas de competição entre as unidades processadoras. Como competição entende-se uma interação entre as unidades de forma a estabelecer núcleos de ativação quando da presença de um determinado padrão de entrada. Assim, de acordo com as caracterísiticas deste núcleo de ativação (posição, intensidade, dimensão, etc.), os padrões de entrada podem ser adequadamente classificados.

Os dois paradigmas de aprendizado utilizam o conceito de informação local para realizar o aprendizado. No caso supervisionado, o erro deve ser definido localmente, enquanto que no caso não-supervisionado, a competição se estabelece entre unidades processadoras que obedecem a algum critério de proximidade. Dessa forma, um aspecto importante do aprendizado em redes neurais é fazer com que a rede apresente um determinado comportamento global através da implementação local do aprendizado (Barto-89)[5].

Os algoritmos de aprendizado não-supervisionado geralmente exigem um carga de processamento menor, mas apresentam uma precisão numérica inferior ao caso supervisionado. O critério de escolha entre os dois paradigmas de aprendizado vai depender da aplicação. A utilização do aprendizado não-supervisionado é mandatória em aplicações onde não existe tempo, informação e precisão computacional suficientes para a utilização de métodos supervisionados.

Nas aplicações desenvolvidas neste trabalho, serão estabelecidas as condições mínimas para a aplicação de algoritmos de aprendizado supervisionado.

2.6.5. Aprendizagem por ajuste da intensidade das conexões

De acordo com o escopo definido, o processo de aprendizado a ser adotado restringe-se ao ajuste da intensidade das conexões (pesos) entre os neurônios. Em sua grande maioria, os métodos são baseados na regra de aprendizado proposta por Hebb em 1949 [29], pela qual se duas unidades **i** e **j** têm saídas $\mathbf{y_i}$ e $\mathbf{y_j}$ de mesmo sinal, ou seja, ambas inibidoras (sinal negativo) ou excitadoras (sinal positivo), os pesos de suas conexões devem ser aumentados. Caso contrário, os pesos devem ser reduzidos. Algebricamente, esta regra

pode ser expressa na forma:

$$\Delta w_{ij} = \lambda\, y_i\, y_j \qquad \lambda > 0 : \text{fator de proporcionalidade}$$

Uma das mais importantes variações utilizadas é o algoritmo de Widrow-Hoff, desenvolvido em 1960 para aplicações a sistemas adaptativos (Widrow&Hoff-60)[74]; (Widrow&Lehr-90)[75]. Este algoritmo, também denominado **regra delta**, pode ser expresso na forma:

$$\Delta w_{ij} = \lambda\, (d_i - y_i)\, y_j \qquad d_i\text{: valor correto ou desejado para a saída } y_i$$

2.7. O estudo de redes neurais como mapeadores não-lineares

O problema do OU-exclusivo tem uma importância histórica no estudo de redes neurais, pois foi utilizado por Minsky e Papert (69)[48] como um exemplo da incapacidade de neurônios do tipo perceptron (veja figura 2.2) realizarem mapeamentos não-lineares. Este problema será discutido na seqüência, e utilizado como ponto de partida para justificar a utilização de redes neurais multi-camadas na realização de mapeamentos não-lineares, muito comuns em problemas de reconhecimento de padrões, aproximação de funções e identificação de sistemas.

2.7.1. O problema do OU-exclusivo

Considere os pontos (0,0),(0,1),(1,0) e (1,1) no plano $\Re^2$, conforme apresentado na figura 2.11. O objetivo é determinar uma rede com duas entradas $x_i \in \{0,1\}$ (i=1,2), e uma saída $y \in \{0,1\}$ de maneira que:

$$\begin{cases} (x_1, x_2) = (0,0) \text{ ou } (1,1) \text{ implica em } y = 0 \\ (x_1, x_2) = (1,0) \text{ ou } (0,1) \text{ implica em } y = 1 \end{cases}$$

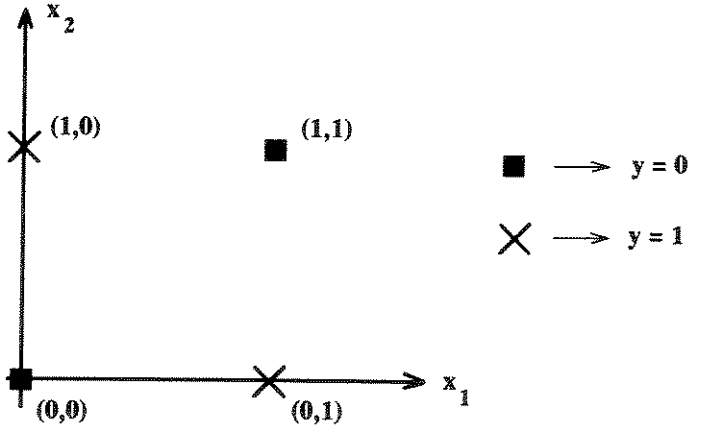


Figura 2.11 - O problema do OU-exclusivo

Inicialmente será analisado o comportamento de um perceptron de uma camada (veja figura 2.12) no processo de solução do problema exposto acima. A saída **y** pode ser representada na forma:

$$\mathbf{y} = \mathbf{g}(\mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 + \mathbf{w}_0) \qquad \text{onde} \begin{cases} \mathbf{g}(\mathbf{u}) = 1 & \text{se } \mathbf{u} > 0 \\ \mathbf{g}(\mathbf{u}) = 0 & \text{se } \mathbf{u} \leq 0 \end{cases}$$

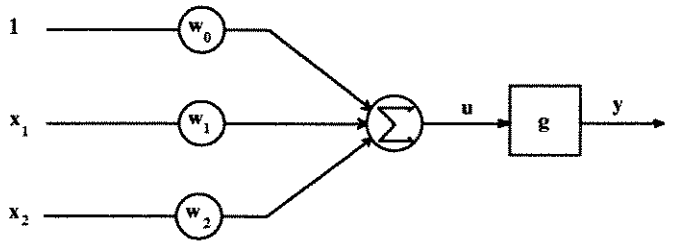


Figura 2.12 - Perceptron de uma camada com duas entradas

É evidente que para qualquer valor dos parâmetros $\mathbf{w}_0$, $\mathbf{w}_1$ e $\mathbf{w}_2$ a função $\mathbf{g}(\mathbf{u})$ separa o espaço de entrada em duas regiões e a curva de separação é uma linha reta. No problema do OU-exclusivo dado, pode-se constatar que não existe uma única linha reta divisória de forma que os pontos (0,0) e (1,1) se posicionem de um lado enquanto que (0,1) e (1,0) permaneçam do outro lado. Logo, pode-se imediatamente concluir que um perceptron de uma única camada não apresenta grau de liberdade suficiente para resolver o problema proposto, o que foi corretamente constatado por Minsky e Papert em seu livro "Perceptrons".

No entanto, esses autores também acreditavam que não havia razão para supor que redes multi-camadas pudessem conduzir a uma solução para o problema proposto, tanto que introduziram esta afirmação em seu livro como uma hipótese a ser adequadamente comprovada ou rejeitada. Esta hipótese só foi definitivamente rejeitada com o desenvolvimento do algoritmo de retro-propagação, já nos anos 80, o qual permite o ajuste automático de pesos para redes neurais multi-camadas, arquitetura necessária para a realização de mapeamentos não-lineares, como será verificado no desenvolvimento que se segue. Em 1988, foi publicada a 3ª edição do livro "Perceptrons" (Minsky&Papert-88)[47]; (Pollack-88)[59], onde os autores procuram rejeitar a idéia de que as afirmações contidas na 1ª edição (Minsky&Papert-69)[48] provocaram a interrupção da pesquisa na área de aprendizagem em redes neurais por aproximadamente uma década. Segundo Minsky e Papert, a área de redes neurais já carecia, à época, de fundamentos teóricos adequados para sustentar a obtenção de novos resultados, sendo que suas afirmações apenas contribuíram para difundir esta idéia.

Considere, agora, o problema de mapeamento de um perceptron de duas camadas (veja figura 2.13) aplicado ao problema do OU-exclusivo.

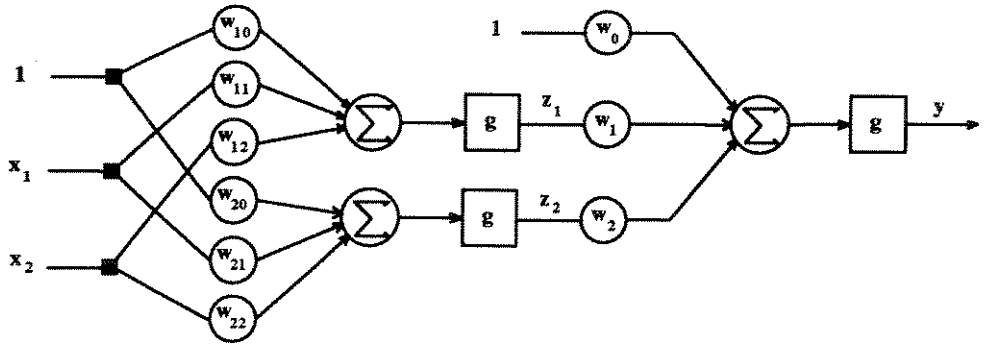


Figura 2.13 - Perceptron de duas camadas

A primeira camada do perceptron consiste de um vetor de entrada (x_1,x_2) e duas saídas $z_1=\text{sgn}(w_{10}+w_{11}x_1+w_{12}x_2)$ e $z_2=\text{sgn}(w_{20}+w_{21}x_1+w_{22}x_2)$. Na segunda camada, o vetor de entrada é dado por (z_1,z_2) e o sinal de saída é dado por $y=\text{sgn}(w_0+w_1z_1+w_2z_2)$.

A questão que surge é se existem parâmetros w_{ij} (i,j = 0,1,2) e w_k (k = 0,1,2) tais que $y = 0$ para as entradas (0,0) e (1,1) e $y = 1$ para as entradas (1,0) e (0,1). As saídas da primeira camada (z_1 e z_2) podem ser consideradas como variáveis intermediárias utilizadas na geração da saída y.

Do que já foi visto a respeito do perceptron de uma camada, segue-se que existem pesos w_{1j} (j=0,1,2) tais que (veja curva de separação L_1 na figura 2.14(a)):

$$(0,1) \text{ produza } z_1 = 1$$
$$(0,0),(1,0),(1,1) \text{ produza } z_1 = 0.$$

De forma similar, existem pesos w_{2j} (j=0,1,2) tais que (veja curva de separação L_2 na figura 2.14(a)):

$$(0,1),(0,0),(1,1) \text{ produza } z_2 = 1$$
$$(1,0) \text{ produza } z_2 = 0$$

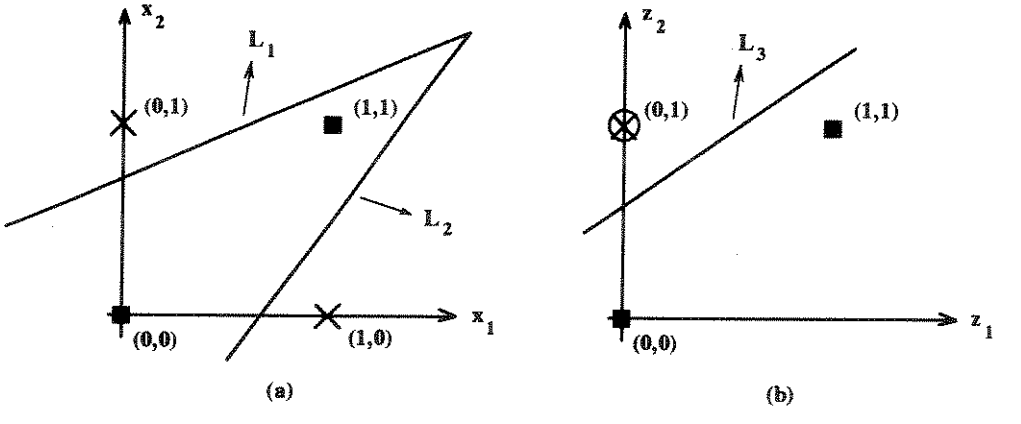


Figura 2.14 - Realização da função OU-exclusivo

A discussão acima mostra que existem pesos w_{ij} (i,j=0,1,2) de maneira que a entrada (0,1) resulte em $z_1 = 1$, $z_2 = 1$, e a entrada (1,0) resulte em $z_1 = 0$, $z_2 = 0$, enquanto que (0,0) e (1,1) produzam $z_1 = 0$, $z_2 = 1$. Já que (0,0) e (1,1) podem ser separados linearmente de (0,1), como mostrado na figura 2.14(b) pela curva de separação L_3, pode-se concluir que a função booleana desejada pode ser obtida utilizando-se perceptrons de camada única conectados em cascata, ou seja, um perceptron de duas camadas. Isto é possível devido à transformação do espaço de entrada (x_1,x_2), onde os padrões não são linearmente separáveis, no espaço (z_1,z_2), onde os padrões são linearmente separáveis.

Em reconhecimento de padrões é bem conhecido que quando classes de padrões podem ser separadas utilizando-se uma função discriminante não-linear, o problema pode ser transformado em um espaço de dimensão maior onde os padrões são linearmente separáveis. Por exemplo, se a curva $a_0+a_1u_1+a_2u_2+a_3u_1u_2 = 0$ no espaço (u_1,u_2) separa duas classes de padrões, então um hiperplano (uma superfície linear) no espaço tridimensional (u_1,u_2,u_1u_2) pode também separá-los. Esta foi a propriedade explorada na solução do problema do OU-exclusivo.

2.7.2. Um problema mais geral de mapeamento não-linear

Considere, agora, um problema mais geral de classificação de padrões em um espaço de dimensão finita. Com base na figura 2.15, assume-se que a região triangular (conjunto **A**) corresponde a uma classe **1**, enquanto que o complemento desta região (conjunto **B**) corresponde a uma classe **2**. O objetivo é determinar os pesos de uma rede neural cuja saída é 1 (simbolizando a classe **1**) quando a entrada $(x_1,x_2)\in A$ e 0 (simbolizando a classe **2**) quando $(x_1,x_2)\in B$.

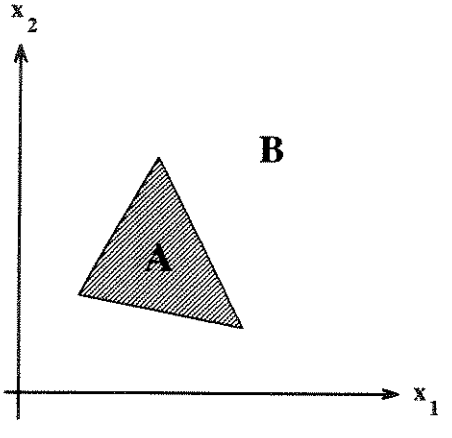


Figura 2.15 - Um problema de reconhecimento de padrões

Partindo das conclusões extraídas da seção anterior, é possível afirmar que cada um dos três segmentos de reta que delimitam a região A pode ser representado por um perceptron de uma camada. Tomando-se a função booleana "AND" das saídas destes três

perceptrons, a saída **y** pode ser feita **1** quando $(x_1,x_2)\in\mathbf{A}$ e **0** quando $(x_1,x_2)\in\mathbf{B}$. A estrutura desta rede é apresentada na figura 2.16.

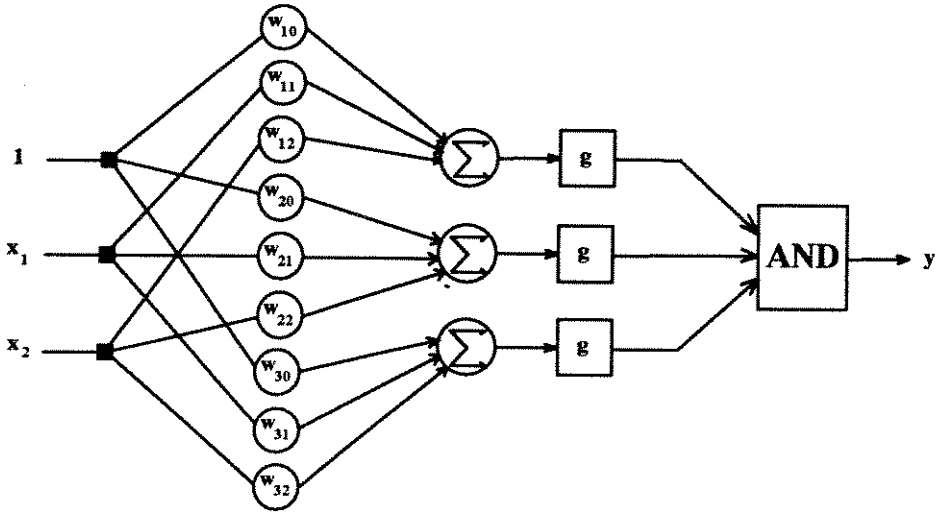


Figura 2.16 - Estrutura da rede para o reconhecimento de padrões

A mesma abordagem pode ser estendida para casos onde o conjunto **A** é limitado por um polígono convexo. O número de "perceptrons" de uma camada necessários na primeira camada da rede é, neste caso, igual ao número de lados do polígono.

Se $\mathbf{A_1}$, $\mathbf{A_2}$ e $\mathbf{A_3}$ são conjuntos desconexos, cada qual limitado por um polígono convexo e o conjunto $\mathbf{A} = \mathbf{A_1}\cup\mathbf{A_2}\cup\mathbf{A_3}$ corresponde à classe **1** e o complemento de **A** (conjunto **B**) corresponde à classe **2**, a mesma abordagem descrita acima pode ser utilizada para reconhecer as duas classes. Neste caso, três redes idênticas àquela apresentada na figura 2.16 são utilizadas. As saídas y_1, y_2 e y_3 das três redes são tomadas como entrada de uma função booleana "OR", cuja saída é 1 (simbolizando classe **1**) quando $(x_1,x_2)\in\mathbf{A}$ e 0 (simbolizando classe **2**) quando $(x_1,x_2)\in\mathbf{B}$.

Com base nos resultados obtidos acima, e levando-se em conta que as funções booleanas "AND" e "OR" podem ser executadas através de um único neurônio do tipo perceptron, conclui-se que um perceptron de três camadas pode executar a tarefa de reconhecer elementos de conjuntos desconexos quando os conjuntos são limitados por segmentos lineares. Situações mais gerais, onde as regiões não são convexas podem também ser adequadamente mapeadas utilizando-se um perceptron de três camadas, já que qualquer região não-convexa pode ser representada como uma união de regiões convexas. No entanto, a seleção automática dos pesos para redes com múltiplas camadas não é uma tarefa elementar. A principal dificuldade provém da natureza descontínua da função de ativação **g** utilizada (função sinal). Uma solução para este problema será proposta quando for apresentado o algoritmo de treinamento para redes neurais multi-camadas.

2.7.3. A capacidade de aproximação universal de redes neurais artificiais

Na seção anterior, verificou-se que redes neurais de três camadas, compostas por unidades processadoras com função de ativação do tipo sinal, são capazes de realizar qualquer tipo de mapeamento não-linear que possa ser representado por uma seqüência de segmentos lineares. No entanto, muitos são os exemplos de mapeamentos não-lineares que não podem ser adequadamente descritos por seqüências de segmentos lineares e, portanto, não podem ser realizados por este tipo de rede neural. Por exemplo, considere o problema de classificação de padrões pertencentes a duas classes distintas, separáveis por uma circunferência, conforme apresentado na figura 2.17.

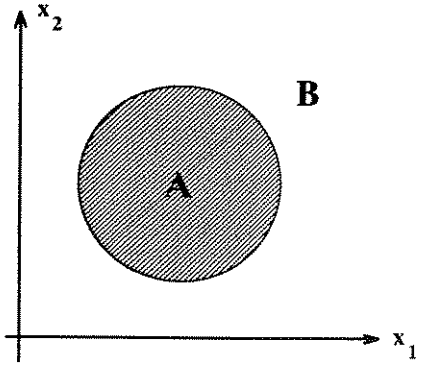


Figura 2.17 - Padrões separáveis por uma circunferência

Se for utilizada uma rede neural com função de ativação do tipo sinal, a melhor solução que pode ser obtida é uma aproximação da circunferência por trechos de segmento de reta. A estrutura da rede neural é a mesma apresentada na figura 2.16, sendo que quanto maior o número de neurônios na primeira camada (camada de entrada), melhor será a aproximação. Esta relação é indesejável, pois cria uma dependência linear entre a dimensão da rede e a capacidade de mapeamento.

A conclusão que se pode extrair é que, no caso das redes neurais em camadas já apresentadas, funções de ativação lineares por partes, como é o caso da função sinal, vão conduzir a mapeamentos lineares por partes. Portanto, existe aparentemente uma correspondência muito forte entre o tipo de não-linearidade da função de ativação e a capacidade de mapeamento de redes neurais artificiais. Se esta correspondência se aplicar a todos os tipos de função de ativação, fica bastante comprometida a utilização de redes neurais em problemas de mapeamento, pois passa a ser necessário conhecer exatamente a característica do mapeamento para que se possa definir a função de ativação a ser utilizada, ou seja, é preciso aplicar a solução do problema na composição da rede neural que, a princípio, iria solucionar o problema.

Felizmente, devido à estruturação da rede em camadas, o processamento dos sinais pela rede faz com que os sinais de saída sejam uma composição de funções de

ativação. Tomando como base métodos de aproximação universal para funções não-lineares, como séries de Taylor (composição de funções polinomiais) e séries de Fourier (composição de funções trigonométricas), não deve surpreender o fato de que redes neurais cujas funções de ativação são dotadas de algum tipo de não-linearidade sejam capazes de realizar qualquer tipo de mapeamento entre os espaços de entrada e saída. Esta propriedade é fundamental para viabilizar a aplicação de redes neurais a problemas de mapeamento, sendo que sua demonstração pode ser encontrada no Apêndice A.

2.8. Treinamento de redes neurais artificiais

Uma forma de se viabilizar o processo de ajuste automático dos pesos, ou seja, o processo de treinamento de redes neurais artificiais, é utilizar uma função de ativação contínua e diferenciável em lugar da função sinal. Este tipo de função de ativação permite a obtenção da sensibilidade dos sinais de saída da rede em relação a cada peso de qualquer camada, como será verificado na seqüência. Com a consideração adicional de não-linearidade para a função de ativação, é possível estabelecer um comportamento arbitrário para os sinais de saída da rede, ou seja, é possível manipular o potencial de aproximação de redes neurais artificiais de acordo com as necessidades de cada aplicação.

2.8.1. Análise do processo de treinamento supervisionado

O conjunto de dados processados pelo algoritmo de treinamento supervisionado deve satisfazer certas propriedades para que o ajuste de parâmetros produza bons resultados. Por exemplo, em tarefas de identificação, o sistema a ser identificado deve ser submetido a um conjunto de entradas que revelem todos os modos de atividade do sistema. Uma situação análoga é encontrada no caso de processos convencionais de estimação de parâmetros, onde a condição para que o resultado da estimação seja representativo é a garantia de excitação não-nula do sistema.

Uma outra propriedade do processo de treinamento supervisionado é a influência exercida pelo estado inicial da rede no resultado do treinamento, ou seja, mesmo no caso da garantia de minimização do erro (função custo), o mínimo encontrado pode não ser o mínimo global. Considere, por exemplo, o mapeamento apresentado na figura 2.18, relacionando o estado da rede com a função custo. O processo de minimização da função custo vai produzir resultados diferentes para os estados iniciais dados por **A** e **B**, sendo que o mínimo global só é obtido caso o estado inicial seja dado por **B**.

Outra propriedade importante é a possibilidade da existência de descontinuidades no processo de ajuste de pesos, provocando alterações significativas no procedimento de convergência. Por exemplo, ainda com o auxílio da figura 2.18, pode-se verificar que, mesmo para estados da rede muito próximos, como é o caso dos estados **C** e **D**, o estado de convergência é totalmente diferente.

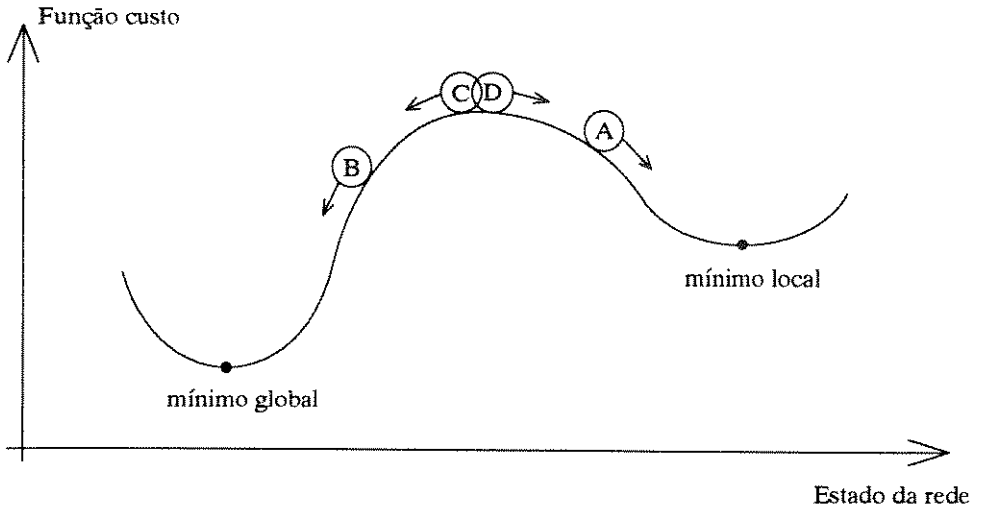


Figura 2.18 - Mapeamento da função custo em relação ao estado da rede neural

Portanto, o resultado do processo de treinamento passa a depender dos valores iniciais dos pesos e dos padrões de treinamento, não sendo possível garantir a convergência para o mínimo global. Entretanto, resultados análogos aos produzidos pelos métodos clássicos de estimação de parâmetros aplicados a sistemas lineares estão sendo obtidos com a aplicação do algoritmo de retro-propagação no processo de ajuste de parâmetros de redes neurais artificiais aplicadas a sistemas não-lineares.

É possível distinguir dois tipos principais de treinamento supervisionado:

- treinamento anterior à operação do sistema ("off-line"): é utilizado quando os dados para treinamento já estão disponíveis.
- treinamento durante a operação do sistema ("on-line"): é utilizado quando os dados para treinamento só estão disponíveis durante o processo de aprendizagem.

Uma comparação entre os dois métodos de treinamento permite afirmar que o treinamento "on-line", ao contrário do caso "off-line", exige a aplicação de técnicas de tempo real, além de geralmente não permitir a existência de nenhum tipo de controle sobre a seqüência e composição dos dados para treinamento.

2.8.2. Representação dos dados para treinamento supervisionado

Nesta seção será discutido o pré-processamento dos dados de entrada e representação de tempo no caso de redes neurais recorrentes e não-recorrentes.

Embora redes neurais artificiais possam realizar qualquer tipo de mapeamento, a forma com que os dados são apresentados à rede pode simplificar a tarefa de representação. Mesmo que não seja possível a realização de nenhum tipo de pré-processamento, há ainda a questão de como os dados serão apresentados à rede neural. Um procedimento padrão é utilizar neurônios sensoriais, ou seja, unidades processadoras cuja tarefa é receber estímulos do mundo exterior e propagá-los pela rede neural.

Para entradas binárias, é comum reservar um neurônio sensorial para cada sinal de entrada, o qual sofre ativação máxima ou mínima, dependendo do valor binário presente no sinal de entrada.

Para entradas contínuas, existem vários métodos de recepção dos sinais de entrada. O procedimento mais comum é normalizar os sinais de entrada de acordo com o intervalo de variação da função de ativação dos neurônios sensoriais. A importância da existência deste intervalo de variação para os valores de entrada está no controle que se pode exercer sobre a região do espaço onde o mapeamento será realizado, ou seja, a normalização pode transladar e escalonar (duas operações lineares) a região de aproximação para pontos de maior flexibilidade de operação por parte da rede neural. Se a função de ativação for a tangente hiperbólica, a qual apresenta um intervalo de variação em torno da origem, então a origem do espaço é o ponto de maior flexibilidade de operação.

Devido à sua analogia com sistemas biológicos, é interessante citar um outro método de recepção de sinais de entrada por parte da rede neural, conhecido por janelas de ativação (Kosko-92)[35]. Neste método, um dado sinal de entrada irá estimular mais de um neurônio sensorial, sendo que cada neurônio sofrerá uma ativação de intensidade diferente (veja exemplo da figura 2.19). As células foto-receptoras existentes no olho humano apresentam este comportamento com relação aos comprimentos de onda da luz.

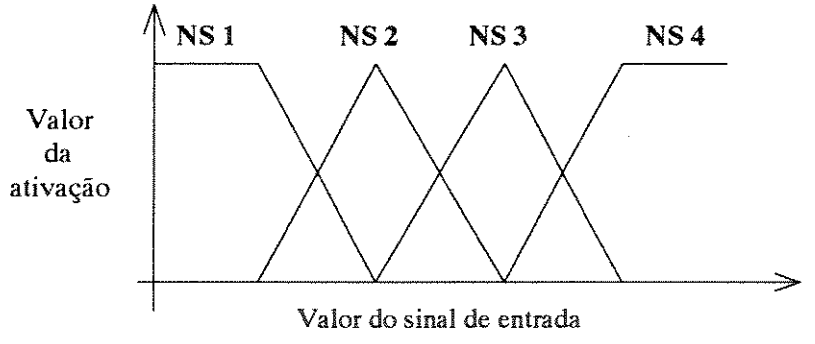


Figura 2.19 - Janelas de ativação (NS = neurônio sensorial)

A representação de tempo, por sua vez, é importante para aplicações em problemas de reconhecimento de voz, processamento de sinais e controle de processos. Os dois principais métodos de representação de tempo, já detectados em sistemas nervosos reais, são:

- expansão espacial do vetor de entrada;
- utilização de recorrência de informação no processamento do sinal.

A expansão espacial utiliza atrasos de tempo para representar a história temporal do sinal de entrada e considera estes sinais defasados no tempo como novos valores de entrada. Neste caso, o vetor de entrada recebe a denominação de linha de derivação de atraso. Redes neurais recorrentes resolvem o problema de representação do fator tempo de uma forma completamente diferente. Neste caso, o tempo é representado

implicitamente por seu efeito no processamento, já que os estados de ativação passados da rede neural afetam o processamento no instante atual. Estas diferentes representações do fator tempo devem ser consideradas no desenvolvimento dos algoritmos de treinamento.

2.8.3. O algoritmo de treinamento

No início dos anos 80, as redes não-recorrentes de múltiplas camadas foram introduzidas e imediatamente encontraram larga aplicação em muitas áreas e com grande sucesso. A função sinal que caracterizava, no início, as funções de ativação dos modelos de perceptron (veja figura 2.2), foram substituídas por funções contínuas e suaves (veja figuras 2.7, 2.8 e 2.9). Com isso, devido à natureza suave da nova função de ativação, a sensibilidade do sinal de saída da rede neural em relação à variação da intensidade de suas interconexões passa a admitir apenas valores finitos, podendo ser obtida pela aplicação de técnicas de derivação numérica. Apesar deste fato ter sido primeiramente observado por Werbos (74)[70], ele foi redescoberto por Parker (85)[56] e tornou-se largamente conhecido devido ao trabalho de Rumelhart, Hinton e Williams (86)[61], que chegaram ao mesmo algoritmo de ajuste dos pesos desconhecendo a existência dos trabalhos anteriores. A técnica de derivação então empregada estava baseada no método do gradiente, o qual era calculado de uma maneira elegante por meio da retro-propagação através das camadas da rede, passando, assim, a ser denominado de algoritmo de retro-propagação.

O sistema composto pela rede neural multi-camadas juntamente com o algoritmo de retro-propagação encontrou inúmeras aplicações em problemas que requerem a realização de mapeamentos lineares ou não-lineares em um espaço compacto e de dimensão finita. A propriedade mais atrativa deste sistema pode ser considerada o fato de que, na maioria dos problemas práticos, os mapeamentos requeridos para se obter a solução não são linearmente separáveis, fazendo com que métodos bem definidos baseados em separação linear não possam ser adequadamente aplicados.

A figura 2.20(a), a seguir, mostra dois conjuntos **A** e **B** que não são linearmente separáveis e a figura 2.20(b) apresenta um conjunto **A** desconexo.

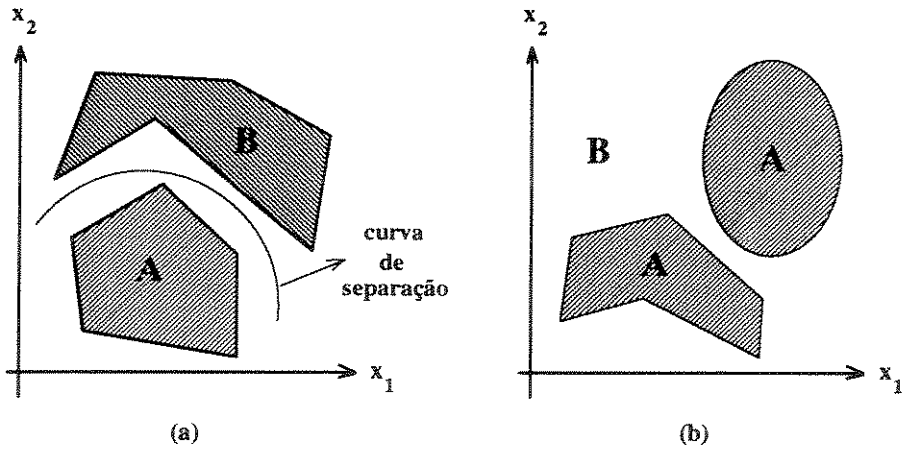


Figura 2.20 - Dois problemas de reconhecimento de padrões

Em ambos os casos, a curva de separação (curva discriminante) pode ser obtida através de uma rede neural de múltiplas camadas, como descrito em seções anteriores. No entanto, não está definido o algoritmo de treinamento responsável pelo ajuste dos pesos da rede neural que executará esta separação.

Um dos fatores mais importantes na definição de um algoritmo de treinamento é a possibilidade de se superar uma das questões fundamentais no tratamento de qualquer tipo de problema, ou seja, a aquisição de conhecimento. O método de aquisição de conhecimento pode ser interpretado como o processo de determinação das informações relevantes acerca do problema a ser resolvido e subseqüente incorporação deste conhecimento ao processo de obtenção da solução.

Com a aplicação do algoritmo de treinamento supervisionado, a rede neural vai solucionar o problema de forma análoga àquela muitas vezes utilizada pelo ser humano e outros animais, ou seja, pelo aprendizado a partir de exemplos.

O algoritmo de treinamento será apresentado utilizando-se a notação clássica de problemas de otimização, já que os critérios empregados para o ajuste dos pesos estão baseados fundamentalmente na minimização de uma função quadrática. De acordo com o nível de informação considerado na definição da direção de ajuste dos pesos, os algoritmos de treinamento podem ser divididos em:

- algoritmos de 1ª ordem: estão baseados no método do gradiente, considerando aproximações de 1ª ordem na definição da direção de ajuste.
- algoritmos de 2ª ordem: estão baseados no método do gradiente conjugado, considerando aproximações de 2ª ordem na definição da direção de ajuste.

2.8.3.1. Algoritmos de 1ª ordem

Caso 1 : O treinamento de um neurônio não-recorrente

Na figura 2.21, é apresentado um neurônio do tipo perceptron, unidade de processamento básica das redes neurais a serem utilizadas neste trabalho.

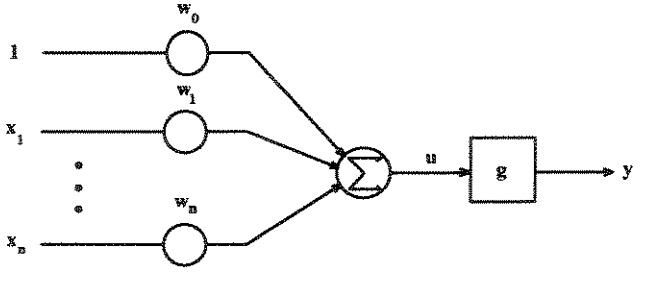


Figura 2.21 - Neurônio do tipo perceptron

onde $\mathbf{x} = [x_0\ x_1\ \ldots\ x_n]^T,\ x_0 = 1$

$\mathbf{w} = [w_0\ w_1\ \ldots\ w_n]^T$

$$u = \mathbf{x}^T\mathbf{w} = \sum_{i=0}^{n} \mathbf{w}_i \mathbf{x}_i$$

$y = g(u)$

Assumindo que os componentes do vetor de estados **w**, denominados pesos do neurônio, são os únicos parâmetros ajustáveis, pode-se formular o seguinte problema:

Problema: Dado um estado inicial $\mathbf{w}_0$, conduzir o neurônio para um estado final $\mathbf{w}_f$ tal que, para um determinado vetor de entrada **x**, a saída **y** da rede assuma um valor desejado $\mathbf{y}_d$.

Este problema pode ser reformulado como segue:

Reformulação do Problema: Dado um estado inicial $\mathbf{w}_0$, conduzir o neurônio para um estado final $\mathbf{w}_f$ tal que, para um determinado vetor de entrada **x**, a função:

$$J = \frac{1}{2}(y - y_d)^2 \qquad (2.1)$$

seja minimizada.

Solução proposta: Para encontrar a direção (ou a seqüência de direções) de ajuste que conduza o estado do neurônio de $\mathbf{w}_0$ a $\mathbf{w}_f$, é necessário e suficiente conhecer os valores das derivadas parciais de **J** em relação a cada $\mathbf{w}_i$ (i=0,...,n), que são dadas na forma:

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_i} = \frac{\partial \mathbf{J}}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{w}_i} = \frac{\partial \mathbf{J}}{\partial \mathbf{g}} \cdot \frac{\mathbf{dg}}{\mathbf{du}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{w}_i} = (y - y_d) \cdot \frac{\mathbf{dg}}{\mathbf{du}} \cdot \mathbf{x}_i, \qquad i=0,...,n$$

e ajustar cada peso $\mathbf{w}_i$ (i=0,...,n) no sentido de minimizar a função **J**. Para tanto, basta que a variação de cada peso tenha um sinal contrário àquele da derivada parcial correspondente, como na equação abaixo, onde α representa a taxa de convergência do processo de ajuste (taxa de .aprendizado):

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) - \alpha \frac{\partial \mathbf{J}}{\partial \mathbf{w}_i} = \mathbf{w}_i(k) - \alpha[y(k) - y_d]\frac{\mathbf{dg}}{\mathbf{du}}\mathbf{x}_i, \quad i = 0,\ldots,n \qquad (2.2)$$

Análise de Convergência:

Definindo $\varepsilon(k)$ na forma:

$$\varepsilon(k) = y_d - y(k) = y_d - g[u(k)] = y_d - g[x^T w(k)] \tag{2.3}$$

após a aplicação no instante k da equação (2.2) para i=0,...,n, a variação $\Delta\varepsilon(k)$ é dada na forma:

$$\Delta\varepsilon(k) = \Delta\{y_d - g[x^T w(k)]\} \cong -x^T \Delta w(k) \frac{dg}{du} \tag{2.4}$$

onde

$$\Delta w(k) = w(k+1) - w(k) = \alpha\varepsilon(k)x\frac{dg}{du} \tag{2.5}$$

Substituindo (2.5) em (2.4) obtém-se:

$$\Delta\varepsilon(k) \cong -\alpha x^T x\varepsilon(k)\left[\frac{dg}{du}\right]^2 = -\alpha\|x\|^2\varepsilon(k)\left[\frac{dg}{du}\right]^2 \tag{2.6}$$

A equação (2.6) pode ser reescrita na forma:

$$\varepsilon(k+1) \cong (1 - \alpha\|x\|^2\left[\frac{dg}{du}\right]^2)\varepsilon(k) \tag{2.7}$$

Logo, pode-se concluir que:

$$\varepsilon(k)\to 0 \text{ para } k\to\infty \text{ se e somente se } \alpha\|x\|^2\left[\frac{dg}{du}\right]^2 \in (0,2)$$

Com isso, para uma solução do tipo apresentado na equação (2.2), se forem utilizados valores fixos para α, não é possível garantir a convergência de ε para 0, ou seja, não é possível garantir a minimização da função **J**. A razão para isto é que não se tem controle sobre o valor de $\|x\|$, podendo conduzir o produto $\alpha\|x\|^2(dg/du)^2$ a valores que não pertençam ao intervalo aberto (0,2).

Como **g** é uma função conhecida e na quase totalidade dos casos monotonicamente não-decrescente (derivada não-negativa), o valor máximo de sua derivada **M** pode ser previamente calculado. Assim, uma forma de contornar este problema de convergência é substituir, na equação (2.2), a taxa de aprendizado α por $\alpha / M^2.\|x\|^2$, obtendo:

$$w_i(k+1) = w_i(k) - \frac{\alpha}{M^2.\|x\|^2}[y(k) - y_d]\frac{dg}{du}x_i, \quad i=0,...,n \tag{2.8}$$

com isso, a equação (2.7) assume a forma:

$$\varepsilon(k+1) \cong (1 - \frac{\alpha}{M^2}\left[\frac{dg}{du}\right]^2)\varepsilon(k) \tag{2.9}$$

e a convergência é garantida para $\alpha \in (0,2)$.

A equação (2.8) é uma equação a diferenças de 1ª ordem que permite obter a solução do problema proposto. Alguns autores, no entanto, incluem um termo adicional a esta equação, que adquire a forma:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) - \frac{\alpha}{\mathbf{M}^2.\|\mathbf{x}\|^2}[y(k)-y_d]\frac{dg}{du}x_i + \beta\Delta\mathbf{w}_i(k-1),\ i=0,...,n \qquad (2.10)$$

O termo $\Delta\mathbf{w}_i(k-1) = \mathbf{w}_i(k)-\mathbf{w}_i(k-1)$ representa a inércia do processo de ajuste de pesos e β é denominado coeficiente de momento. A consequência imediata da inclusão deste termo é a transformação da equação de ajuste de pesos em uma equação a diferenças de 2ª ordem. Uma vantagem da equação (2.10) em relação à equação (2.8) está no fato de que, dependendo dos valores de α e β, o processo de convergência pode adquirir um comportamento oscilatório, o que geralmente reduz a possibilidade de ocorrência de mínimos locais. Pode-se interpretar este procedimento como uma tentativa de introduzir informações de 2ª ordem até então desprezadas na formulação do processo de ajuste.

Desta forma, a equação (2.10) representa um método de ajuste de pesos para o neurônio da figura 2.21 que vai permitir a obtenção da solução do problema de minimização proposto.

Caso 2 : <u>O treinamento de uma rede neural não-recorrente</u>

Considere, agora, uma rede neural de duas camadas, conforme apresentado na figura 2.22. Esta rede neural tem as seguintes características de processamento:

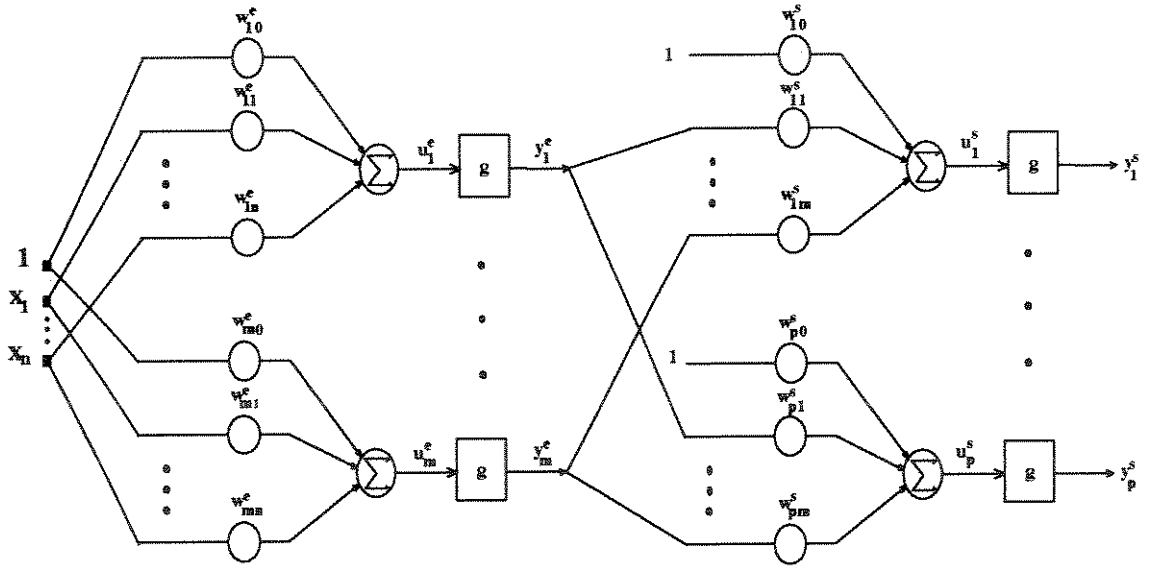


Figura 2.22 - Rede Neural tipo perceptron de duas camadas

- cada camada contém um determinado número de neurônios do tipo perceptron (figura 2.21), onde todos possuem a entrada adicional de polarização $\mathbf{x}_0$=1.
- o vetor de entrada de cada camada da rede neural é distribuído a todos os neurônios da mesma camada.
- as saídas de todos os neurônios de uma camada formam o vetor de entrada para a camada seguinte.

Considere, então, o seguinte problema:

Problema: Ajustar os pesos da rede neural da figura 2.22 de forma a minimizar a função:

$$\mathbf{J} = \frac{1}{2}\sum_{i=1}^{p}\left(\mathbf{y}_i^s - \mathbf{y}_{d_i}\right)^2 \qquad (2.11)$$

Solução: Neste caso, a solução também pode ser obtida com o auxílio da equação (2.10) do caso anterior, que assume a forma:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\alpha}{\mathbf{M}^2.\|\mathbf{x}\|^2}\frac{\partial \mathbf{J}}{\partial \mathbf{w}} + \beta\Delta\mathbf{w}(k-1) \qquad (2.12)$$

Para aplicar a equação (2.12) no processo de ajuste dos pesos da rede neural, basta calcular $\partial\mathbf{J}/\partial\mathbf{w}$ para todo peso $\mathbf{w}$ da rede:

Camada de saída: $\mathbf{w}_{ij}^s$, i=1,...,p; j=0,...,m

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{ij}^s} = \frac{\partial \mathbf{J}}{\partial \mathbf{u}_i^s}\frac{\partial \mathbf{u}_i^s}{\partial \mathbf{w}_{ij}^s} = \frac{\partial \mathbf{J}}{\partial \mathbf{g}\left(\mathbf{u}_i^s\right)}\frac{\mathrm{dg}}{\mathrm{du}_i^s}\frac{\partial \mathbf{u}_i^s}{\partial \mathbf{w}_{ij}^s} = \left(\mathbf{y}_i^s - \mathbf{y}_{d_i}\right)\frac{\mathrm{dg}}{\mathrm{du}_i^s}\mathbf{x}_{ij}^s$$

Camada de entrada: $\mathbf{w}_{ij}^e$, i=1,...,m; j=0,...,n

Primeiramente, é necessário calcular as derivadas parciais em relação a $\mathbf{x}_{kl}^s$, (k=1,...,p; l=0,...,m):

$$\frac{\partial \mathbf{J}}{\partial \mathbf{x}_{kl}^s} = \frac{\partial \mathbf{J}}{\partial \mathbf{u}_k^s}\frac{\partial \mathbf{u}_k^s}{\partial \mathbf{x}_{kl}^s} = \frac{\partial \mathbf{J}}{\partial \mathbf{g}\left(\mathbf{u}_k^s\right)}\frac{\mathrm{dg}}{\mathrm{du}_k^s}\frac{\partial \mathbf{u}_k^s}{\partial \mathbf{x}_{kl}^s} = \left(\mathbf{y}_k^s - \mathbf{y}_{d_k}\right)\frac{\mathrm{dg}}{\mathrm{du}_k^s}\mathbf{w}_{kl}^s$$

e então obtém-se:

$$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_{ij}^e} = \frac{\partial \mathbf{J}}{\partial \mathbf{u}_i^e}\frac{\partial \mathbf{u}_i^e}{\partial \mathbf{w}_{ij}^e} = \frac{\partial \mathbf{J}}{\partial \mathbf{g}\left(\mathbf{u}_i^e\right)}\frac{\mathrm{dg}}{\mathrm{du}_i^e}\frac{\partial \mathbf{u}_i^e}{\partial \mathbf{w}_{ij}^e} = \left[\sum_{k=1}^{p}\frac{\partial \mathbf{J}}{\partial \mathbf{x}_{ki}^s}\right]\frac{\mathrm{dg}}{\mathrm{du}_i^e}\mathbf{x}_{ij}^e = \left[\sum_{k=1}^{p}\left(\mathbf{y}_k^s - \mathbf{y}_{d_k}\right)\frac{\mathrm{dg}}{\mathrm{du}_k^s}\mathbf{w}_{ki}^s\right]\frac{\mathrm{dg}}{\mathrm{du}_i^e}\mathbf{x}_{ij}^e$$

Se na rede neural da figura 2.22 forem incluídas múltiplas camadas entre as camadas de entrada e saída, o procedimento de obtenção das derivadas parciais é o mesmo empregado acima, apenas exigindo um maior número de aplicações da regra da cadeia.

Este método de obtenção das derivadas parciais é conhecido como algoritmo de retro-propagação, porque, como pode ser verificado, a aplicação da regra da cadeia vai propagar a função **J**, definida na saída da rede neural, pelo caminho inverso, até a entrada da rede.

Dessa forma, para qualquer rede não-recorrente do tipo perceptron com múltiplas camadas, o problema de minimização da função **J** (equação (2.11)) pode ser solucionado com o ajuste dos pesos realizado por intermédio da equação (2.12), sendo que as derivadas parciais são obtidas com a aplicação do algoritmo de retro-propagação.

Caso 3 : O treinamento de um neurônio recorrente

Na figura 2.23 é apresentada uma modificação do neurônio encontrado na figura 2.21, sendo que uma entrada adicional, proveniente da realimentação do sinal de saída, está presente. A ocorrência desta realimentação permite classificar este neurônio como sendo do tipo recorrente.

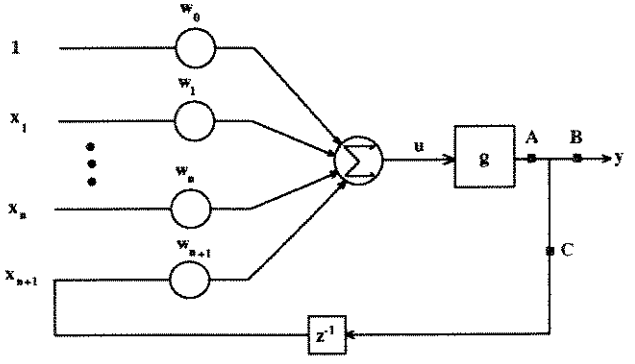


Figura 2.23 - Neurônio recorrente

Problema: Ajustar os pesos do neurônio da figura 2.23 de forma a minimizar a função

$$\mathbf{J} = \frac{1}{2}(\mathbf{y} - \mathbf{y}_d)^2 \qquad (2.13)$$

Solução: A mesma do caso não-recorrente:

$$\mathbf{w}_i(k+1) = \mathbf{w}_i(k) - \alpha\frac{\partial \mathbf{J}}{\partial \mathbf{w}_i} + \beta\Delta\mathbf{w}_i(k-1),\ i=0,...,n+1 \qquad (2.14)$$

onde $$\frac{\partial \mathbf{J}}{\partial \mathbf{w}_i} = \frac{\partial \mathbf{J}}{\partial \mathbf{g}}\frac{\mathbf{dg}}{\mathbf{du}}\mathbf{x}_i$$

A diferença em relação ao caso não-recorrente (equação (2.10)) está no cálculo de $\partial \mathbf{J}/\partial \mathbf{g}(\mathbf{u}) = \partial \mathbf{J}/\partial \mathbf{y}$, que deve considerar a influência da realimentação.

Num instante de tempo k, o sinal **y** definido no ponto **A** vai ser propagado através de dois percursos distintos:

- o percurso **AB** vai conduzir o sinal **y** para a saída do neurônio (percurso também realizado no caso não-recorrente).
- o percurso **AC** vai conduzir o sinal **y** para a entrada do neurônio (percurso recorrente)

Na aplicação do algoritmo de retro-propagação, $\partial \mathbf{J}/\partial \mathbf{y}$ no ponto **A** deve considerar a participação do termo proveniente da saída do neurônio (como no caso não-recorrente) e do termo proveniente da realimentação, produzindo:

$$\left.\frac{\partial \mathbf{J}}{\partial \mathbf{y_A}}\right|_{t=k} = \left.\frac{\partial \mathbf{J}}{\partial \mathbf{y_B}}\right|_{t=k} + \left.\frac{\partial \mathbf{J}}{\partial \mathbf{x}_{n+1}}\right|_{t=k} = \left.\frac{\partial \mathbf{J}}{\partial \mathbf{y_B}}\right|_{t=k} + \left[\left.\frac{\partial \mathbf{J}}{\partial \mathbf{y_A}}\right|_{t=k-1} \frac{\mathbf{dg}}{\mathbf{du}} \mathbf{w}_{n+1}(k)\right]$$

O algoritmo para o cálculo das derivadas parciais, neste caso, é denominado retro-propagação no tempo, porque é utilizada uma equação recorrente no tempo para efetivar os cálculos (Werbos-90)[69].

Caso 4 : <u>O treinamento de uma rede neural recorrente</u>

No caso de uma rede neural recorrente, existem dois tipos possíveis de realimentações:

- realimentação interna: a saída de um neurônio é realimentada como entrada para outros neurônios da rede, na mesma camada ou em camadas anteriores. O tipo mais simples de realimentação interna ocorre quando a saída de um neurônio é realimentada como entrada apenas para o próprio neurônio.
- realimentação externa: neste caso, apenas as saídas da rede neural são realimentadas, e somente na forma de entradas da própria rede.

Para efeito de aplicação, será estudado apenas o caso de realimentação externa. Na figura 2.24, é apresentada uma rede neural de duas camadas com realimentações externas.

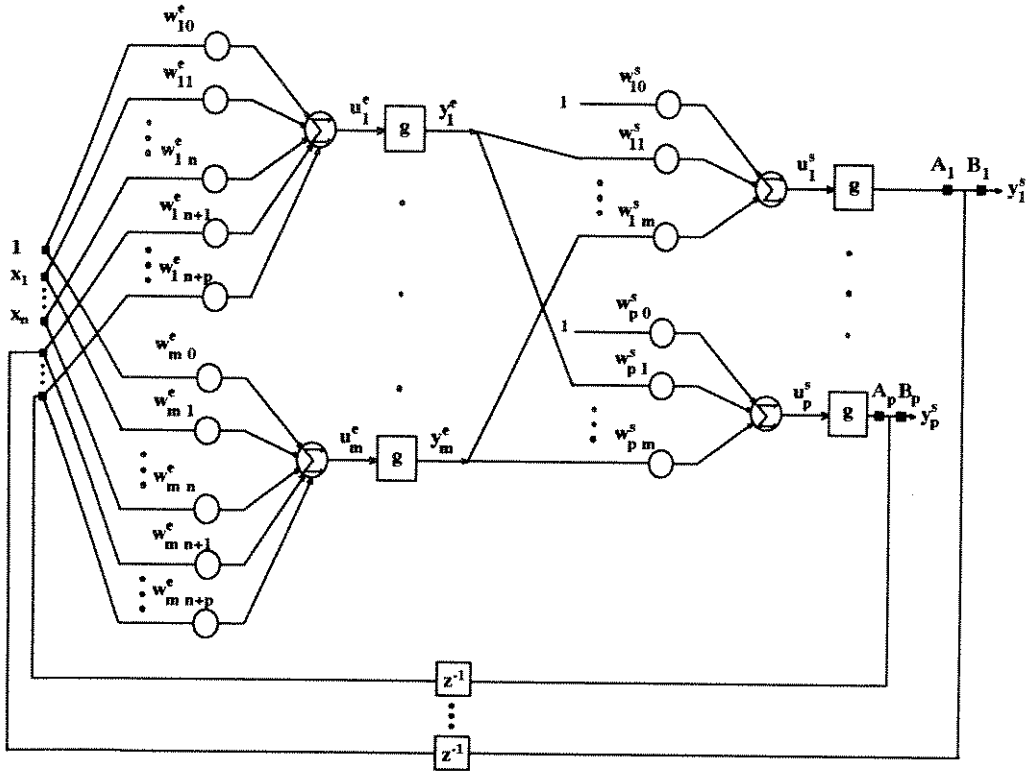


Figura 2.24 - Rede neural recorrente

Problema: Ajustar os pesos da rede neural da figura 2.24 de forma a minimizar a função:

$$\mathbf{J} = \frac{1}{2}\sum_{i=1}^{p}\left(\mathbf{y}_i^s - \mathbf{y}_{d_i}\right)^2 \tag{2.15}$$

Solução: Neste caso, a solução pode ser obtida com o auxílio da equação (2.12) do caso não-recorrente, que assume a forma:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\alpha}{\mathbf{M}^2.\|\mathbf{x}\|^2}\frac{\partial \mathbf{J}}{\partial \mathbf{w}} + \beta\Delta\mathbf{w}(k-1) \tag{2.16}$$

A diferença em relação ao caso não-recorrente novamente está no cálculo de $\frac{\partial \mathbf{J}}{\partial \mathbf{g}(\mathbf{u}_i^s)} = \frac{\partial \mathbf{J}}{\partial \mathbf{y}_i^s}$, i=1,...,p, que passa a ser descrito por:

$$\left.\frac{\partial \mathbf{J}}{\partial \mathbf{y}_{A_i}^s}\right|_{t=k} = \left.\frac{\partial \mathbf{J}}{\partial \mathbf{y}_{B_i}^s}\right|_{t=k} + \left.\frac{\partial \mathbf{J}}{\partial \mathbf{x}_{n+i}}\right|_{t=k} \tag{2.17}$$

onde

$$\left.\frac{\partial \mathbf{J}}{\partial \mathbf{y}_{B_i}^s}\right|_{t=k} = \left(\mathbf{y}_i^s(k) - \mathbf{y}_{d_i}\right)$$

$$\left.\frac{\partial J}{\partial x_{n+i}}\right|_{t=k} = \sum_{l=1}^{m}\left[\sum_{k=1}^{p}\left(\left.\frac{\partial J}{\partial y^{s}_{A_k}}\right|_{t=k-1}\frac{dg}{du^{s}_{k}}w^{s}_{kl}\right)\right]\frac{dg}{du^{e}_{l}}w^{e}_{l\,n+i}$$

Para uma rede com múltiplas camadas entre as camadas de entrada e saída, é necessário aplicar a regra da cadeia mais vezes para obter o segundo termo do lado direito da equação (2.17). Esta é a versão do algoritmo de retro-propagação no tempo que será utilizada nos procedimentos de aplicação que vierem a utilizar redes neurais recorrentes.

2.8.3.2. As limitações do algoritmo de treinamento de 1ª ordem baseado no método do gradiente

No caso de processos de minimização baseados no método do gradiente, a única informação a respeito da função a ser minimizada e que é utilizada no processo de ajuste de parâmetros está baseada na expansão por série de Taylor de 1ª ordem, ou seja:

$$J(w_{k+1}) \cong J(w_k) + \left[\frac{\partial J(w_k)}{\partial w}\right]^{T}[w_{k+1}-w_k] \qquad (2.18)$$

Como o gradiente informa a direção de máxima variação da função e como a função é linearizada (aproximação de 1ª ordem), então, em problemas de minimização, a direção de ajuste é dada pelo valor negativo do vetor gradiente. Baseando-se na equação (2.10), a regra de ajuste pode ser dada na forma:

$$w_{k+1} = = w_k + \frac{\alpha_k}{M^2.\|x\|^2}\left(-\frac{\partial J(w_k)}{\partial w}\right) + \beta\Delta w_{k-1} \qquad (2.19)$$

A aproximação de 1ª ordem é muitas vezes responsável pela baixa taxa de convergência apresentada por esta estratégia de minimização, principalmente em fases evoluídas do processo de ajuste, onde o erro é pequeno. Além disso, a utilização de taxas de ajuste (α) arbitrárias (constantes ou normalizadas) faz com que o algoritmo perca eficiência e robustez. Neste caso, a inclusão do termo de momento é importante, à medida que o algoritmo de ajuste é forçado a considerar informações de 2ª ordem. Apesar deste fator positivo, a inclusão do termo de momento torna o algoritmo menos robusto ainda, pois é introduzido mais um parâmetro a ser definido arbitrariamente (β), já que não existem fundamentos teóricos que permitam a definição de métodos automáticos de ajuste. Este é um ponto crítico de algoritmos baseados no método do gradiente, já que os valores desses parâmetros acabam sendo decisivos para o sucesso do processo de treinamento.

2.8.3.3. O algoritmo de treinamento de 2ª ordem

Nesta seção, será apresentado um algoritmo de treinamento supervisionado de 2ª ordem desenvolvido com o objetivo de eliminar ou, pelo menos, reduzir as limitações do algoritmo de 1ª ordem, discutidas na seção anterior. Considerando-se o fato de que os algoritmos de 1ª ordem são mais eficientes nas fases iniciais de treinamento, enquanto que os de 2ª ordem são mais eficientes nas fases avançadas de treinamento, o algoritmo a ser apresentado na seqüência vai procurar uma forma de introduzir estas características no seu desenvolvimento. Como conseqüência, será produzido um algoritmo híbrido que, devido à sua maior complexidade, apresenta propriedades de convergência ainda não totalmente avaliadas. Com isso, nos procedimentos de implementação será utilizado preferencialmente o algoritmo de 1ª ordem, sendo que em situações onde seu desempenho não for satisfatório, o algoritmo de 2ª ordem será utilizado como uma alternativa viável.

2.8.3.3.1. Nomenclatura e conceitos a serem utilizados no desenvolvimento do algoritmo

Do ponto de vista de otimização, o aprendizado supervisionado em redes neurais artificiais é equivalente à minimização de uma função multivariável dependente dos parâmetros variáveis da rede. Baseado neste princípio, o desenvolvimento de algoritmos de treinamento efetivos pode estar fundamentado em conceitos bem estabelecidos em outras áreas de atuação científica, como por exemplo, a aplicação da teoria convencional de análise numérica no processo de minimização de funções multivariáveis.

Já que processos de treinamento em redes neurais envolvem invariavelmente o ajuste de um grande número de parâmetros, apenas métodos de otimização capazes de operar com problemas de larga escala representam alternativas viáveis. Dentro da teoria de análise numérica, um método de otimização de 2ª ordem capaz de atuar em problemas de larga escala é o método do gradiente conjugado (Fletcher-75)[24].

Dada uma rede neural multi-camadas, os parâmetros ajustáveis são os pesos, representando a intensidade de cada conexão entre as unidades processadoras da rede. Agrupando os pesos da rede na forma de um vetor, podemos construir um vetor de pesos de dimensão igual ao número de pesos da rede (n). Dessa forma, o vetor de pesos será referido como um ponto no $\Re^n$. O ordenamento dos pesos pode respeitar qualquer regra, desde que esta regra, uma vez definida, seja utilizada na composição das outras variáveis que vão participar no desenvolvimento do algoritmo.

A regra de ordenamento a ser adotada neste desenvolvimento está baseada na ordem crescente dos índices das unidades processadoras de cada camada e na inclusão dos pesos por camadas, a partir da camada de entrada até a camada de saída. Tomando como referência a rede neural recorrente da figura 2.24, o vetor de pesos vai assumir a forma:

$$w = \left[w^e_{10} \;\; w^e_{11} \;\; \cdots \;\; w^e_{1n+p} \;\; w^e_{20} \;\; \cdots \;\; w^e_{mn+p} \;\; w^s_{10} \;\; w^s_{11} \;\; \cdots \;\; w^s_{1m} \;\; w^s_{20} \;\; \cdots \;\; w^s_{pm} \right]^T$$

Dado um número de *r* vetores padrões y_j^{pad} (j=1,...,r) a serem apresentados à rede neural da figura 2.24 durante o processo de treinamento supervisionado, é possível definir uma função erro na forma:

$$J_j(y^s) = \frac{1}{2}\sum_{i=1}^{p}\left[\left(y^s - y_j^{pad}\right)^2\right] \qquad (j=1,...,r)$$

onde as componentes do vetor de saída $y^s = [y^s_1 \; y^s_2 \; ... \; y^s_p]$ são funções dos pesos da rede. Como $J_j(.)$ é função apenas do vetor de saída, pode-se afirmar que a função erro também é função dos pesos da rede, ou seja, $J_j(y^s) = J_j(w)$. Dessa forma, é possível tomar a derivada parcial de $J_j(.)$ com relação a cada peso da rede e contruir o vetor gradiente considerando-se a participação de todos os *r* vetores padrões:

$$\frac{\partial J}{\partial w} = \left[\sum_{j=1}^{r}\left(\frac{\partial J_j}{\partial w^e_{10}}\right) \cdots \sum_{j=1}^{r}\left(\frac{\partial J_j}{\partial w^e_{mn+p}}\right) \sum_{j=1}^{r}\left(\frac{\partial J_j}{\partial w^s_{10}}\right) \cdots \sum_{j=1}^{r}\left(\frac{\partial J_j}{\partial w^s_{pm}}\right) \right]^T$$

Seja w^i a i-ésima componente do vetor de pesos w, então o comprimento do vetor w é dado na forma:

$$\|w\| = \left[\sum_{j=1}^{n}\left(w^i\right)^2 \right]^{\frac{1}{2}}$$

Dados um estado inicial w_0 para o vetor de pesos, a função erro $J(w_0)$ e suas derivadas parciais em relação ao vetor de pesos, então o valor da função erro para o estado $w^*=w_0+\varepsilon$, pode ser obtida por expansão em série de Taylor na forma:

$$J(w^*) = J(w_0+\varepsilon) = J(w_0) + \left[\frac{\partial J(w_0)}{\partial w}\right]^T \varepsilon + \frac{1}{2}\varepsilon^T\left[\frac{\partial^2 J(w_0)}{\partial w^2}\right]\varepsilon + \ldots$$

Uma matriz Q_{nXn} é dita ser positiva definida (Q>0) se a forma quadrática:

$$w^T Q w > 0 \qquad \forall\, w \in \Re^n$$

Seja w_1, w_2, ..., w_k um conjunto de vetores de pesos no $\Re^n$. Estes vetores são ditos Q-conjugados com relação a uma matriz não-singular e simétrica Q_{nXn} se:

$$w_i^T Q w_j = 0 \qquad (i \neq j,\; i=1,\; ...,\; k)$$

A notação apresentada nesta seção servirá como base para os desenvolvimentos apresentados nas seções seguintes.

2.8.3.3.2. Estratégia de otimização

A maior parte dos métodos de otimização aplicados a processos de minimização de funções utilizam a mesma estratégia: a minimização está baseada em um processo iterativo local, em que uma aproximação da função em uma vizinhança do ponto de operação atual é minimizada. A idéa básica da estratégia de minimização é ilustrada através do seguinte algoritmo:

Passo 1: Escolha um vetor de pesos inicial w_1 e faça k=1;

Passo 2: Determine uma direção de ajuste p_k e uma taxa de ajuste α_k de maneira que $J(w_k+\alpha_k p_k)<J(w_k)$;

Passo 3: Atualize o vetor de pesos: $w_{k+1}=w_k+\alpha_k p_k$;

Passo 4: Se $\frac{\partial J(w_k)}{\partial w} \neq 0$, então faça k=k+1 e volte ao **Passo 2**;
Senão, w_{k+1} é o mínimo desejado.

A determinação do próximo ponto no espaço $\Re^n$ dos pesos envolve a realização de dois procedimentos independentes. Primeiramente, uma direção de ajuste deve ser determinada; uma vez encontrada esta direção, deve-se decidir o quanto caminhar nesta direção no espaço dos pesos, ou seja, deve-se definir a taxa de ajuste α.

2.8.3.3.3. A aplicação do método do gradiente conjugado no desenvolvimento do algoritmo de treinamento de 2ª ordem

Tanto o método do gradiente, como o método do gradiente conjugado utilizam a mesma estratégia de otimização descrita na seção anterior. A diferença fundamental está no fato de que, agora, para a definição da direção e da taxa de ajuste, um conjunto mais completo de informações é utilizado. Neste caso, a aproximação da função a ser minimizada está baseada na expansão por série de Taylor de 2ª ordem:

$$J(w_{k+1}) \cong J(w_k) + \left[\frac{\partial J(w_k)}{\partial w}\right]^T [w_{k+1}-w_k] + \frac{1}{2}[w_{k+1}-w_k]^T \left[\frac{\partial^2 J(w_k)}{\partial w^2}\right][w_{k+1}-w_k]$$

que, em notação simplificada, assume a forma:

$$J(w_{k+1}) = J(w_k) + (J')^T \Delta w + \frac{1}{2}\Delta w^T J'' \Delta w \quad \text{onde } \Delta w = w_{k+1}-w_k$$

Esta expansão de 2ª ordem resulta em uma função quadrática. As funções quadráticas apresentam propriedades muito interessantes do ponto de vista de aplicação a problemas de minimização. Por exemplo, se a matriz hessiana de uma função quadrática é

positiva definida, quando um ponto de mínimo é encontrado para a função, então este mínimo é global (Luenberger-69)[42].

Como é utilizada uma aproximação quadrática para a função J(.), então para determinar (aproximadamente) o valor do ponto de mínimo de J(.), é preciso analisar os pontos críticos w^*, dados nos instantes onde a derivada primeira se anula, ou seja:

$$\frac{\partial J(w^*)}{\partial w} = J''(w^*-w_1) + J' = 0 \qquad \text{onde } J''=(J'')^T \tag{2.20}$$

A solução pode ser simplificada consideravelmente se a análise do problema for baseada na geração de *n* vetores J''-conjugados (Moller-90)[49].

Seja um conjunto de vetores J''-conjugados p_1, p_2, ..., p_n. Como este conjunto forma uma base do $\Re^n$, o processo de obtenção de um ponto crítico w^*, a partir de um ponto inicial w_1 pode ser expresso como uma combinação linear de p_1, p_2, ..., p_n, ou seja:

$$w^* = w_1 + \sum_{i=1}^{n} \alpha_i p_i \,,\ \alpha_i \in \Re \tag{2.21}$$

Multiplicando a equação (2.21) por $p_j^T J''$, substituindo J' da equação (2.20) e utilizando a propriedade de que p_1, p_2, ..., p_n são J''-conjugados, é possível obter α_j na forma:

$$\alpha_j = \frac{-p_j^T J'}{p_j^T J'' p_j} \tag{2.22}$$

Utilizando a equação (2.22) para j=1,...,n e substituindo na equação (2.21), o ponto crítico w^* pode ser obtido. Além disso, se J''>0, w^* minimiza a função erro dada pela aproximação de 2ª ordem da série de Taylor. Para a aplicação das equações (2.21) e (2.22), foi assumida a existência de um conjunto de *n* vetores J''-conjugados. Felizmente não é necessário o conhecimento deste conjunto por antecipação, pois os *n* vetores podem ser determinados recursivamente (Moller-90)[49]:

- inicialmente, defina $p_1 = -J'(w_1)$;
- então, para j=2,...,n, p_j é determinado recursivamente como uma combinação linear de $-J'(w_j)$ e dos vetores J''-conjugados já calculados.

Uma 1ª versão do algoritmo do gradiente conjugado é apresentada a seguir, onde novas variáveis foram introduzidas para efeito de estruturação da seqüência de procedimentos.

Algoritmo do gradiente conjugado - 1ª versão (Moller-90)[49]

1. Escolha um vetor de pesos inicial w_1 e faça:

   $p_1 = r_1 = -J'(w_1)$;
   $k = 1$.

2. Calcule as informações de 2ª ordem:

   $s_k = J''(w_k)p_k$;
   $\delta_k = p_k^T s_k$.

3. Calcule a taxa de ajuste:

   $\mu_k = p_k^T r_k$;
   $\alpha_k = \mu_k/\delta_k$.

4. Atualize o vetor de pesos:

   $w_{k+1} = w_k + \alpha_k p_k$;
   $r_{k+1} = -J'(w_{k+1})$.

5. Se k mod $n = 0$, então reinicialize o algoritmo:

   $p_{k+1} = r_{k+1}$.

   Senão, defina uma nova direção conjugada:

   $$\beta_k = \frac{\|r_{k+1}\|^2 - r_{k+1} r_k}{\mu_k};$$

   $p_{k+1} = r_{k+1} + \beta_k p_k$.

6. Se $r_k > \varepsilon$, onde $\varepsilon \to 0$, então faça:

   $k = k+1$;
   retorne ao item 2.

   Senão, o procedimento de ajuste chegou ao fim e w_{k+1} é o ponto de mínimo.

Para o cálculo de uma nova direção conjugada no item 5 do algoritmo, o coeficiente β_k da combinação linear foi obtido utilizando-se a fórmula de Hestenes-Stiefel (Moller-90)[49], sendo que muitas outras fórmulas poderiam ser utilizadas.

Na forma como foi implementado, este algoritmo apresenta três problemas principais:

- para cada iteração, a matriz hessiana $J''(w_k)$ deve ser calculada e armazenada.
- O algoritmo assume que a matriz hessiana $J''(w_k)$ é sempre positiva definida, o que não pode ser garantido.
- A aproximação quadrática da função erro, na qual o algoritmo se baseia, pode ser inadequada quando o erro se encontra distante do valor mínimo.

É interessante do ponto de vista computacional não obter a matriz hessiana explicitamente, devido à demanda de cálculo e memória. Uma alternativa viável é estimar $J''(w_k)p_k$ na forma:

$$s_k = J''(w_k)p_k \cong \frac{J'(w_k + \sigma_k p_k) - J'(w_k)}{\sigma_k}, \qquad 0<\sigma_k<<1$$

Uma modificação deve ser imposta ao algoritmo para garantir que a matriz hessiana seja sempre positiva definida. Dentre as alternativas existentes, uma possibilidade é realizar, a cada iteração, uma busca da melhor taxa de ajuste. No entanto, a adoção deste procedimento pode aumentar consideravelmente a carga computacional por iteração. Uma segunda possibilidade, baseada no método de Levenberg-Marquardt (Fletcher-75)[24], conduz à introdução de um fator adicional na expressão do cálculo da matriz hessiana:

$$s_k \cong \frac{J'(w_k + \sigma_k p_k) - J'(w_k)}{\sigma_k} + \lambda_k p_k \tag{2.23}$$

sendo que a cada iteração λ_k é ajustado com base no sinal de δ_k, que indica quando a matriz hessiana não é positiva definida. Se $\delta_k \leq 0$, então λ_k é aumentado para λ_k^N e o novo valor de s_k (s_k^N) é dado por:

$$s_k^N = s_k + (\lambda_k^N - \lambda_k)p_k$$

É possível determinar-se quanto λ_k deve ser aumentado para tornar δ_k positivo:

$$\delta_k^N = p_k^T s_k^N = p_k^T[s_k + (\lambda_k^N - \lambda_k)p_k] = \delta_k + (\lambda_k^N - \lambda_k)\|p_k\|^2$$

então, para $\delta_k^N > 0$, tem-se $\lambda_k^N > \lambda_k - \delta_k/\|p_k\|^2$.

No entanto, não existe um critério automático para definir-se quanto λ_k deve ser aumentado para se obter uma solução ótima. Uma escolha interessante e que será adotada neste desenvolvimento é dada na forma:

$$\lambda_k^N = 2\left(\lambda_k - \frac{\delta_k}{\|p_k\|^2}\right), \text{ produzindo } \delta_k^N = -\delta_k + \lambda_k\|p_k\|^2 > 0$$

Com isso, a taxa de ajuste assume a forma:

$$\alpha_k = \frac{\mu_k}{\delta_k} = \frac{\mu_k}{p_k^T s_k + \lambda_k\|p_k\|^2} \tag{2.24}$$

Agora, com relação ao terceiro problema apresentado pelo algoritmo, além da possibilidade de inadequação da aproximação quadrática da função erro quando o erro se encontra distante do valor mínimo, existe também o termo adicional na definição da matriz hessiana (veja equação (2.23)). Portanto, para obter uma boa aproximação, pode ser necessário reduzir ou aumentar o fator λ_k, mesmo quando a matriz hessiana já for positiva definida. Definindo:

$$\Delta_k = \frac{2\delta_k\left[J\left(y^s(w_k)\right) - J\left(y^s(w_k + \alpha_k p_k)\right)\right]}{\mu_k^2} \qquad (2.25)$$

como um fator de comparação que indica o quanto a função quadrática se aproxima da função real (para boas aproximações, Δ_k se aproxima de 1), é possível estabeler o seguinte critério de atualização para λ_k (Moller-90)[49]:

Se $\Delta_k > 0.75$, então $\lambda_k = 0.5 * \lambda_k$
Se $\Delta_k < 0.25$, então $\lambda_k = 4.0 * \lambda_k$

A versão final do algoritmo do gradiente conjugado é apresentada a seguir:

Algoritmo do gradiente conjugado - Versão Final (Moller-90)[49]

1. Escolha um vetor de pesos inicial w_1 e escalares $\sigma>0$, $\lambda_1^N>0$ e $\lambda_1=0$.
2. Faça:
   $p_1 = r_1 = -J(w_1)$;
   $k = 1$, sucesso = 1.
3. Se sucesso = 1, calcule as informações de 2ª ordem:

$$\sigma_k = \frac{\sigma}{\|p_k\|};$$

$$s_k = \frac{J'(w_k + \sigma_k p_k) - J'(w_k)}{\sigma_k};$$

$$\delta_k = p_k^T s_k.$$

4. Ajuste s_k:
   $s_k^N = s_k + (\lambda_k^N - \lambda_k)p_k$;
   $\delta_k = \delta_k + (\lambda_k^N - \lambda_k)\|p_k\|^2$.
5. Se $\delta_k \leq 0$, então faça a matriz hessiana positiva definida:

$$s_k = s_k + \left(\lambda_k - 2\frac{\delta_k}{\|p_k\|^2}\right)p_k;$$

$$\lambda_k^N = 2\left(\lambda_k - \frac{\delta_k}{\|p_k\|^2}\right);$$

$$\delta_k^N = -\delta_k + \lambda_k\|p_k\|^2;$$

$$\lambda_k = \lambda_k^N.$$

6. Calcule a taxa de ajuste:

$\mu_k = p_k^T r_k$;

$\alpha_k = \mu_k / \delta_k$.

7. Calcule o parâmetro de comparação:

$$\Delta_k = \frac{2\delta_k \left[J\left(y^s(w_k)\right) - J\left(y^s(w_k + \alpha_k p_k)\right) \right]}{\mu_k^2}$$

8. Se $\Delta_k \geq 0$ (o erro pode ser reduzido), então atualize o vetor de pesos:

$w_{k+1} = w_k + \alpha_k p_k$;

$r_{k+1} = -J'(w_{k+1})$;

$\lambda_k^N = 0$;

sucesso = 1.

8a. Se k mod n = 0, então reinicialize o algoritmo:

$p_{k+1} = r_{k+1}$.

Senão, defina uma nova direção conjugada:

$$\beta_k = \frac{\|r_{k+1}\|^2 - r_{k+1} r_k}{\mu_k};$$

$p_{k+1} = r_{k+1} + \beta_k p_k$.

8b. Se $\Delta_k \geq 0.75$, então faça $\lambda_k = 0.5\lambda_k$.

Senão, uma redução no erro não é possível:

sucesso = 0;

$\lambda_k^N = \lambda_k$.

9. Se $\Delta_k \leq 0.25$, então faça $\lambda_k = 4\lambda_k$.

10. Se $r_k > \varepsilon$, onde $\varepsilon \to 0$, então faça:

k = k+1;

retorne ao item 2.

Senão, o procedimento de ajuste chegou ao fim e w_{k+1} é o ponto de mínimo.

2.8.3.4. Comparação entre os algoritmos de 1ª e 2ª ordem

A diferença básica entre os algoritmos de 1ª e 2ª ordem está no tipo de aproximação da função erro: algoritmos de 1ª ordem consideram uma aproximação linear, enquanto os de 2ª ordem consideram uma aproximação quadrática. Portanto, o desempenho dos algoritmos vai depender, além da carga de cálculo requerida, do nível de adequação da aproximação empregada.

Com relação à carga de cálculo por iteração, o algoritmo de 2ª ordem requer, em média, 50% a mais de operações por iteração. Já com relação à adequação de aproximação, em geral, aproximações lineares têm um melhor desempenho em regiões de

operação afastadas dos pontos críticos (pontos onde o gradiente se anula),ou seja, na fase inicial do treinamento, enquanto aproximações de 2ª ordem são mais adequadas em pontos de operação próximos aos pontos críticos, ou seja, em fases avançadas do treinamento.

Portanto, um procedimento interessante seria fazer uma composição de algoritmos de 1ª e 2ª ordem durante o processo de treinamento, de tal forma que a direção de ajuste privilegiada na fase inicial de treinamento seja a fornecida pelo algoritmo de 1ª ordem, enquanto que na fase final de treinamento seja a fornecida pelo algoritmo de 2ª ordem. De certa forma, no algoritmo de 2ª ordem descrito acima, a aplicação do método de Levenberg-Marquardt objetiva exatamente a implementação desta composição de algoritmos. À medida que o fator de comparação Δ_k se aproxima de 0, ou seja, a aproximação quadrática deixa de ser adequada, a taxa de adaptação é reduzida, pois λ_k aumenta com Δ_k pequeno e, da equação (2.24) pode-se concluir que α_k diminui. Com a adoção deste processo de compensação, pode-se concluir que o algoritmo de 2ª ordem não deve perder em eficiência para o algoritmo de 1ª ordem em fases iniciais de treinamento.

O exemplo de simulação apresentado na figura 2.25, comparando o desempenho dos dois algoritmos nas fases inicial e avançada de treinamento, ajuda a validar a conclusão obtida acima, pois os dois algoritmos apresentam um desempenho semelhante no início do treinamento, enquanto que na fase final do treinamento o desempenho do algoritmo de 2ª ordem mostra-se muito superior. O objetivo do treinamento[2] é aproximar a função y=sen(w) para w $\in$ [-1.5,1.5] utilizando uma rede neural não-recorrente[3] com três camadas, sendo 10 unidades processadoras na camada de entrada, 5 na camada oculta e 1 na camada de saída. O conjunto inicial de pesos é o mesmo para os dois algoritmos de treinamento. Na figura 2.25, nos gráficos da coluna da esquerda são mostrados os sinais de saída da rede neural y(RN) e da função y(padrão), enquanto na coluna da direita são mostrados os valores da função custo J (erro quadrático médio) para os algoritmos de 1ª e 2ª ordem em função do número de iterações. Os gráficos da primeira linha correspondem à fase inicial do treinamento, enquanto que os da segunda linha correspondem a uma fase avançada do treinamento.

[2]O procedimento de implementação de algoritmos de treinamento para aproximação de funções será apresentado no capítulo 4.

[3]Observe que o algoritmo de 2ª ordem independe do tipo de rede neural utilizada, desde que J' e J" sejam calculados adequadamente.

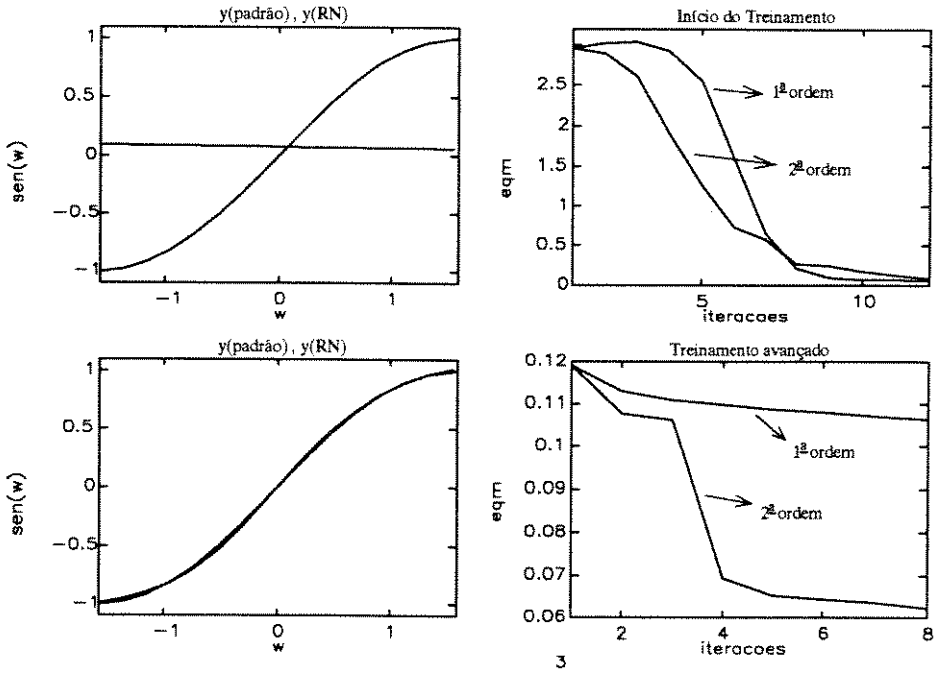


Figura 2.25 - Comparação entre algoritmos de treinamento de 1ª e 2ª ordem

Capítulo 3

Teoria de Sistemas de Controle

Os sistemas de controle têm desempenhado um papel fundamental no desenvolvimento científico e tecnológico. A partir dos anos 20, resultados importantes foram obtidos com o entendimento e utilização de técnicas de realimentação. Em resposta aos desafios impostos pelos problemas de controle existentes durante a 2ª Guerra Mundial, avanços tecnológicos foram obtidos e orientaram o desenvolvimento da teoria de controle por mais de duas décadas. A partir do início dos anos 60, novos desafios surgiram com o estabelecimento de projetos de conquista espacial, sendo que resultados impressionantes foram obtidos. A condução da espaçonave Apollo em uma trajetória ótima da Terra até a Lua, e o pouso suave na Lua foram exemplos de aplicações bem sucedidas de engenharia de controle.

A aplicação de metodologias de controle em todos os aspectos da vida das sociedades industrializadas vem contribuindo decisivamente para a melhoria da qualidade de vida. Avanços na teoria e na prática de controle automático propiciam meios para se atingir desempenho ótimo de sistemas dinâmicos, além de sustentarem o desenvolvimento de processos industriais e de fabricação mais modernos. A operação adequada dos mais variados dispositivos, desde equipamentos domésticos e automóveis, até reatores nucleares e veículos espaciais, só pode ser obtida através da atuação de controladores automáticos.

No entanto, a maior parte dos sistemas de controle automático em operação é resultado de projetos simples, que incorporam um conjunto reduzido de especificações em seu desenvolvimento. O controle de processos industriais é um bom exemplo: embora tenham ocorrido avanços significativos na teoria de controle e na obtenção de sensores e atuadores mais eficientes, ainda predominam os controladores automáticos do tipo proporcional-integral-derivativo (PID). Como conseqüência, estes sistemas de controle apresentam uma região de operação limitada, sendo que a operação fora desta região pode conduzir o sistema à perda de eficiência ou mesmo à instabilidade.

O principal problema relacionado com as novas técnicas desenvolvidas é que elas requerem a incorporação de uma quantidade significativa de conhecimento para serem adequadamente implementadas. Portanto, é necessário o desenvolvimento de técnicas mais gerais, mais fáceis de implementar e que, ainda assim, apresentem um desempenho superior aos controladores PID convencionais.

Este capítulo descreve sistemas lineares e não-lineares e o controle de sistemas físicos. A intenção é apresentar os conceitos fundamentais necessários para o entendimento desta dissertação. São discutidos os fundamentos de sistemas dinâmicos, as diferenças entre sistemas lineares e não-lineares e os conceitos relacionados com bifurcação, atratores e caos. A seguir, o conceito de estabilidade de sistemas dinâmicos é apresentado, incluindo a

definição de estabilidade de Lyapunov. Os resultados recentes na teoria de sistemas não-lineares são então utilizados para demonstrar que controle adaptativo de sistemas não-lineares é teoricamente factível. Os fundamentos e métodos clássicos e modernos de controle de sistemas físicos também são apresentados, enfatizando métodos de controle adaptativo. Finalmente são discutidas as tendências dentro da área de controle de sistemas dinâmicos.

3.1. Sistemas Dinâmicos

O estudo de sistemas dinâmicos é a investigação das variações do sistema com o tempo. Sistemas dinâmicos podem ser derivados do estudo de fluxo de informação, fluxo de energia, ou mesmo de sistemas puramente matemáticos. Esta dissertação está concentrada no estudo do fluxo de energia em sistemas físicos.

O tratamento matemático de sistemas dinâmicos utiliza o conceito de equações diferenciais. Dessa forma, os fundamentos da análise de sistemas dinâmicos estão relacionados com a geração e análise de equações diferenciais.

Em aplicações de controle, a distinção entre sistemas lineares e não-lineares ou, equivalentemente, entre equações diferenciais lineares e não-lineares, é fundamental. Uma equação diferencial pode ser definida na forma:

> *Uma equação diferencial é linear se, e somente se, cada termo da equação é de primeiro grau com relação a suas variáveis dependentes ou com relação a qualquer derivada de suas variáveis dependentes.*

Por outro lado, uma equação diferencial não-linear é qualquer equação diferencial que não é linear com relação a alguma de suas variáveis dependentes. Sistemas físicos reais são geralmente não-lineares, representando uma forte motivação para o estudo de sistemas não-lineares.

A maior parte da pesquisa em controle automático tem se concentrado em sistemas lineares, principalmente devido à carência de ferramentas adequadas para a análise de sistemas não-lineares. Com isso, o controle de boa parte dos sistemas não-lineares tem sido realizado através da obtenção de uma aproximação linear para o sistema, seguido da aplicação da metodologia de controle linear. Considerando-se que qualquer não-linearidade pode ser representada aproximadamente por uma série de Taylor truncada, caso a operação do sistema permaneça suficientemente próxima do estado em torno do qual o sistema foi linearizado, a aproximação linear pode ser adequada. No entanto, este procedimento pode resultar em um desempenho insatisfatório, especialmente se a não-linearidade for significativa e a região de operação for ampla. Assim, uma representação não-linear é desejada, e com a crescente disponibilidade de recursos por parte dos computadores digitais, torna-se cada vez mais viável.

3.2. Sistemas não-lineares

Durante as últimas duas décadas, tem havido um considerável progresso no entendimento de sistemas não-lineares, particularmente devido à aplicação de conceitos matemáticos derivados do campo da geometria diferencial (Thompson&Stewart-89)[65]. No entanto, a teoria e as ferramentas de aplicação encontram-se em seus primeiros estágios de desenvolvimento, apresentando muitas limitações e questões a serem melhor resolvidas, como, por exemplo, uma melhor formulação dos conceitos de sensibilidade, robustez e índice de desempenho. Um resultado que poderia contribuir para a solução de muitas destas questões é o desenvolvimento de uma extensão para o caso não-linear dos métodos frequenciais desenvolvidos para sistemas lineares. Este desenvolvimento poderia fornecer ferramentas gráficas e de auxílio computacional apropriadas para o projeto de controladores (Levis *et al.*-87)[40].

Uma representação não-linear de um sistema é desejável, já que um modelo linear não é capaz de representar adequadamente a dinâmica complexa de um sistema não-linear. Um sistema não-linear pode apresentar comportamentos totalmente distintos conforme seus parâmetros variam. Pelo fato de ferramentas matemáticas formais para análise de sistemas não-lineares não apresentarem um nível de desenvolvimento equiparado ao alcançado no caso de sistemas lineares, uma ferramenta primária para análise é a simulação computacional. Com o auxílio de simulação computacional, o comportamento dinâmico de um sistema pode ser explorado através da variação de parâmetros e de condições externas.

3.2.1. Diferenças entre sistemas lineares e não-lineares

A dinâmica de um sistema não-linear pode ser muito complexa. Geralmente, este comportamento se deve à presença de não-linearidades na equação, e não à interação complexa de muitas equações diferenciais. Para ilustrar esta afirmação e demonstrar as diferenças existentes entre sistemas lineares e não-lineares, é possível considerar um modelo simples de evolução populacional (modelo logístico) dado na forma (Heermann-92)[30]:

$$x_{t+1} = r\, x_t(1-x_t),$$

onde x representa a população e r é um parâmetro de controle que representa o ambiente. Esta equação representa a mudança sazonal na população de uma espécie com um ciclo anual de reprodução. Ele determina a população do próximo ciclo x_{t+1} a partir da população atual x_t e do efeito ambiental r.

Esta equação não-linear ilustra muitos fenômenos interessantes que ocorrem em sistemas não-lineares ao apresentar mudanças de comportamento abruptas quando o parâmetro de controle r é alterado. A figura 3.1 a seguir apresenta o comportamento

populacional com relação a alterações no parâmetro r. Pode-se observar que a população evolui de um estado estacionário para bifurcação, e de bifurcação para caos.

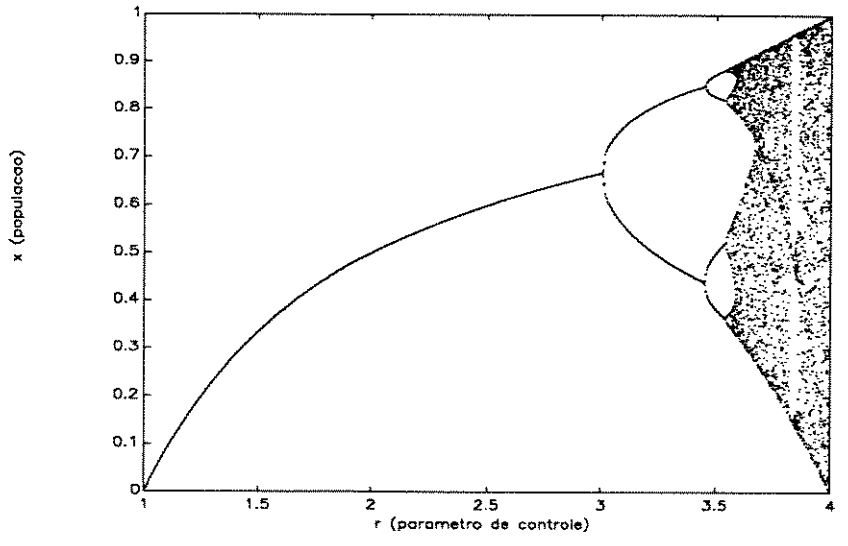


Figura 3.1 - Efeito da variação de parâmetros em um modelo logístico

Na figura 3.1, a região compreendida entre $1 \leq r \leq 3$ é uma região de estado estacionário. Para qualquer valor inicial $0 \leq x \leq 1$, a população x vai apresentar uma convergência assintótica para um valor constante e dependente de r. Por exemplo, para um fator de controle ambiental $r=2$, a população de estado estacionário vai assumir o valor $x=0.5$. Este comportamento também é apresentado por uma equação diferencial linear de primeira ordem.

Para $r>3$, o comportamento dinâmico da população se modifica, não mais apresentando um único valor de estado estacionário. Na região compreendida entre $3<r<3,4$, para qualquer valor inicial $0 \leq x \leq 1$, a população x vai convergir para um ciclo limite de dois pontos. Estes dois valores de convergência são constantes e dependentes de r. Esta transição de um único valor para dois valores de estado estacionário é conhecida como bifurcação. Uma única equação diferencial linear de primeira ordem não pode representar este comportamento oscilatório. Para representar os dois estados estacionários de população em termos de uma formulação linear, é necessária a utilização de um sistema de duas equações diferenciais lineares de primeira ordem. Portanto, no caso linear, a maior complexidade provém da interação de múltiplas equações diferenciais, enquanto que no caso não-linear, pode resultar da simples existência de uma não-linearidade em uma única equação de primeira ordem.

Retornando à figura 3.1, para $r \geq 3,4$, a população passa a convergir para um ciclo limite de quatro pontos. O processo de bifurcação se acelera à medida que r continua a aumentar. Finalmente, os valores de convergência da população passam a apresentar um

comportamento aleatório. Este comportamento aleatório originado a partir de uma equação determinística é denominado caos, não sendo encontrado em nenhum tipo de sistema linear.

O fenômeno de caos, contudo, não é simplesmente uma curiosidade matemática. Caos determinístico tem sido encontrado em muitos sistemas físicos, como reações químicas e turbulência em fluidos. Na verdade, comportamentos dinâmicos caóticos são comuns e em muitos casos comportamentos aparentemente aleatórios são, na realidade, fenômenos caóticos.

3.2.2. Atratores

Se trajetórias não-perturbadas de estados de um sistema dinâmico aproximam um subconjunto de um espaço de estados, então este subconjunto é denominado um atrator. Um atrator pode ser um único ponto, um grupo de pontos, uma única trajetória ou um lugar geométrico de trajetórias. Quando um atrator é um único ponto, ele é denominado atrator de ponto fixo. Sendo uma única trajetória, é denominado ciclo limite. Se o atrator engloba trajetórias caóticas, é denominado atrator estranho.

Atratores são geralmente estudados via análise de trajetórias de um sistema dinâmico no espaço de fase, permitindo visualizar a estrutura do atrator. Por exemplo, o movimento em torno de um atrator de ponto fixo geralmente se assemelha a uma espiral que converge para o ponto de atração.

3.3. Estabilidade de sistemas dinâmicos

A estabilidade de um sistema é uma propriedade muito importante. No caso de sistemas físicos, a instabilidade vai significar, na melhor das hipóteses, um comportamento imprevisível e, na pior das hipóteses, uma falha irreversível e muitas vezes catastrófica. Para enfatizar este fato, B. Kuo classificou os sistemas dinâmicos como sendo viáveis para aplicação, se estáveis, e inviáveis, se instáveis (Kuo-82)[38]. Esta classificação é particularmente significativa no caso dos controladores automáticos. Um controlador pode prover estabilidade para um sistema instável ou desestabilizar um sistema estável. Portanto, estabilidade é um tópico importante no projeto de qualquer tipo de controlador.

Com o objetivo de apresentar uma definição formal do conceito de estabilidade em sistemas dinâmicos, considere um sistema dinâmico autônomo descrito pela seguinte equação diferencial:

$$\frac{dx}{dt} = f(x,t) \ , \quad x(t_0) = x_0 \tag{3.1}$$

onde x é o vetor de estados e t é o tempo. Então, x^e é um ponto de equilíbrio do sistema se $f(x^e,t)=0$ para todo t, ou seja, o sistema não altera o seu estado na ausência de forças externas resultantes. Definindo os conjuntos $S(\mu)$ e $S(\lambda)$ na forma:

$$S(\mu) = \left\{ x(t) \in \Re^n : \left\| x(t) - x^e \right\| \leq \mu,\ \mu \geq 0,\ t \geq t_0 \right\}$$

$$S(\lambda) = \left\{ x_0 \in \Re^n : \left\| x_0 - x^e \right\| \leq \lambda,\ \lambda \geq 0 \right\}$$

então, x^e é estável no sentido de Lyapunov se qualquer trajetória que tenha origem em S(λ) permanece em S(μ) quando t→∞ (veja figura 3.2). Além disso, x^e é assintoticamente estável se qualquer trajetória com origem em S(λ) converge para x^e, sem abandonar S(μ), para t→∞.

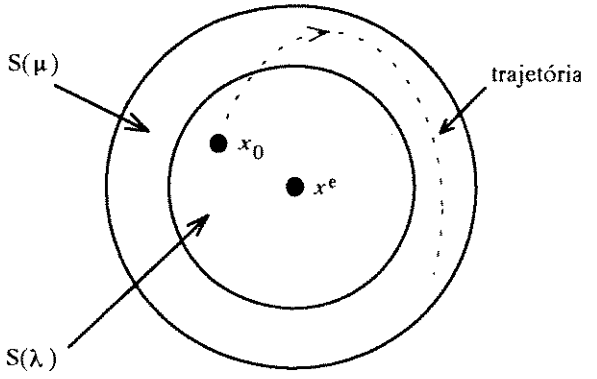


Figura 3.2 - Estado de equilíbrio no sentido de Lyapunov

No caso de sistemas lineares, existem muitos métodos para se determinar estabilidade. Técnicas como Routh-Hurwitz, Nyquist, Lugar das Raízes e Diagramas de Bode são métodos muito conhecidos (Ogata-82)[54].

No entanto, provar estabilidade no caso de sistemas não-lineares é uma tarefa muito mais difícil. Uma técnica empregada é baseada no segundo método de Lyapunov, um método direto para investigação da estabilidade da solução de uma equação diferencial não-linear. O equilíbrio será estável se for possível encontrar uma função real V(x) no espaço de estados cujas curvas de nível englobem o ponto de equilíbrio tal que as derivadas das variáveis de estado sempre apontem para o interior destas curvas de nível. O princípio básico está ilustrado na figura 3.3, onde o vetor de estados é de dimensão dois.

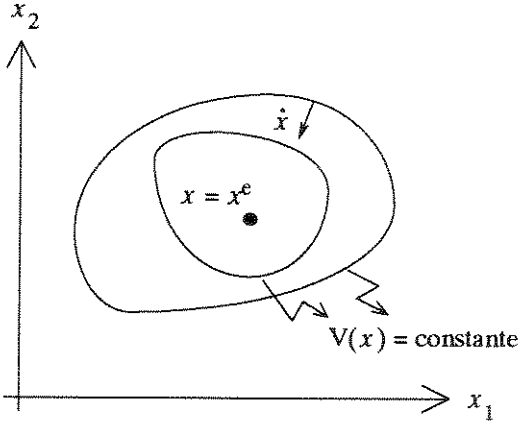


Figura 3.3 - Ilustração do método de Lyapunov para investigação de estabilidade

A demonstração da estabilidade é obtida analisando-se o comportamento da função de Lyapunov no tempo. Utilizando os conhecimentos atualmente disponíveis, encontrar uma função de Lyapunov para um sistema não-linear pode ser muito difícil. Existem técnicas para algumas classes de sistemas, mas nenhum método geral para todos os sistemas.

No caso dos sistemas reais, a demonstração de estabilidade é ainda mais complicada, já que muitos sistemas reais apresentam não-linearidades que não são modeladas matematicamente. Além disso, os parâmetros presentes no modelo matemático estão sujeitos a erros de medida. Devido a estes problemas, a prova de estabilidade para modelos matemáticos não garante a estabilidade para sistemas físicos reais. Portanto, nestes casos, a estabilidade é, de certa forma, mais uma questão de projeto e experimentação que propriamente de demonstração analítica.

3.4. Sistemas dinâmicos não-estacionários

Um sistema dinâmico não-estacionário é um sistema que possui parâmetros estruturais que variam com o tempo. Por exemplo, no caso do sistema descrito pela equação logística (3.1), se o parâmetro r variar com o tempo, o sistema será não-estacionário. No caso da descrição geral via equações de estado, a não-estacionariedade de um sistema irá se refletir na presença de parâmetros variantes com o tempo na matriz de estado[1].

Boa parte das alterações ocorridas em sistemas dinâmicos, como, por exemplo, o decréscimo de massa de um foguete em vôo, são atribuídas à evolução no tempo das variáveis de estado. Apesar disso, existem outras alterações, como, por exemplo, a variação do desempenho de um motor como função da temperatura, que são considerados efeitos de não-estacionariedade do sistema (veja Capítulo 6).

Sistemas físicos reais sempre apresentam parâmetros não-estacionários, diferindo apenas no grau de variação. O aspecto importante no contexto de sistemas de controle é como considerar estas variações de parâmetros no projeto dos controladores, ou seja, projetar controladores capazes de operar de forma satisfatória dentro de certos níveis de variação. Os efeitos não-estacionários são geralmente compensados via realimentação de informação. No entanto, se estes efeitos são muito amplos ou não-mensuráveis por parte do controlador, então o controlador vai necessitar de reajustes de forma a assegurar desempenho satisfatório.

[1]A conceituação de matriz de estado é apresentada no Apêndice B.

3.5. Controle de sistemas físicos

Pode-se definir controle de um sistema dinâmico como a regulação do sistema para executar uma determinada tarefa. O termo controle automático é geralmente utilizado para enfatizar que o controle não envolve a atuação direta do homem.

Os métodos de controle são divididos em duas grandes classes: controle de malha aberta e controle de malha fechada. A entrada destes dois tipos de controle é representada pelo conjunto de objetivos a serem atingidos, determinados externamente ao controle. As figuras 2.4(a) e 2.4(b) ilustram as estruturas básicas destes dois métodos de controle.

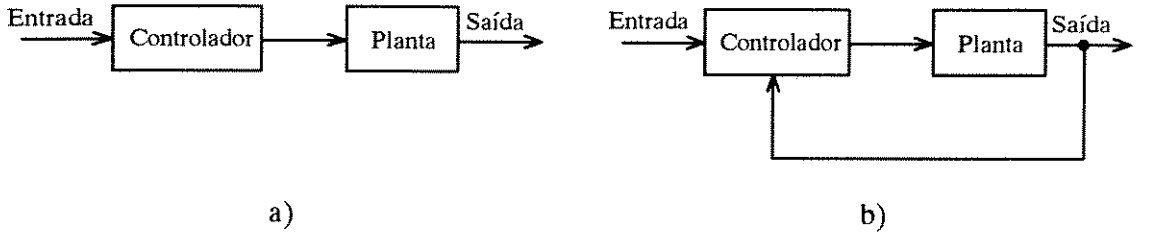


Figura 3.4 - Métodos de controle de malha aberta (a) e malha fechada (b)

O método de controle em malha aberta requer que o controlador possua um conhecimento suficiente a respeito do processo (planta) e de suas entradas para exercer o controle sem necessitar das informações referentes à saída do sistema. Já no caso do método de controle de malha fechada, o controlador utiliza informações da saída do sistema para exercer o controle, pois possui menos conhecimento prévio acerca do sistema. O controle em malha fechada, também denominado controle por realimentação, é geralmente mais adequado para anular efeitos de perturbações não-modeladas que o controle em malha aberta.

3.5.1. O conceito de realimentação em teoria de controle

Na abordagem de estratégias de controle, o conceito de realimentação é fundamental. Talvez o primeiro dispositivo a utilizar explicitamente controle por realimentação foi o relógio d'água criado por Ktesibios na antiga Alexandria. A teoria de realimentação passou a ser melhor fundamentada a partir dos estudos realizados por James C. Maxwell. Importantes resultados com relação ao entendimento e utilização de técnicas de realimentação foram obtidos a partir dos trabalhos de Black, Nyquist e Bode, no decorrer dos anos 20 (Black-77)[9]. Atualmente, métodos de controle por realimentação tornam viáveis projetos de alto desempenho, inclusive para sistemas originalmente instáveis.

A noção fundamental do conceito de realimentação contribuiu não apenas para avanços tecnológicos na área de controle, mas também auxiliou no desenvolvimento de muitas teorias em outras áreas de atuação científica, como biologia e economia. Fenômenos causais, por exemplo, são agora analisados não apenas como cadeias estáticas de causa-efeito, mas também como cadeias de causa-efeito-causa, introduzindo a noção de dinâmica na formulação deste tipo de problema.

Em sistemas de controle e tomada de decisão, o conceito de realimentação está relacionado à questão da utilização de informações a respeito da saída do sistema (efeito) para definir ou modificar a entrada (causa) com base na definição de uma tarefa a ser executada.

3.5.2. Controle clássico

As técnicas de controle clássico foram desenvolvidas por volta dos anos 30. A maior parte das técnicas se concentra na abordagem de sistemas lineares de coeficientes constantes, com uma saída e uma entrada (SISO) (Ogata-82)[54]. As ferramentas matemáticas mais aplicadas são a transformada de Laplace e a transformada-Z, com base na análise freqüencial e representação por funções de transferência.

Os controladores clássicos representam a grande maioria dos controladores em uso atualmente. Os dois tipos mais comum de controladores são o controlador por chaveamento entre limites ("bang-bang") e o controlador proporcional-integral-derivativo (PID), que ilustram o nível de sofisticação dos sistemas de controle atuais.

Um bom exemplo de controle por chaveamento entre limites é o sistema de refrigeração doméstica à base de condicionadores de ar: em modo de aquecimento, caso a temperatura atinja um valor abaixo de um certo limite inferior, o condicionador de ar é ativado até que um determinado limite superior de temperatura seja atingido; em modo de resfriamento, caso a temperatura atinja um valor acima de um certo limite superior, o condicionador de ar é ativado até que um determinado limite inferior de temperatura seja atingido.

A principal característica destes controladores é a necessidade de existência de algum tipo de histerese ou zona morta em torno do ponto de operação, de forma a prevenir ciclos de chaveamento muito rápidos. A principal vantagem é o baixo custo e a operação simples. A principal desvantagem é o fato de o sistema nunca operar exatamente no ponto desejado.

Por outro lado, os controladores PID são geralmente utilizados quando é necessária uma maior precisão que a fornecida pelos controladores por chaveamento. A ação de controle, neste caso, é o resultado da soma ponderada do erro, da derivada do erro e da integral do erro, como apresentado na figura 3.5.

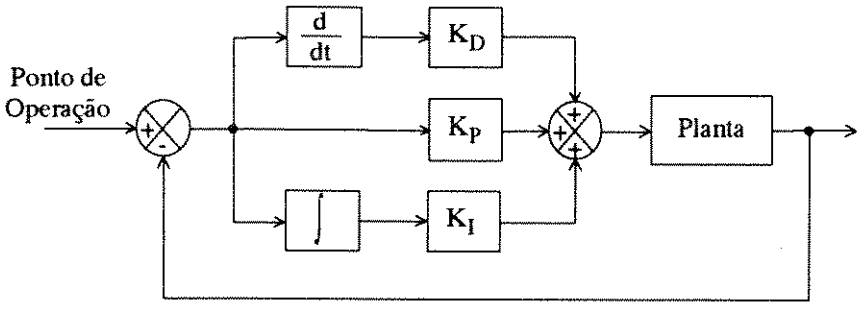


Figura 3.5 - Controlador Proporcional-Integral-Derivativo (PID)

Estes controladores são muito utilizados em indústrias de manufatura e de processamento. Controladores PID são também utilizados em conjunto com outros tipos de controladores. A sua principal vantagem é o baixo custo dos componentes e a grande familiariedade por parte dos usuários com relação a sua operação. A principal desvantagem é que a escolha dos coeficientes de ponderação pode ser difícil, além do fato de que a ocorrência de mudanças nas condições de operação acabam por exigir um reajuste no valor destes coeficientes.

3.5.3. Controle Moderno

Controle moderno se refere às técnicas de controle desenvolvidas a partir do início dos anos 60. Dentre os conceitos que influenciaram substancialmente a pesquisa na área de controle neste período, pode-se citar o princípio do mínimo (máximo) de Pontryagin, o princípio da programação dinâmica de Bellman e os filtros de Kalman-Bucy e Wiener. Combinados com os conceitos de realimentação, sensibilidade e estabilidade dinâmica, resultados abrangentes e de grande importância prática têm sido obtidos.

Associadas a estes conceitos, encontram-se as teorias de controlabilidade, observabilidade e estabilização de sistemas dinâmicos lineares, baseadas nos conceitos de espaço de estados e realização de modelos de entrada-saída. A aplicação destas teorias, embora produza resultados impressionantes, tem sido limitada devido às hipóteses restritivas de que os modelos matemáticos dos sistemas são completamente conhecidos e têm a forma de equações diferenciais (a diferenças) lineares (Levis *et al.*-87)[40].

O aspecto fundamental da teoria de controle moderno é a análise no domínio do tempo utilizando modelos de variáveis de estado. Na discussão que se segue, são enfocadas duas técnicas de controle moderno: controle ótimo e controle adaptativo. Também é incluída uma discussão sobre sistemas de controle especialistas baseado em regras.

Controle ótimo engloba uma variedade de técnicas de controle moderno, as quais têm em comum o objetivo de otimizar um índice de desempenho especificado para cada problema. Portanto, controle ótimo está baseado na aplicação de técnicas de

otimização como programação dinâmica, programação linear e cálculo variacional. Geralmente devido à sua complexidade computacional, técnicas de controle ótimo são aplicadas anteriormente à fase de operação ("off-line"). Os requisitos para implementação de técnicas de controle ótimo englobam um modelo do sistema e uma medida escalar de desempenho. O planejamento de percursos de veículos espaciais é um exemplo de aplicação de técnicas de controle ótimo.

Por outro lado, controle adaptativo combina estimação de parâmetros e controle, ambos a nível de operação ("on-line"). A idéia básica é projetar controladores auto-ajustáveis com relação a mudanças no sistema. Exemplos de controladores adaptativos incluem planejamento de ganhos, sistemas adaptativos baseados em modelos de referência, reguladores auto-ajustáveis e outros controladores com aprendizado (Äström-83)[2]; (Wittenmark&Äström-84)[79]; (Narendra&Parthasarathy-91)[51].

Já no caso de controle baseado em regras, são utilizados sistemas especialistas ou sistemas de produção no desenvolvimento de controladores. A principal característica destes sistemas que os distingue dos já citados anteriormente, é que o controle é descrito em termos linguísticos, introduzindo a possibilidade de processamento simbólico. Sistemas especialistas, no entanto, são comumente utilizados em tarefas de supervisão, regulação de controladores e no processo de adaptação de controladores adaptativos, sendo que a sua aplicação no controle direto de sistemas físicos é uma área a ser melhor explorada. Aplicações e projetos detalhados de sistemas especialistas são encontrados na literatura (Äström-86)[1]; (Bernard-88)[8].

3.5.4. Controle Adaptativo

Adaptação pode ser definida como uma mudança de comportamento com o objetivo de ajustar-se a novas circunstâncias. Intuitivamente, um controlador adaptativo é um sistema que pode modificar sua própria dinâmica em resposta a mudanças na dinâmica do processo a ser controlado e em resposta a perturbações externas. Este conceito é de muito interesse para projetistas de sistemas de controle, já que possibilita a um sistema adaptativo ajustar-se a erros no projeto de engenharia e a possíveis falhas e incertezas moderadas de componentes secundários do sistema, aumentando portanto a confiabilidade do projeto.

Apesar da evidência de que procedimentos de realimentação com ganho constante não apresentam características adaptativas, uma definição abrangente de controle adaptativo ainda não existe. O caráter um tanto vago da maioria das definições e classificações de sistemas adaptativos se deve à grande variedade de mecanismos através dos quais a adaptação pode ser conseguida e também, devido a equívocos na diferenciação entre as manifestações externas do comportamento adaptativo e os mecanismos internos

utilizados para implementá-lo (Ästrom&Wittenmark-89)[3]. Neste trabalho, controle adaptativo será considerado um caso particular de controle não-linear por realimentação.

Em sistemas de controle adaptativo, é sempre possível considerar os estados do processo que variam mais lentamente como parâmetros. Neste contexto, são evidenciadas algumas características de sistemas adaptativos que são fundamentalmente diferentes daquelas de sistemas convencionais com realimentação. Em sistemas de controle adaptativo ocorre uma medição contínua e automática das características dinâmicas do processo, uma comparação com as características dinâmicas desejadas ou previamente definidas e a utilização dos resultados desta comparação para ajuste de parâmetros variáveis do controlador ou geração de um sinal atuante de tal forma que o desempenho ótimo possa ser mantido. Neste processo, o ajuste de parâmetros pode não ser função apenas de medições realizadas, mas pode depender também da existência de características de auto-organização do sistema.

3.5.4.1. Um breve histórico

O termo controle adaptativo tem sido utilizado pelo menos desde 1950, quando W. I. Caldwell teve registrada uma patente de um sistema de controle com ajuste de resposta automático. Os primeiros progressos nesta área foram motivados por projetos de piloto-automático para aeronaves de alto desempenho, onde controladores mais sofisticados que aqueles baseados em realimentação linear com ganho constante mostravam-se necessários. Tais aeronaves operavam em uma larga faixa de velocidade e altitude, exigindo o desenvolvimento de controladores capazes de atuar em condições de operação variáveis. Os primeiros projetos nesta área eram caracterizados pela inexistência de fundamentos teóricos (Ästrom-83)[2].

Nos anos 60, muitos resultados em teoria de controle foram obtidos, incluindo o desenvolvimento da teoria de espaço de estados, teoria da estabilidade e teoria de controle estocástico. Estes resultados foram de grande importância para o desenvolvimento de técnicas de controle adaptativo. A teoria de programação dinâmica, introduzida por Bellman (57)[7], e a teoria de controle dual, introduzida por Feldbaum (60/61)[19,20,21,22], ajudaram a compreender melhor os processos adaptativos. Uma contribuição fundamental foi feita por Tsypkin (71)[66], que mostrou que muitos esquemas de aprendizado e controle adaptativo podem ser descritos por equações recursivas, nos moldes dos métodos de aproximação estocástica. Muitos progressos também foram obtidos em identificação de sistemas e estimação de parâmetros.

Atualmente, importantes resultados quanto à robustez e estabilidade do controle adaptativo já estão disponíveis, viabilizando o desenvolvimento de processos de modelagem e projeto automáticos de sistemas de controle mais complexos.

3.5.4.2. Métodos de Controle Adaptativo

A teoria de controle convencional atua predominantemente em sistemas lineares com parâmetros constantes. Esta é geralmente uma boa aproximação para sistemas de regulação com pontos de operação fixos. No entanto, esta aproximação linear com coeficientes constantes pode apresentar dificuldades em condições de operação variável, pois o desempenho do processo de controle pode não se manter adequado para conjuntos de parâmetros distintos, podendo até provocar instabilidade.

Se o processo é conhecido apenas imperfeitamente, talvez devido a parâmetros variando aleatoriamente no tempo ou a perturbações externas, então a identificação inicial e a decisão de controle associada não produzirão os resultados esperados. Dessa forma, torna-se necessária a realização dos seguintes procedimentos continuamente ou em intervalos de tempo, dependendo do tipo de variação na dinâmica do processo :

- identificação das características dinâmicas do processo
- decisão baseada na identificação do processo
- modificação ou atuação baseada na decisão tomada.

Tomando como base uma estrutura de controle convencional em malha fechada, composta por um processo e um controlador com parâmetros ajustáveis, uma representação em diagrama de blocos de um sistema geral de controle adaptativo é mostrada na figura 3.6.

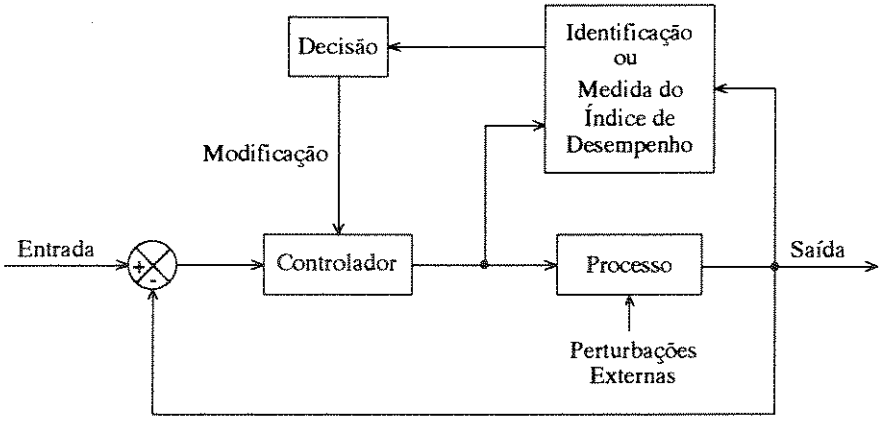


Figura 3.6 - Diagrama de blocos de um sistema de controle adaptativo

Neste sistema, o processo é identificado e o índice de desempenho medido contínua ou periodicamente. Tendo isto sido feito, o índice de desempenho é comparado com o ótimo e uma decisão é tomada. O ajuste dos parâmetros é, portanto, uma operação em malha fechada.

Com o objetivo de evidenciar os princípios envolvidos na definição do esquema geral da figura 3.6, serão apresentados a seguir três esquemas particulares de controle adaptativo de parâmetros: planejamento de ganho, controle por modelo de referência e

UNICAMP
BIBLIOTECA CENTRAL

regulador auto-ajustável. A mesma estrutura de controle convencional em malha fechada da figura 3.6 será tomada como referência, sendo que o problema central é encontrar um modo de ajuste dos parâmetros do controlador (regulador) em resposta a mudanças ocorridas no processo ou a perturbações externas. Cada um dos esquemas apresenta um política distinta de ajuste dos parâmetros.

3.5.4.2.1. Planejamento de ganho

Em algumas aplicações, é possível definir variáveis auxiliares que se correlacionam bem com as mudanças na dinâmica do processo. Para a redução dos efeitos de variação de parâmetros do processo, os parâmetros do controlador podem ser ajustados como função destas variáveis auxiliares. Dessa forma, planejamento de ganho é visto como uma técnica de controle em malha fechada em que os ganhos de realimentação são ajustados em malha aberta, como apresentado na figura 3.7.

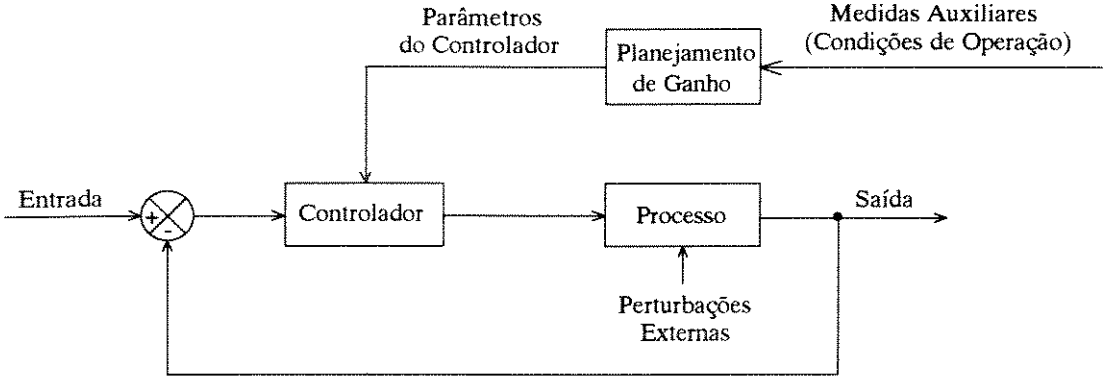


Figura 3.7 - Sistema de controle baseado em planejamento de ganho

Existe uma controvérsia quanto à consideração de sistemas com planejamento de ganho como caso particulares de controle adaptativo, já que os parâmetros do controlador são ajustados em malha aberta. Apesar disso, planejamento de ganho é uma técnica eficiente de redução dos efeitos de variação de parâmetros. Esta técnica teve sua origem no desenvolvimento de sistemas de controle de vôo, sendo empregada com sucesso em problemas de compensação de não-linearidades estáticas e dinâmicas (Ästrom&Wittenmark-89)[3]. Existem situações em que é possível obter planejamento de ganhos através da introdução de transformações não-lineares, de forma que o sistema transformado não dependa das condições de operação. As variáveis de controle transformadas são então calculadas e retransformadas antes de serem aplicadas ao processo. O controlador assim obtido pode ser considerado como uma composição de duas transformações não-lineares que atuam sobre um controlador linear.

O problema principal no projeto de sistemas com planejamento de ganho é encontrar variáveis auxiliares adequadas. Isto é geralmente realizado a partir do conhecimento da física do sistema. Quando estas variáveis auxiliares são obtidas, o controlador é ajustado ou calibrado para cada condição de operação. A estabilidade e o desempenho do sistema são geralmente avaliados através de simulações. Portanto, não há realimentação do índice de desempenho do sistema de controle em malha fechada na definição dos parâmetros do controlador. Esta é a razão pela qual muitos autores defendem a idéia de que o planejamento de ganho não deve ser considerado uma técnica de controle adaptativo, mas um caso especial de controle não-linear.

Quanto aos aspectos favoráveis e desfavoráveis da técnica de planejamento de ganho, é possível destacar:

aspectos favoráveis:

- os parâmetros do controlador podem ser ajustados rapidamente em resposta a mudanças nas condições de operação, dependendo apenas do tempo de resposta das medidas auxiliares às novas condições de operação

aspectos desfavoráveis:

- o planejamento de ganho é uma técnica de compensação em malha aberta, não existindo nenhuma forma de compensação para eventuais planejamentos incorretos
- a análise de desempenho deve ser realizada através de extensivas simulações

3.5.4.2.2. Controle adaptativo por modelo de referência

O método de controle adaptativo por modelo de referência é um dos principais procedimentos de controle adaptativo já desenvolvidos, tendo sido originalmente apresentado por Whitaker *et al.* em 1958 [72]. Os dois conceitos fundamentais envolvidos no desenvolvimento do método são:

- o desempenho desejado para o sistema é especificado por um modelo de referência;
- os parâmetros do controlador são ajustados com base no erro existente entre as saídas do modelo de referência e do processo.

A figura 3.8 apresenta uma das muitas configurações possíveis para implementação deste método, denominada configuração paralela. Pode-se observar a existência de duas malhas de realimentação, sendo que a malha interna é responsável pelo controle convencional por realimentação, e a malha externa é responsável pelo ajuste dos parâmetros do controlador presente na malha interna. Portanto, a malha interna deve apresentar uma dinâmica mais rápida que a malha externa.

Os parâmetros do controlador são ajustados no sentido de reduzir o erro (e) entre as saídas do modelo de referência e do processo, ou seja, a malha externa de realimentação também é uma malha de controle. A questão fundamental é determinar o mecanismo de ajuste de tal forma a conduzir o erro para zero, mantendo a estabilidade do sistema.

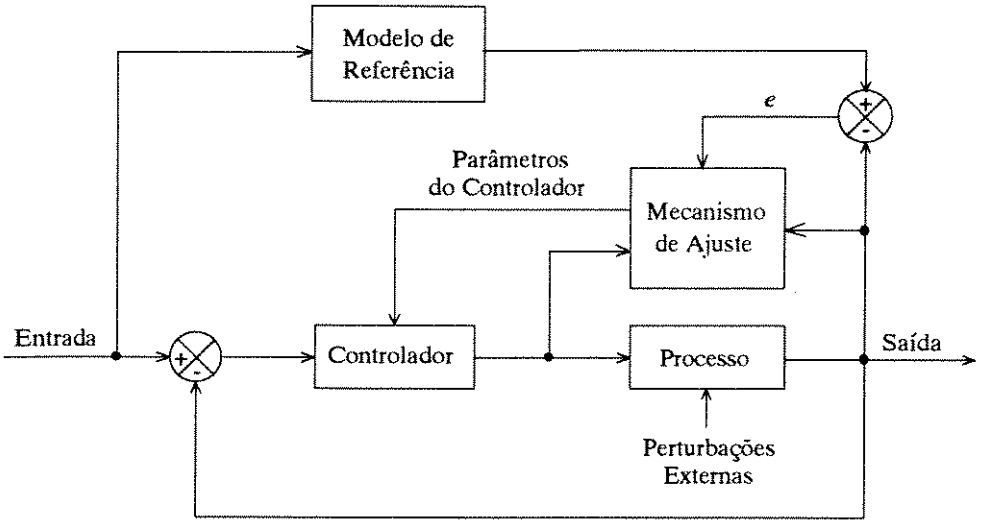


Figura 3.8 - Sistema de controle adaptativo por modelo de referência

Este problema não pode ser resolvido via realimentação linear (Ästrom& Wittenmark-89)[3], sendo necessário o emprego do método do gradiente para o ajuste de parâmetros. Este método, já utilizado no trabalho de Whitaker *et al.* (58)[72], está baseado no princípio de que a variação dos parâmetros é muito mais lenta que a variação apresentada pelas outras variáveis do sistema. Esta hipótese conduz a um tratamento quase-estacionário do sistema, sendo essencial para o cálculo das derivadas de sensibilidade necessárias para a aplicação do mecanismo de ajuste. O mecanismo de ajuste é dado na forma:

$$\frac{d\theta}{dt} = \alpha e \nabla_\theta e \tag{3.2}$$

Nesta equação, os componentes do vetor θ são os parâmetros ajustáveis do controlador e os componentes do vetor gradiente $\nabla_\theta e$ são as derivadas de sensibilidade do erro com relação aos parâmetros ajustáveis, calculadas a partir das entradas e saídas do processo. O número α representa a taxa de adaptação. No sentido de reduzir o erro (e), ajusta-se os parâmetros na direção do gradiente negativo, sendo que a equação (3.2) assume a forma:

$$\theta(t) = -\alpha \int^t e(s) \nabla_\theta e(s) ds \tag{3.3}$$

O mecanismo de ajuste pode ser considerado como uma composição de três módulos: um filtro para o cálculo das derivadas de sensibilidade a partir das entradas e

saídas do processo, um multiplicador e um integrador. Esta configuração é típica de muitos sistemas adaptativos e apresenta uma forte analogia com algoritmos de treinamento para redes neurais, apresentados no Capítulo 2.

Quanto aos aspectos favoráveis e desfavoráveis da técnica de controle adaptativo por modelo de referência, é possível destacar:

aspectos favoráveis:

- é aplicável a sistemas não-lineares
- é aplicável a sistemas parcialmente modelados

aspectos desfavoráveis:

- não é possível garantir a estabilidade do sistema em malha fechada. O ajuste tem um bom desempenho para taxas de adaptação (α) pequenas, mas esta taxa é também função da magnitude do erro e do ganho do processo, não sendo possível estabelecer limites que garantam estabilidade. Técnicas derivadas da teoria de estabilidade podem ser aplicadas na derivação de outras regras de ajuste (Ästrom&Wittenmark-89)[3].

3.5.4.2.3. Regulador auto-ajustável

O regulador auto-ajustável é uma estrutura de controle adaptativo baseada na separação entre a estimação de parâmetros desconhecidos e o projeto do controlador, tendo sido originalmente proposto por Kalman (58)[34]. A estrutura básica é ilustrada na figura 3.9.

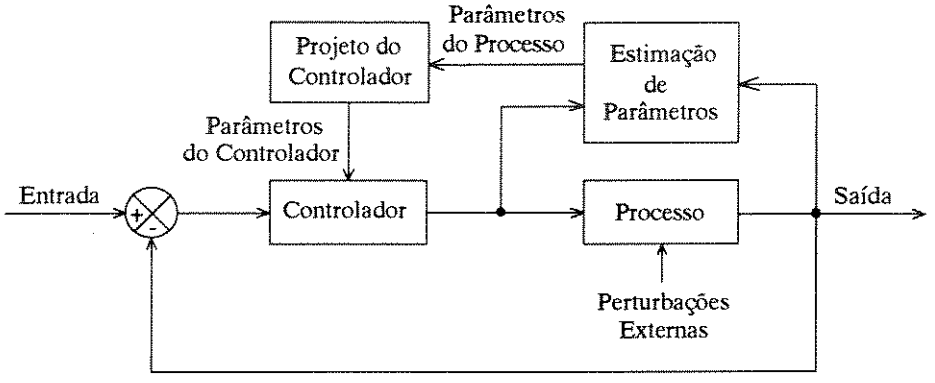


Figura 3.9 - Regulador auto-ajustável

Os parâmetros desconhecidos do processo são estimados durante a operação do sistema, sendo que os valores estimados são utilizados no ajuste dos parâmetros do controlador, desprezando-se os erros de estimação. Este procedimento é denominado princípio da equivalência certa. Diferentes esquemas de estimação podem ser utilizados,

incluindo aproximação estocástica, mínimos quadrados, variável instrumental e princípio da máxima verossimilhança. Métodos recursivos de estimação são os mais empregados. O bloco de projeto do controlador representa uma solução em tempo real do problema de controle para parâmetros do processo conhecidos. Os métodos de projeto do controlador que podem ser utilizados incluem método de variância mínima, alocação de pólos e seguidor de modelo, sendo que a escolha do método depende das especificações do sistema em malha fechada. Combinações diferentes de métodos de estimação e controle conduzem a sistemas de controle com propriedades distintas.

O controlador apresentado na figura 3.9 é denominado regulador auto-ajustável explícito, pois os parâmetros do controlador são ajustados indiretamente através da estimação de um modelo explícito do processo. No entanto, é geralmente possível re-parametrizar o processo de tal forma que ele possa ser expresso em termos dos parâmetros do controlador. Desta forma, os parâmetros do controlador podem ser ajustados diretamente, com base em um modelo implícito do processo, sendo que o controlador passa a ser denominado regulador auto-ajustável implícito.

Quanto aos aspectos favoráveis e desfavoráveis do regulador auto-ajustável, é possível destacar:

aspectos favoráveis:

- flexibilidade
- adequado para implementação computacional

aspectos desfavoráveis:

- é aplicável apenas a sistemas lineares
- exige o conhecimento exato da ordem do processo

3.5.4.3. Resultados recentes da teoria de controle adaptativo

Nesta seção, serão discutidos resultados importantes obtidos na área de controle adaptativo. Particularmente, os resultados a serem apresentados demonstram que os objetivos desta dissertação são teoricamente possíveis de serem atingidos.

A idéia central presente no desenvolvimento de controladores mais avançados é aperfeiçoar o desempenho do controle pela incorporação de mais conhecimento acerca do sistema físico a ser controlado. Dependendo da quantidade de conhecimento incorporada pelo controlador, ele pode predizer ou se antecipar ao sistema físico, habilidade importante no cumprimento de determinadas tarefas de controle. Neste contexto, um controlador adaptativo é simplesmente um controlador capaz de adquirir conhecimento acerca do sistema físico em operação.

Portanto, para que um controlador adaptativo seja factível, ele deve ser capaz de predizer direta ou indiretamente o comportamento do sistema físico por intermédio da obtenção de informações a respeito deste sistema, além de possuir dimensões adequadas para implementação. Para sistemas lineares, estes requisitos foram demonstrados e testados com sucesso. Já no caso de sistemas não-lineares, fenômenos como bifurcação e caos trazem dúvidas com relação à capacidade de predição de comportamento dinâmico. Nas discussões que se seguem, são apresentadas evidências de que controle adaptativo de sistemas não-lineares é um procedimento factível.

A visão clássica de sistemas atribui comportamentos complexos a sistemas com um número elevado de graus de liberdade. No entanto, os resultados proveniente da teoria de caos indicam que mesmo sistemas com poucos graus de liberdade podem apresentar comportamentos muito complexos. Mais importante ainda é a constatação de que sistemas físicos de dimensão elevada podem se auto-organizar de forma que o comportamento global do sistema possa ser descrito por um modelo de ordem reduzida. Particularmente, sistemas de dimensão infinita que são analisados através de equações diferenciais parciais não-lineares apresentam atratores de dimensão finita. Por exemplo, sistemas de fluxo de fluido possuem espaço de estados de dimensão infinita, mas seu modelo de movimento pode ser representado por atratores de dimensão finita (Heermann-92)[30]. Este fator é importante por dois motivos:

- representa uma visão diferente da natureza de não-linearidades complexas;
- controladores não-lineares acabam sendo capazes de representar tais fenômenos.

Assim, processos de predição de sistemas de dimensões elevadas não requerem o cálculo e a armazenagem de todos os graus de liberdade do sistema. Já que um modelo matemático não-linear de ordem finita pode predizer o comportamento de um sistema físico de dimensão infinita, um controlador não-linear apresenta-se como uma alternativa viável para predição de sistemas e aperfeiçoamento da ação de controle.

Um outro fator importante refere-se à determinação ou reconstrução de um atrator. Pode-se demonstrar que o limite inferior da dimensão do vetor de entrada necessário para reconstruir um atrator é igual à dimensão do atrator (Heermann-92)[30]. Lapedes e Farber (87)[39] demonstraram que uma rede neural pode aproximar suficientemente um atrator para efeito de predição. Na verdade, Lapedes e Farber também demonstraram que a predição utilizando redes neurais é muito mais precisa que a fornecida por outros métodos convencionais. Isto significa que um atrator pode ser reconstruído com a utilização de um controlador adequadamente dimensionado.

Um último resultado importante refere-se à questão da observabilidade de sistemas dinâmicos. Um sistema é observável se os estados internos do sistema podem ser determinados a partir do conhecimento das entradas e saídas do sistema (veja Apêndice B). Assumindo que as entradas são fornecidas pelo controlador e, portanto, são conhecidas,

pode-se concluir que a observabilidade representa o acoplamento entre os estados internos e as saídas do sistema. Packard *et al.* (80)[55] demonstrou que um atrator pode ser reconstruído a partir de uma única coordenada, através da utilização de atrasos de tempo. Assim, mesmo quando nem todos os estados do sistema puderem ser medidos, o atrator pode ser reconstruído com o acréscimo junto ao vetor de entrada de valores dos estados mensuráveis defasados no tempo. Este resultado é importante, pois nem sempre os estados que participam na formação do atrator são conhecidos ou mensuráveis.

Estes resultados ajudam a demonstrar que controle adaptativo de sistemas não-lineares é possível. Apesar da complexidade dos sistemas não-lineares, um controle mais eficiente pode ser obtido pela determinação ou reconstrução do atrator do sistema físico e sua utilização na predição do comportamento do sistema.

3.6. Necessidades atuais e perpectivas futuras na área de sistemas de controle

Os projetistas de sistemas de controle defrontam-se atualmente com a necessidade de expandir seus conceitos e métodos de modo a torná-los aplicáveis a sistemas cujos modelos são incompletos ou inicialmente mal definidos, mas que podem ser aperfeiçoados em fase de operação. Em outras palavras, estes projetistas buscam incorporar a capacidade de modelagem como parte do projeto do sistema. Dessa forma, os modelos passam a ser considerados como entidades em evolução, a partir de uma estrutura inicial fornecida.

Ao mesmo tempo, novas classes de modelos precisam ser desenvolvidas, envolvendo não apenas dados analíticos e numéricos, mas também qualitativos e simbólicos. Paralelamente ao surgimento de novas classes de modelos, métodos de controle precisam ser estendidos a uma ampla classe de modelos já tradicionais, como: modelos não-lineares, estocásticos e com parâmetros distribuídos.

Em síntese, é possível constatar-se um aumento de demanda nos seguintes aspectos:

- **maior flexibilidade e autonomia**: para sistemas de controle que operam com pouca ou nenhuma participação humana, flexibilidade e operação autônoma são necessárias, já que, na maior parte dos casos, todas as situações não podem ser previstas.
- **menor custo de operação em termos de energia, recursos e ambiente**: uma maior eficiência de operação é importante não apenas para os sistemas de controle futuros, mas também para os atualmente em operação. O sistema de injeção de combustível dos automóveis modernos é um bom exemplo: maior potência de saída e menor emissão de gases são o resultado de um controle mais eficiente do fluxo de combustível.

- **custo de manutenção decrescente**: uma ação de controle mais eficiente pode reduzir o custo de manutenção tanto do sistema controlado como do controlador, pois reduz a ocorrência de condições de operação indesejáveis, como superaquecimento e outros efeitos provocados por saturação de componentes. A manutenção do controlador pode também ser reduzida caso este apresente uma larga faixa de operação, pois reduz a necessidade de reajuste de seus parâmetros para atender a mudanças no processo.
- **aumento da faixa de operação**: atualmente, muitos controladores são projetados via linearização em torno de um dado ponto de operação. A linearização resulta em sérios problemas caso a operação se desloque para outros pontos. O controlador de vôo dos ônibus espaciais é um exemplo de limite da faixa de operação, já que vários pousos e decolagens já foram adiados até que as condições atmosféricas, como a velocidade do vento, estivessem dentro de um intervalo de valores aceitável.
- **operação robusta para sistemas não-lineares sujeitos a perturbações de diversos tipos**: a necessidade de controle de sistemas não-lineares sujeitos a altos índices de perturbação já é uma realidade em áreas como a exploração espacial. Neste caso, a operação robusta de sistemas de controle é condição essencial para a realização de novas descobertas e o desenvolvimento de novas teorias.

Na verdade, cientistas e engenheiros da área de controle estarão sempre envolvidos na solução de problemas cada vez mais complexos. As pesquisas futuras devem fornecer as metodologias, teorias e algoritmos que tornem possível o planejamento, gerenciamento e controle de sistemas de complexidade sem precedentes. É fundamental que as pesquisas na área de controle sejam encorajadas, pois é a única forma de garantir que a teoria e as ferramentas necessárias para sustentar os avanços tecnológicos futuros estejam disponíveis.

Capítulo 4

Redes neurais aplicadas à identificação e controle de sistemas dinâmicos

Sistemas dinâmicos de tempo discreto podem ser construídos utilizando-se os operadores de soma, multiplicação e atraso de tempo. Redes neurais estáticas realizam as duas primeiras operações, além de uma operação não-linear conhecida. Com a inclusão do operador de atraso de tempo, as redes neurais podem ser utilizadas em tarefas de identificação e controle de sistemas dinâmicos lineares e não-lineares de interesse prático. O fato de ser utilizada apenas uma função não-linear conhecida na ativação das unidades processadoras faz das redes neurais sistemas interessantes do ponto de vista de aplicação em contextos dinâmicos. Além disso, por causa de sua habilidade na aproximação de funções e classificação de padrões e seu potencial de processamento paralelo, redes neurais artificais parecem ser capazes de implementar muitas das funções essenciais para o controle de sistemas com alto grau de autonomia.

Os dois fatores fundamentais envolvidos na formulação da solução de problemas via aplicação de redes neurais são a definição da arquitetura da rede e do algoritmo de treinamento associado. Redes neurais recorrentes e não-recorrentes têm sido aplicadas com sucesso em inúmeros problemas de controle, sendo que a escolha do tipo de arquitetura mais adequada vai depender da natureza da aplicação e dos objetivos a serem atingidos. Algoritmos de treinamento supervisionado são mais comumente empregados, e estão baseados na disponibilidade de informação para treinamento. Nos casos em que as informações necessárias para treinamento não estão disponíveis, algoritmos de treinamento não-supervisionado são empregados.

A seguir, é apresentada uma série de esquemas gerais de identificação e controle utilizando redes neurais, onde são enfatizados vários tipos de formulações já desenvolvidas para o tratamento de sistemas dinâmicos. Com a definição destes esquemas envolvendo sistemas dinâmicos e redes neurais conectados em diversas configurações, ou seja, com a especificação das tarefas a serem executadas pelas redes neurais em cada caso, torna-se necessário estabelecer os critérios de implementação e as arquiteturas de redes neurais mais adequadas.

4.1. Configurações baseadas na aplicação de algoritmos de treinamento supervisionado

Boa parte dos problemas de controle requerem a definição de sinais de comando que conduzam um processo a apresentar um comportamento desejado. O conhecimento envolvido na definição do comportamento desejado deve ser utilizado no desenvolvimento do sistema de controle. A questão que surge diz respeito à procedência desta informação, ou seja, à possibilidade de aplicação direta deste conhecimento no projeto do controlador.

No caso da utilização de redes neurais, este conhecimento é adquirido através da aplicação de algoritmos de treinamento baseado no ajuste de parâmetros da rede neural. Como o ajuste de parâmetros durante o processo de treinamento supervisionado é realizado com base na informação disponível imediatamente na saída da rede (veja Capítulo 2), a figura 4.1 a seguir ilustra um problema que pode surgir em aplicações de controle utilizando redes neurais, descrito na forma (Barto-89)[5]:

Problema de controle: Dado um processo que não pode ser devidamente controlado através da aplicação de técnicas convencionais de controle, seja por seu alto grau de complexidade ou devido à existência de poucas informações a respeito de seu comportamento dinâmico, treinar uma rede neural de forma que ela aprenda a controlar este processo com base em algum critério de desempenho.

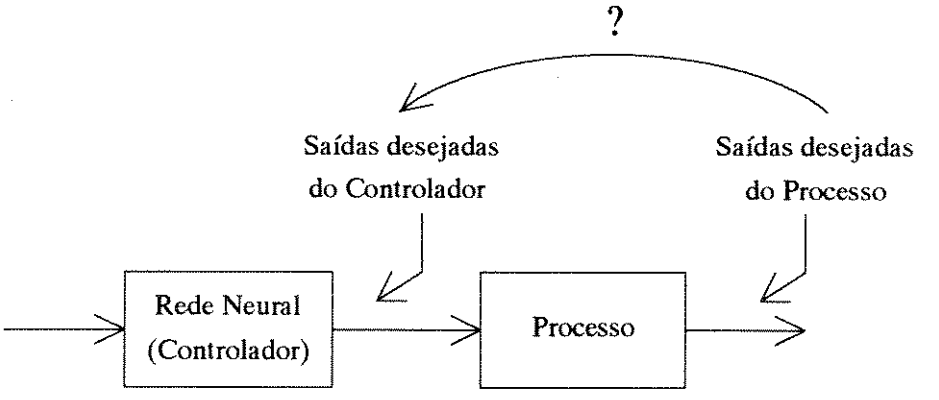


Figura 4.1 - O problema da procedência de informação para treinamento

Em um problema típico de controle, geralmente estão disponíveis as saídas desejadas do processo e não as saídas desejadas do controlador. Como o processo de ajuste dos parâmetros da rede neural está baseado na definição de uma função objetivo a partir de informações disponíveis diretamente na saída da rede, parece ser evidente que o conhecimento das saídas desejadas do controlador é, a princípio, fundamental no processo de geração dos padrões para treinamento. Deste modo, fica estabelecida uma contradição, já que as saídas desejadas do controlador representam a solução do problema e ao mesmo tempo devem ser utilizadas na geração da solução. Portanto, a pergunta que surge é como fornecer os padrões para treinamento da rede neural na ausência de informações a respeito das saídas desejadas do controlador.

A solução deste impasse pode ser obtida por intermédio da análise do problema representado pelo ajuste dos pesos das camadas intermediárias de uma rede neural multi-camadas. Como os padrões para treinamento estão definidos apenas na saída da rede, dada qualquer camada intermediária, a porção da rede neural interposta entre esta camada e a camada de saída é análoga ao bloco correspondente ao processo da figura 4.1. O problema

da determinação dos padrões de treinamento para uma camada intermediária a partir dos padrões definidos na camada de saída é, portanto, análogo ao problema de treinamento de um controlador com base no comportamento desejado para a saída do processo. Logo, pode-se concluir que os mesmos procedimentos aplicados a um caso também serão aplicáveis ao outro.

Estes procedimentos, assim como outras formas de obtenção da informação necessária para treinamento de redes neurais em tarefas relevantes para controle são descritos nas próximas seções.

4.1.1. Emulação de controladores existentes

Caso já exista um dispositivo capaz de controlar o processo, então a informação necessária para treinar a rede neural pode ser obtida deste controlador, conforme apresentado na figura 4.2. Dada uma entrada para a rede neural, a saída desejada é igual à saída do controlador submetido à mesma condição de entrada, ou seja, a rede neural aprende a emular o controlador. Naturalmente é necessário justificar a obtenção de um controlador baseado em redes neurais a partir de um controlador efetivo previamente existente. Uma situação em que a adoção deste procedimento faz-se necessária é dada quando o controlador existente não pode ser utilizado na prática. Uma outra situação ocorre quando a estrutura de controle implementada pela rede neural apresenta-se mais adequada para aplicação que a estrutura de controle previamente existente.

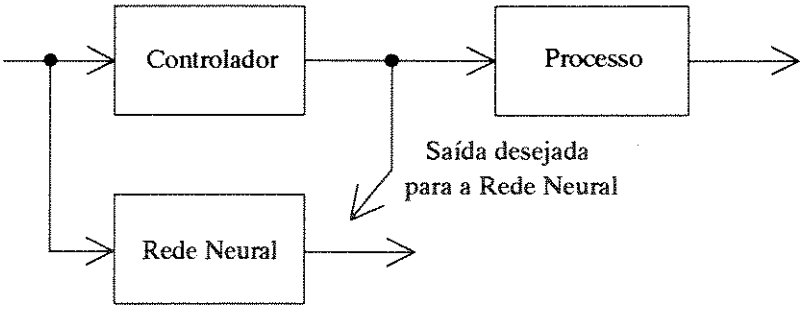


Figura 4.2 - Emulação de um controlador existente utilizando redes neurais

4.1.2. Preditor adaptativo

A figura 4.3 ilustra a idéia básica de treinar uma rede neural para a tarefa de predição. Assume-se que as linhas na parte superior da figura conduzem sinais cujos valores devem ser previstos. Para propósitos de treinamento, as entradas da rede consistem dos valores desses sinais defasados no tempo, e as saídas desejadas correspondem aos seus valores atuais. Dessa forma, a rede neural deve ser capaz de predizer os valores atuais através da aproximação de uma função dos valores defasados no tempo. O esquema

apresentado na figura 4.3 é apenas o caso mais simples de predição baseado na construção do vetor de entrada a partir de defasagens no tempo de sinais de entrada. O tratamento de modelos auto-regressivos, por exemplo, exige a utilização das defasagens dos sinais de saída na composição do vetor de entrada. O preditor adaptativo é geralmente aplicado à identificação de sistemas discretos e predição de séries temporais.

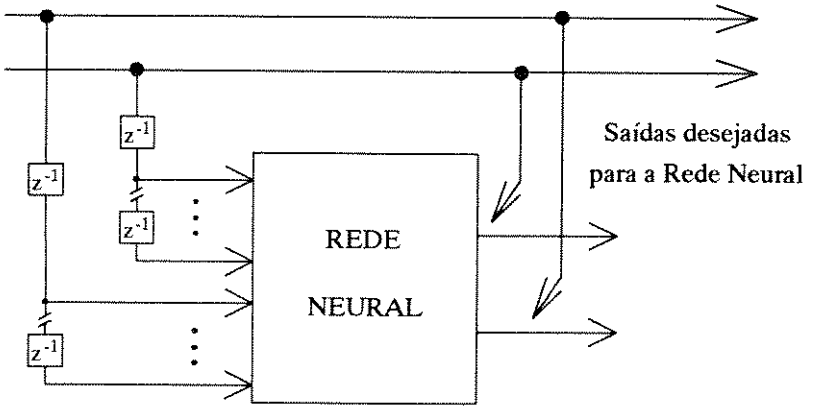


Figura 4.3 - Predição adaptativa utilizando redes neurais

4.1.3. Identificação de sistemas dinâmicos

Em tarefas de identificação de sistemas dinâmicos, a informação para treinamento pode ser obtida a partir da observação do comportamento de entrada-saída do processo, como sugerido na figura 4.4. Neste caso, a rede neural recebe a mesma entrada que o processo e a saída desejada para a rede é dada pela saída do processo. Outros tipos de identificação são também empregados, não se restringindo ao caso do mapeamento de entrada-saída, sendo que o preditor adaptativo descrito na seção anterior (figura 4.3) pode ser considerado um caso particular de identificação de sistemas.

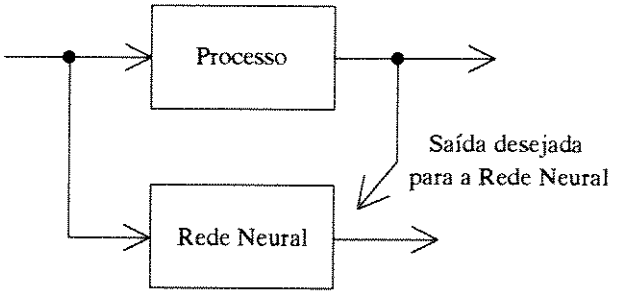


Figura 4.4 - Identificação de sistemas utilizando redes neurais

Identificação de sistemas é uma etapa fundamental na implementação de muitos métodos de controle adaptativo. Por exemplo, dado um conjunto de classes de modelos para identificação e um correspondente conjunto de controladores projetados de acordo

com as especificações de cada classe, caso o processo possa ser adequadamente identificado como pertencente a uma destas classes, uma lei de controle efetiva estará sempre disponível. Esta é uma versão discreta para o caso de adaptação estrutural de controladores em tempo de operação para atender a alterações significativas na dinâmica do processo (Narendra&Mukhopadhyay-92)[50].

É importante observar que a capacidade de identificação de classes de sistemas não-lineares apresentada pelas redes neurais (veja Apêndice A) não implica na capacidade de controle destes sistemas. Portanto, é necessário desenvolver técnicas de projeto de controladores que se apliquem às classes de sistemas não-lineares identificáveis pelas redes neurais.

4.1.4. Identificação da dinâmica inversa de um sistema não-linear

A figura 4.5 ilustra a utilização de redes neurais na identificação da dinâmica inversa de um processo não-linear, já que para o caso linear soluções analíticas geralmente estão disponíveis. Em contraste com a situação apresentada na figura 4.4, neste caso a entrada da rede é a saída do processo, e os valores desejados de saída para a rede correspondem às entradas do processo.

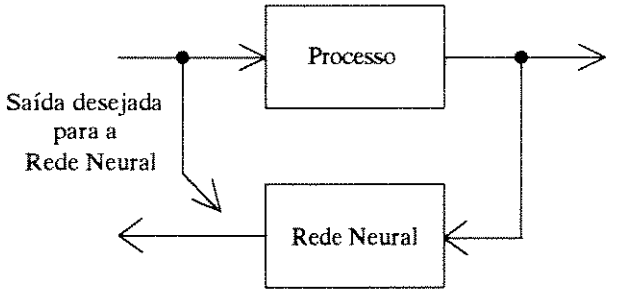


Figura 4.5 - Identificação da dinâmica inversa de um sistema utilizando redes neurais

Tendo sido treinada, a rede neural pode ser utilizada para propósitos de controle, conforme apresentado na figura 4.6. A saída desejada para o processo é fornecida à entrada da rede neural, cuja saída resultante é fornecida como entrada do processo. Pelo fato da rede neural representar a dinâmica inversa, a saída da rede vai fazer com que o processo produza o comportamento de saída desejado.

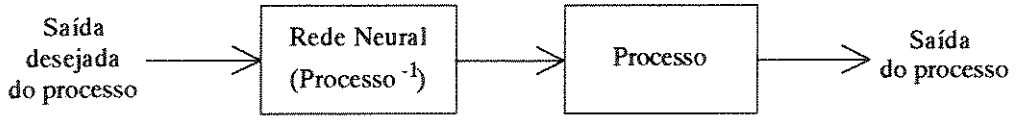


Figura 4.6 - Controle de um processo com base na sua dinâmica inversa

Um dos problemas deste procedimento de identificação ocorre quando mais de uma entrada do processo produz a mesma saída, ou seja, quando a dinâmica inversa do processo não é bem comportada. Neste caso, a rede neural, como também outros métodos de estimação, vai procurar responder com o valor médio das diferentes saídas desejadas correspondentes a uma mesma entrada, produzindo um mapeamento que não corresponde necessariamente à dinâmica inversa do processo.

4.1.5. A obtenção de um modelo diferenciável para o processo

O método a ser apresentado nesta seção ilustra uma aplicação aos problemas de controle da identificação de sistemas utilizando redes neurais. Este é um dos métodos mais gerais de solução para o problema de procedência de informação para treinamento, discutido anteriormente (veja figura 4.1). Neste caso, os modelos de identificação obtidos são utilizados de uma forma diferente daquela empregada em métodos convencionais de controle adaptativo (veja Capítulo 3). O procedimento de identificação da dinâmica do processo permitindo a geração do modelo diferenciável é o mesmo apresentado na figura 4.4. Como ilustrado na figura 4.7, o algoritmo de retro-propagação é empregado no cálculo das derivadas das saídas do modelo com relação às suas entradas, possibilitando a propagação do erro existente entre as saídas do modelo e as saídas desejadas de volta para a entrada do modelo. Este procedimento permite a obtenção do erro na saída do controlador, que pode ser utilizado para treinar uma outra rede neural para a execução da tarefa de controle. Este método de diferenciação de um processo foi originalmente proposto por Werbos (87)[71]. Observe-se que quando o modelo de referência realiza a função identidade, este procedimento pode também ser aplicado em substituição à identificação direta da dinâmica inversa do processo (veja figura 4.5), principalmente quando esta dinâmica é mal comportada.

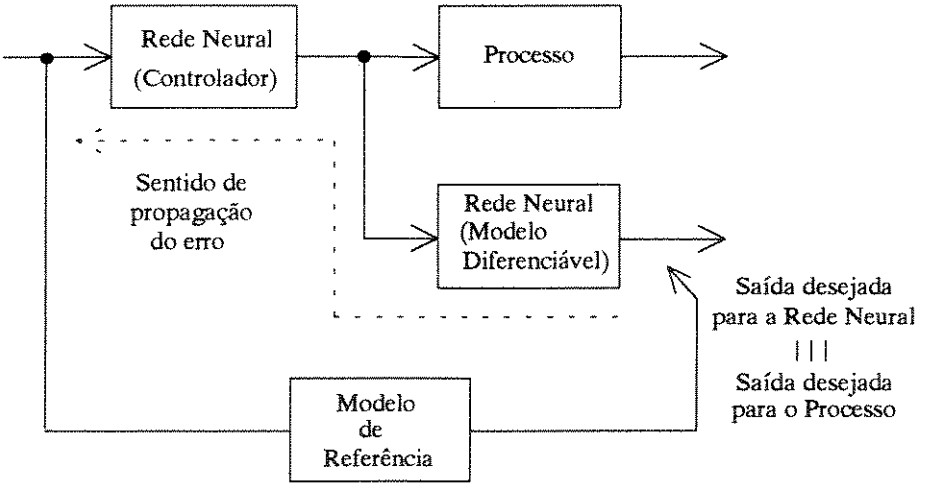


Figura 4.7 - A utilização de um modelo diferenciável no projeto de controladores baseados em redes neurais (Versão I)

É importante observar que a aplicação deste método requer um modelo do processo que seja diferenciável, não exigindo a utilização de uma rede neural na geração do modelo diferenciável. No entanto, a utilização de redes neurais treinadas com base no algoritmo de retro-propagação é um procedimento coerente, já que este mecanismo pode ser utilizado tanto para o ajuste dos parâmetros da rede, como para a diferenciação do modelo resultante.

Um procedimento equivalente ao apresentado na figura 4.7 foi proposto por Narendra e Parthasarathy (91)[53] e também faz uso de um modelo diferenciável para o processo. Neste caso, conforme ilustrado na figura 4.8, no lugar da propagação do erro da saída para a entrada do sistema de controle, a propagação se dá no sentido contrário, fazendo com que as derivadas parciais em relação aos parâmetros do controlador sejam conduzidas da entrada para a saída do sistema de controle. Com isso, a regra de ajuste dos parâmetros do controlador pode ser aplicada na saída do processo, onde o erro está definido.

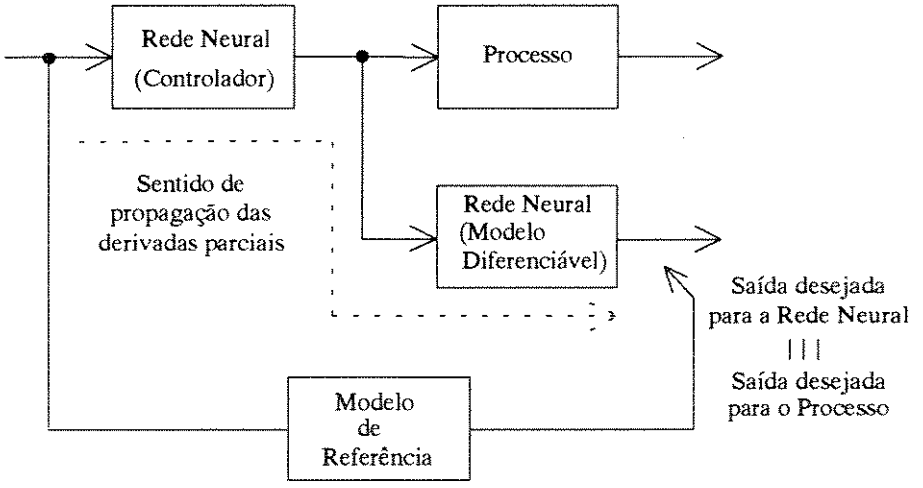


Figura 4.8 - A utilização de um modelo diferenciável no projeto de controladores baseados em redes neurais (Versão II)

4.1.6. A rede neural como um filtro não-linear adaptativo

O filtro linear adaptativo apresentado na figura 4.9 é uma estrutura básica para a maioria dos sistemas adaptativos (Widrow&Winter-88)[77].

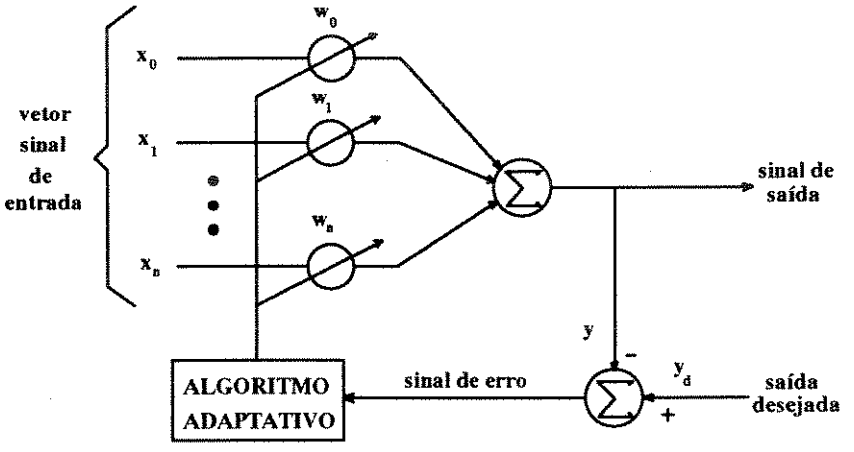


Figura 4.9 - Filtro linear adaptativo

O sinal de saída y é uma combinação linear dos sinais de entrada x_i (i=1,...,n), que formam o vetor de entrada **x**. Os coeficientes de ponderação w_i (i=1,...,n) formam o vetor de pesos **w**. O sinal de saída pode, então, ser descrito pelo produto escalar do vetor de entrada pelo vetor de pesos. Cada sinal de saída y produzido é comparado com um sinal desejado y_d, produzindo uma diferença representada pelo sinal de erro. De posse do sinal de erro, um algoritmo adaptativo é aplicado com o objetivo de aproximar o sinal de saída y do sinal desejado y_d. De todos os algoritmos adaptativos para ajuste automático dos pesos, o mais popular é o algoritmo de erro quadrático médio de Widrow-Hoff (60)[74].

Na figura 4.10, temos a estrutura básica de treinamento de um neurônio do tipo perceptron, onde g é a função de ativação.

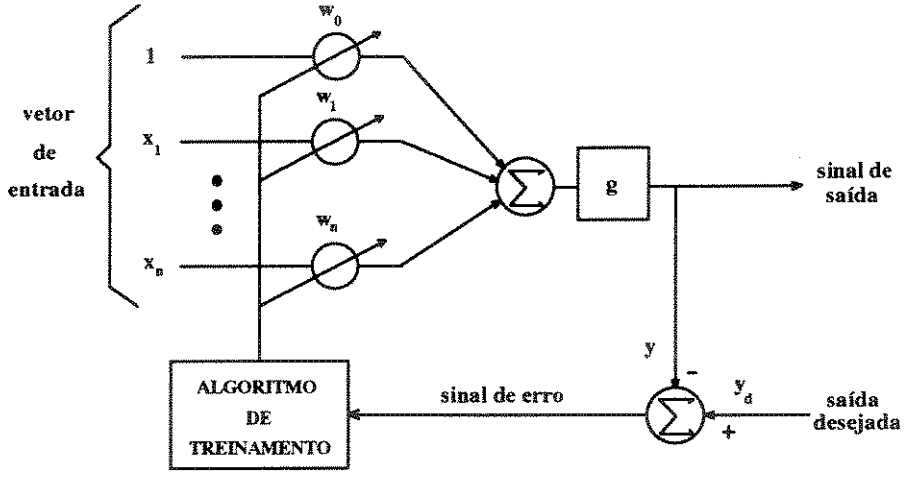


Figura 4.10 - Estrutura básica de treinamento de um neurônio do tipo perceptron

Comparando-se as figuras 4.9 e 4.10, temos que a diferença básica entre as duas estruturas está na função de ativação g, que, na figura 4.9 é a função identidade e na figura 4.10 geralmente é uma função que representa o efeito de saturação do neurônio, o que introduz uma não-linearidade no processamento do sinal. Na figura 4.10, temos também $x_0=1$, representando a entrada de polarização do neurônio.

Ainda na figura 4.10, sendo g uma função não-linear, se no lugar do único neurônio apresentado introduzirmos uma rede neural com pelo menos uma camada oculta, teremos um filtro não-linear adaptativo capaz de, através da aplicação de um algoritmo de treinamento apropriado, mapear todo tipo de função não-linear contínua definida em um conjunto compacto de dimensão finita (veja Apêndice A).

4.2. Análise de critérios de implementação

O sucesso da aplicação de redes neurais está geralmente vinculado a uma definição correta da arquitetura e do algoritmo de treinamento associado. Infelizmente, com base no nível atual de desenvolvimento das pesquisas, não existem métodos automáticos de definição dos critérios gerais de implementação adequados para todo tipo de aplicação. Por exemplo, no caso da utilização de redes neurais em problemas de aproximação de mapeamentos contínuos e definidos em um universo compacto, a necessidade de se arbitrar uma série de parâmetros considerados fundamentais para a garantia de sucesso da aproximação faz com que a capacidade de aproximação universal de redes neurais, analisada no Apêndice A, seja uma condição apenas necessária para permitir a obtenção da solução do problema de aproximação. As condições suficientes só podem ser obtidas para classes específicas de problemas que atendem a um conjunto de restrições muito fortes.

Por outro lado, com base em critérios específicos desenvolvidos para algumas classes de problemas e em procedimentos iterativos de tentativa e erro, conceitos intuitivos e de natureza puramente heurística têm sido desenvolvidos, possibilitando a ampliação do universo de aplicação de redes neurais. É importante salientar que o rigor matemático e a fundamentação teórica permanecem como etapas importantes a serem atingidas com a evolução dos processos de amadurecimento das idéias, estruturação dos conceitos e desenvolvimento de ferramentas de apoio. Enquanto estas etapas não forem superadas, a aplicação de métodos heurísticos continuará a representar uma alternativa atraente, inclusive como fonte de conhecimento para a formulação de novos conceitos matematicamente rigorosos e teoricamente bem fundamentados.

4.2.1. As etapas envolvidas no processo de implementação

Uma vez definida a tarefa a ser executada pela rede neural, os seguintes tópicos[1] devem ser analisados de forma a se estabelecer os critérios de implementação a serem adotados:

1) Qualificação do treinamento supervisionado[2]:

- escopo do conhecimento: como definir qualitativa e quantitativamente o conhecimento necessário para executar a tarefa especificada.
- representação do conhecimento: como representar na forma de padrões de treinamento este conhecimento.
- natureza dos padrões de treinamento: como representar numericamente padrões de natureza não-numérica.
- disponibilidade dos padrões de treinamento: qual o critério de acesso aos padrões de treinamento.
- pré-processamento dos padrões: como manipular o conhecimento descrito na forma de padrões de treinamento.
- apresentação dos padrões na fase de treinamento: qual o critério de apresentação dos padrões à rede neural.
- representação do aprendizado: como a rede neural vai representar o conhecimento adquirido via treinamento.
- avaliação do treinamento: como comparar o conhecimento já aprendido pela rede neural com o conhecimento necessário para executar a tarefa especificada.

2) Quantificação do treinamento supervisionado

- escopo do processo de treinamento: qual o conjunto de parâmetros da rede neural passível de ajuste por parte do processo de treinamento.
- implementação do treinamento: definidos os procedimentos qualitativos, como implementá-los na forma de um algoritmo de treinamento apropriado para simulação computacional.
- arbitramento de parâmetros: como arbitrar os parâmetros do algoritmo de treinamento como a taxa de aprendizado (α) e a constante de momento (β).

---

[1]Este conjunto de tópicos está restrito ao escopo deste trabalho, retratando apenas uma parcela das inúmeras alternativas de implementação citadas na literatura.

[2]Neste trabalho não são considerados os métodos de treinamento não-supervisionados.

- definição do processamento: de acordo com a disponibilidade de "software" e "hardware", como definir as caracterísitcas de processamento de forma a explorar eficientemente os recursos disponíveis.

3) Qualificação da rede neural

- definição das principais características da unidade processadora (neurônio).
- definição da topologia ou arquitetura da rede neural.
- definição do critério de processamento.

4) Quantificação da rede neural

- implementação numérica das principais características da unidade processadora.
- dimensionamento da rede neural.
- especificações numéricas do processamento.

4.2.2. Um exemplo de implementação

Um exemplo de aplicação de redes neurais é apresentado a seguir, com o objetivo de ilustrar o processo de implementação através da descrição detalhada das etapas envolvidas. Com relação àqueles procedimentos a serem executados não apenas na presente implementação, mas em todas as implementações incluídas no desenvolvimento deste trabalho, além de uma descrição detalhada, é apresentada uma discussão a respeito das razões que fundamentaram tal escolha.

Problema: Aproximação utilizando redes neurais de um mapeamento estático não-linear f: $\Re^1 \rightarrow \Re^1$ realizado por um processo desconhecido apresentado na figura 4.11, onde são ilustrados os canais de entrada e saída, únicos meios de comunicação com o mundo exterior[3].

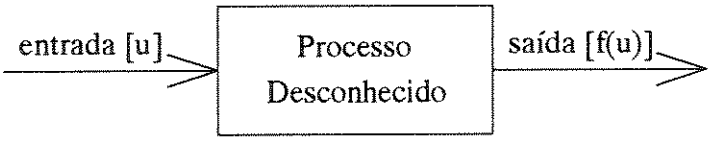


Figura 4.11 - Mapeamento de entrada-saída a ser aproximado via redes neurais

[3]A escolha de um mapeamento do tipo f: $\Re^1 \rightarrow \Re^1$ é motivada pela possibilidade de representação no plano $\Re^2$, adequado para efeito de visualização gráfica dos resultados.

4.2.2.1. Hipóteses iniciais

- o mapeamento realizado pelo processo é dado por f(u)=sen(2πu). Esta função é considerada desconhecida e deve ser aproximada utilizando-se redes neurais. A função f(u) será utilizada apenas na geração dos pares de entrada-saída a serem utilizados como padrões de treinamento.
- a característica estática do mapeamento é uma informação considerada disponível (caso contrário, métodos de detecção de características dinâmicas devem ser aplicados).
- a comunicação do processo com o mundo exterior de acordo com o esquema da figura 4.11 se dá através de sinais de natureza numérica, isto é, qualquer que seja a natureza do processamento interno, dispositivos de conversão devem atuar na comunicação com o mundo exterior de forma a garantir a compatibilidade de processamento.

4.2.2.2. Escopo do conhecimento

Qualitativamente, o conhecimento é representado pela capacidade de aproximação do mapeamento realizado pelo processo desconhecido, ou seja, pela capacidade de reprodução dos pares de entrada-saída provenientes do processo. Quantitativamente, a capacidade de aproximação deve ser restrita a uma região de interesse, ou seja, deve ser definido um universo compacto para o problema de aproximação. No caso de f(u)=sen(2πu), a saída já está limitada ao intervalo [-1,1], sendo que a entrada será restrita ao intervalo [0,1], de forma que o mapeamento desejado corresponda a um período completo da função senoidal, conforme apresentado na figura 4.12.

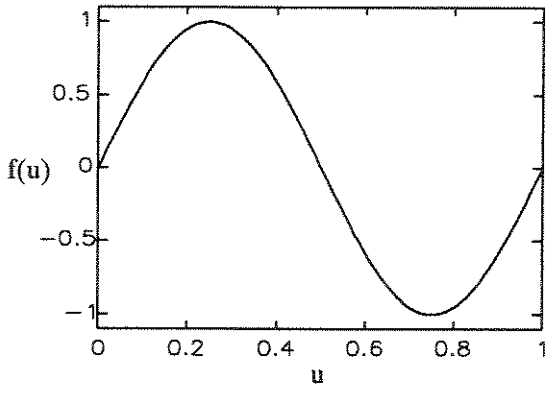


Figura 4.12 - Características do mapeamento na região de aproximação

4.2.2.3. Representação do conhecimento

O conhecimento é representado pelos pares de entrada-saída. Portanto, os padrões para treinamento correspondem aos valores de saída produzidos pelo processo desconhecido a partir de um dado conjunto de valores de entrada.

4.2.2.4. Natureza dos padrões de treinamento

Por hipótese, os padrões de treinamento são de natureza numérica.

4.2.2.5. Disponibilidade dos padrões de treinamento

A obtenção de novos padrões de treinamento pode ser realizada a todo momento, bastando realizar a leitura dos valores de entrada e saída do processo desconhecido, de acordo com o esquema da figura 4.11.

4.2.2.6. Pré-processamento dos padrões

Com base na disponibilidade dos padrões de treinamento, é interessante obter um conjunto de padrões que atendam aos seguintes critérios:

- capacidade de representação global do conhecimento;
- mínima ocupação de memória;

O objetivo é garantir a eficiência do processo de treinamento e a minimização dos custos computacionais. No entanto, esses dois critérios acima mencionados são, a princípio, contraditórios, já que um aumento na capacidade de representação do conhecimento via ampliação do conjunto de padrões de treinamento vai acarretar uma maior ocupação de memória. Portanto, deve-se buscar o maior índice de representação do conhecimento por unidade de memória ocupada. Este índice é tanto maior quanto menor for o nível de redundância presente no conjunto de padrões.

A minimização dos custos computacionais não é conseqüência apenas de uma redução na ocupação de memória, mas também de uma redução na carga de processamento, pois uma menor quantidade de informação é apresentada à rede neural na etapa de treinamento. Logicamente, esta menor quantidade deve ser compensada por uma maior qualidade de informação, de forma a manter o nível de representação.

Além da questão da redundância, outro fator importante na caracterização dos padrões de treinamento é a presença de padrões conflitantes. Do ponto de vista de treinamento, a presença de padrões conflitantes pode ser entendida como um indicativo de ausência de conhecimento, criando dificuldades para a convergência do processo de treinamento.

No caso específico do problema de aproximação proposto, um conflito pode ser caracterizado pela presença no conjunto de padrões de treinamento de pares de entrada-saída tais que:

$$u_1=u_2 \quad e \quad f(u_1)\neq f(u_2)$$

conforme ilustrado na figura 4.13, indicando a presença de uma descontinuidade no mapeamento a ser aproximado.

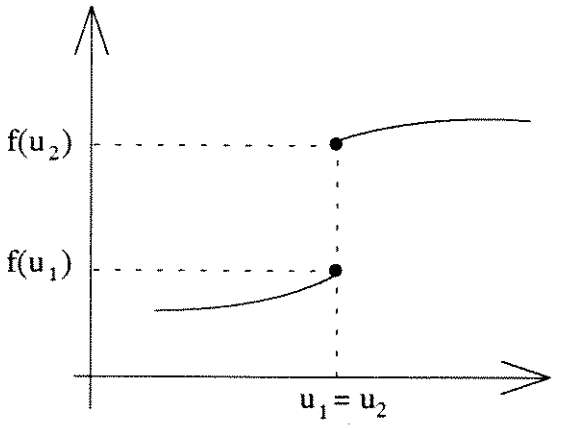


Figura 4.13 - A presença de conflito nos padrões de treinamento

Deve-se salientar que a presença de padrões conflitantes pode ser conseqüência tanto de mapeamentos mal definidos como de incorreções na geração dos padrões.

Com base nesta discussão, pode-se concluir que o melhor conjunto de padrões para treinamento deve ser gerado levando-se em conta a capacidade de representação do conhecimento e a redução dos níveis de redundância e conflito. Para garantir a capacidade de representação, um conjunto de padrões de treinamento que parece ser adequado para o problema de aproximação do mapeamento apresentado na figura 4.12 é dado pelos sinais de saída correspondentes a sinais de entrada definidos de forma a varrer toda a região de aproximação. A fim de reduzir o nível de redundância, é interessante tomar padrões de entrada adequadamente espaçados entre si. É certo que seria interessante a utilização de espaçamentos menores em regiões do mapeamento que concentram uma maior quantidade de informação (obtenção de um maior número de padrões de treinamento por unidade de espaço), e espaçamentos maiores em outras regiões, mas assumindo o desconhecimento de qualquer característica do mapeamento, a geração de padrões igualmente espaçados representa um procedimento inicial coerente, por considerar toda a região de aproximação igualmente representativa.

Portanto, tomando-se sinais de entrada igualmente espaçados e varrendo toda a região de aproximação (intervalo [0,1]), os padrões de treinamento gerados são apresentados na figura 4.14.

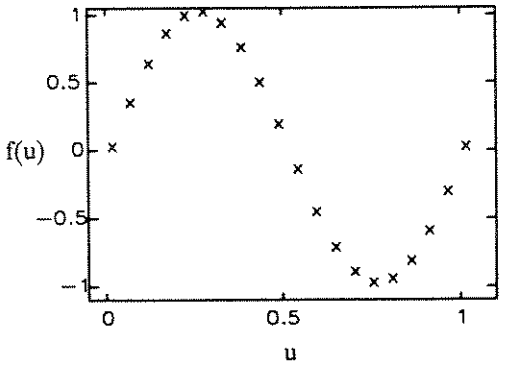


Figura 4.14 - Padrões de treinamento

4.2.2.7. Apresentação dos padrões na fase de treinamento

Em problemas onde os padrões para treinamento só se encontram disponíveis durante a fase de treinamento propriamente dita, como por exemplo, em técnicas de treinamento "on-line", não existe outra alternativa senão a de utilizar os padrões para treinamento conforme estes forem sendo produzidos. Já no caso da aplicação descrita acima, como também em boa parte das aplicações de redes neurais que utilizam técnicas de treinamento "off-line", a prévia disposição do conjunto de padrões para treinamento permite o estabelecimento de políticas distintas de apresentação dos padrões à rede neural, permitindo a definição de ciclos de treinamento distintos. Cada política procura privilegiar algum critério, e a escolha da política mais adequada vai depender das especificações de cada aplicação. Dentre as políticas mais difundidas é possível citar:

- Política 1: aplicação do treinamento somente após a apresentação de todos os padrões à rede neural: esta política privilegia o desempenho global do treinamento em detrimento do desempenho apresentado com relação a cada padrão específico. Analisando-se em termos da evolução do erro existente entre os padrões desejados e aqueles produzidos pela rede neural, é possível afirmar que esta política privilegia a média dos erros em detrimento da variância. Neste caso, o ciclo de treinamento é definido pelas seguintes etapas: apresentação de todos os padrões à rede neural seguida pela aplicação do processo de ajuste dos pesos.
- Política 2: aplicação do treinamento com base somente nos padrões que produzem um desempenho abaixo de um determinado limiar: esta política é exatamente o oposto da apresentada anteriormente, privilegiando a variância em detrimento da média dos erros existentes entre os padrões desejados e aqueles produzidos pela rede neural. Neste caso, o ciclo de treinamento é definido pelas seguintes etapas: definição dos padrões a serem apresentados à rede neural, apresentação dos padrões à rede neural e aplicação do processo de ajuste dos pesos após a apresentação de cada padrão.

- Política 3: apresentação aleatória dos padrões para treinamento: é uma alternativa intermediária com relação às duas políticas anteriores, atribuindo igual importância para a média e a variância do erro. Neste caso, o ciclo de treinamento é definido pelas seguintes etapas: escolha aleatória de um padrão a ser apresentado à rede neural, apresentação do padrão à rede neural e aplicação do processo de ajuste dos pesos após a apresentação de cada padrão.
- Política 4: apresentação seqüencial dos padrões para treinamento: ordenação dos padrões disponíveis para treinamento e apresentação seqüencial à rede neural. Esta ordenação pode seguir uma série de critérios, como por exemplo, níveis de complexidade, similaridade e correlação. Neste caso, o ciclo de treinamento é definido pelas seguintes etapas: apresentação seqüencial dos padrões à rede neural e aplicação do processo de ajuste dos pesos após a apresentação de cada padrão.

4.2.2.8. Representação do aprendizado

Com a utilização de processamento de natureza numérica, a representação do aprendizado, ou seja, a representação do conhecimento adquirido pela rede neural em fase de treinamento, é dada pelos valores das intensidades das conexões entre as unidades processadoras.

4.2.2.9. Avaliação do treinamento

Tanto para avaliar a eficiência como para orientar a própria execução do algoritmo de treinamento é necessário definir um critério de comparação entre o conhecimento já aprendido pela rede neural e o conhecimento considerado necessário para executar a tarefa especificada. Já foi utilizado anteriormente o conceito de erro ou diferença existente entre os padrões desejados e aqueles produzidos pela rede neural. Métodos numéricos clássicos de comparação entre grandezas de mesma natureza permitem o estabelecimento de níveis numéricos de proximidade ou similaridade. Na aplicação de algoritmos de treinamento supervisionado, o erro quadrático médio é geralmente utilizado como critério de avaliação do desempenho do treinamento, sendo definido na forma:

- submetendo-se o processo desconhecido e a rede neural ao mesmo vetor de entrada (u), o erro (e) existente entre a saída desejada proveniente do processo desconhecido (y_d) e a correspondente saída da rede neural (y) pode ser calculado com base no esquema da figura 4.15:

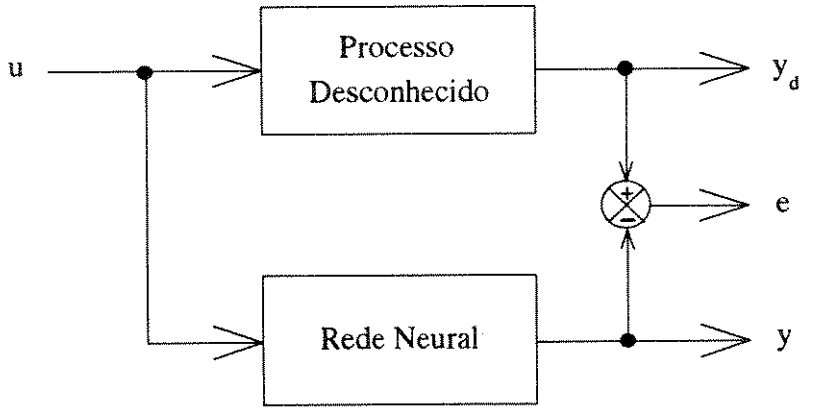


Figura 4.15 - Procedimento de definição do erro existente entre a saída do processo e aquela produzida pela rede neural para um mesmo vetor de entrada

• dado um conjunto de n padrões de avaliação, definindo o erro produzido para cada padrão i (i=1,...n) como sendo e_i, o erro quadrático médio (EQM) é calculado na forma:

$$EQM = \sum_{i=1}^{n} \frac{(e_i)^2}{n} \qquad (4.1)$$

4.2.2.10. Escopo do processo de treinamento

Neste trabalho, os únicos parâmetros da rede neural passíveis de ajuste por parte do processo de treinamento são os pesos, cujos valores indicam a intensidade das conexões entre as unidades processadoras.

4.2.2.11. Implementação do treinamento

Com base nos critérios de avaliação de treinamento apresentados anteriormente, pode-se concluir que quanto menor o erro quadrático médio calculado com base no conjunto de padrões de treinamento, melhor é o nível de aproximação do mapeamento. Portanto, o processo de treinamento deve atuar no sentido de minimizar o erro quadrático médio a partir do ajuste dos pesos da rede neural., ou seja, deve atuar no sentido de atender à seguinte função objetivo:

$$\min_{w} \; J(w) = \sum_{i=1}^{n} \frac{(e_i)^2}{n} \qquad \text{onde w é o vetor de pesos.}$$

Os algoritmos de treinamento de 1ª e 2ª ordem desenvolvidos para redes neurais recorrentes e não-recorrentes e apresentados no Capítulo 2, foram implementados com base neste mesmo tipo de função objetivo, sendo de aplicação imediata.

4.2.2.12. Arbitramento de parâmetros

Como discutido no Capítulo 2, os algoritmos de treinamento desenvolvidos não são totalmente automatizados, exigindo a definição por parte do projetista de alguns parâmetros fundamentais para o sucesso do processo de convergência, como a taxa de aprendizado (α) e a constante de momento (β). Apesar da existência de alguns critérios de manipulação dos valores desses parâmetros a partir de valores inicialmente arbitrados, neste trabalho estes parâmetros, uma vez arbitrados, serão mantidos constantes. Com relação ao problema de aproximação proposto, os valores numéricos destes parâmetros assim como outras especificações são apresentados no Apêndice F, seção 1.

4.2.2.13. Definição do processamento

Os algoritmos de treinamento são implementados em computadores digitais do tipo IBM-PC, apresentando, portanto, processamento serial. Com isso, não são exploradas as potencialidades de processamento paralelo apresentadas pelas redes neurais, já que aquelas operações que poderiam ser executadas ao mesmo tempo, representando uma economia de tempo de computação, acabam sendo executadas sequencialmente. O desenvolvimento de ferramentas de "software" e "hardware" para processamento paralelo são importantes passos no aumento de eficiência dos algoritmos de treinamento.

4.2.2.14. Definição das principais características da unidade processadora

Uma discussão detalhada das principais características da unidade processadora é apresentada no Capítulo 2.

4.2.2.15. Definição da topologia ou arquitetura da rede neural

Por se tratar de um problema de mapeamento estático, será discutido ainda neste capítulo quais as redes neurais não-recorrentes em camadas que representam as arquiteturas mais adequadas.

4.2.2.16. Definição do critério de processamento

Em razão da implementação em computadores digitais, vão existir instantes preferenciais de processamento, ou seja, vai ser utilizado processamento discreto e execução síncrona (veja Capítulo 2).

4.2.2.17. Implementação numérica das principais características da unidade processadora

A configuração básica da unidade processadora resultante, desenvolvida no Capítulo 2 e reapresentada na figura 4.16 a seguir, é caracterizada por processamento puramente numérico, ou seja, as principais características da unidade processadora já estão implementadas numericamente.

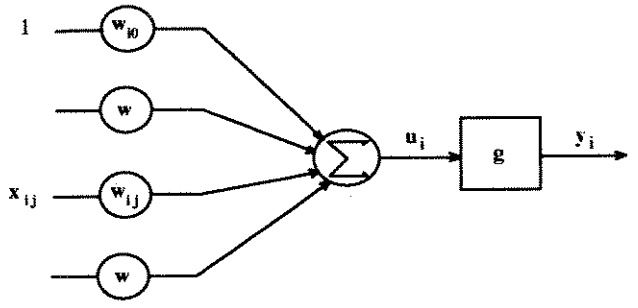


Figura 4.16 - Configuração básica da unidade processadora

4.2.2.18. Dimensionamento da rede neural

O dimensionamento da rede neural é caracterizado pela definição do número de camadas e do número de unidades processadoras por camada.

Quanto maior a dimensão da rede neural, maior é a capacidade de representação. Logo, quanto maior a complexidade da tarefa a ser executada pela rede neural maior deve ser a sua dimensão, de forma a garantir a existência de um potencial de representação adequado.

No estudo do potencial de mapeamento não-linear de redes neurais artificiais, desenvolvido no Capítulo 2, ficou claro que, com relação ao potencial de representação da rede neural, o efeito da inclusão de mais uma unidade processadora na rede vai depender da camada em que ela é introduzida. Portanto, a distribuição de unidades processadoras pela rede neural é um fator importante na definição de seu potencial de representação.

Neste trabalho, o procedimento adotado em todas as simulações é utilizar um maior número de unidades processadoras nas camadas mais próximas à entrada da rede e ir reduzindo o número de unidades processadoras por camada a partir da camada de entrada até a camada de saída. A razão para adoção deste critério está baseada em resultados experimentais e em alguns resultados teóricos presentes na literatura, indicando que a existência de uma maior capacidade de representação nas camadas mais próximas à entrada da rede vai facilitar a obtenção de representações internas a partir das informações provenientes da entrada da rede.

No caso do problema de aproximação proposto, os parâmetros correspondentes à dimensão da rede neural são apresentados no Apêndice F, seção 1.

4.2.2.19. Especificações numéricas do processamento

A característica não-linear da função de ativação g, representada pela função tangente hiperbólica (veja Capítulo 2), introduz o efeito de saturação na saída das unidades processadoras. Pelo fato da derivada parcial $\partial g/\partial w$ utilizada na definição da intensidade de ajuste dos pesos assumir valores muito próximos de zero na região de saturação, a unidade processadora operando nesta região acaba sendo pouco sensível ao processo de ajuste de seus pesos, comandado pelo algoritmo de treinamento. Neste caso, já que o algoritmo de treinamento não é capaz de alterar significativamente o estado da unidade processadora, esta última deixa de participar do processo de aprendizado, reduzindo assim a capacidade de representação da rede neural. Logo, para manter a sua capacidade máxima de representação, ao menos no estágio inicial do processo de treinamento, é necessário que todas as unidades processadoras estejam operando fora da região de saturação, conforme ilustrado na figura 4.17.

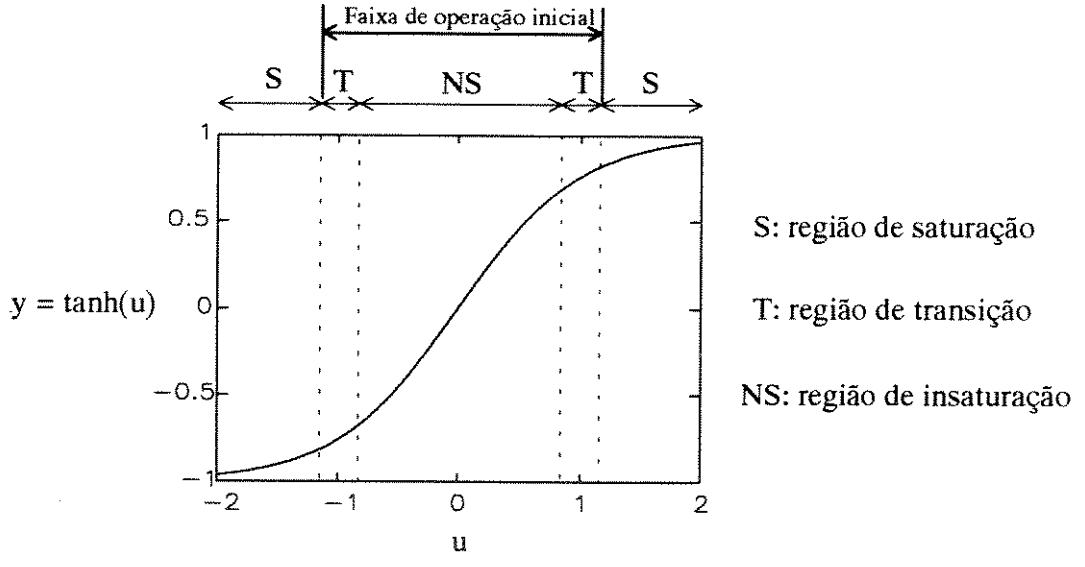


Figura 4.17 - Definição dos pontos de operação da unidade processadora

Um forma de se garantir uma condição inicial para a rede neural onde todas as suas unidades processadoras estejam operando fora da região de saturação é gerar os pesos iniciais aleatoriamente, com média zero e variância baixa.

Existindo uma região do espaço de representação onde a capacidade de representação da rede neural é maior, dada pela operação fora da região de saturação, é interessante que os sinais de entrada e de saída assumam valores em intervalos interiores a esta região, ou seja em intervalos definidos em torno de zero, como por exemplo, o intervalo [-1,1]. Portanto, é um procedimento recomendável a normalização (escalonamento e translação) dos sinais de entrada e saída de forma que o intervalo onde eles assumem valores seja coerente com a faixa de operação fora da região de saturação.

4.2.2.20. Resultado do processo de treinamento

A seguir são apresentados os resultados do processo de treinamento da rede neural utilizando os parâmetros definidos no Apêndice F, seção 1. Utilizando-se uma mesma condição inicial para a rede neural, são realizadas quatro simulações diferindo com relação à política de apresentação dos padrões à rede. Para efeito de comparação, foram realizados 1.000.000 de ciclos de treinamento para cada caso. Devido à simplicidade do problema de aproximação, as quatro políticas produziram bons resultados, conforme apresentado na figura 4.18.

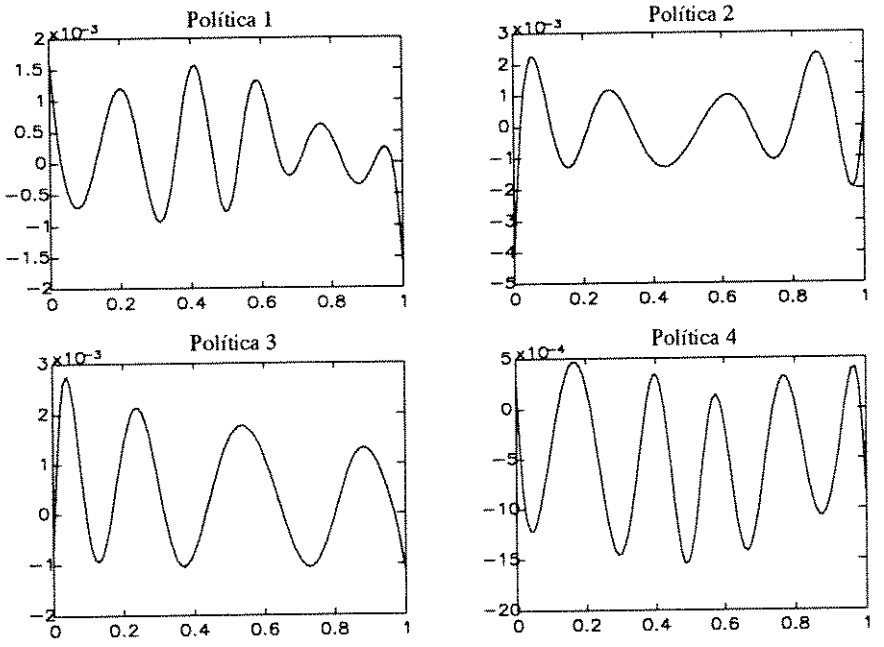


Figura 4.18 - Resultado do processo de treinamento utilizando as quatro políticas de apresentação dos padrões à rede neural

Nesta figura, os gráficos apresentados correspondem ao erro existente entre o mapeamento realizado pelo processo desconhecido (figura 4.12) e o mapeamento realizado pela rede neural após o treinamento. Devido à utilização de políticas distintas, apesar de partir de uma mesma condição inicial, do mesmo conjunto de padrões para treinamento e de um mesmo número de ciclos de treinamento, o resultado final do processo de treinamento para cada política mostrou-se qualitativamente distinto. Dependendo da aplicação, esta diferença qualitativa no comportamento de convergência pode influir decisivamente no sucesso do processo de treinamento, indicado quantitativamente pelo erro quadrático médio (EQM) existente entre as saídas do processo e aquelas produzidas pela rede neural. No caso do problema de aproximação proposto, devido à sua simplicidade, não é verificada uma diferença quantitativa significativa em relação ao erro quadrático médio resultante, conforme indicado pelos dados da tabela 4.1.

	EQM
Política 1	0,00005072
Política 2	0,00013927
Política 3	0,00013425
Política 4	0,00006176

Tabela 4.1 - Erro quadrático médio resultante

É importante verificar que na geração dos gráficos da figura 4.18, o desempenho da rede neural foi comparado não apenas com os padrões de treinamento, mas também com outros padrões pertencentes à região de aproximação. A manutenção do erro em valores reduzidos demonstra a capacidade de generalização apresentada pelas redes neurais artificiais, neste caso, representada pela capacidade de interpolação, já que, apesar de ser treinada com base em um conjunto limitado de padrões (veja figura 4.14), a rede neural acaba sendo capaz de representar os demais padrões não utilizados na fase de treinamento.

Na figura 4.19, a seguir, é apresentada a evolução do erro quadrático médio em função dos ciclos de treinamento para a aplicação da política 1 de apresentação dos padrões de treinamento à rede neural.

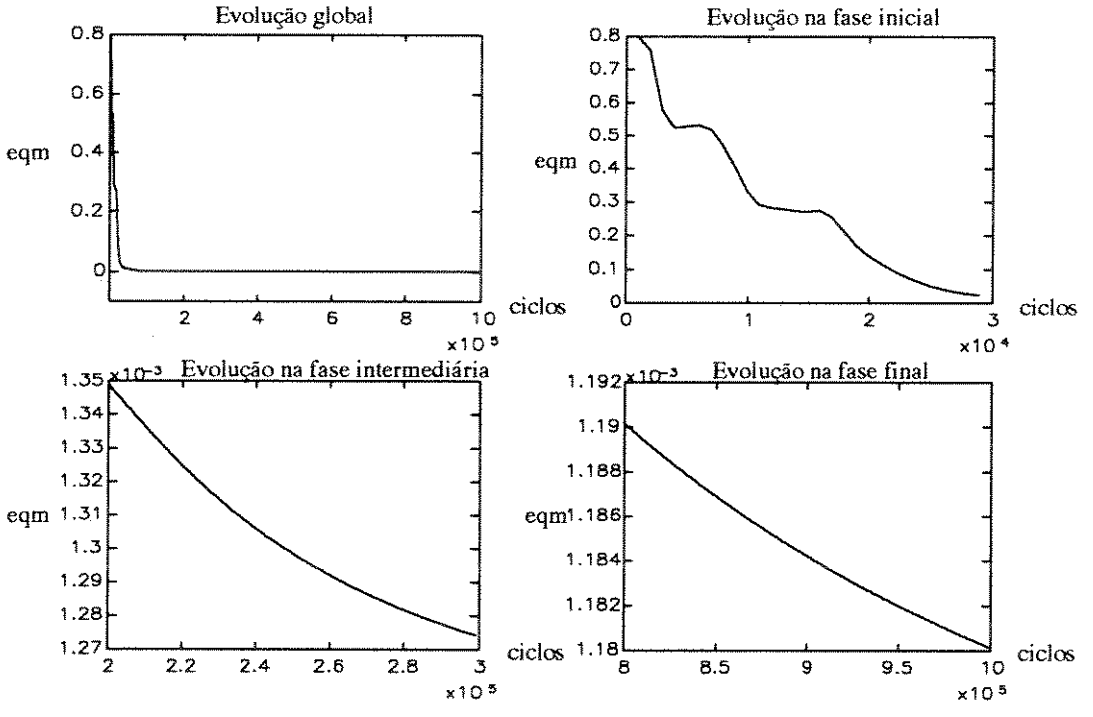


Figura 4.19 - Evolução do processo de treinamento

Com base na evolução global do treinamento, pode-se concluir que a convergência do erro quadrático médio segue um perfil aproximadamente exponencial, com reduções acentuadas do erro quadrático médio na fase inicial do treinamento e uma queda contínua da velocidade de convergência conforme o treinamento evolui. Uma comparação entre os gráficos referentes às fases inicial, intermediária e final permite constatar mais claramente esta tendência. Além disso pode-se verificar que na fase inicial o processo de convergência apresenta períodos de queda acentuada seguidos por períodos de estagnação, indicando a busca de novas direções de ajuste. Já nas fases intermediária e final, o processo de convergência evolui de forma bastante homogênea, indicando não haver mudanças abruptas de direção de ajuste, definida no espaço dos pesos.

4.3. Considerações a respeito de arquiteturas de redes neurais utilizadas na representação de sistemas dinâmicos

Na representação de sistemas dinâmicos, duas arquiteturas de redes neurais são amplamente utilizadas:

- redes neurais recorrentes;
- redes neurais não-recorrentes.

Os conceitos básicos e as propriedades relacionadas com estes dois tipos de arquiteturas foram apresentados no Capítulo 2, sendo que a diferença básica está na presença de conexões de realimentação no caso das redes neurais recorrentes, o que não ocorre no caso não-recorrente.

4.3.1. Comparação entre as duas arquiteturas de redes neurais na representação de comportamentos dinâmicos

Redes neurais não-recorrentes têm sido amplamente utilizadas na identificação de sistemas dinâmicos. Devido ao fato de as redes não-recorrentes não apresentarem memória dinâmica, a representação de sistemas dinâmicos só é possível com a inclusão, no vetor de entradas da rede, de entradas e saídas do sistema no instante atual e em instantes anteriores de tempo, conforme apresentado na figura 4.20. Este vetor de entradas é conhecido como linha de derivação de atraso (LDA) e transforma um problema de modelagem temporal (aprendizagem do comportamento dinâmico de um sistema no domínio do tempo) em um problema de modelagem espacial (mapeamento estático das entradas e saídas, juntamente com seus respectivos retardos no tempo, na próxima saída).

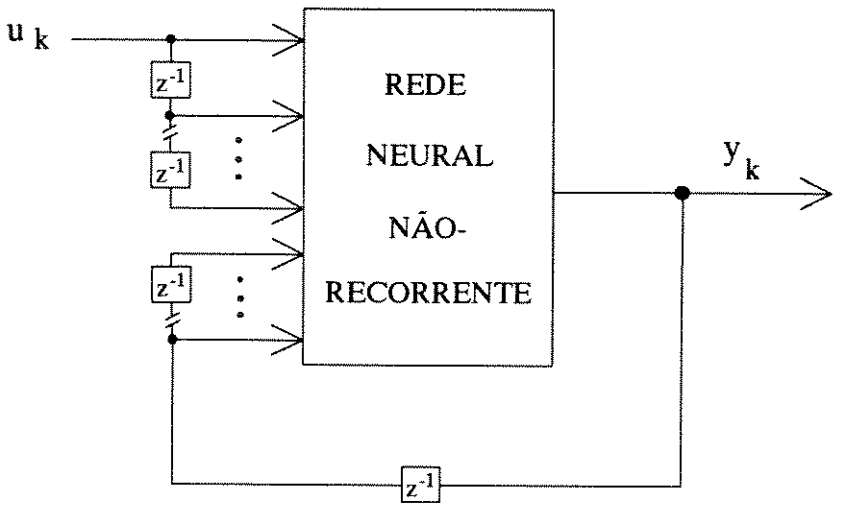


Figura 4.20 - Representação de comportamento dinâmico utilizando rede neural não-recorrente

De acordo com Elman (90)[18], existe uma série de desvantagens na utilização da LDA:

- a necessidade da existência de uma interface com o ambiente para registrar temporariamente uma sequência de valores de forma a apresentá-los ao mesmo tempo na entrada da rede;
- a ausência de evidências de que os sistemas biológicos fazem uso destes registradores;
- a imposição por parte dos registradores de um limite rígido à duração das sequências de dados, sugerindo que todos os vetores de entrada têm o mesmo comprimento;
- a dificuldade na distinção entre posições temporais relativas e posições temporais absolutas. Considere o exemplo de detecção de seqüências temporais a seguir:

  Dadas duas sequências temporais na forma : ... 0 0 1 1 0 0 ...

  ... 0 1 1 0 0 0 ...

  um mesmo padrão (...11...) é encontrado nas duas sequências, sendo que redes neurais não-recorrentes com LDA somente poderão tratá-los como padrões similares se esta condição for imposta à rede neural em fase de treinamento. Assim, a similaridade não seria uma consequência da estrutura dos padrões em si. Além disso, não haveria uma generalização para o caso de outras sequências que também apresentem padrões similares mas deslocados no tempo.
- a necessidade de um grande número de sinais de entrada para a rede neural, aumentando a dimensão do espaço de entrada-saída. A execução de mapeamentos em espaços de maior dimensão geralmente são mais complexos, dificultando a tarefa de aprendizado e exigindo redes neurais de dimensões elevadas. Estes fatores contribuem para aumentar a carga de processamento, mesmo com a utilização de dispositivos computacionais dotados de processamento paralelo.

Redes neurais recorrentes não estão sujeitas a estas limitações. Elas permitem que o tempo seja representado pelo seu efeito real no processamento. As conexões recorrentes provêm à rede a possibilidade de construir representações dinâmicas internas, o que vai permitir a existência de memória dinâmica.

4.3.2. Redes neurais recorrentes

As redes neurais recorrentes podem ser classificadas como totalmente ou parcialmente recorrentes. As redes totalmente recorrentes podem possuir todo tipo de conexão recorrente, todas ajustáveis. Já no caso das redes parcialmente recorrentes, a estrutura principal é não-recorrente e suas conexões são todas ajustáveis, ao contrário das conexões recorrentes, que são fixas.

Dentre as arquiteturas de redes neurais recorrentes, a rede de Elman é a mais simples. Um diagrama de blocos da rede de Elman é apresentado na figura 4.21.

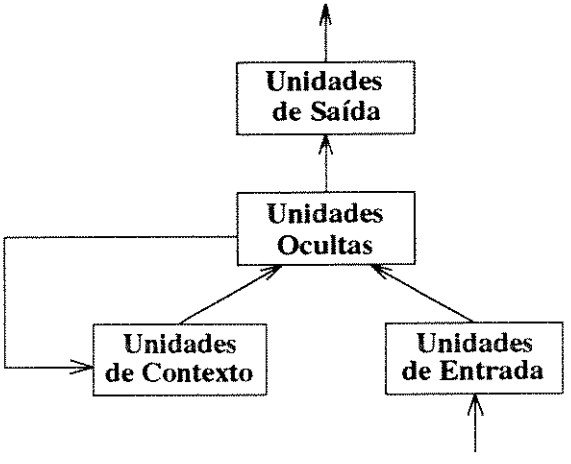


Figura 4.21 - Diagrama de blocos da rede de Elman

Observando-se a figura 4.21, pode-se constatar que a rede de Elman, além das unidades de entrada, ocultas e de saída, possui também unidades de contexto, como no caso geral de redes neurais parcialmente recorrentes. As unidades de entrada e saída interagem com o ambiente externo, enquanto que as unidades ocultas e de contexto não o fazem. Além disso, é possível definir:

Unidades de entrada: são apenas unidades registradoras que propagam o sinal de entrada sem alterá-lo.

Unidades de saída: são unidades lineares cujas saídas são somas ponderadas dos sinais que alimentam suas respectivas entradas.

Unidades ocultas: são unidades que apresentam funções de ativação lineares ou não-lineares, dependendo do tipo de aplicação.

Unidades de contexto: são unidades utilizadas apenas para memorizar as ativações prévias das unidades ocultas, podendo ser consideradas como atrasadores de um instante de tempo.

A intensidade de cada conexão é indicada por um valor numérico denominado peso. As conexões não-recorrentes são ajustáveis, enquanto que as recorrentes são fixas. Portanto, a rede de Elman é uma rede parcialmente recorrente.

Em um instante de tempo específico k, a ativação prévia da camada oculta no instante k-1 e as entradas correntes são utilizadas como entradas da rede. Neste estágio de processamento, a rede é puramente não-recorrente, sendo que estas entradas são propagadas de forma a produzir as saídas no instante k. Após a propagação dos sinais até a saída, a ativação atual das camadas ocultas é enviada às unidades de contexto através das conexões recorrentes, onde é armazenada para a realização do próximo passo de processamento no instante k+1. No início da operação, a ativação das unidades ocultas não é conhecida. Com isso, um procedimento usual é tomar o intervalo em que a função de ativação das unidades ocultas assume valores e inicializá-las com o valor que corresponde ao ponto central deste intervalo.

Uma arquitetura detalhada da rede de Elman com uma entrada e uma saída é apresentada na figura 4.22.

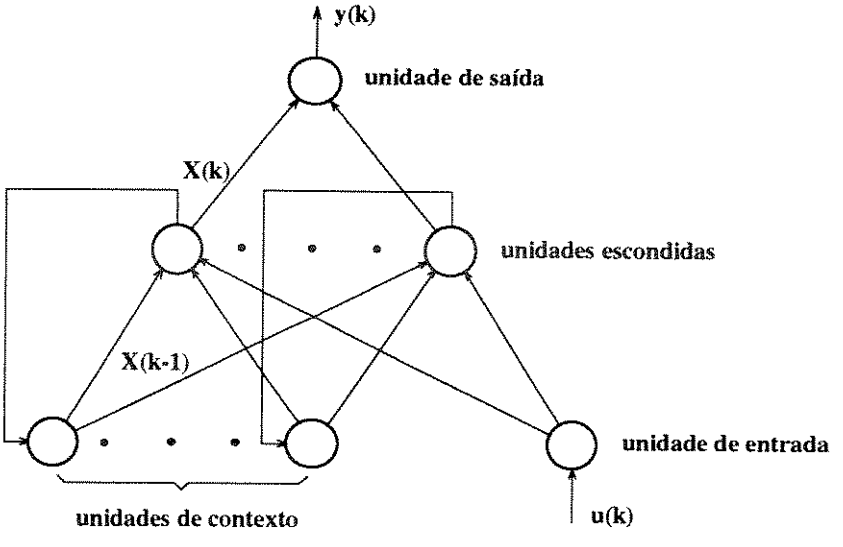


Figura 4.22 - Estrutura detalhada da rede de Elman

4.3.2.1. Aplicação da rede de Elman linear em contextos dinâmicos

Embora muitos esforços tenham sido empregados na aplicação da rede de Elman em processamento de voz, poucos trabalhos foram feitos na investigação das potencialidades deste tipo de rede em aplicações de modelagem de sistemas dinâmicos (Elman-90)[18]; (Pham-92)[58]. A arquitetura da rede de Elman linear mostra-se adequada

para a representação de sistemas dinâmicos lineares invariantes no tempo, como é demonstrado a seguir.

Na figura 4.22, a entrada externa da rede neural é representada por u(k) e a saída da rede por y(k). As ativações prévias e atuais das unidades ocultas são **X**(k-1) e **X**(k) respectivamente. Analisando-se o processamento de sinais na rede da figura 4.22, pode-se concluir que estas variáveis estão relacionadas na forma:

$$\begin{cases} \mathbf{X}(k) = \mathbf{F}[\mathbf{W}_{xx}\mathbf{X}(k-1), \mathbf{W}_{xu}\mathbf{u}(k)] \\ \mathbf{y}(k) = \mathbf{W}_{yx}\mathbf{X}(k) \end{cases} \tag{4.2}$$

onde $\mathbf{W}_{xx}$, $\mathbf{W}_{xu}$ e $\mathbf{W}_{yx}$ são matrizes de pesos e **F** é uma função multidimensional.

Em particular, se as unidades ocultas são lineares e a polarização das unidades ocultas e de saída é zero, a equação (4.2) pode ser expressa na forma:

$$\begin{cases} \mathbf{X}(k) = \mathbf{W}_{xx}\mathbf{X}(k-1) + \mathbf{W}_{xu}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{W}_{yx}\mathbf{X}(k) \end{cases} \tag{4.3}$$

Quando a entrada u é defasada de um instante de tempo na unidade de entrada, a equação (4.3) assume a forma:

$$\begin{cases} \mathbf{X}(k) = \mathbf{W}_{xx}\mathbf{X}(k-1) + \mathbf{W}_{xu}\mathbf{u}(k-1) \\ \mathbf{y}(k) = \mathbf{W}_{yx}\mathbf{X}(k) \end{cases} \tag{4.4}$$

A equação (4.4) corresponde à representação de sistemas dinâmicos por espaço de estados (veja Apêndice B). A ordem do modelo vai depender, portanto, do número de unidades ocultas presentes na rede de Elman.

Assim, uma rede de Elman pode ser capaz de modelar um sistema dinâmico de ordem n, onde n é o número de unidades na camada oculta, desde que ela seja adequadamente treinada para realizar esta tarefa.

A equação (4.4) pode ser desenvolvida na forma (veja Apêndice B):

$$y(k) = \mathbf{A}_1 y(k-1) + \ldots + \mathbf{A}_n y(k-n) + \mathbf{B}_1 u(k-1) + \ldots + \mathbf{B}_n u(k-n) \tag{4.5}$$

Comparando-se as equações (4.4) e (4.5), fica evidente a diferença no número de entradas necessárias para a modelagem de sistemas dinâmicos de ordem n feita por uma rede não-recorrente com LDA e uma rede parcialmente recorrente (rede de Elman). Se o método de LDA é aplicado, 2n entradas serão necessárias (veja equação (4.5)). Para uma rede de Elman, o número de entradas é 1 (ou n+1 caso as unidades de contexto sejam consideradas como unidades de entrada - veja equação (4.4)). Assim, redes recorrentes irão requerer uma carga de processamento menor, principalmente quando n assume valores elevados.

A diferença entre uma rede não-recorrente com LDA e uma rede de Elman pode também ser considerada como sendo análoga à diferença entre métodos de identificação de sistemas dinâmicos baseados na descrição de entrada-saída e na descrição por espaço de estados.

Com base na representação por espaço de estados e tomando-se o caso de identificação utilizando-se uma rede de Elman, se todos os estados do sistema estiverem disponíveis no processo de treinamento, ou seja, se o vetor de saída é igual ao vetor de estados[4], as unidades de saída podem ser excluídas da rede.

4.3.2.2. Simulações computacionais - Rede de Elman linear

Com base no diagrama de blocos da figura 4.23, a estrutura de treinamento para a rede de Elman está baseada na apresentação de um mesmo vetor de entrada u(k) tanto para a rede neural como para o sistema a ser identificado. O erro, proveniente da comparação entre o vetor de saída $y_d(k)$ produzido pelo sistema (saída desejada) e aquele produzido pela rede y(k), será utilizado para o ajuste dos pesos dos neurônios de cada camada da rede, através da aplicação do algoritmo de retro-propagação, completando, assim, um ciclo de treinamento.

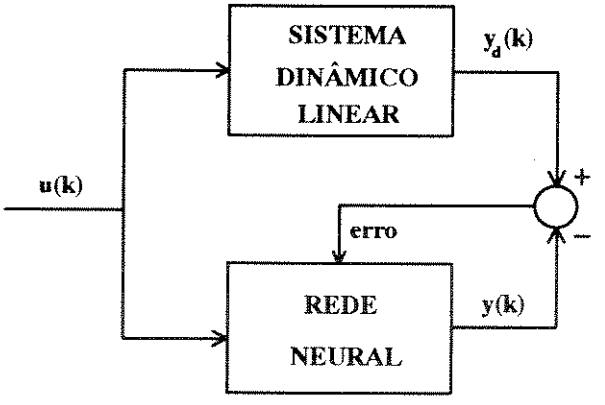


Figura 4.23 - Estrutura do processo de treinamento

A linguagem C de programação foi utilizada para desenvolver o programa de simulação computacional do treinamento da rede de Elman linear para identificação de sistemas dinâmicos lineares. Em todas as simulações, a entrada u(k) é do tipo ruído aleatório uniformemente distribuído no intervalo [-1,1].

[4]Esta equivalência é válida sempre que o vetor de entrada não participa diretamente na obtenção do vetor de saída, o qual é função apenas do vetor de estados (veja equação (4.2))

Exemplo 1 : Dada a representação por espaço de estados de um sistema linear de tempo contínuo de 1ª ordem na forma:

$$\begin{cases} \dot{x}(t) = -x(t) + u(t) \\ y(t) = x(t) \end{cases} \qquad \text{(E1.1)}$$

então, a sua representação equivalente de tempo discreto pode ser obtida através da amostragem da equação de tempo contínuo com um perído de amostragem T (veja Apêndice B):

$$\begin{cases} x(k+1) = e^{-T}x(k) + (1-e^{-T})u(k) \\ y(k) = x(k) \end{cases} \qquad \text{(E1.2)}$$

Expressando a equação (E1.2) na forma de diagrama de fluxo de sinais obtém-se:

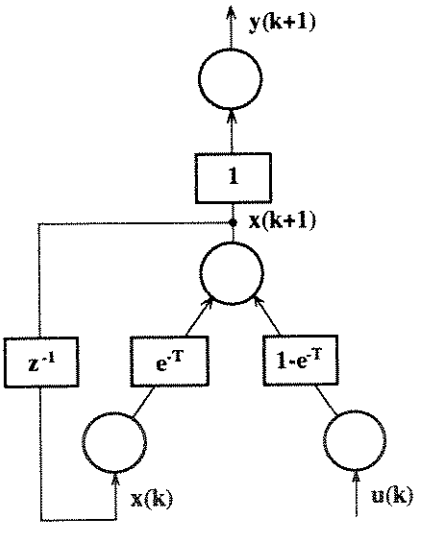


Figura E1.1 - Diagrama de fluxo de sinais do sistema

A figura E1.1 nada mais é que uma rede de Elman linear com uma única unidade na camada oculta. Portanto, tomando-se uma rede de Elman com estas mesmas dimensões e pesos gerados aleatoriamente, o processo de treinamento deverá ajustar os pesos de forma a que a dinâmica da rede de Elman se aproxime arbitrariamente bem da dinâmica do sistema. De fato, para um período de amostragem T = 0,1 segundos, pesos gerados aleatoriamente no intervalo [-0,1;0,1] foram conduzidos para os valores encontrados na figura E1.1 em 8 ciclos de treinamento.

Exemplo 2 : Dada a representação por espaço de estados de um sistema linear de tempo contínuo de 2ª ordem na forma:

$$\begin{cases} \dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix}u(t) \\ y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{x}(t) \end{cases} \qquad \text{(E2.1)}$$

então, a sua representação equivalente de tempo discreto, com período de amostragem T=0,2 segundos, é dada na forma:

$$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} 0,9671 & 0,1484 \\ -0,2968 & 0,5219 \end{bmatrix}\mathbf{x}(k) + \begin{bmatrix} 0,0164 \\ 0,1484 \end{bmatrix}u(k) \\ y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{x}(k) \end{cases} \qquad \text{(E2.2)}$$

Tomando-se uma rede de Elman linear com duas unidades na camada oculta e pesos gerados aleatoriamente no intervalo [-0,1;0,1], a partir da análise dos gráficos da figura E2.1 pode-se concluir que, após 350 ciclos de treinamento, a rede neural foi capaz de aproximar o comportamento dinâmico do sistema.

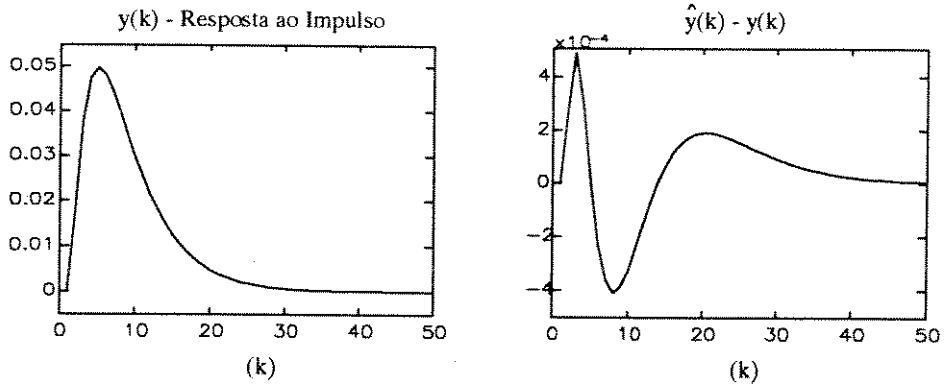


Figura E2.1 - Comparação entre o comportamento dinâmico do sistema e da rede neural

A equação de estado de tempo discreto representando o comportamento dinâmico da rede neural é dada na forma (os elementos das matrizes são os pesos da rede neural após o treinamento):

$$\begin{cases} \hat{\mathbf{x}}(k+1) = \begin{bmatrix} 0,3348 & -0,6940 \\ 0,2229 & 1,1413 \end{bmatrix}\hat{\mathbf{x}}(k) + \begin{bmatrix} -0,0401 \\ -0,0322 \end{bmatrix}u(k) \\ \hat{y}(k) = \begin{bmatrix} 0,2235 & -0,7963 \end{bmatrix}\hat{\mathbf{x}}(k) \end{cases} \qquad \text{(E2.3)}$$

Uma comparação entre as equações (E2.2) e (E2.3) permite concluir que a representação por espaço de estados não é única (veja demonstração no Apêndice B). Os vetores de estado destas duas equações estão relacionados por:

$$\mathbf{x} = \begin{bmatrix} 0,2236 & -0,7964 \\ -2,1486 & -1,9274 \end{bmatrix}\hat{\mathbf{x}} \qquad \text{(E2.4)}$$

Exemplo 3 : Dada a representação por espaço de estados de um sistema linear de tempo discreto de 3ª ordem na forma:

$$\begin{cases} \mathbf{x}(k+1) = \begin{bmatrix} 0,5 & 0,3 & -0,4 \\ -0,7 & 0,4 & 0,2 \\ 0,4 & -0,9 & -0,1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0,7 \\ -0,8 \\ 0,3 \end{bmatrix} u(k) \\ y(k) = [0,6 \quad 0,2 \quad -0,1]\mathbf{x}(k) \end{cases} \qquad \text{(E3.1)}$$

então, tomando-se uma rede de Elman linear com três unidades na camada oculta e pesos gerados aleatoriamente no intervalo [-0,1;0,1], após 1200 ciclos de treinamento, a rede neural foi capaz de aproximar o comportamento dinâmico do sistema, como indicado pelos gráficos da figura E3.1.

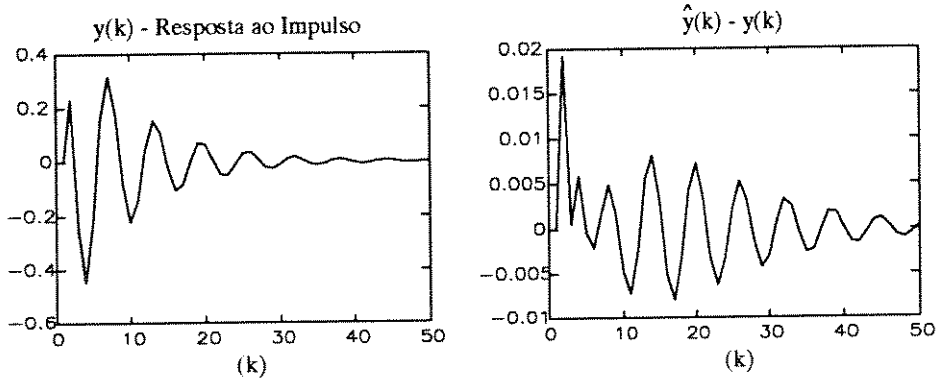


Figura E3.1 - Comparação entre o comportamento dinâmico do sistema e da rede neural

A equação de estado de tempo discreto representando o comportamento dinâmico da rede neural é dada na forma (os elementos das matrizes são os pesos da rede neural após o treinamento):

$$\begin{cases} \hat{\mathbf{x}}(k+1) = \begin{bmatrix} -0,0322 & -1,0011 & -0,0573 \\ 0,8359 & 0,9837 & 0,0385 \\ 0,0129 & 0,0464 & 0,0008 \end{bmatrix} \hat{\mathbf{x}}(k) + \begin{bmatrix} -0,1820 \\ -0,0264 \\ -0,0050 \end{bmatrix} u(k) \\ \hat{y}(k) = [-1,5404 \quad 1,1336 \quad 0,0450]\hat{\mathbf{x}}(k) \end{cases} \qquad \text{(E3.2)}$$

Estes exemplos ilustram o potencial das redes de Elman em identificar, em uma forma análoga à representação por espaço de estados, sistemas dinâmicos lineares de qualquer ordem descritos apenas pelos seus pares de vetores de entrada-saída. É certo que a tarefa vai se tornando mais complexa com o aumento da ordem do sistema a ser identificado (aumento do número de unidades na camada oculta) e da dimensão dos vetores de entrada e saída (aumento do número de unidades nas camadas de entrada e de saída). No entanto, com a utilização destas redes parcialmente recorrentes, a carga computacional é menor que no caso da utilização de redes não-recorrentes com LDA, principalmente devido à menor dimensão das redes recorrentes, as quais exigem um número bem menor de entradas.

A questão da dimensão da rede neural a ser empregada é decisivo no caso de não se conhecer a ordem do sistema a ser identificado, pois nesta situação é usual considerar-se valores superestimados para a ordem do sistema, de forma a garantir o sucesso da tarefa de identificação por parte da rede neural.

4.3.2.3. Extensão para o caso não-linear

A rede de Elman apresentada na figura 4.22 pode aproximar mapeamentos lineares ou não-lineares, dependendo do tipo de função de ativação presente em seus neurônios. Caso estas funções de ativação sejam lineares, existe, como verificado anteriormente, uma analogia perfeita com representações por espaço de estado de sistemas dinâmicos lineares. No entanto, esta analogia não se mantém no caso não-linear.

Apesar da capacidade de aproximar alguns tipos de mapeamentos não-lineares, redes de Elman cujos neurônios apresentam funções de ativação não-lineares não podem ser descritas na forma geral de representações por espaço de estado de sistemas não-lineares. Com isso, é sempre possível encontrar-se um sistema dinâmico não-linear cujo comportamento dinâmico não possa ser identificado por uma rede de Elman não-linear de qualquer dimensão. Esta limitação é compreensível, já que a rede de Elman, como apresentado anteriormente, representa a arquitetura mais simples de redes neurais recorrentes.

Portanto, é interessante desenvolver uma nova arquitetura de rede recorrente capaz de realizar esta tarefa de identificação de sistemas não-lineares. Para tanto, é suficiente a obtenção de uma arquitetura de rede cujo comportamento dinâmico possa ser descrito na forma geral de representações por espaço de estados de sistemas dinâmicos não-lineares.

A representação por espaço de estados mais geral de sistemas dinâmicos não-lineares é dada na forma:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}[\mathbf{x}(k), \mathbf{u}(k)] \\ \mathbf{y}(k) = \mathbf{G}[\mathbf{x}(k), \mathbf{u}(k)] \end{cases} \tag{4.6}$$

onde $\mathbf{x} \in R^n$, $\mathbf{y} \in R^p$, $\mathbf{u} \in R^r$, $\mathbf{F} \in F^{nX1}$ e $\mathbf{G} \in G^{pX1}$ de modo que:

$$\begin{cases} x_i(k+1) = f_i[\mathbf{x}(k), \mathbf{u}(k)] & \text{com } i = 1,\ldots,n \\ y_j(k) = g_j[\mathbf{x}(k), \mathbf{u}(k)] & \text{com } i = 1,\ldots,p \end{cases} \tag{4.7}$$

Expressando a equação (4.7) na forma de um diagrama de blocos obtém-se:

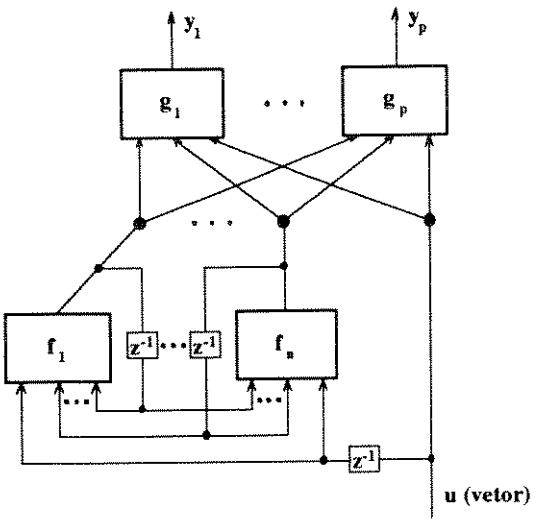


Figura 4.24 - Diagrama de blocos do sistema

Sabendo-se que redes neurais não-recorrentes de múltiplas camadas com pelo menos uma camada oculta são capazes de aproximar qualquer função contínua definida em um conjunto compacto no R^n (veja Apêndice A), então, se f_i (i=1,...,n) e g_j (j=1,...,p) são funções que atendem a esta condição, redes neurais não-recorrentes multi-camadas podem substituir, na figura 4.24, os blocos correspondentes às funções f_i (i=1,...,n) e g_j (j=1,...,p). Além disso, como redes neurais são processadores inerentemente paralelos, é fato que $\underline{k}$ redes neurais operando em paralelo são equivalentes a uma única rede neural devidamente dimensionada, com o mesmo número de saídas que a somatória das saídas das $\underline{k}$ redes e submetida a um conjunto de entradas que corresponda à união dos $\underline{k}$ conjuntos de entrada de cada rede em particular.

De posse destes conceitos estabelecidos acima, pode-se concluir que o mapeamento realizado pelo diagrama de blocos da figura 4.24 pode ser igualmente expresso na forma (N_1 e N_2 são redes neurais não-recorrentes):

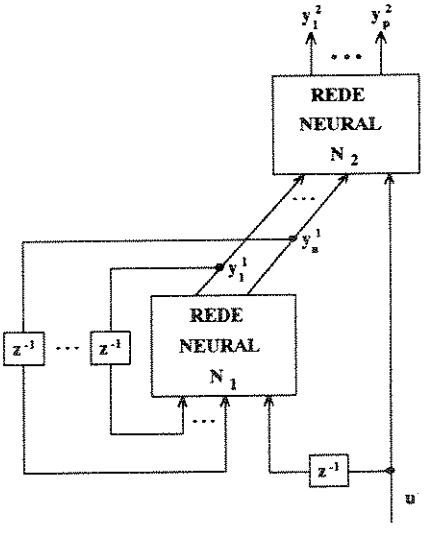


Figura 4.25 - Diagrama de fluxo de sinais do sistema utilizando-se redes neurais

4.3.2.4. Um exemplo de simulação - identificação de um sistema não-linear

A representação por espaço de estados de um sistema não-linear é apresentada na equação (4.8) a seguir:

$$\begin{cases} x(k+1) = f[x(k),u(k)] = \dfrac{u(k)^2 - x(k)}{1+x(k)^2} \\ y(k) = h[x(k),u(k)] = \dfrac{x(k) - u(k)}{1+4u(k)^2} \end{cases} \tag{4.8}$$

Tomando uma rede neural recorrente do tipo apresentado na figura 4.25, a aplicação da regra delta generalizada (equação (2.10), Capítulo 2) e do algoritmo de retro-propagação no tempo vai permitir a identificação do comportamento dinâmico deste sistema. Os parâmetros de treinamento são apresentados no Apêndice F, seção 2.

A implementação deste processo de treinamento foi desenvolvida utilizando-se a linguagem C de programação. Dado um conjunto de valores de entrada u(k) (k=1,...,50) gerados aleatoriamente no intervalo [-1,1], uma comparação entre o valor de saída produzido pela rede neural ($\hat{y}(k)$) e o correspondente valor de saída do sistema dinâmico da equação (4.8) (y(k)) é apresentado na figura 4.26.

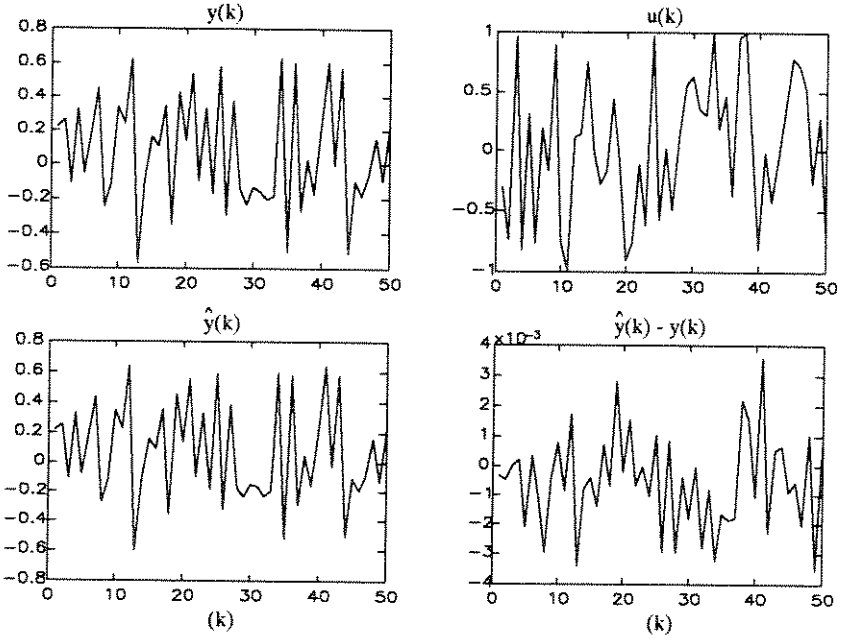


Figura 4.26 - Comparação entre o comportamento dinâmico do sistema e da rede neural

4.3.3. Análise do processo de treinamento de redes neurais recorrentes

A capacidade de representação de comportamentos dinâmicos não-lineares apresentada por redes neurais recorrentes constitui um forte atrativo para sua utilização em

problemas de identificação e controle de sistemas dinâmicos via treinamento por ajuste da intensidade das conexões. No entanto, a presença de recorrências pode levar a comportamentos dinâmicos não-lineares muito acentuados, dificultando a análise e implementação do processo de treinamento.

O processo de treinamento vai realizar o ajuste da intensidade das conexões (pesos) no sentido de otimizar uma função custo definida na saída da rede neural. No caso das redes neurais não-recorrentes cujas unidades processadoras são dotadas de funções de ativação do tipo sigmoidal, o processo de ajuste dos pesos é bem comportado, pois os sinais de saída utilizados no cálculo da função custo variam continuamente com os pesos. Por outro lado, o comportamento assintótico do ajuste de parâmetros de redes neurais recorrentes pode sofrer alterações qualitativas em certos pontos do espaço de parâmetros, conduzindo à ocorrência do fenômeno de bifurcação, com possíveis transições para o comportamento caótico. Portanto, o estudo dos problemas que podem surgir durante o processo de treinamento de redes neurais recorrentes é de grande importância prática.

Nas próximas seções, será investigado o fenômeno de bifurcação no processo de treinamento de redes neurais recorrentes, com o objetivo de detectar e apresentar soluções para os possíveis problemas daí advindos.

4.3.3.1. O fenômeno de bifurcação

O fenômeno de bifurcação no processo de treinamento de redes neurais recorrentes é geralmente investigado a nível de redes neurais de processamento contínuo, modeladas através de equações diferenciais de primeira ordem para cada unidade processadora (Doya-93)[17]; (Pearlmutter-90)[57]. Nesta dissertação, devido ao fato de serem estudadas e aplicadas apenas redes neurais de processamento discreto, toda a análise será desenvolvida utilizando-se equações a diferenças.

Um exemplo de ocorrência de bifurcação no processo de treinamento de redes neurais recorrentes pode ser obtido investigando-se o comportamento de uma rede composta por uma única unidade processadora, com função de ativação do tipo tangente hiperbólica e entrada nula, conforme apresentado na figura 4.27.

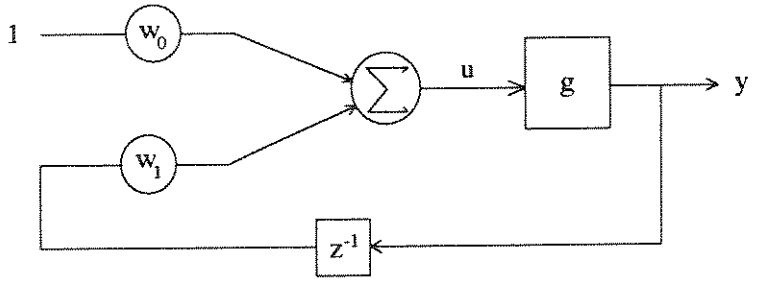


Figura 4.27 - Unidade processadora recorrente

O comportamento dinâmico da unidade processadora da figura 4.27 é descrito pela seguinte equação a diferenças:

$$y(k+1)=g(u(k+1)) = g(w_0+w_1y(k)) \tag{4.9}$$

sendo que, para valores constantes de w_0 e w_1 e uma determinada condição inicial y_I, a saída y(k) converge para um ponto fixo $y_f(w_0,w_1,y_I)$ quando k tende a infinito.

Fixando-se o valor do peso de recorrência w_1, a aplicação do processo de ajuste ao peso w_0 a partir da condição inicial y_I produz resultados qualitativamente distintos de acordo com a região de operação. O comportamento de $y_f(w_0,w_1,y_I)$ em função de w_0, com w_1 fixo e igual a 2, é apresentado na figura 4.28, onde os pontos fixos estáveis são representados pelas curvas de traço cheio, e os instáveis pela curva tracejada. Dada uma condição inicial y_I, as setas indicam os sentidos de convergência para os pontos fixos estáveis e divergência dos pontos fixos instáveis. Este comportamento pode conduzir a dinâmica de ajuste do peso w_0 ao fenômeno de bifurcação, como será descrito na seqüência.

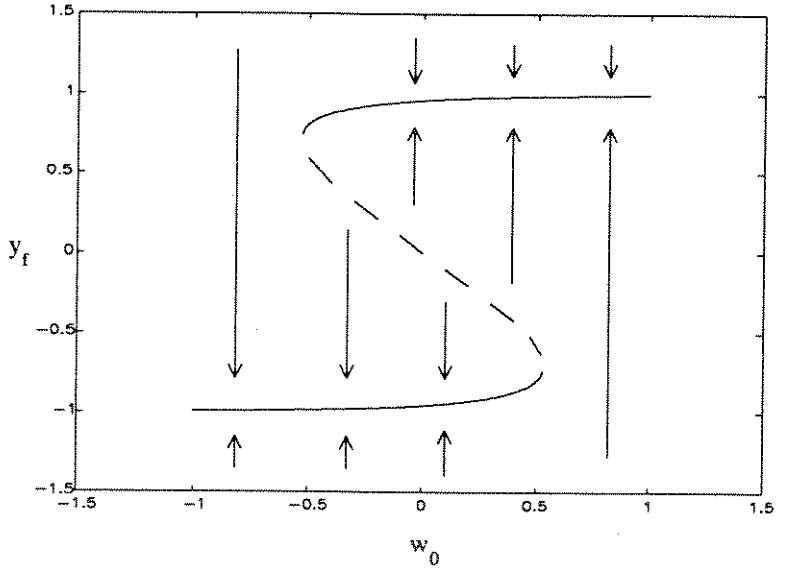


Figura 4.28 - Pontos fixos em função da variação do parâmetro w_0

Problema: Dados $w_0(0)=0$, $w_1(0)=2$ e $y_I=1,5$, ajustar o parâmetro w_0, mantendo w_1 constante, de forma a minimizar a função:

$$J(k) = \frac{1}{2}(y(k)-y_d)^2 \tag{4.10}$$

A solução deste problema através do ajuste do peso w_0 pode ser obtida aplicando-se a regra delta generalizada:

$$w_0(k+1) = w_0(k) - \alpha\frac{\partial J}{\partial w_0} + \beta\Delta w_0(k-1) \tag{4.11}$$

onde $\partial J/\partial w_0$ é calculado com a aplicação do algoritmo de retro-propagação no tempo produzindo:

$$\frac{\partial J}{\partial w_0} = \frac{\partial J}{\partial y}\frac{dg}{du}x_0 = (y - y_d)\frac{dg}{du}x_0 \qquad (4.12)$$

Caso 1: Suponha que y_d=0.85. Neste caso, a aplicação da equação (4.11) para o ajuste do peso w_0, com α=0.8 e β=0, produz o resultado apresentado na figura 4.29.

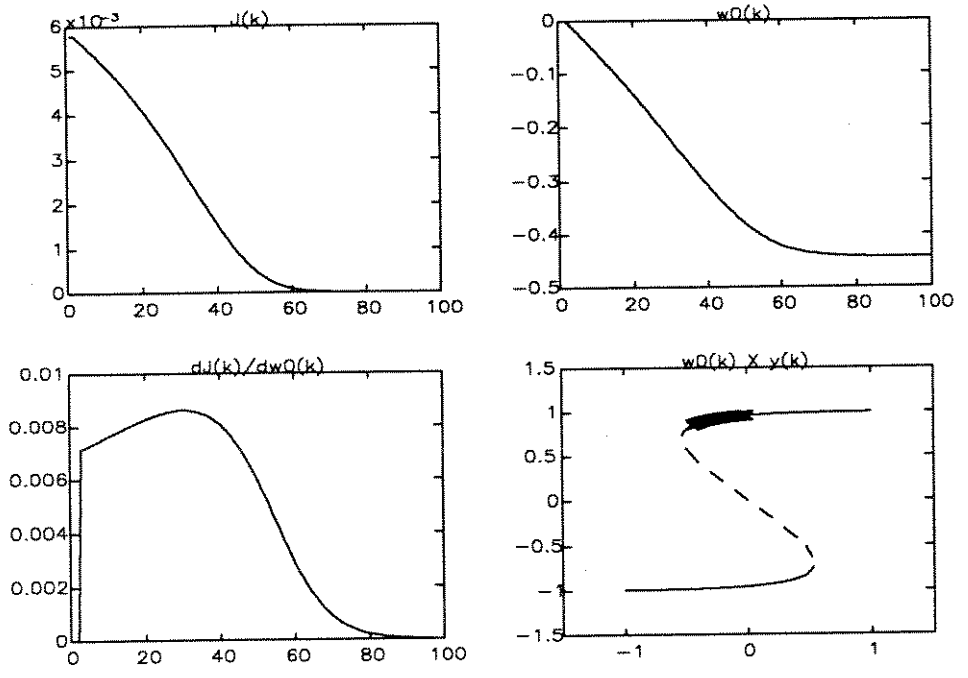


Figura 4.29 - Resultado do processo de ajuste do parâmetro w_0 para y_d=0.85

Pode-se verificar que o processo de ajuste foi bem sucedido, promovendo a minimização da função J. O gráfico **w0(k) X y(k)** permite acompanhar a evolução do saída y(k) em função do peso w_0, onde os pontos fixos são tomados como referência. Como a seqüência de pontos fixos obtida permanece na região dos pontos fixos estáveis, o processo de treinamento apresenta uma convergência assintótica.

Caso 2: Suponha que y_d=0. Assim, a aplicação da equação (4.11) para o ajuste do parâmetro w_0, com α=0.2 e β=0, produz o resultado apresentado na figura 4.30.

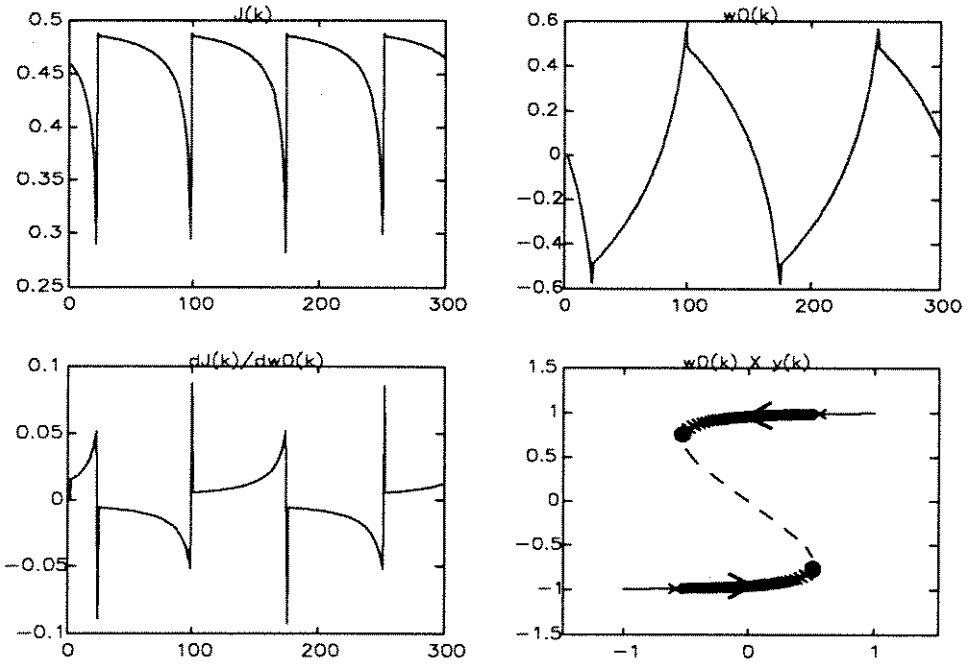


Figura 4.30- Resultado do processo de ajuste do parâmetro w_0 para y_d=0

Neste caso, o processo de ajuste não é capaz de promover a minimização da função J, convergindo para um ciclo limite de múltiplos pontos. O gráfico **w0(k) X y(k)** permite acompanhar a evolução do saída y(k) em função do peso w_0, dada pelos pontos em '**x**'. Os pontos fixos são tomados como referência e as setas indicam o sentido de variação. Quando um ponto de bifurcação '•' é atingido, o próximo ponto fixo estável se desloca para a outra curva de pontos fixos estáveis, fazendo com que a saída y(k) varie abruptamente, provocando, por sua vez, uma variação descontínua da função J e, consequentemente, do valor do gradiente. Dessa forma, a seqüência de pontos fixos obtida pelo processo de treinamento fica alternando entre as duas curvas de pontos fixos estáveis. Certamente nunca haverá convergência assintótica, já que o ponto fixo $y_f(w_0,w_1,y_1)$=0 que minimiza globalmente a função J é um ponto fixo instável.

Mesmo tendo sido utilizada uma rede neural composta por apenas uma unidade processadora aplicada a um problema de minimização trivial, foi possível detectar a ocorrência dos seguintes comportamentos não encontrados no caso de redes não-recorrentes:

- descontinuidades na definição do gradiente;
- variações abruptas do sinal de saída;
- impossibilidade de convergência do processo de treinamento.

Estes comportamentos indesejáveis não encontrados no caso de redes não-recorrentes podem apresentar efeitos ainda mais acentuados no caso de redes neurais recorrentes de dimensões elevadas aplicadas a problemas mais complexos.

4.3.3.2. Procedimentos de minimização dos efeitos de instabilidade no treinamento

Se um ponto for tomado aleatoriamente no espaço de parâmetros, é muito improvável que ele seja um ponto de bifurcação. No entanto, estando os parâmetros sendo ajustados continuamente durante o processo de treinamento, há uma possibilidade não desprezível de evolução para um ponto de bifurcação. O espaço de parâmetros de uma rede neural recorrente é dividido em muitas regiões com diferentes propriedades qualitativas, como por exemplo, regiões em que o espaço de parâmetros tem apenas um ponto de atração, um ciclo limite, dois pontos de atração, um ponto de atração e um ciclo limite, e assim por diante. Estas regiões qualitativamente distintas são limitadas por pontos de bifurcação. Portanto, alguns tipos de bifurcação são etapas inevitáveis na construção de redes neurais recorrentes com comportamentos dinâmicos desejados.

Como ilustrado no exemplo da seção anterior, existem alguns tipos de bifurcação que fazem com que o espaço de parâmetros apresente descontinuidades. Neste caso, os valores do gradiente podem ser muito elevados, provocando acentuadas variações de parâmetros. Mesmo que estas variações não conduzam à instabilidade numérica, o método de ajuste pode ter comprometida sua eficiência no caso de acentuadas variações de parâmetros em direções distintas. Alguns procedimentos têm sido desenvolvidos com o objetivo de minimizar ou até eliminar estes efeitos danosos ao processo de treinamento (Doya-92)[16], dentre os quais pode-se destacar:

- **utilização de algoritmos de treinamento não-recorrentes**: embora originalmente aplicados com o objetivo de reduzir a complexidade do processamento em redes neurais recorrentes, estes algoritmos têm sido capazes de evitar a passagem por pontos de bifurcação, já que não consideram o efeito das recorrências no processo de ajuste. Como conseqüência, sua aplicação é bastante restrita, pois o processo de ajuste não é capaz de explorar todas as potencialidades das redes neurais recorrentes.

- **pré-processamento do espaço de parâmetros**: se for previamente conhecida a estrutura qualitativa do espaço de parâmetros adequada para solucionar determinado problema (por exemplo, a quantidade e tipo de atratores que devem estar presentes no espaço de parâmetros) é possível realizar um pré-processamento dos parâmetros da rede neural de forma a introduzir este conhecimento inicial (Doya-92)[16].

- **realimentação utilizando os valores desejados ("teaching forcing")**: no caso de treinamento supervisionado e redes com recorrência externa (veja Capítulo 2), um procedimento comum é realimentar os valores desejados de saída e não as correspondentes saídas da rede neural. Assim, quando os valores reais de saída da rede apresentarem-se próximos dos valores desejados, é de se esperar que o comportamento não se altere caso se volte a utilizá-los na realimentação. Neste caso, embora não exista uma demonstração teórica de convergência assintótica, a existência de bifurcação na dinâmica do treinamento torna-se menos provável, pois reduz-se o grau de liberdade da dinâmica da rede.

Capítulo 5

Máquina de Indução - Modelagem e Técnicas de Controle

A aplicação de acionamentos elétricos de alto desempenho está vinculada à rapidez e precisão de operação, exigindo a atuação de controladores altamente eficientes. Em aplicações de controle de velocidade ou posição, a atuação sobre o conjugado em configurações de malha fechada representa uma operação básica. Na maioria dos casos, é interessante que haja um desacoplamento dos controles de conjugado e fluxo, de maneira que o máximo conjugado possível por unidade de corrente de estator possa ser obtido em diferentes condições de operação. O objetivo principal deste controle desacoplado é produzir uma resposta dinâmica rápida e um bom desempenho em estado estacionário.

Enquanto os motores de corrente contínua têm sido largamente utilizados em aplicações de controle de velocidade ou posição, os motores de corrente alternada têm sido tradicionalmente utilizados em aplicações de malha aberta e velocidade constante, onde as características de estado estacionário são as mais importantes. Isto se deve ao fato de que, em configurações de malha fechada e velocidade variável, tanto o comportamento estático quanto o dinâmico devem ser considerados, sendo que o comportamento dinâmico de uma máquina de corrente alternada é notadamente mais complexo que o de uma máquina de corrente contínua, por envolver não somente o controle direto de magnitudes de corrente ou tensão, mas também da freqüência elétrica e da posição relativa entre os fluxos magnéticos.

No entanto, apesar de apresentarem um comportamento dinâmico mais simples, as máquinas de corrente contínua, quando comparadas a alguns tipos de máquinas de corrente alternada, são inferiores em vários aspectos como custo, confiabilidade, potência, peso e velocidade máxima. Além disso, dentre as máquinas de corrente alternada, as máquinas de indução trifásicas com rotor tipo gaiola de esquilo se destacam em aplicações industriais devido à sua simplicidade, robustez e menor custo de fabricação.

Portanto, o desenvolvimento de técnicas de acionamento de alto desempenho para máquinas de corrente alternada, como no caso do motor de indução trifásico com rotor tipo gaiola de esquilo, é um procedimento desejável do ponto de vista de aplicação, sendo que a implementação em tempo real das técnicas já desenvolvidas tem sido viabilizada devido à evolução tecnológica dos dispositivos de potência. A configuração básica de sistemas de acionamento elétrico em corrente alternada é composta por um máquina de corrente alternada, um inversor e dispositivos de controle, conforme apresentado no esquema geral da figura 5.1.

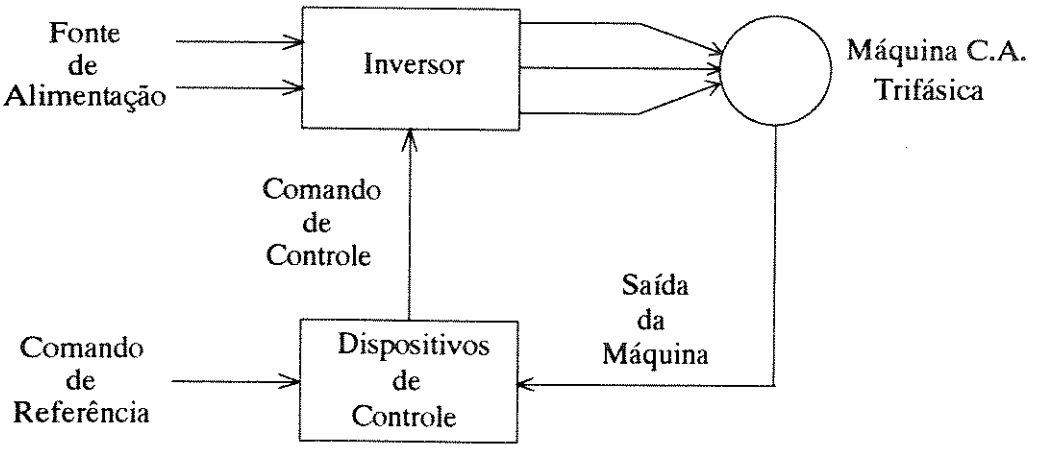


Figura 5.1 - Esquema geral de acionamento elétrico em corrente alternada

O objetivo deste capítulo é descrever cada um destes módulos constituintes, procurando enfatizar os conceitos fundamentais relacionados com a aplicação da máquina de indução com rotor tipo gaiola de esquilo e com o desenvolvimento de métodos de controle vetorial utilizados em aplicações de velocidade variável.

5.1. Máquina de Indução - Conceitos básicos

Nas máquinas de indução trifásicas, uma corrente alternada trifásica balanceada é fornecida diretamente aos enrolamentos do estator e, por ação de transformador, aos enrolamentos do rotor. Portanto, existe um acoplamento entre as fases de estator e rotor. Este acoplamento varia de acordo com a posição do rotor, implicando em um aumento de complexidade na descrição do comportamento dinâmico da máquina, que passa a ser baseado em equações diferenciais com coeficientes variantes no tempo.

Sempre que se possa garantir o suprimento balanceado de potência trifásica, o que ocorre quando a máquina é alimentada via conversores, a teoria dos dois eixos é normalmente empregada para modelagem dinâmica. Neste caso, as variáveis e parâmetros passam a ser representados em um sistema bifásico composto por dois eixos ortogonais mutuamente desacoplados, denominados eixo direto (d) e eixo em quadratura (q). As relações de transformação entre sistemas de referência trifásicos e bifásicos é apresentada no Apêndice D.

A adoção de um sistema de referência d-q girando à velocidade síncrona permite representar parâmetros que variam senoidalmente com o tempo como grandezas constantes, conduzindo à obtenção de modelos matemáticos mais simples para a máquina de indução.

5.1.1. O Modelo d-q da Máquina de Indução

No modelo da máquina de indução a ser apresentado a seguir, é utilizada uma série de parâmetros. Com o objetivo de facilitar o acompanhamento da dedução do modelo, optou-se por apresentar antecipadamente a tabela 5.1, contendo o símbolo e a correspondente descrição de cada parâmetro.

R_s	resistência de estator
R_r	resistência de rotor (referida ao estator)
L_m	indutância mútua (estator/rotor)
L_{ls}	indutância de dispersão do estator
$L_s = L_m + L_{ls}$	indutância própria do estator
L_{lr}	indutância de dispersão do rotor (referida ao estator)
$L_r = L_m + L_{lr}$	indutância própria do rotor (referida ao estator)
P	número de pólos
K	coeficiente de atrito viscoso
J	momento de inércia da máquina

Tabela 5.1 - Descrição dos parâmetros da máquina de indução

A equação de estator de uma máquina trifásica, tomando como referência as coordenadas estacionárias a^s-b^s-c^s, pode ser expressa na forma vetorial como segue[1]:

$$\overline{V}_s^s = R_s \overline{I}_s^s + \frac{d\overline{\psi}_s^s}{dt} \tag{5.1}$$

onde $\overline{V}_s^s$, $\overline{I}_s^s$ e $\overline{\psi}_s^s$ são, respectivamente, os vetores de tensão, corrente e fluxo concatenado instantâneos.

Aplicando as relações de transformação dadas pelas equações (D.1) e (D.3) (veja Apêndice D), é possível obter a representação das variáveis da equação (5.1) em um sistema de referência de dois eixos ortogonais, eixo direto (d) e eixo em quadratura (q), girando à velocidade síncrona w_e, na forma[2]:

$$\overline{V}_s^e = R_s \overline{I}_s^e + \frac{d\overline{\psi}_s^e}{dt} + \overline{w}_e \times \overline{\psi}_s^e \tag{5.2}$$

onde o termo adicional $\overline{w}_e \times \overline{\psi}_s^e$ é definido como uma tensão de velocidade, originada da rotação do sistema de referência. Assumindo que as componentes de seqüência zero são nulas, os vetores são representados em termos de suas componentes de eixo direto e eixo em quadratura na forma (Matsuo&Lipo-85)[43]:

$$\overline{V}_s^e = [v^e_{qs} \;\; v^e_{ds} \;\; 0]^T \qquad \overline{I}_s^e = [i^e_{qs} \;\; i^e_{ds} \;\; 0]^T$$

$$\overline{\psi}_s^e = [\psi^e_{qs} \;\; \psi^e_{ds} \;\; 0]^T \qquad \overline{w}_e = [0 \;\; 0 \;\; w_e]^T$$

---

[1] O superíndice s indica representação em um sistema de referência estático

[2] O superíndice e indica representação em um sistema de referência girando à velocidade síncrona w_e

A equação de rotor, com todas as variáveis e parâmetros referidos ao estator, pode ser analogamente expressa na forma:

$$\overline{V}_r^e = R_r \overline{I}_r^e + \frac{d\overline{\psi}_r^e}{dt} + \overline{w}_e \times \overline{\psi}_r^e \qquad (5.3)$$

A equação (5.3) corresponde à situação de rotor estacionário. Estando o rotor se movendo à velocidade angular w_r, a velocidade relativa do sistema de referência síncrono é dada por w_e-w_r, e a equação (5.3) assume a forma:

$$\overline{V}_r^e = R_r \overline{I}_r^e + \frac{d\overline{\psi}_r^e}{dt} + (\overline{w}_e - \overline{w}_r) \times \overline{\psi}_r^e \qquad (5.4)$$

onde $\quad \overline{V}_r^e = [v^e_{qr} \;\; v^e_{dr} \;\; 0]^T \qquad \overline{I}_r^e = [i^e_{qr} \;\; i^e_{dr} \;\; 0]^T$

$\overline{\psi}_r^e = [\psi^e_{qr} \;\; \psi^e_{dr} \;\; 0]^T \qquad \overline{w}_r = [0 \;\; 0 \;\; w_r]^T$

Considerando-se que o fluxo varia linearmente com a corrente, é possível definir as seguintes expressões algébricas para os fluxos concatenados de estator e rotor:

$$\overline{\psi}_r^e = L_{lr}\overline{I}_r^e + \overline{\psi}_m^e = L_{lr}\overline{I}_r^e + L_m(\overline{I}_r^e + \overline{I}_s^e) = L_r\overline{I}_r^e + L_m\overline{I}_s^e \qquad (5.5)$$

$$\overline{\psi}_s^e = L_{ls}\overline{I}_s^e + \overline{\psi}_m^e = L_{ls}\overline{I}_s^e + L_m(\overline{I}_r^e + \overline{I}_s^e) = L_s\overline{I}_s^e + L_m\overline{I}_r^e \qquad (5.6)$$

onde $\overline{\psi}_m^e$ é o fluxo de magnetização do entreferro.

Com isso, as equações (5.2) e (5.4) assumem a forma:

$$\overline{V}_s^e = \left[R_s + L_s\left(\frac{d}{dt} + \overline{w}_e \times\right)\right]\overline{I}_s^e + L_m\left(\frac{d}{dt} + \overline{w}_e \times\right)\overline{I}_r^e \qquad (5.7)$$

$$\overline{V}_r^e = L_m\left(\frac{d}{dt} + (\overline{w}_e - \overline{w}_r) \times\right)\overline{I}_s^e + \left[R_r + L_r\left(\frac{d}{dt} + (\overline{w}_e - \overline{w}_r) \times\right)\right]\overline{I}_r^e \qquad (5.8)$$

ou, de forma equivalente, desenvolvendo os produtos vetoriais, obtém-se:

$$\begin{bmatrix} v^e_{qs} \\ v^e_{ds} \\ v^e_{qr} \\ v^e_{dr} \end{bmatrix} = \begin{bmatrix} R_s + L_s\frac{d}{dt} & w_eL_s & L_m\frac{d}{dt} & w_eL_m \\ -w_eL_s & R_s + L_s\frac{d}{dt} & -w_eL_m & L_m\frac{d}{dt} \\ L_m\frac{d}{dt} & (w_e - w_r)L_m & R_r + L_r\frac{d}{dt} & (w_e - w_r)L_r \\ -(w_e - w_r)L_m & L_m\frac{d}{dt} & -(w_e - w_r)L_r & R_r + L_r\frac{d}{dt} \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \\ i^e_{qr} \\ i^e_{dr} \end{bmatrix} \qquad (5.9)$$

No Apêndice C é obtida uma série de expressões para o conjugado eletromagnético em termos de componentes de corrente e fluxo. Para efeito de controle, a

expressão mais adequada para representar o conjugado eletromagnético (T_e) é dada pela equação (C.22), que é reproduzida a seguir:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\bar{I}_s^e \times \bar{\psi}_r^e\right) = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}[\psi^e_{dr} i^e_{qs} - \psi^e_{qr} i^e_{ds}] \qquad (5.10)$$

A velocidade do rotor é relacionada ao conjugado eletromagnético T_e pela seguinte expressão:

$$\frac{dw_r}{dt} = -\frac{K}{J}w_r + \frac{P}{2J}(T_e - T_L) = -\frac{K}{J}w_r + \frac{P}{2J}(\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}[\psi^e_{dr} i^e_{qs} - \psi^e_{qr} i^e_{ds}] - T_L) \qquad (5.11)$$

onde T_L é o conjugado resistente de carga.

A equação (5.9), representando a parte eletromagnética, e a equação (5.11), representando a parte mecânica, definem completamente o modelo d-q da máquina de indução trifásica. Este modelo pressupõe uma série de idealizações na descrição física da máquina. Com relação à parte eletromagnética, são desprezados os efeitos de saturação, efeito pelicular e perdas no ferro (Fitzgerald,Kingsley&Kusko-75)[23]. Já com relação à parte mecânica, o conjugado resistente de carga T_L é fornecido diretamente, não sendo considerada a equação mecânica de produção de conjugado resistente a partir de uma carga especificada. Dependendo da aplicação, alguns fenômenos físicos normalmente desprezados devem ser incorporados através da utilização de modelos mais detalhados.

5.1.1.1. A Representação por Espaço de Estados

Tomando a equação (5.9), que corresponde à parte eletromagnética da dinâmica da máquina, é possível rearranjar os termos de maneira a obter equações que conduzam a uma representação por espaço de estados[3] do modelo d-q da máquina de indução na forma:

$$\begin{bmatrix} \frac{di^e_{qs}}{dt} \\ \frac{di^e_{ds}}{dt} \\ \frac{di^e_{qr}}{dt} \\ \frac{di^e_{dr}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{\sigma L_s} & w_e L_s L_r \sigma & \frac{L_m R_r}{L_s L_r \sigma} & 0 \\ -w_e L_s L_r \sigma & -\frac{R_s}{\sigma L_s} & 0 & \frac{L_m R_r}{L_s L_r \sigma} \\ \frac{L_m R_r}{L_s L_r \sigma} & 0 & -\frac{R_r}{\sigma L_r} & w_e L_s L_r \sigma \\ 0 & \frac{L_m R_r}{L_s L_r \sigma} & -w_e L_s L_r \sigma & -\frac{R_r}{\sigma L_r} \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \\ i^e_{qr} \\ i^e_{dr} \end{bmatrix} + \frac{1}{\sigma L_s L_r} \begin{bmatrix} -L_r & 0 & L_m & 0 \\ 0 & -L_r & 0 & L_m \\ L_m & 0 & -L_s & 0 \\ 0 & L_m & 0 & -L_s \end{bmatrix} \begin{bmatrix} v^e_{qs} \\ v^e_{ds} \\ v^e_{qr} \\ v^e_{dr} \end{bmatrix} \qquad (5.12)$$

onde $\sigma = \frac{L_s L_r - L_m^2}{L_s L_r}$ é denominado fator de acoplamento.

[3] Veja equação B.2, Apêndice B.

Com base nas relações entre fluxo e corrente dadas pelas equações (5.5) e (5.6), é possível expressar os vetores de corrente em termos dos vetores de fluxo na forma:

$$\bar{I}_s^e = \frac{1}{\sigma L_s}\bar{\psi}_s^e + \frac{\sigma-1}{\sigma L_m}\bar{\psi}_r^e \tag{5.13}$$

$$\bar{I}_r^e = \frac{1}{\sigma L_r}\bar{\psi}_r^e + \frac{\sigma-1}{\sigma L_m}\bar{\psi}_s^e \tag{5.14}$$

Substituindo as equações (5.13) e (5.14) na equação (5.12), obtém-se outro modelo equivalente na forma de equação de estado, onde as componentes de fluxo formam o conjunto de variáveis de estado. A equação de estado correspondente assume a forma:

$$\begin{bmatrix} \frac{d\psi_{qs}^e}{dt} \\ \frac{d\psi_{ds}^e}{dt} \\ \frac{d\psi_{qr}^e}{dt} \\ \frac{d\psi_{dr}^e}{dt} \end{bmatrix} = \frac{-1}{\sigma}\begin{bmatrix} \frac{R_s}{L_s} & w_e\sigma & \frac{(\sigma-1)R_s}{L_m} & 0 \\ -w_e\sigma & \frac{R_s}{L_s} & 0 & \frac{(\sigma-1)R_s}{L_m} \\ \frac{(\sigma-1)R_r}{L_m} & 0 & \frac{R_r}{L_r} & (w_e-w_r)\sigma \\ 0 & \frac{(\sigma-1)R_r}{L_m} & -(w_e-w_r)\sigma & \frac{R_r}{L_r} \end{bmatrix}\begin{bmatrix} \psi_{qs}^e \\ \psi_{ds}^e \\ \psi_{qr}^e \\ \psi_{dr}^e \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} v_{qs}^e \\ v_{ds}^e \\ v_{qr}^e \\ v_{dr}^e \end{bmatrix} \tag{5.15}$$

Para se obter uma representação que tenha somente grandezas de estator como variáveis de estado, basta isolar o vetor $\bar{I}_r^e$ na equação (5.6) e substituir as componentes d-q resultantes na equação (5.12), produzindo:

$$\begin{bmatrix} \frac{d\psi_{qs}^e}{dt} \\ \frac{d\psi_{ds}^e}{dt} \\ \frac{di_{qs}^e}{dt} \\ \frac{di_{ds}^e}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -w_e & -R_s & 0 \\ w_e & 0 & 0 & -R_s \\ \frac{R_r}{\sigma L_s L_r} & -\frac{w_r}{\sigma L_s} & -\frac{1}{\sigma}\left(\frac{R_s}{L_s}+\frac{R_r}{L_r}\right) & -(w_e-w_r) \\ \frac{w_r}{\sigma L_s} & \frac{R_r}{\sigma L_s L_r} & (w_e-w_r) & -\frac{1}{\sigma}\left(\frac{R_s}{L_s}+\frac{R_r}{L_r}\right) \end{bmatrix}\begin{bmatrix} \psi_{qs}^e \\ \psi_{ds}^e \\ i_{qs}^e \\ i_{ds}^e \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{\sigma L_s} & 0 & \frac{\sigma-1}{\sigma L_m} & 0 \\ 0 & \frac{1}{\sigma L_s} & 0 & \frac{\sigma-1}{\sigma L_m} \end{bmatrix}\begin{bmatrix} v_{qs}^e \\ v_{ds}^e \\ v_{qr}^e \\ v_{dr}^e \end{bmatrix} \tag{5.16}$$

As equações (5.12), (5.15) e (5.16) têm como entrada as componentes de tensão de estator e rotor, sendo que para rotor em gaiola as componentes de tensão de rotor são sempre nulas. Portanto, estas equações representam a máquina de indução alimentada por tensão.

Neste trabalho, a máquina de indução será alimentada por corrente, através da utilização de um inversor fonte de corrente[4]. Neste caso, nenhuma das equações acima é adequada para representar o modelo de estado da parte eletromagnética da máquina. Além disso, assumindo que o inversor fonte de corrente atua diretamente sobre as componentes de corrente de estator, o comportamento dinâmico do circuito de estator fica totalmente vinculado à atuação do circuito de alimentação. Portanto, a equação (5.8), representando o comportamento dinâmico do circuito de rotor, é suficiente para descrever a dinâmica da parte eletromagnética da máquina. A nova representação por espaço de estados pode ser obtida utilizando-se o mesmo procedimento aplicado na obtenção das equações (5.12), (5.15) e (5.16), apresentado detalhadamente na seqüência.

Tomando a equação (5.8), é possível desenvolver o produto vetorial e expressar as correntes e tensões em termos de suas componentes d-q, produzindo a seguinte equação:

$$\begin{bmatrix} v^e_{qr} \\ v^e_{dr} \end{bmatrix} = \begin{bmatrix} L_m \frac{d}{dt} & (w_e - w_r)L_m & R_r + L_r \frac{d}{dt} & (w_e - w_r)L_r \\ -(w_e - w_r)L_m & L_m \frac{d}{dt} & -(w_e - w_r)L_r & R_r + L_r \frac{d}{dt} \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \\ i^e_{qr} \\ i^e_{dr} \end{bmatrix} \tag{5.17}$$

A partir das equações (5.5) e (5.6), é possível expressar as componentes d-q de corrente de rotor em termos de componentes de corrente de estator e de fluxo de estator, de entreferro ou de rotor, na forma:

$$\begin{bmatrix} i^e_{qr} \\ i^e_{dr} \end{bmatrix} = \frac{1}{L_m} \begin{bmatrix} \psi^e_{qs} \\ \psi^e_{ds} \end{bmatrix} - \frac{L_s}{L_m} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \end{bmatrix} \tag{5.18}$$

$$\begin{bmatrix} i^e_{qr} \\ i^e_{dr} \end{bmatrix} = \frac{1}{L_r} \begin{bmatrix} \psi^e_{qr} \\ \psi^e_{dr} \end{bmatrix} - \frac{L_m}{L_r} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \end{bmatrix} \tag{5.19}$$

$$\begin{bmatrix} i^e_{qr} \\ i^e_{dr} \end{bmatrix} = \frac{1}{L_m} \begin{bmatrix} \psi^e_{qm} \\ \psi^e_{dm} \end{bmatrix} - \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \end{bmatrix} \tag{5.20}$$

Portanto, com a substituição das equações (5.18), (5.19) e (5.20) na equação (5.17), é possível obter representações de estado para a dinâmica da parte eletromagnética da máquina tendo componentes de fluxo como variáveis de estado. No entanto, para variáveis de estado dadas pelas componentes de fluxo de estator (substituição da equação (5.18)) ou de entreferro (substituição da equação (5.20)), as representações de estado

[4] Na verdade, a máquina é alimentada em tensão, mas quando esta alimentação é definida por algum tipo de inversor fonte de corrente, ela é considerada como alimentação em corrente.

correspondentes são muito complexas. Isto de deve basicamente ao fato de se buscar uma equação para o circuto de rotor que não envolva diretamente grandezas elétricas de rotor. Por outro lado, para variáveis de estado dadas pelas componentes de fluxo de rotor (substituição da equação (5.19)) e considerando rotor em gaiola onde as componentes de tensão de rotor são sempre nulas, uma nova representação da equação (5.17) pode ser obtida na forma:

$$\begin{bmatrix} \frac{d\psi^e_{qr}}{dt} \\ \frac{d\psi^e_{dr}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -(w_e - w_r) \\ (w_e - w_r) & -\frac{R_r}{L_r} \end{bmatrix} \begin{bmatrix} \psi^e_{qr} \\ \psi^e_{dr} \end{bmatrix} + \begin{bmatrix} \frac{L_m R_r}{L_r} & 0 \\ 0 & \frac{L_m R_r}{L_r} \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \end{bmatrix} \qquad (5.21)$$

O modelo completo da máquina de indução alimentada por corrente, composto pela parte eletromagnética (equação (5.21)) e parte mecânica (equação (5.11)), pode ser representado na forma de equação de estado, como segue:

$$\begin{bmatrix} \frac{d\psi^e_{qr}}{dt} \\ \frac{d\psi^e_{dr}}{dt} \\ \frac{dw_r}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_r}{L_r} & -(w_e - w_r) & 0 \\ (w_e - w_r) & -\frac{R_r}{L_r} & 0 \\ -\frac{P}{2J}\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} i^e_{ds} & \frac{P}{2J}\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} i^e_{qs} & -\frac{K}{J} \end{bmatrix} \begin{bmatrix} \psi^e_{qr} \\ \psi^e_{dr} \\ w_r \end{bmatrix} + \begin{bmatrix} \frac{L_m R_r}{L_r} & 0 & 0 \\ 0 & \frac{L_m R_r}{L_r} & 0 \\ 0 & 0 & \frac{-P}{2J} \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \\ T_L \end{bmatrix} \qquad (5.22)$$

5.1.1.2. Obtenção do modelo discreto

Simulações computacionais utilizando o modelo da máquina de indução requerem uma versão discreta para os modelos de tempo contínuo obtidos na seção anterior. Abordagens de tempo discreto são também comuns na implementação de sistemas de acionamento de máquinas de indução, devido à utilização predominante de processadores digitais.

A obtenção de modelos discretos a partir de sistemas inerentemente contínuos é comumente realizada através de processos de amostragem. Observando a equação (5.22), correspondente ao modelo da máquina de indução alimentada por corrente, é possível verificar a presença, na matriz de estado, de parâmetros variantes no tempo. Portanto, a máquina de indução é um sistema não-estacionário[5]. Com isso, o procedimento de discretização numérica deve ser feito a cada passo de amostragem[6], com a consideração adicional de que os parâmetros variantes da matriz de estado são constantes entre dois instantes consecutivos de amostragem. Reescrevendo a equação (5.22) na forma:

[5]A definição de sistema não-estacionário é apresentada no Capítulo 3.
[6]O processo de discretização é apresentado no Apêndice B.

$$\begin{bmatrix} \frac{d\psi_{qr}^{e}}{dt} \\ \frac{d\psi_{dr}^{e}}{dt} \\ \frac{dw_r}{dt} \end{bmatrix} = A_n \begin{bmatrix} \psi_{qr}^{e} \\ \psi_{dr}^{e} \\ w_r \end{bmatrix} + B \begin{bmatrix} i_{qs}^{e} \\ i_{ds}^{e} \\ T_L \end{bmatrix} \tag{5.23}$$

com

$$A_n = \begin{bmatrix} -\frac{R_r}{L_r} & -w_{slip}(nT) & 0 \\ w_{slip}(k) & -\frac{R_r}{L_r} & 0 \\ -\frac{P}{2J}\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} i_{ds}^{e}(nT) & \frac{P}{2J}\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r} i_{qs}^{e}(nT) & -\frac{K}{J} \end{bmatrix} \qquad B = \begin{bmatrix} \frac{L_m R_r}{L_r} & 0 & 0 \\ 0 & \frac{L_m R_r}{L_r} & 0 \\ 0 & 0 & \frac{-P}{2J} \end{bmatrix}$$

onde T é o período de amostragem, $w_{slip}=w_e-w_r$ é a velocidade de escorregamento e A_n é a matriz de estado definida no n-ésimo período de amostragem, obtém-se:

$$\begin{bmatrix} \psi_{qr}^{e}(k+1) \\ \psi_{dr}^{e}(k+1) \\ w_r(k+1) \end{bmatrix} = G(T) \begin{bmatrix} \psi_{qr}^{e}(k) \\ \psi_{dr}^{e}(k) \\ w_r(k) \end{bmatrix} + H(T) \begin{bmatrix} i_{qs}^{e}(k) \\ i_{ds}^{e}(k) \\ T_L(k) \end{bmatrix} \tag{5.24}$$

onde:

$$G(T) = e^{A_n T} = \begin{bmatrix} e^{-T\frac{R_r}{L_r}}\cos\left[w_{slip}(nT).T\right] & -e^{-T\frac{R_r}{L_r}}\text{sen}\left[w_{slip}(nT).T\right] & 0 \\ e^{-T\frac{R_r}{L_r}}\text{sen}\left[w_{slip}(nT).T\right] & e^{-T\frac{R_r}{L_r}}\cos\left[w_{slip}(nT).T\right] & 0 \\ \zeta\frac{3}{2}\left(\frac{P}{2}\right)^2\frac{1}{J}\frac{L_m}{L_r} i_{ds}^{e}(nT) & \zeta\frac{3}{2}\left(\frac{P}{2}\right)^2\frac{1}{J}\frac{L_m}{L_r} i_{qs}^{e}(nT) & e^{-\frac{K}{J}T} \end{bmatrix}$$

$$H(T) = \int_0^T e^{A_n t}dt.B = \begin{bmatrix} h_1 & h_2 & 0 \\ -h_2 & h_1 & 0 \\ 0 & 0 & -\zeta\frac{P}{2} \end{bmatrix}$$

com

$$h_1 = \frac{R_r L_m L_r}{w_{slip}^2 L_r^2 + R_r^2}\left[e^{-T\frac{R_r}{L_r}}\left(w_{slip}\,sen(w_{slip}t) - \frac{R_r}{L_r}\cos(w_{slip}T)\right) + \frac{R_r}{L_r}\right]$$

$$h_2 = -\frac{R_r L_m L_r}{w_{slip}^2 L_r^2 + R_r^2}\left[e^{-T\frac{R_r}{L_r}}\left(w_{slip}\cos(w_{slip}T) + \frac{R_r}{L_r}sen(w_{slip}T)\right) + w_{slip}\right]$$

$$\zeta = \begin{cases} \frac{1}{K}\left(1 - e^{-\frac{K}{J}T}\right) & \text{se } K \neq 0 \\ T & \text{se } K = 0 \end{cases}$$

5.2. Inversores - Conceitos Básicos

A evolução tecnológica dos dispositivos de potência permitiu a obtenção de avanços significativos na área de acionamento de máquinas assíncronas. Particularmente, o desenvolvimento de dispositivos semicondutores com chaveamento em alta freqüência tem viabilizado a implementação em tempo real da grande maioria dos sistemas de acionamento já desenvolvidos.

A alimentação da máquina é geralmente realizada via inversores, os quais podem ser classificados de acordo com suas características em dois tipos básicos:

- Inversor de chaveamento por histerese
- Inversor de chaveamento por modulação de largura de pulso (PWM)

Neste trabalho será utilizado um inversor PWM para a geração das correntes de alimentação, o qual, conforme apresentado na figura 5.2, recebe como entrada a seqüência de chaveamento para as fases (Brod&Novotny-85)[12].

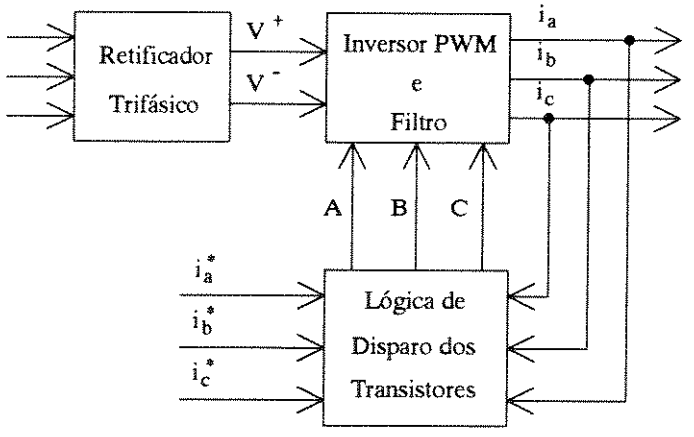


Figura 5.2 - Inversor fonte de corrente alimentado por corrente

As correntes comandadas i_a^*, i_b^* e i_c^* são comparadas com as correntes atuais de alimentação i_a, i_b e i_c, e o resultado desta comparação é utilizado na definição da lógica de disparo dos transistores. Uma vez definida, esta seqüência de disparo é utilizada para chavear o conjunto de transistores do inversor PWM. Os sinais de corrente então gerados, correspondendo a uma seqüência de pulsos, são filtrados e utilizados para alimentar a máquina.

5.3. Técnicas de controle para operação em velocidade variável

As técnicas de controle da máquinas de indução operando como motor podem ser dividido em dois tipos:

- técnicas escalares
- técnicas vetoriais

Na aplicação de técnicas escalares de controle de máquinas de indução através de inversores de tensão ou corrente, amplitude e freqüência são os parâmetros básicos de controle. Quando o sinal de controle atua, por exemplo, sobre a tensão, tanto o conjugado quanto o fluxo de entreferro são funções da amplitude e freqüência da tensão. Este acoplamento existente entre as variáveis a serem controladas é responsável pela dinâmica lenta da máquina em operações de velocidade variável, conforme descrito a seguir:

- um aumento (redução) no valor do conjugado, pode ser obtido através do aumento (redução) da freqüência da tensão de alimentação;
- como a constante de tempo mecânica é maior que a constante de tempo elétrica, o efeito sobre a velocidade de rotação da máquina não será sentido imediatamente, provocando um aumento (redução) da frequência de escorregamento;
- com uma freqüência de escorregamento maior (menor), ocorre uma redução (aumento) do fluxo de entreferro (Bose-86)[11];
- esta queda (elevação) do nível de excitação da máquina vai provocar, como conseqüência, uma redução (aumento) no valor do conjugado;
- sendo assim, para se chegar ao aumento (redução) de conjugado desejado, é preciso corrigir o valor do fluxo por meio de uma malha de compensação;
- este processo certamente vai aumentar o tempo de resposta do motor ao comando de controle.

Um comportamento semelhante ocorre quando o controle atua sobre a corrente (Bose-86)[11]. Como será apresentado nas próximas seções, este efeito indesejável causado pelo acoplamento entre variáveis pode ser evitado quando se utilizam métodos vetoriais de controle baseados no princípio de orientação de fluxo. Estes métodos são geralmente aplicado às máquinas de corrente alternada, como máquinas de indução e máquinas síncronas.

5.3.1. A obtenção do desacoplamento entre as variáveis de controle

As máquinas de corrente contínua com excitação independente possuem um modelo dinâmico extremamente simples e desacoplado[7], isto é, a interação entre as variáveis de controle praticamente inexiste, possibilitando que os sinais de controle atuem independentemente sobre cada variável a ser controlada. Sendo assim, conjugado e fluxo podem ser determinados isoladamente, sendo que o comportamento do conjugado pode ser descrito pela equação a seguir (Fitzgerald,Kingsley&Kusko-75)[23]:

$$T_e = K_{cc} I_f I_a = K_{cc} K_f \psi_d I_a = K'_{cc} \psi_d I_a \qquad (5.25)$$

onde a corrente de armadura (I_a) e a corrente de campo ($I_f = K_f \psi_d$) são as variáveis de controle. Normalmente, a corrente de campo é fixada de forma a se estabelecer uma determinada intensidade constante de fluxo (ψ_d), sendo que o conjugado passa, então, a ser função apenas da corrente de armadura, ou seja, as variáveis de controle estão desacopladas.

Esta mesma estrutura de controle pode também ser obtida para uma máquina de indução quando sua representação é definida em um sistema de referência bifásico girando à velocidade síncrona, e os seguintes procedimentos são adotados:

Tomando, por exemplo, uma das expressões de conjugado obtidas no Apêndice C, onde as variáveis estão todas referidas ao estator (equação (C.22)):

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\psi^e_{dr} i^e_{qs} - \psi^e_{qr} i^e_{ds}\right) \qquad (5.26)$$

e assumindo, agora, que o eixo direto d^e está sempre posicionado na direção do vetor fluxo de rotor[8] ψ_r^e, então:

$$\psi^e_{dr} = \left\|\psi^e_r\right\| \qquad \psi^e_{qr} = 0$$

e o conjugado pode, então, ser reescrito na forma:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\psi^e_{dr} i^e_{qs} = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left\|\psi^e_r\right\| i^e_{qs} \qquad (5.27)$$

A partir da equação (5.27), pode-se obter uma expressão para o conjugado totalmente análoga àquela apresentada para uma máquina de corrente contínua com excitação independente (equação (5.25)), ou seja:

$$T_e = K_{ca}\left\|\psi^e_r\right\| i^e_{qs} \qquad (5.28)$$

---

[7] O desacoplamento é válido sempre que for possível desprezar-se os efeitos de saturação de campo e desmagnetização de armadura.

[8] De acordo com a expressão de torque considerada, a orientação do eixo direto d^e girando à velocidade síncrona pode tomar como referência o vetor fluxo de rotor, de entreferro ou de estator.

As variáveis de controle i^e_{qs} e $\|\psi^e_r\| = \psi^e_{dr}$ são mutuamente desacopladas e podem ser independentemente variadas sem afetar a componente ortogonal. Normalmente, como na máquina de corrente contínua, a componente $\|\psi^e_r\|$ é mantida constante e o conjugado fica apenas como função da componente i^e_{qs}.

5.3.2. O Princípio da Orientação de Fluxo

O princípio da orientação de fluxo é um método de controle vetorial que vai atuar sobre correntes ou tensões de estator com base no controle da posição espacial do fluxo de rotor, de entreferro ou de estator[9]. O alinhamento entre o eixo direto do referencial d^e-q^e síncrono e o vetor de fluxo, como demonstrado na seção anterior, vai permitir o desacoplamento entre as variáveis de controle, possibilitando a obtenção de respostas dinâmicas mais rápidas, importantes para operação da máquina de indução em velocidade variável. Os fundamentos da implementação do controle vetorial, indicando a seqüência de transformações necessárias para conversão entre o sistema de referência d^e-q^e e o sistema trifásico original podem ser explicados com o auxílio da figura 5.3, onde as correntes são tomadas como variáveis de controle (Bose-86)[11].

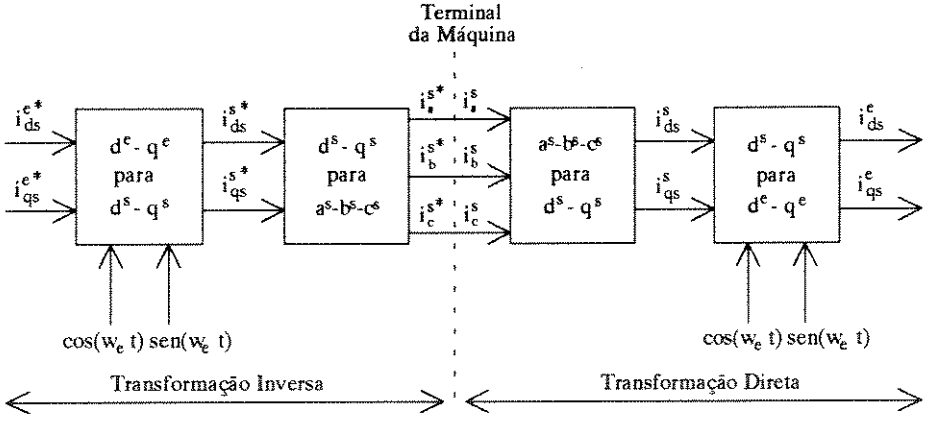


Figura 5.3 - Seqüência de transformação para implementação do controle vetorial

Transformação inversa

A partir das componentes d-q de corrente provenientes do comando de controle, duas transformações são realizadas para obter as correntes de fase que vão alimentar a máquina:

- as componentes i^{e*}_{ds} e i^{e*}_{qs}, definidas em um referencial d^e-q^e síncrono, são rebatidas em um sistema de referência d^s-q^s estacionário (veja equação (D.4), Apêndice D),

[9]As representações obtidas com orientação no fluxo de rotor são muito mais simples e fáceis de manipular, conforme discutido anteriormente.

produzindo as componentes i_{ds}^{s*} e i_{qs}^{s*}, sendo que o sincronismo é determinado pela posição instantânea do vetor de fluxo ($\theta_e = w_e t$), na forma de um vetor unitário:

$$v_\theta = [\cos(w_e t) \;\; \text{sen}(w_e t)]^T \tag{5.29}$$

- uma transformação bifásica/trifásica (veja equação (D.1), Apêndice D) é, então, aplicada para converter as componentes de corrente i_{ds}^{s*} e i_{qs}^{s*}, em correntes de fase i_a^{s*}, i_b^{s*} e i_c^{s*}.

Transformação direta

As correntes de referência i_a^{s*}, i_b^{s*} e i_c^{s*} são convertidas nas correntes de fase i_a^s, i_b^s e i_c^s pelo inversor fonte de corrente (veja figura 5.2), que não é apresentado para facilitar a visualização do processo de transformação. As correntes de fase i_a^s, i_b^s e i_c^s são utilizadas diretamente para alimentar a máquina. No entanto, quando a máquina é descrita pelo seu modelo d-q, é requerida a aplicação de duas transformações para a obtenção das correntes de alimentação:

- uma transformação trifásica/bifásica (veja equação (D.2), Apêndice D) é aplicada para rebater as correntes de fase i_a^s, i_b^s e i_c^s em um sistema de eixos d^s-q^s estacionário, produzindo as componentes i^s_{ds} e i^s_{qs}.
- as componentes i^s_{ds} e i^s_{qs} são, por sua vez, rebatidas em um sistema de referência d^e-q^e girando à velocidade síncrona (veja equação (D.3), Apêndice D), produzindo as componentes i^e_{ds} e i^e_{qs}, sendo que o sincronismo é determinado pela posição instantânea do vetor de fluxo.

É importante observar que as transformações diretas e inversas são dadas por relações algébricas, assegurando que as componentes de corrente comandadas atuem no mesmo instante em que foram produzidas. Outro fator importante é a necessidade de utilização da posição determinada para o vetor de fluxo na obtenção das relações de transformação direta e inversa.

5.3.2.1. Orientação baseada no fluxo de rotor

Com base na equação (5.22), que descreve o modelo da máquina de indução em um referencial bifásico d^e-q^e girando à velocidade síncrona, se o eixo d^e for posicionado na direção do vetor fluxo de rotor (que gira à velocidade síncrona), então é possível afirmar que:

$$\psi_{qr}^e = \frac{d\psi_{qr}^e}{dt} = 0 \tag{5.30}$$

$$\psi_{dr}^e = \left\| \psi_r^e \right\| \tag{5.31}$$

Substituindo as equações (5.30) e (5.31) na equação (5.22), obtém-se o seguinte conjunto de três equações:

$$w_{slip} = \frac{L_m R_r}{L_r} \frac{i^e_{qs}}{\left\|\psi^e_r\right\|} = \frac{2}{3}\frac{2}{P} R_r \frac{T_e}{\left\|\psi^e_r\right\|^2} \tag{5.32}$$

$$\frac{L_r}{R_r}\frac{d\left\|\psi^e_r\right\|}{dt} + \left\|\psi^e_r\right\| = L_m i^e_{ds} \tag{5.33}$$

$$\frac{dw_r}{dt} = -\frac{K}{J} w_r + \frac{P}{2J}\left[T_e - T_L\right] = -\frac{K}{J} w_r + \frac{P}{2J}\left[\left(\frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left\|\psi^e_r\right\| i^e_{qs}\right) - T_L\right] \tag{5.34}$$

Se a orientação de fluxo for mantida a todo instante, as equações (5.32), (5.33) e (5.34) passam a representar o modelo da máquina de indução. Da equação (5.33), pode-se concluir que o valor do fluxo passa a ser controlado apenas pela componente de corrente de estator de eixo direto (i^e_{ds}), enquanto que da equação (5.34), deduz-se que o comportamento do conjugado eletromagnético é função apenas da componente em quadratura da corrente de estator (i^e_{qs}). Ficam, assim, estabelecidas as condições para aplicação do método de controle vetorial baseado no princípio da orientação de fluxo:

1) obtenção de uma representação por espaço de estados do modelo d-q da máquina de forma que as variáveis de estado e as variáveis de entrada sejam adequadas para aplicação. Por exemplo, se a máquina for alimentada via inversor fonte de corrente, as componentes de corrente de alimentação devem participar do conjunto de variáveis de entrada. Analogamente, o tipo de fluxo escolhido no processo de orientação (fluxo de rotor, de entreferro ou de estator) deve ter suas componentes d-q participando na composição do conjunto de variáveis de estado.
2) adoção de um referencial bifásico síncrono com eixos direto (d^e) e em quadratura (q^e) mutuamente desacoplados;
3) aplicação de relações de transformação para obter a representação das variáveis da máquina neste referencial;
4) orientação do referencial síncrono de forma que seu eixo direto d^e coincida com o vetor de fluxo escolhido no processo de orientação;

De acordo com a condição 4, para possibilitar a orientação do referencial síncrono, é preciso definir métodos de determinação da posição do fluxo de forma a viabilizar o desacoplamento entre as variáveis de controle.

5.3.2.2. Métodos de determinação da posição do fluxo

Há essencialmente três métodos gerais de controle baseados no princípio da orientação de fluxo. Estes métodos diferem basicamente na forma de obtenção da posição do fluxo, a qual vai definir a orientação do sistema de coordenadas de referência. Quando a posição do fluxo é determinada a partir da velocidade de rotação do eixo da máquina e da estimativa do escorregamento a denominação dada é de método indireto, que foi desenvolvido por K. Hasse em 1969 [28]. Quando a posição do fluxo é medida através de sensores, a denominação dada é de método direto, desenvolvido por F. Blaschke em 1972 [10]. Quando a posição do fluxo é obtida através de um observador ou estimador de fluxo, a denominação dada é de método por realimentação de estado (Verghese&Sanders-88)[67]; (Bellini,Figalli&Ulivi-88)[6].

5.3.2.2.1. O método direto

O método direto obtém a posição instantânea do vetor de fluxo através da inclusão no estator da máquina de sensores de efeito "hall" ou bobinas exploratórias. Como apresentado na figura 5.4, a posição espacial do vetor de fluxo é representada por um vetor unitário definido a partir das componentes de fluxo de entreferro ψ^s_{dm} e ψ^s_{qm}, na forma:

$$\cos(w_e t) = \frac{\psi^s_{dm}}{\sqrt{\psi^{s\,2}_{dm} + \psi^{s\,2}_{qm}}} \qquad \text{sen}(w_e t) = \frac{\psi^s_{qm}}{\sqrt{\psi^{s\,2}_{dm} + \psi^{s\,2}_{qm}}}$$

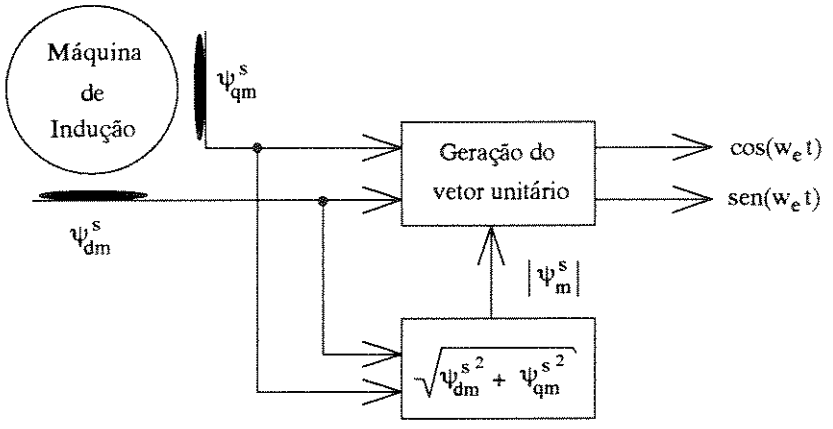


Figura 5.4 - Método direto de determinação da posição do fluxo

As componentes do vetor fluxo de entreferro podem ser obtidas diretamente, como ilustrado na figura 5.4, ou então calculadas através de sinais de tensão e corrente de estator. Caso necessário, o cálculo das componentes de fluxo de rotor a partir das componentes de fluxo de entreferro podem ser obtidas como o auxílio da equação (5.5), na forma:

$$\psi^s_{qr} = \frac{L_r}{L_m}\psi^s_{qm} - L_{lr} i^s_{qs} \tag{5.35}$$

$$\psi^s_{dr} = \frac{L_r}{L_m}\psi^s_{dm} - L_{lr} i^s_{ds} \tag{5.36}$$

A colocação dos sensores requer a abertura da máquina e, portanto, esta acaba perdendo a sua configuração padrão. Este procedimento invariavelmente ocasiona perda de robustez e simplicidade da máquina de indução. Além disso, devido à dificuldade de sensoriamento adequado do fluxo em baixas velocidades, este método não pode ser aplicado em condições de velocidade abaixo de 10% da velocidade nominal (Bose-86)[11].

5.3.2.2.2. O método indireto

O método indireto obtém a posição instantânea do fluxo com base no modelo da dinâmica inversa da máquina de indução. A posição espacial do fluxo de rotor (θ_e) é obtida a partir da velocidade do rotor (w_r) e da estimativa do escorregamento ($\hat{w}_{slip}$), na forma:

$$\theta_e = \int_{t_0}^{t} (w_r + \hat{w}_{slip})dt \tag{5.37}$$

O esquema da figura 5.5 apresenta o princípio de operação do sistema de controle[10]. Dados os comandos desejados de fluxo (ψ^*_r) e conjugado eletromagnético (T_e^*) e a medida da velocidade de rotação do eixo da máquina (w_r), o objetivo do bloco de controle é determinar a nova posição do referencial síncrono (θ_e) (alinhado com o vetor fluxo de rotor) e as componentes de corrente i^{e*}_{qs} e i^{e*}_{ds} (calculadas com base na hipótese de alinhamento). Depois de serem convertidas em correntes de fase i^s_a, i^s_b e i^s_c pelo bloco de inversão[11], as correntes são utilizadas para alimentar a máquina de forma a conduzir o fluxo e o conjugado aos valores comandados.

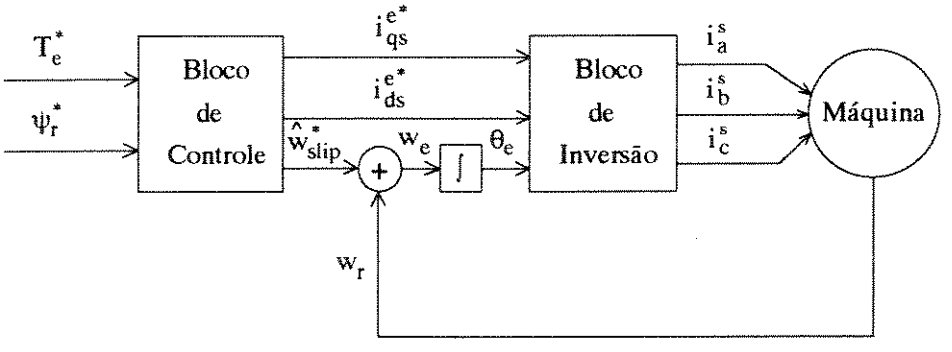


Figura 5.5 - Princípio de operação do método indireto de controle vetorial

[10]As variáveis de controle são identificadas com um asterisco.

[11]O bloco de inversão realiza a seqüência de transformação inversa apresentada na figura 5.3 e utiliza o inversor PWM apresentado na figura 5.2.

A velocidade w_r do rotor pode ser obtida com a inclusão de um sensor de freqüência no eixo da máquina. A estimativa do escorregamento é obtida a partir das equações do modelo dinâmico desacoplado da máquina, ou seja, o modelo obtido tomando-se como referência um referencial bifásico d^e-q^e girando à velocidade síncrona, com o eixo d^e posicionado na direção do vetor fluxo de rotor. Se o escorregamento for estimado de acordo com a equação (5.32), o alinhamento entre vetor fluxo de rotor e eixo d^e está garantido, pois:

$$\hat{w}_{slip} = \frac{L_m R_r}{L_r} \frac{i^e_{qs}}{\psi^e_{dr}} \tag{5.38}$$

e da equação (5.22):

$$\frac{d\psi^e_{qr}}{dt} = -\frac{R_r}{L_r}\psi^e_{qr} - w_{slip}\psi^e_{dr} + \frac{L_m R_r}{L_r} i^e_{qs} \tag{5.39}$$

Substituindo a equação (5.38) na equação (5.39), obtém-se:

$$\frac{d\psi^e_{qr}}{dt} = -\frac{R_r}{L_r}\psi^e_{qr} = -\frac{1}{\tau_r}\psi^e_{qr} \tag{5.40}$$

ou seja, mesmo que se tenha inicialmente $\psi^e_{qr}>0$ (ausência de alinhamento), da equação (5.40) pode-se concluir que ψ^e_{qr} vai para zero em um tempo determinado pela constante de tempo do rotor $\tau_r=L_r/R_r$.

A partir das informações de entrada (T_e^* e ψ^*_r), a geração dos comandos de corrente para alimentação da máquina (i^{e*}_{qs} e i^{e*}_{ds}) deve ser baseada na aplicação das equações do modelo d^e-q^e da máquina para controle desacoplado (equações (5.32) e (5.33)). Portanto, substituindo os valores de comando de fluxo e conjugado nas equações (5.32) e (5.33), é possível extrair as componentes de corrente na forma:

$$i^{e*}_{qs} = \frac{2}{3}\frac{2}{P}\frac{L_r}{L_m}\frac{T^*_e}{\psi^*_r} \tag{5.41}$$

$$i^{e*}_{ds} = \frac{1}{L_m}\left[\frac{L_r}{R_r}\frac{d\psi^*_r}{dt} + \psi^*_r\right] \tag{5.42}$$

O cálculo de w_{slip} em termos dos comandos de fluxo e conjugado pode ser obtido das equações (5.38) e (5.41) na forma:

$$\hat{w}_{slip} = \frac{L_m R_r}{L_r}\frac{i^{e*}_{qs}}{\psi^*_r} = \frac{2}{3}\frac{2}{P} R_r \frac{T^*_e}{\left(\psi^*_r\right)^2} \tag{5.43}$$

A figura 5.6 apresenta o diagrama de blocos correspondente à geração dos comandos de controle e de posição do sistema de referência síncrono a partir das referências de fluxo, conjugado e velocidade de rotação do eixo da máquina, tomando como base as equações (5.37), (5.41), (5.42) e (5.43).

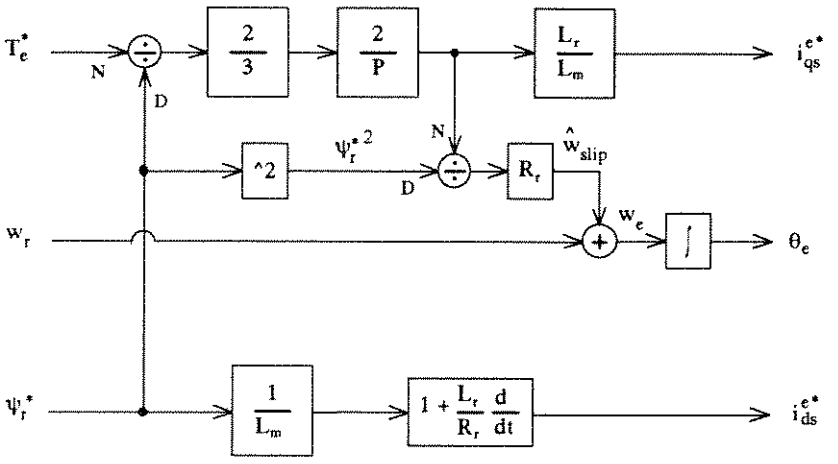


Figura 5.6 - Geração dos comandos de controle pela aplicação do método indireto de determinação da orientação de fluxo

O diagrama fasorial da figura 5.7 apresenta o princípio de operação do método indireto.

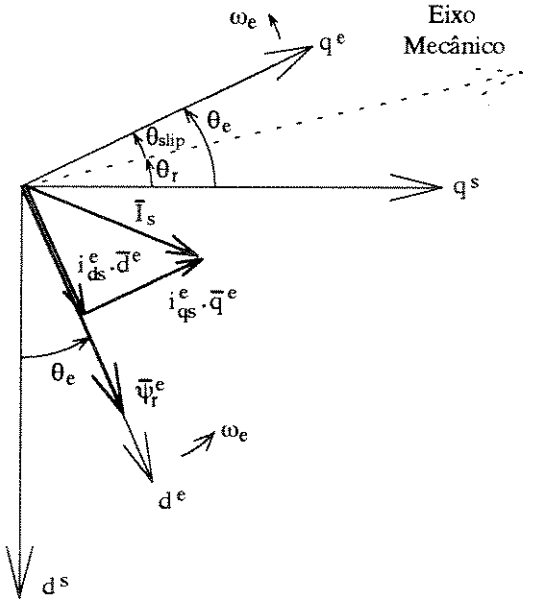


Figura 5.7 - Princípio de operação do método indireto

Os eixos d^s-q^s são fixos no estator enquanto que os eixos d^e-q^e giram à velocidade síncrona w_e. Em um determinado instante, o eixo q^e está a uma posição angular θ_e com relação ao eixo q^s. O ângulo θ_e é dado pela soma da posição angular do rotor $\theta_r=w_r t$

e do ângulo de escorregamento $\theta_{slip}=w_{slip}t$. Para efeito de controle desacoplado, o vetor fluxo de rotor deve estar alinhado com o eixo d^e, sendo que as componentes de corrente de estator são definidas com base neste alinhamento.

No método indireto de determinação da orientação de fluxo, ao contrário do método direto, a máquina mantém sua configuração original. Além disso, o método indireto pode ser aplicado em qualquer condição de velocidade. Outro fator interessante é a pronta aplicabilidade deste método vetorial a sistemas de controle já implementados via métodos escalares, sem exigir maiores investimentos ou modificações (Lima-93)[41].

5.3.2.2.3. O método por realimentação de estado

O método por realimentação de estado utiliza técnicas de controle moderno (veja Capítulo 3) na determinação da orientação do sistema de referência. As componentes de fluxo são definidas como variáveis de estado da máquina e são realimentadas para a entrada do controlador. Quando as componentes de fluxo não puderem ser medidas diretamente, procedimentos de estimação são empregados. Estes procedimentos de estimação geralmente utilizam representações por espaço de estados e são denominados observadores de estado (Verghese&Sanders-88)[67]; (Bellini,Figalli&Ulivi-88)[6]; (Silvino-92)[62]. A técnica de controle a ser adotada no desenvolvimento que se segue está baseada nestes princípios básicos, mas não vai utilizar a representação convencional por espaço de estados na implementação do observador de fluxo. O princípio básico de operação é apresentado na figura 5.8, na forma de um diagrama de blocos.

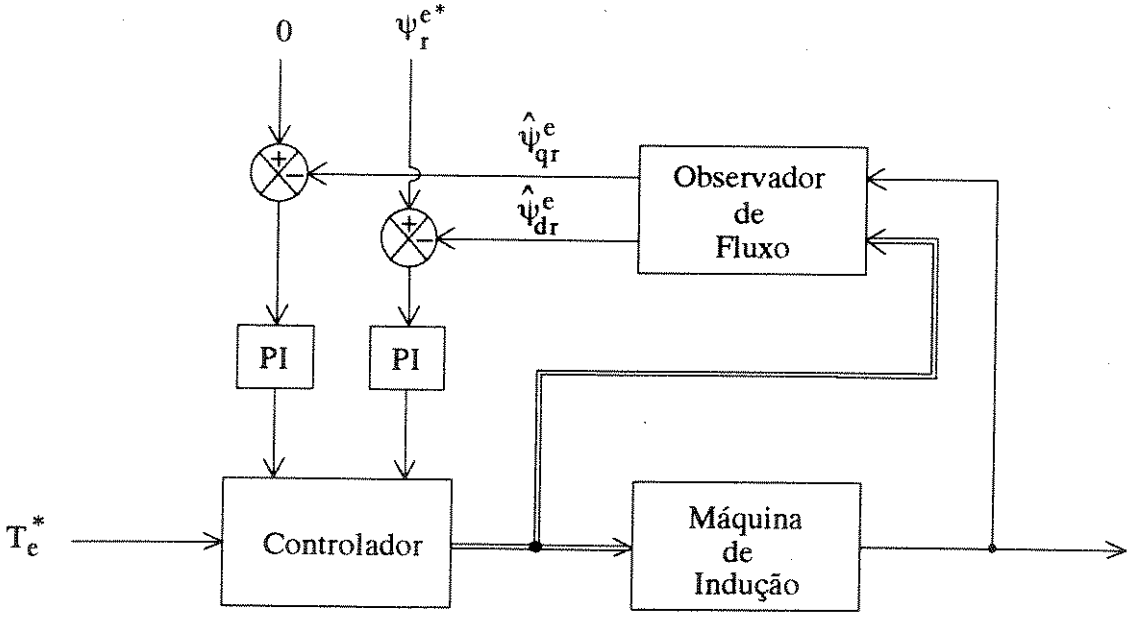


Figura 5.8 - Princípio de operação do método por realimentação de estado

Neste caso, o vetor de estados composto pelas componentes d-q do fluxo de rotor é estimado através de um observador de fluxo e, na seqüência, realimentado para a entrada do controlador. Um aspecto importante deste método é o fato de que a garantia de alinhamento entre o eixo direto e o vetor fluxo de rotor não está baseada no conhecimento da posição instantânea do vetor fluxo de rotor, como ocorre no caso dos métodos direto (posição obtida via medição direta) e indireto (posição obtida via estimação). Com a disponibilidade dos valores estimados $\hat{\psi}_{qr}^{e}$ e $\hat{\psi}_{dr}^{e}$, um procedimento equivalente pode ser utilizado, ou seja, a adoção de valores de referência para as componentes d-q do fluxo de rotor na forma:

$$\psi_{qr}^{e*} = 0 \text{ e } \psi_{dr}^{e*} = \psi_{r}^{e*}$$

Se as componentes de fluxo seguirem os respectivos valores de referência a todo instante, o alinhamento entre o eixo direto e o vetor fluxo de rotor está garantido. Para tanto, os sinais de erro produzidos a partir da comparação entre os valores de referência e os respectivos valores estimados formam a entrada para um par de controladores do tipo proporcional-integral (PI). Os sinais de saída destes controladores são utilizados juntamente com o sinal de comando de conjugado (T_e^*) na geração dos comandos de alimentação da máquina. Portanto, a função do controlador é providenciar o atendimento dos comandos de fluxo (ψ_r^{e*}) e conjugado (T_e^*) preservando, assim, a condição de alinhamento entre o eixo direto e o vetor fluxo de rotor.

As equações do controlador estão baseadas no modelo inverso da parte eletromagnética da máquina de indução, que pode ser obtido a partir do modelo direto (equação (5.21)) e da equação de conjugado eletromagnético (equação (5.10)), na forma:

$$i_{ds}^{e*} = \frac{L_r}{L_m R_r}\left[\frac{d\hat{\psi}_{dr}^{e}}{dt} - w_{slip}^{*}\hat{\psi}_{qr}^{e} + \frac{R_r}{L_r}\hat{\psi}_{dr}^{e}\right] \tag{5.44}$$

$$i_{qs}^{e*} = \frac{2}{3}\frac{2}{P}\frac{L_r}{L_m}\frac{T_e^*}{\hat{\psi}_{dr}^{e}} + \hat{\psi}_{qr}^{e} i_{ds}^{e*} \tag{5.45}$$

$$w_{slip}^{*} = \frac{1}{\hat{\psi}_{dr}^{e}}\left[-\frac{d\hat{\psi}_{qr}^{e}}{dt} - \frac{R_r}{L_r}\hat{\psi}_{qr}^{e} + \frac{L_m R_r}{L_r} i_{qs}^{e*}\right] \tag{5.46}$$

onde:

$$\frac{d\hat{\psi}_{qr}^{e}}{dt} \cong \frac{1}{T}\left[K_{Pq}\left(\psi_{qr}^{e*} - \hat{\psi}_{qr}^{e}\right) - K_{Iq}\int_0^t\left(\psi_{qr}^{e*} - \hat{\psi}_{qr}^{e}\right)dt\right]$$

$$\frac{d\hat{\psi}_{dr}^{e}}{dt} \cong \frac{1}{T}\left[K_{Pd}\left(\psi_{dr}^{e*} - \hat{\psi}_{dr}^{e}\right) - K_{Id}\int_0^t\left(\psi_{dr}^{e*} - \hat{\psi}_{dr}^{e}\right)dt\right]$$

com T representando o período de amostragem e K_P e K_I representando respectivamente as constantes de ganho proporcional e integral. É certo que, na definição das aproximações para $d\hat{\psi}^e_{qr}/dt$ e $d\hat{\psi}^e_{dr}/dt$ poderia ter sido utilizado o modelo discreto dado pela equação (5.24), mas uma significativa redução de cálculos é obtida com a utilização dos controladores PI.

O desempenho do método de determinação da posição do fluxo por realimentação de estado é dependente da eficiência do observador de fluxo. Utilizando observadores adaptativos baseados na representação por espaço de estados, é possível definir ganhos de realimentação que conduzam os erros de estimação assintoticamente para zero (Verghese&Sanders-88)[67]; (Bellini,Figalli&Ulivi-88)[6]; (Silvino-92)[62]. Os observadores desenvolvidos utilizando-se a representação por espaço de estados apresentam uma série de limitações quando aplicados a problemas de controle de sistemas não-lineares e não-estacionários, como é o caso da máquina de indução:

- são utilizadas aproximações lineares para representar a dinâmica não-linear da máquina de indução.
- é necessário calcular o ganho de realimentação a todo instante de amostragem, pois a equação dinâmica resultante para a máquina é não-estacionária e a dinâmica do erro de estimação passa a depender da velocidade de rotação da máquina e da freqüência de escorregamento.
- devido à alta sensibilidade à variação de parâmetros, é necessário utilizar métodos de estimação de parâmetros que devem operar em paralelo com o observador de fluxo. Estes métodos geralmente dependem do conhecimento preciso de variáveis como a freqüência de escorregamento para permitir a obtenção de estimativas corretas (Silvino-92)[62].

Com o objetivo de ampliar o potencial de estimação para o observador de fluxo, visando minimizar o efeito das limitações descritas acima e permitir a aplicação de técnicas alternativas de definição de ganho de realimentação e estimação de parâmetros, é proposto, a seguir, um observador de fluxo baseado em redes neurais recorrentes.

5.3.2.2.3.1. Observador de fluxo utilizando redes neurais recorrentes

No Capítulo 4 foi desenvolvida uma rede neural recorrente adequada para representação de sistemas dinâmicos descritos na forma de equações de estado. A intenção é treinar uma rede neural para identificar o comportamento dinâmico de tempo discreto das componentes de fluxo de rotor, dado pela equação (5.24) (parte eletromagnética), reproduzida a seguir[12]:

[12] A unidade de tempo adotada é o segundo e o período de amostragem é T=0,001 segundos.

$$\begin{bmatrix} \psi_{qr}^{e}(k+1) \\ \psi_{dr}^{e}(k+1) \end{bmatrix} = G(T)\begin{bmatrix} \psi_{qr}^{e}(k) \\ \psi_{dr}^{e}(k) \end{bmatrix} + H(T)\begin{bmatrix} i_{qs}^{e}(k) \\ i_{ds}^{e}(k) \end{bmatrix} \tag{5.47}$$

onde:

$$G(T) = \begin{bmatrix} e^{-T\frac{R_r}{L_r}}\cos\left[w_{slip}(nT).T\right] & -e^{-T\frac{R_r}{L_r}}\text{sen}\left[w_{slip}(nT).T\right] \\ e^{-T\frac{R_r}{L_r}}\text{sen}\left[w_{slip}(nT).T\right] & e^{-T\frac{R_r}{L_r}}\cos\left[w_{slip}(nT).T\right] \end{bmatrix}$$

$$H(T) = \begin{bmatrix} h_1 & h_2 \\ -h_2 & h_1 \end{bmatrix}$$

com

$$h_1 = \frac{R_r L_m L_r}{w_{slip}^2 L_r^2 + R_r^2}\left[e^{-T\frac{R_r}{L_r}}\left(w_{slip}\text{sen}(w_{slip}t) - \frac{R_r}{L_r}\cos(w_{slip}T)\right) + \frac{R_r}{L_r}\right]$$

$$h_2 = -\frac{R_r L_m L_r}{w_{slip}^2 L_r^2 + R_r^2}\left[e^{-T\frac{R_r}{L_r}}\left(w_{slip}\cos(w_{slip}T) + \frac{R_r}{L_r}\text{sen}(w_{slip}T)\right) + w_{slip}\right]$$

Da equação (5.47), considerando-se parâmetros constantes, pode-se concluir que as componentes de fluxo no instante (k+1) são obtidas a partir de relações algébricas entre as seguintes variáveis:

- componentes de fluxo que formam o vetor de estados no instante k;
- componentes de corrente de estator que formam o vetor de entrada no instante k;
- velocidade de escorregamento no instante k (responsável pela não-estacionariedade da matriz de estado G(T), assumindo-se parâmetros constantes).

O vetor de saída da rede neural recorrente é dado pelos valores estimados para as componentes de fluxo de rotor, que, neste caso, correspondem ao vetor de estados. Esta equivalência entre vetor de saída e vetor de estados dispensa a representação da equação de saída na estrutura da rede neural recorrente desenvolvida no Capítulo 4. Portanto a rede neural resultante assume a forma apresentada na figura 5.9.

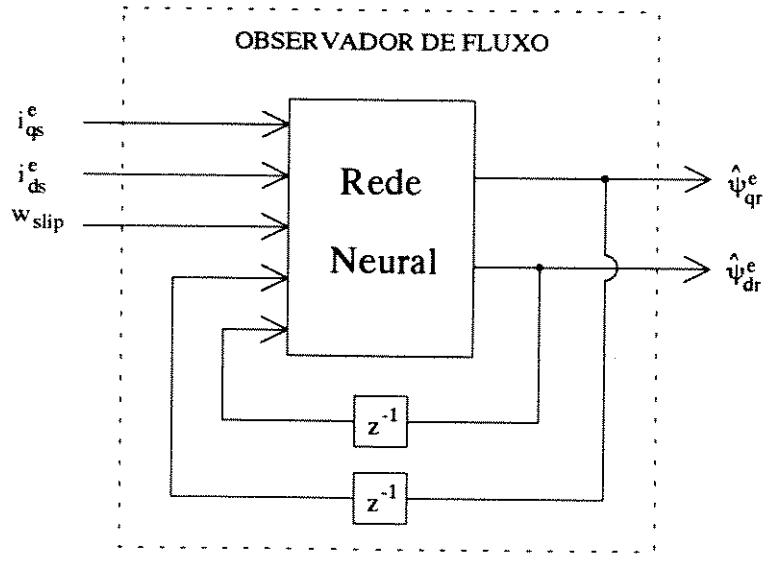


Figura 5.9 - Rede neural recorrente utilizada como observador de fluxo

As principais propriedades de redes neurais recorrentes a serem exploradas neste processo de identificação são as seguintes:

- capacidade de representação de comportamentos dinâmicos discretos;
- capacidade de representação de sistemas multivariáveis;
- capacidade de representação de sistemas lineares e não-lineares;
- capacidade de adaptação a variações de comportamento dinâmico.

Um dos motivos da não utilização do próprio modelo dinâmico como observador de fluxo é dado pelas limitações imposta por representações por espaço de estados na forma da equação (5.47), já discutidas na seção anterior. Além disso, com a aplicação de técnicas de treinamento "on-line", dinâmicas não-modeladas e modificações de comportamento podem ser identificadas pela rede neural em tempo de operação, ampliando a capacidade de adaptação do observador de fluxo.

A geração dos padrões para treinamento da rede neural a partir da equação (5.47) está baseado na seguinte seqüência de procedimentos:

- assumindo referência de fluxo constante e nominal (veja Apêndice E), os valores de referência para as componentes d-q de fluxo são dados por:

$$\psi^{e*}_{qr} = 0 \text{ e } \psi^{e*}_{dr} = 0{,}2990 \text{ Wb}$$

- como é desejado o acompanhamento dessas referências, pode-se assumir que a região de operação está restrita a um pequeno intervalo em torno desses valores de referência;
- sendo assim, é suficiente treinar a rede neural para identificar o comportamento dinâmico da máquina, ditado pela equação (5.47), apenas em torno da condição de alinhamento entre o eixo direto do sistema d-q de referência e o vetor fluxo de rotor.

A figura 5.10, a seguir, apresenta os padrões utilizados para treinamento da rede neural, seguindo critérios de implementação descritos detalhadamente no Capítulo 4, como, por exemplo, a normalização dos padrões no intervalo [-1,1]. Os parâmetros de treinamento são apresentados no Apêndice F, seção 3.

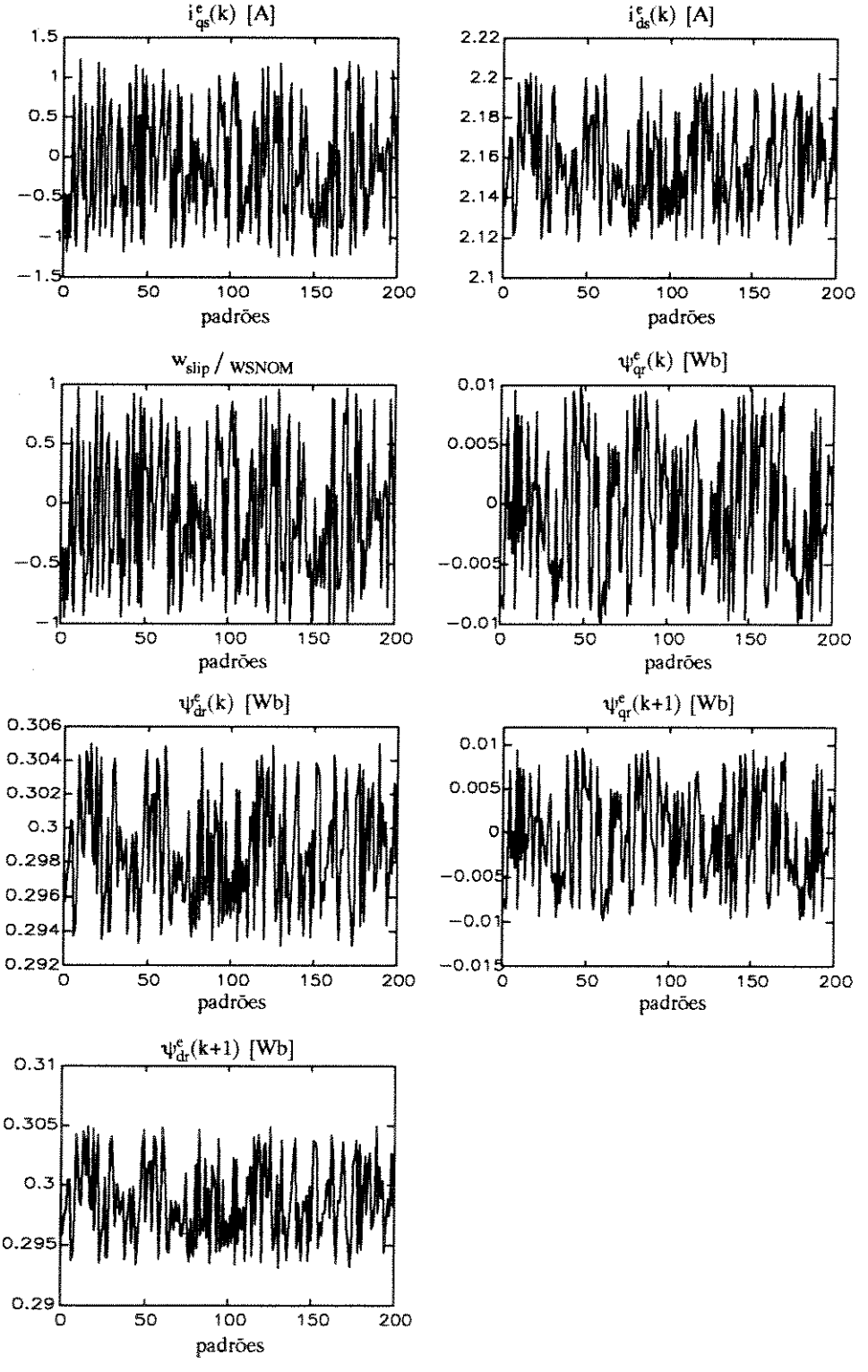


Figura 5.10 - Padrões (não normalizados) para treinamento da rede neural da figura 5.9

Após o treinamento, o erro existente entre as componentes de fluxo $\hat{\psi}^e_{qr}$ e $\hat{\psi}^e_{dr}$ estimadas pela rede neural e as componentes ψ^e_{qr} e ψ^e_{dr} utilizadas como padrões de treinamento é apresentado na figura 5.11.

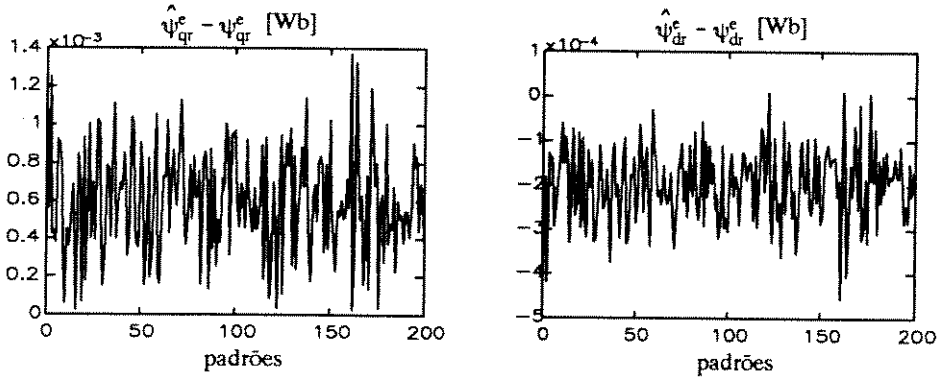


Figura 5.11 - Erro após a execução do treinamento

A seguir, são apresentados resultados de simulação onde é possível comparar as componentes de fluxo estimadas pelo observador de fluxo e as correspondentes componentes de fluxo da máquina de indução em três situações caracterizadas pela operação da máquina na condição de alinhamento entre o eixo direto do sistema d-q de referência e o vetor fluxo de rotor.

Conforme apresentado na figura 5.12, o comando de fluxo é mantido constante no valor nominal em todas as simulações e são utilizados três perfis diferentes de conjugado, um em cada simulação.

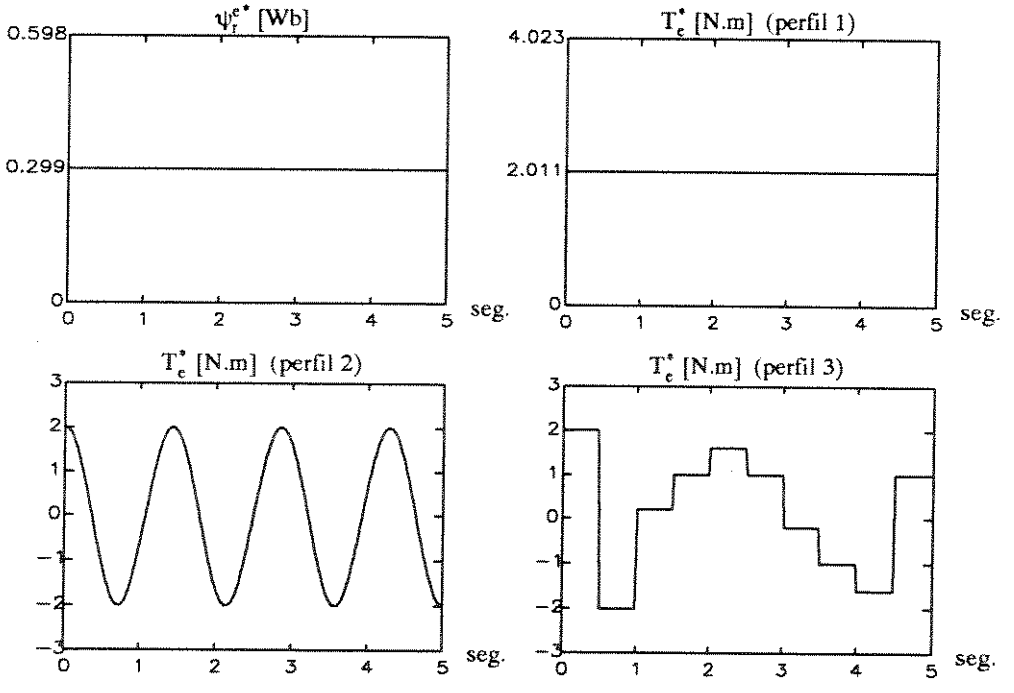


Figura 5.12 - Perfis de fluxo e conjugado utilizados nas simulações

Os perfis de comando de conjugado apresentados na figura 5.12 foram escolhidos com as seguintes peculiaridades:

- perfil 1 - comando de conjugado constante no valor nominal;
- perfil 2 - comando de conjugado do tipo senoidal abrangendo todo o intervalo entre aceleração e frenagem com conjugado nominal;
- perfil 3 - comando de conjugado do tipo onda quadrada (variação abrupta) abrangendo alguns pontos do intervalo entre aceleração e frenagem com conjugado nominal;

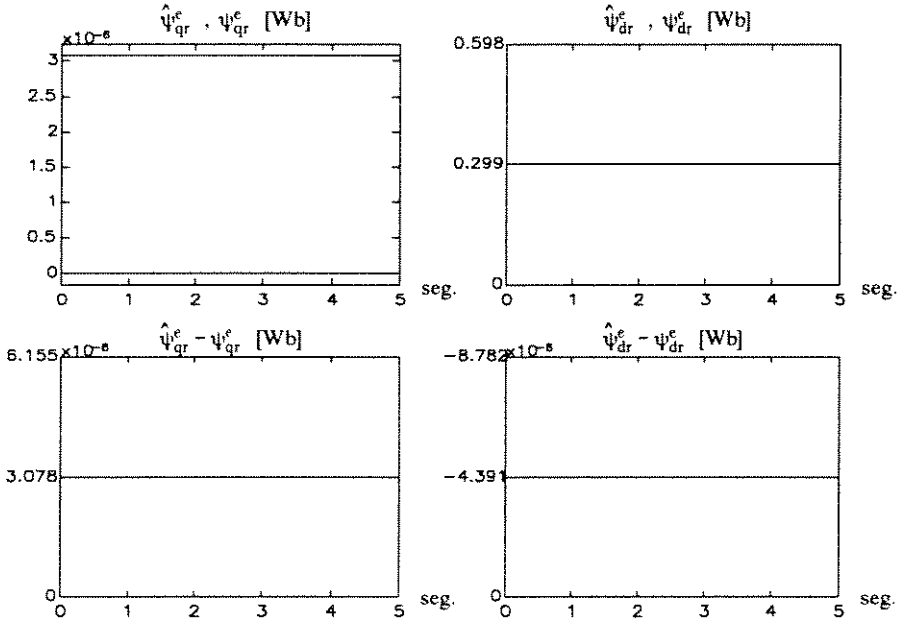


Figura 5.13 - Simulação utilizando o comando de conjugado constante (perfil 1)

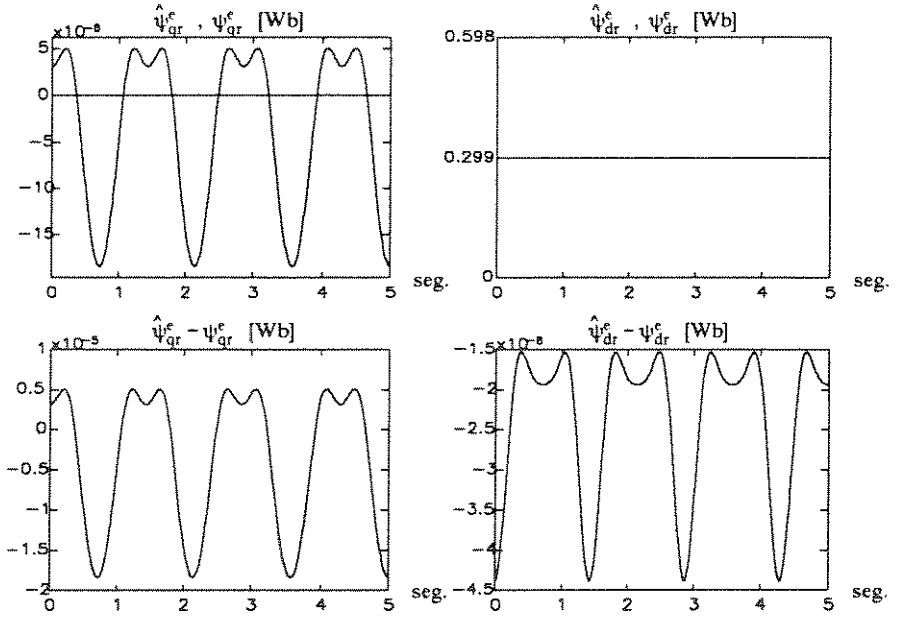


Figura 5.14 - Simulação utilizando o comando de conjugado senoidal (perfil 2)

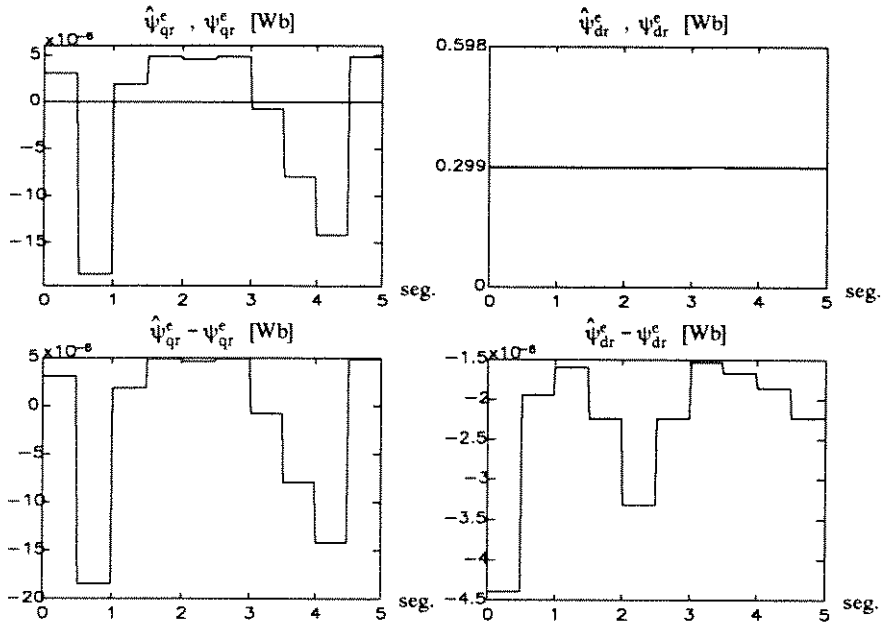


Figura 5.15 - Simulação utilizando o comando de conjugado do tipo onda quadrada (perfil 3)

Com base nos resultados de simulação apresentados nas figuras 5.13, 5.14 e 5.15, pode-se concluir que, independente do comando de conjugado e quando a condição de alinhamento é garantida, o desempenho do observador de fluxo baseado em redes neurais (figura 5.9) é muito bom, pois mesmo com variações acentuadas no ponto de operação da máquina, o processo de estimação apresenta um erro máximo da ordem de 10^{-6}.

5.3.3. Resultados de simulação do processo de controle

Nesta seção, o comportamento dinâmico do controle vetorial por orientação do fluxo de rotor baseado nos métodos indireto e por realimentação de estado atuando sobre uma máquina de indução trifásica é analisado através de exemplos de simulação computacional[13]. A equação (5.24) é usada para representar o modelo discreto da máquina de indução, cujos parâmetros são apresentados no Apêndice E.

O procedimento de simulação segue o seguinte critério:

- Os parâmetros da máquina de indução são considerados invariantes no tempo.
- O inversor fonte de corrente é considerado ideal.
- De posse dos seguintes dados:
  - perfil temporal desejado para o fluxo de rotor (ψ^*_r);
  - perfil temporal desejado para o conjugado eletromagnético (T_e^*);
  - velocidade de rotação do eixo da máquina (w_r) obtida via sensores de freqüência;

[13] Nas simulações a serem apresentadas, é assumido que as componentes de corrente comandadas são efetivamente utilizadas para alimentar a máquina, ou seja, que o inversor fonte de corrente é ideal.

- Gerar os comandos de controle de forma que a máquina apresente um comportamento de fluxo e conjugado de acordo com o perfil desejado, utilizando:
  - método indireto: esquema de controle da figura 5.5 e as equações (5.41), (5.42) e (5.43);
  - método por realimentação de estado: esquema de controle da figura 5.8, observador de fluxo da figura 5.9 e as equações (5.44), (5.45) e (5.46);

5.3.3.1. Simulações utilizando o método indireto

Um primeiro resultado de importância prática está relacionado a um procedimento comumente adotado no tratamento de máquinas de corrente contínua: a pré-excitação. A pré-excitação consiste no estabelecimento do fluxo nominal antes de exigir que a máquina produza conjugado eletromagnético. A adoção deste procedimento nas simulações com máquina de indução se justifica pelo fato de que o método de controle a ser empregado utiliza um modelo para a máquina de indução totalmente análogo aos modelos para máquinas de corrente contínua. De fato, resultados experimentais obtidos por Lima (93)[41] comprovam que a não-observância deste procedimento acarreta dificuldades na partida da máquina, conduzindo a picos de corrente muito elevados, capazes de sobrecarregar a rede de alimentação ou mesmo danificar a máquina de forma irreversível. A simulação computacional da partida da máquina sem pré-excitação permite detectar este fenômeno, já que as correntes comandadas assumem valores infinitos (condição de operação ideal).

Além do problema de instabilidade de partida, a observação dos resultados de simulação da figura 5.16 permite concluir que a única forma de garantir que a componente de eixo em quadratura do vetor fluxo de rotor seja identicamente zero (indicando a existência de alinhamento entre o eixo direto do referencial síncrono e o vetor de fluxo), inclusive em condições de partida, é observar o princípio da pré-excitação. Partindo de um valor inicial de fluxo igual a ψ_r=0.05 Wb de forma a evitar a instabilidade de partida, sem a pré-excitação (SP) a máquina não mantém o alinhamento enquanto o fluxo não atinge seu valor nominal, conduzindo a uma diferença entre o conjugado eletromagnético comandado (T^*_e) e o conjugado produzido na máquina (T_e).

Esta diferença, mesmo quando pequena e mantida durante um intervalo de tempo reduzido, pode ser inaceitável, por exemplo, em aplicações de controle de posição. Com a simples iniciativa de aguardar o estabelecimento do fluxo nominal para a solicitação de conjugado[14], ou seja, com pré-excitação (CP), o alinhamento é preservado e o conjugado produzido segue exatamente o valor comandado (condição de operação ideal).

---

[14] Nos exemplos de simulação o comando de fluxo é mantido constante e igual ao fluxo nominal da máquina.

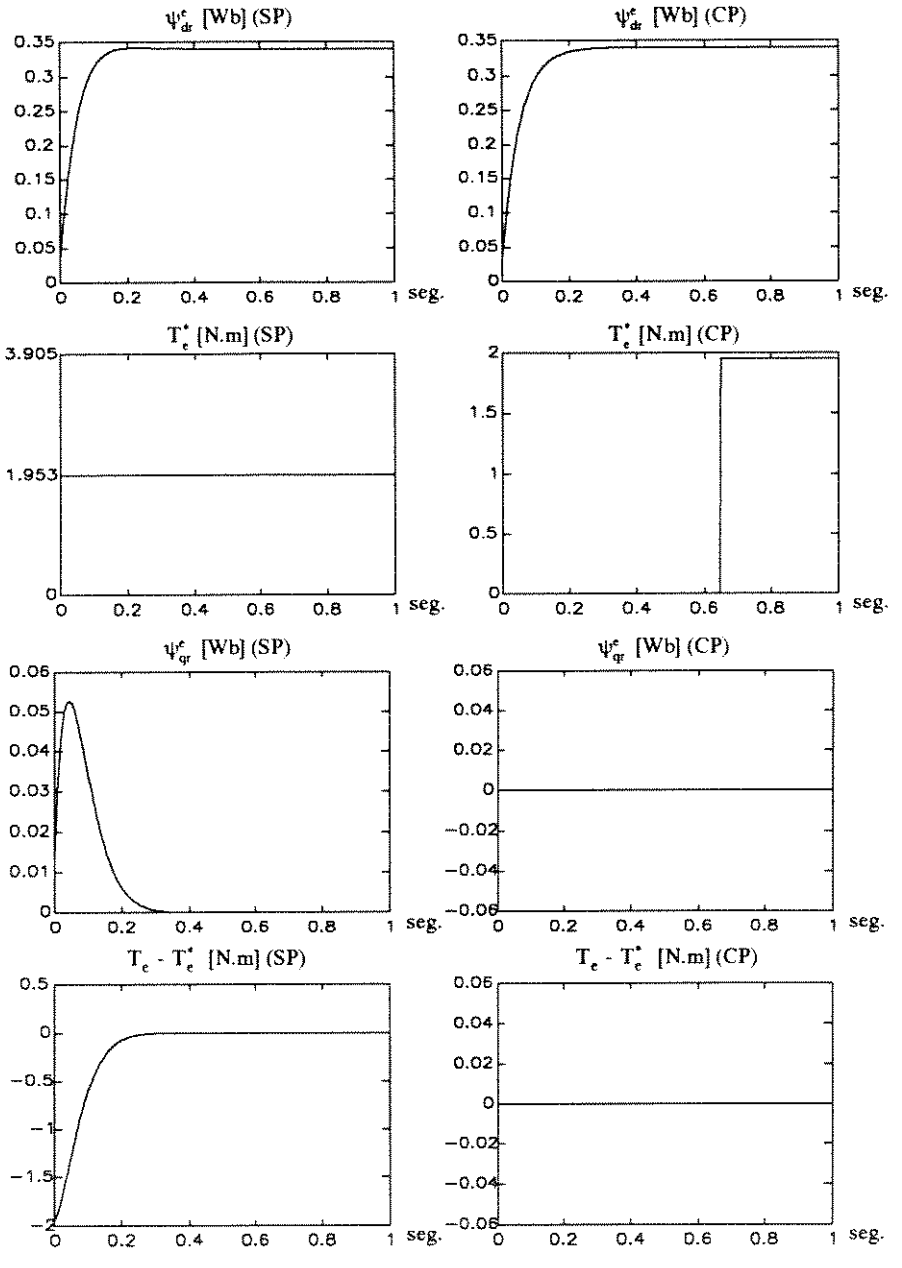


Figura 5.16 - O princípio da pré-excitação

Como um segundo exemplo de simulação, seja um perfil de conjugado do tipo onda quadrada, projetado de forma a exigir que a máquina opere em condições de aceleração, frenagem e inversão de rotação. O atrito viscoso é considerado desprezível (K=0) e a máquina é considerada operando em vazio (T_L=0).

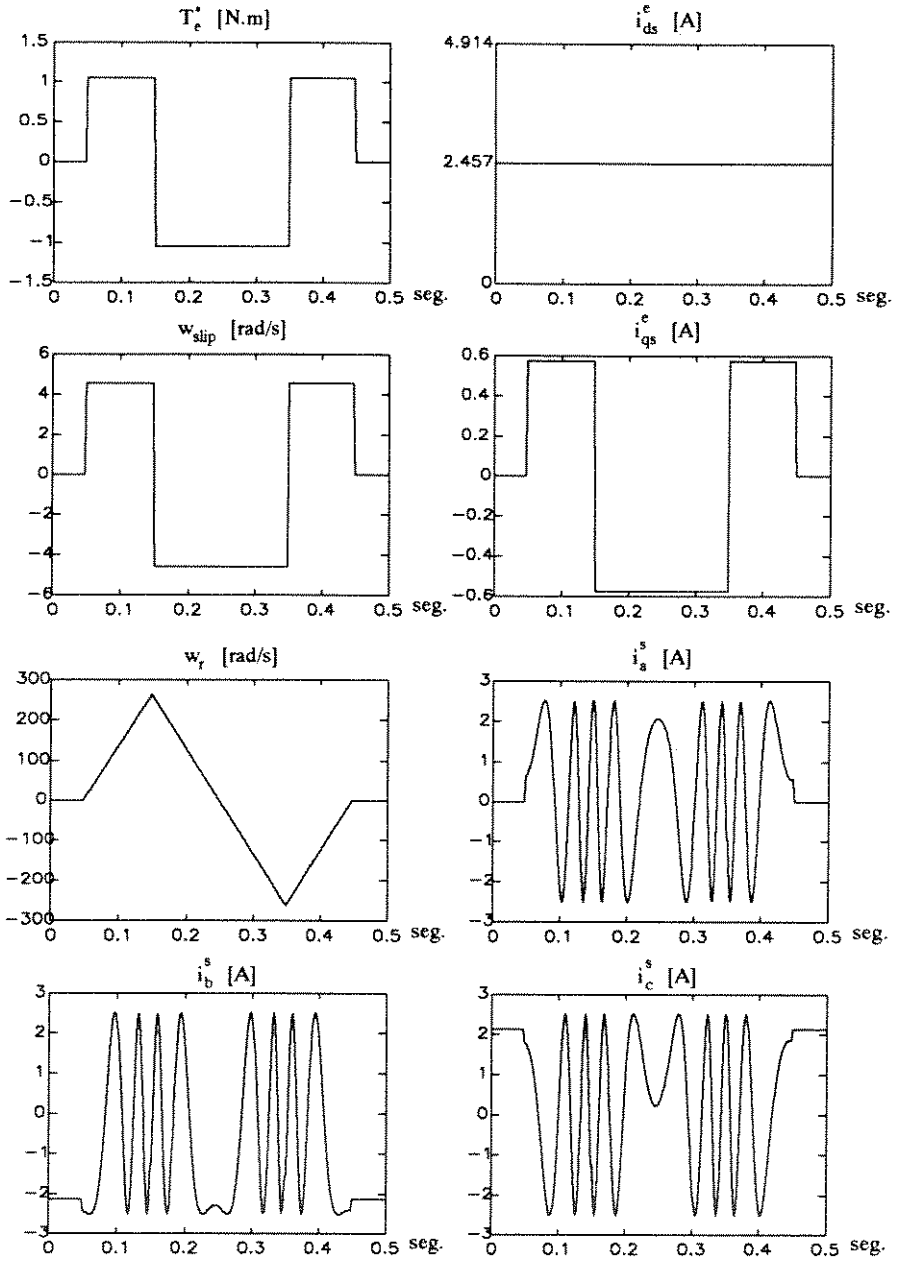


Figura 5.17 - Aceleração, frenagem e reversão de velocidade

Como o comando de fluxo ψ^*_r é mantido constante no valor nominal, na figura 5.17 pode-se observar que i^e_{ds} fica constante, enquanto que i^e_{qs} acompanha a variação do comando de conjugado T^*_e, demonstrando a existência de desacoplamento entre as variáveis de controle. Ainda na figura 5.17, é possível observar o comportamento da freqüência das correntes de fase nos intervalos de aceleração e frenagem. Quando o sentido de rotação é invertido, ocorre também uma inversão de fase das correntes. O comportamento de w_r indica a existência de uma correspondência direta entre conjugado eletromagnético e aceleração, conforme indicado pela equação (5.34), para K=0 e T_L=0.

Mantendo as mesmas condições de operação do exemplo anterior (ψ^*_r constante no valor nominal, K=0 e T_L=0), seja um terceiro exemplo de simulação, onde um perfil de conjugado é escolhido de forma a produzir um comportamento de aceleração e frenagem mais suave para a máquina. Variações suaves de velocidade são desejáveis em veículos elétricos de transporte, como por exemplo, locomotivas e elevadores. A curva de conjugado foi gerada por uma composição de funções do tipo tangente hiperbólica.

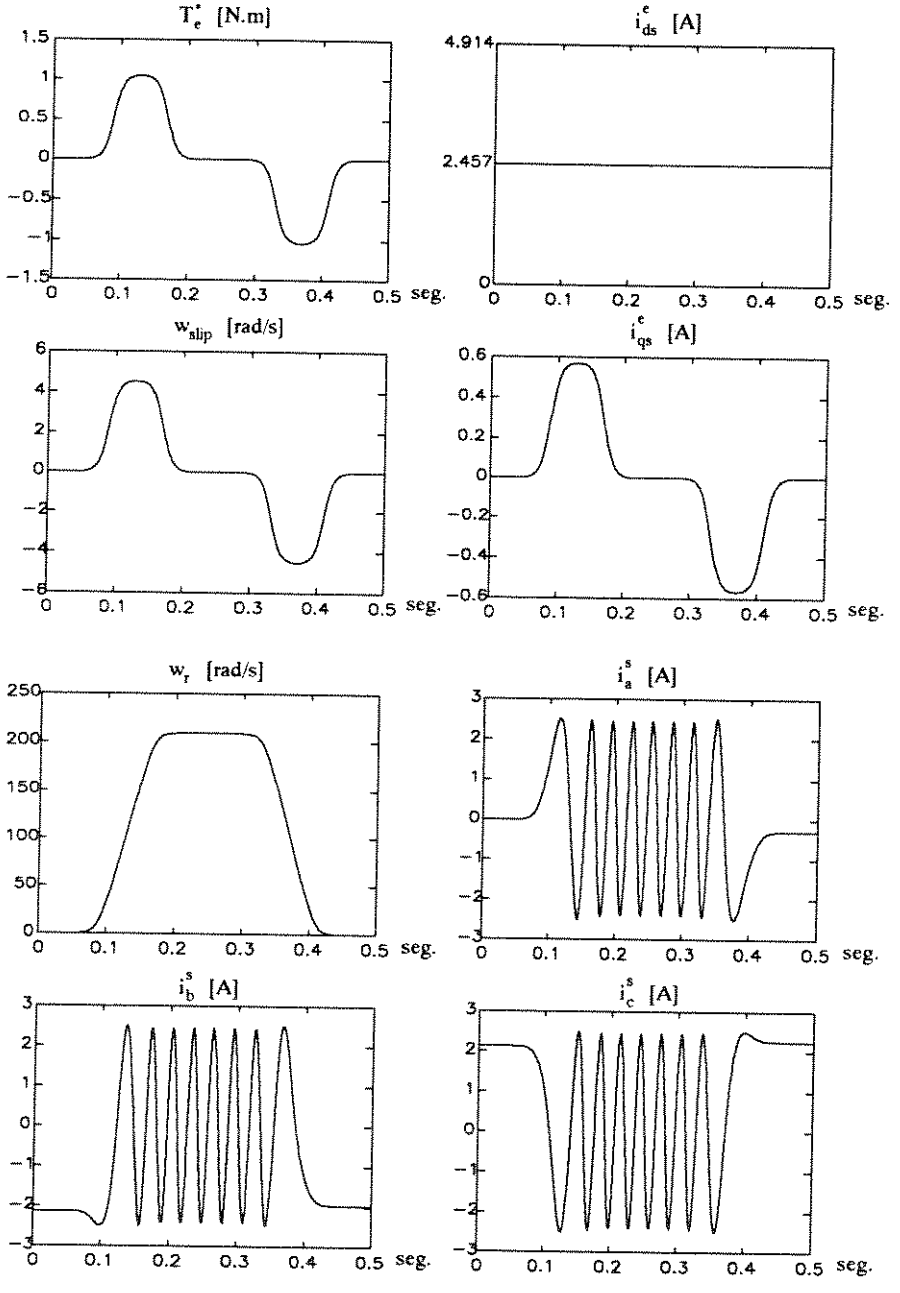


Figura 5.18 - Perfil de aceleração suave

5.3.3.2. Simulações utilizando o método por realimentação de estado

O desempenho do método de controle por realimentação de estado é totalmente vinculado ao desempenho do observador de fluxo. Os controladores PI são responsáveis por conduzir os fluxos estimados para os respectivos valores de referência. Caso os valores estimados consigam representar os valores reais de fluxo a todo instante, então os valores reais também vão ser conduzidos para os respectivos valores de referência. Caso, contrário, mesmo que os valores estimados acompanhem a referência, os valores reais podem se distanciar dos valores de referência, degradando o desempenho do controlador. Como apresentado nas figuras 5.13, 5.14 e 5.15, se a máquina operar próxima da condição de alinhamento, o observador de fluxo vai sempre apresentar um bom desempenho.

Mantendo a referência de fluxo no valor nominal, o primeiro exemplo de simulação está baseado na manutenção da referência de conjugado também no valor nominal. Como pode ser observado, os controladores PI[15] são responsáveis por conduzir os valores estimados de fluxo para as respectivas referências, sendo que o valor real das componentes de fluxo se estabilizam com um erro em relação ao valor estimado da mesma ordem de grandeza que o erro obtido na comparação com os padrões de treinamento (figura 5.11), demonstrando a capacidade de generalização apresentada pelas redes neurais. Um comportamento qualitativamente análogo é dado pela diferença entre o comando de conjugado (T_e^*) e o conjugado efetivamente produzido na máquina (T_e), conforme apresentado na figura 5.19.

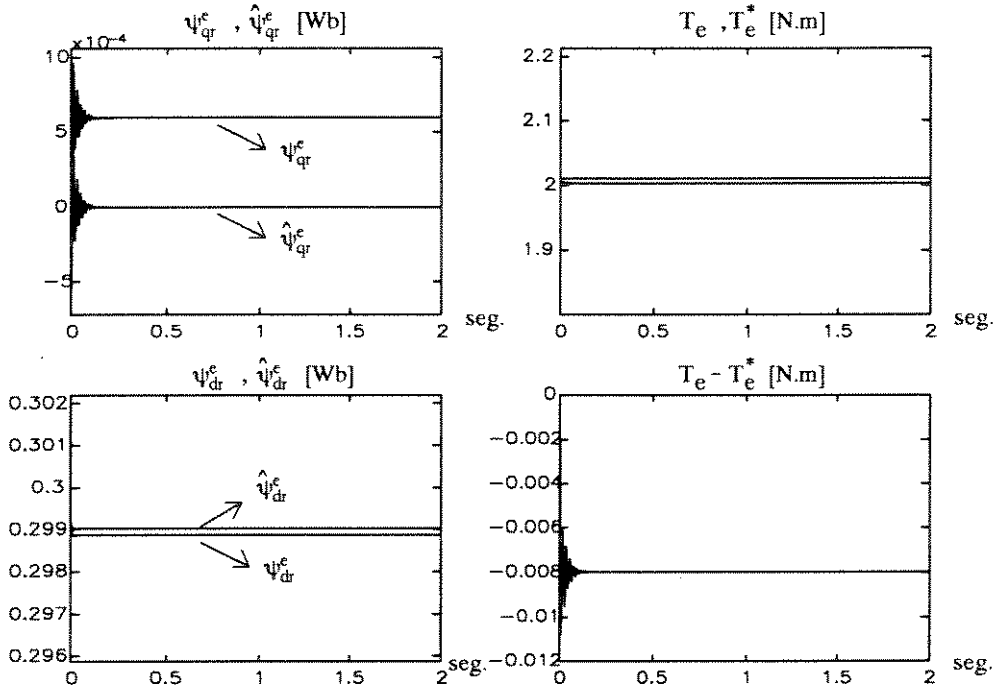


Figura 5.19 - Operação na condição de conjugado nominal

[15] $K_{Pq} = K_{Pd} = 1$, $K_{Iq} = K_{Id} = 2$

Passando, agora, para um perfil de conjugado do tipo senoidal, que percorre todo o intervalo entre aceleração e frenagem nominal, é possível comparar os valores reais e estimados de fluxo. Novamente o comportamento qualitativo do erro de estimação em relação às componentes de fluxo vai se repetir no erro existente entre os valores comandados e efetivos de conjugado, conforme apresentado na figura 5.20.

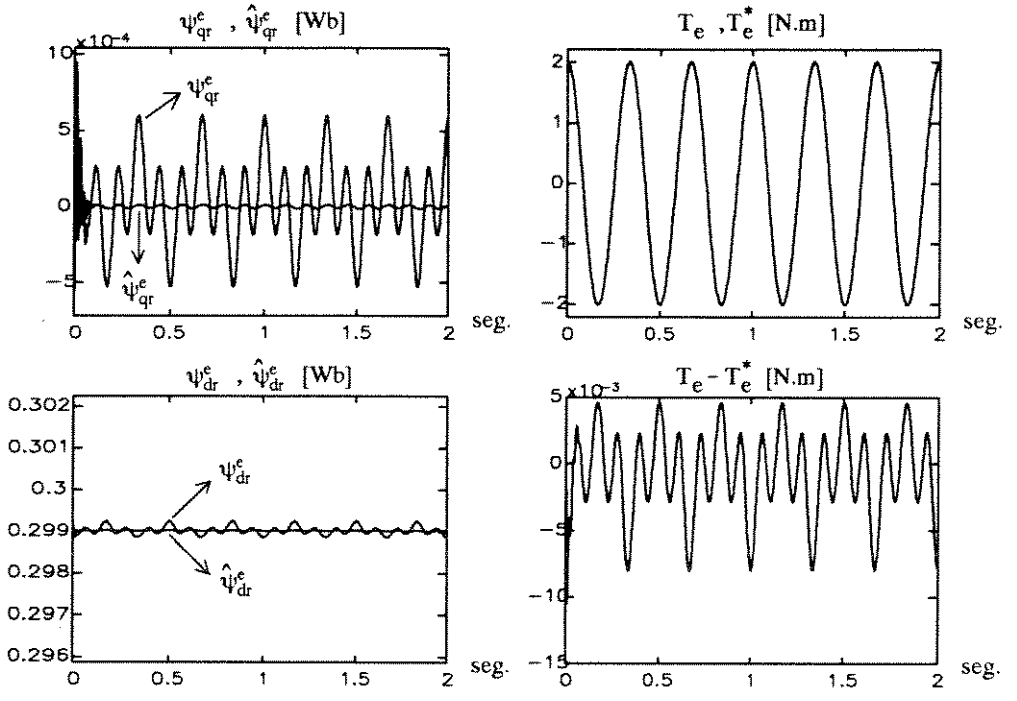


Figura 5.20 - Perfil senoidal explorando os limites máximos de variação de conjugado

Passando agora para um perfil de conjugado do tipo onda quadrada, que utiliza alguns pontos do intervalo entre aceleração e frenagem com conjugado nominal, é possível verificar que o erro máximo existente entre os valores reais e estimados de fluxo são os mesmos obtidos nas simulações cujos resultados são apresentados nas figuras 5.19 e 5.20, ou seja, independente do perfil de conjugado, é possível definir um erro de estimação máximo[16]. É importante verificar também que novamente o perfil de variação no fluxo vai se repetir na variação entre os valores comandados e efetivos de conjugado, conforme apresentado na figura 5.21.

[16] Uma conclusão análoga já havia sido obtida por ocasião da análise dos resultados de simulação apresentados nas figuras 5.13, 5.14 e 5.15.

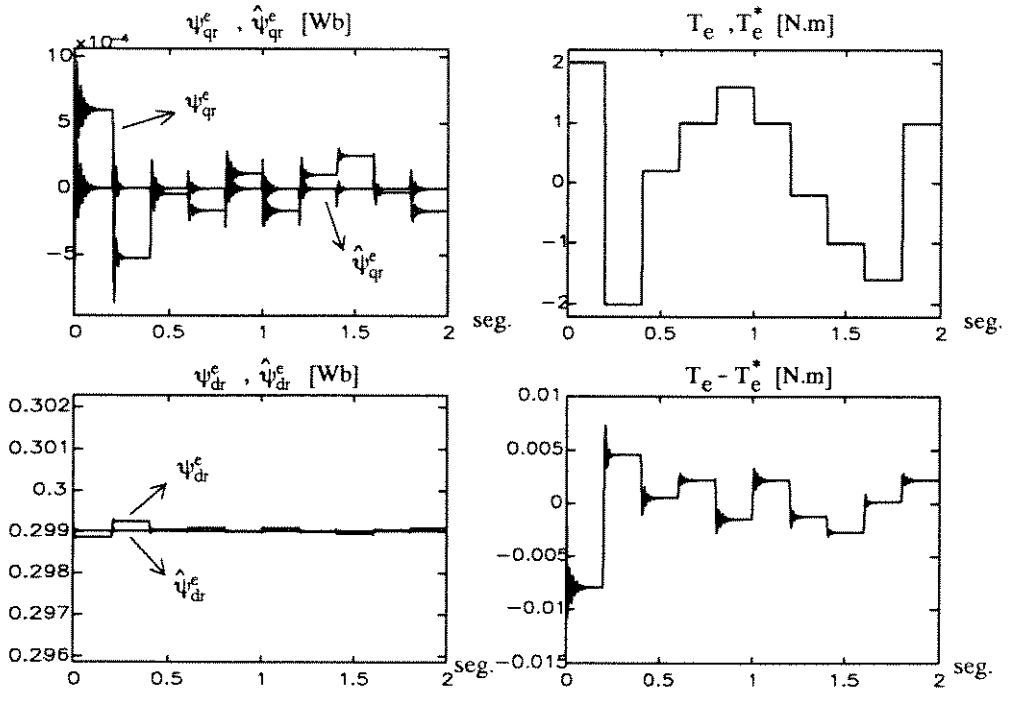


Figura 5.21 - Perfil do tipo onda quadrada explorando os limites máximos de variação de conjugado

Capítulo 6

Análise e solução do problema de sensibilidade à variação de parâmetros

A operação da máquina de indução em velocidade variável exige a manutenção da condição de desacoplamento entre as variáveis de controle que vão atuar sobre fluxo e conjugado eletromagnético. O desacoplamento é condição necessária para permitir a obtenção de respostas dinâmicas rápidas e bom desempenho em regime estacionário. Com a utilização de um sistema de referência d-q girando à velocidade síncrona, a garantia de alinhamento entre o eixo direto do sistema de referência e o vetor fluxo de rotor[1] é condição suficiente para garantir o desacoplamento.

Sendo conhecida a posição espacial do vetor fluxo de rotor, esta é utilizada no processo de orientação do sistema de referência, de forma a posicionar o eixo direto na direção do fluxo de rotor. No Capítulo 5, foram apresentados três métodos utilizados na determinação da posição espacial do vetor fluxo de rotor: método direto, método indireto e método por realimentação de estado. Também foram apresentados resultados de simulação computacional referentes à implementação dos dois últimos métodos.

Independente do método adotado, o sistema de controle resultante pode ser descrito de uma forma genérica de acordo com o esquema da figura 6.1.

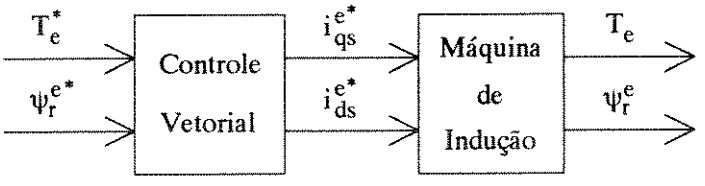


Figura 6.1 - Esquema simplificado do sistema de controle

Neste esquema fica evidenciado o fato de que o controlador realiza a dinâmica inversa da máquina, pois seus sinais de entrada correspondem aos sinais de saída da máquina e seus sinais de saída correspondem aos sinais de entrada da máquina. Portanto, o desempenho do controlador está diretamente vinculado à sua capacidade de representar a dinâmica inversa da máquina. Se o comportamento dinâmico da máquina é invariante com o tempo, o controlador, uma vez projetado, não sofrerá alterações em seu desempenho. Caso contrário, para evitar a degradação de desempenho, o controlador deve ser capaz de se ajustar às possíveis variações ocorridas na dinâmica da máquina.

[1] Neste caso, também são utilizados os vetores correspondentes ao fluxo de entreferro e fluxo de estator.

A representação do comportamento dinâmico de uma máquina de indução genérica em um sistema de referência d-q girando à velocidade síncrona foi desenvolvida no Capítulo 5 e é reproduzida a seguir, onde $\tau_r = (L_m + L_{lr})/R_r$ é a constante de tempo do rotor:

Parte eletromagnética

$$\begin{bmatrix} \frac{d\psi^e_{qr}}{dt} \\ \frac{d\psi^e_{dr}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_r} & -w_{slip} \\ w_{slip} & -\frac{1}{\tau_r} \end{bmatrix} \begin{bmatrix} \psi^e_{qr} \\ \psi^e_{dr} \end{bmatrix} + \frac{L_m}{\tau_r} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} i^e_{qs} \\ i^e_{ds} \end{bmatrix} \tag{6.1}$$

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_m + L_{lr}}\left(\bar{\psi}^e_r \times \bar{I}^e_s\right) = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_m + L_{lr}}[\psi^e_{dr} i^e_{qs} - \psi^e_{qr} i^e_{ds}] \tag{6.2}$$

Parte mecânica

$$\frac{dw_r}{dt} = -\frac{K}{J}w_r + \frac{P}{2J}(T_e - T_L) \tag{6.3}$$

A particularização das equações (6.1), (6.2) e (6.3) na representação do comportamento dinâmico de uma máquina de indução específica se dá pela definição dos valores dos parâmetros utilizados na representação, sendo que para cada máquina vai existir um conjunto específico de valores. Portanto, na determinação da dinâmica inversa da máquina de indução, o controlador deve utilizar um conjunto de parâmetros com valores idênticos àqueles dos parâmetros reais da máquina. Com isso, sempre que os parâmetros reais da máquina sofrerem alterações em seus valores, ou seja, sempre que o comportamento dinâmico da máquina sofrer alterações, os parâmetros correspondentes do controlador devem ser adequadamente atualizados, de forma a promover as alterações correspondentes na dinâmica inversa.

6.1. Estudo da variação de parâmetros

Quanto aos parâmetros utilizados nas equações (6.1), (6.2) e (6.3), é possível classificá-los em dois grupos distintos:

Parâmetros constantes	**Parâmetros passíveis de variação no tempo**
P: Número de pólos	L_m: indutância mútua entre estator e rotor
L_{lr}: Indutância de dispersão do rotor	R_r: resistência do rotor
K: coeficiente de atrito viscoso	J: Momento de inércia do eixo da máquina[2]

[2]O momento de inércia varia em função da carga presente no eixo da máquina.

Intervalo típico de variação para a indutância mútua (L_m)

A indutância mútua varia de acordo com o nível de saturação magnética da máquina. De acordo com a região de saturação, o valor real da indutância mútua (L_m) pode variar cerca de 20% em torno do valor nominal (L_m^*), correspondente ao nível de saturação nominal (Krishnan&Doran-87)[36]. Portanto, definindo $\beta=L_m/L_m^*$, obtém-se:

$$0,8<\beta<1,2$$

Intervalo típico de variação para a resistência do rotor (R_r)

A resistência de rotor varia principalmente em função da temperatura[3]. De acordo com o intervalo típico de variação de temperatura, é possível afirmar que a resistência de rotor (R_r) chega a atingir valores 20% abaixo e 50% acima de seu valor nominal (R_r^*), definido à temperatura ambiente (Krishnan&Doran-87)[36]; (Sul-89)[64]. Portanto, definindo $\gamma=R_r/R_r^*$, obtém-se:

$$0,8<\gamma<1,5$$

Intervalo típico de variação para a constante de tempo do rotor ($\tau_r=(L_m+L_{lr})/R_r$)

Definindo $\alpha=\tau_r/\tau_r^*$ então, com base nos intervalos típicos de variação definidos para R_r e L_m, dados respectivamente por β e γ, obtém-se (Krishnan&Doran-87)[36]:

$$0,5<\alpha<1,5$$

6.1.1. Modelo dinâmico do efeito da variação de parâmetros

As equações (6.1) e (6.2), correspondentes ao modelo dinâmico da parte eletromagnética da máquina de indução, apresentam parâmetros que podem variar no tempo. Para estudar o efeito desta variação de parâmetros no comportamento dinâmico da máquina, considere as seguintes expressões para os parâmetros variantes:

$$\tau_r = \tau_r^* + \Delta\tau_r$$
$$L_m = L_m^* + \Delta L_m$$

ou seja, os parâmetros reais são dados pelos seus valores nominais adicionados de um incremento correspondente à variação.

Sejam ψ_{qr}^{e*} e ψ_{dr}^{e*} os valores de fluxo obtidos com base nos valores nominais dos parâmetros. Com a introdução da variação nos parâmetros, os novos valores de fluxo são dados por (Garcés-80)[27]:

$$\psi_{qr}^{e} = \psi_{qr}^{e*} + \Delta\psi_{qr}^{e}$$
$$\psi_{dr}^{e} = \psi_{dr}^{e*} + \Delta\psi_{dr}^{e}$$

[3] Outros fatores que influenciam na definição do valor de resistência são o efeito pelicular e o nível de saturação.

Assumindo a hipótese de alinhamento entre o eixo direto do sistema de referência e o vetor fluxo de rotor, então $\psi_{qr}^{e*}=0$, o que implica em $\psi_{qr}^{e} = \Delta\psi_{qr}^{e}$.

Substituindo, agora, as expressões para τ_r, L_m, ψ_{qr}^{e} e ψ_{dr}^{e} nas equações (6.1) e (6.2), resulta:

$$\begin{bmatrix} \frac{d\Delta\psi_{qr}^{e}}{dt} \\ \frac{d\Delta\psi_{dr}^{e}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\tau_r} & -w_{slip} \\ w_{slip} & -\frac{1}{\tau_r} \end{bmatrix} \begin{bmatrix} \Delta\psi_{qr}^{e} \\ \Delta\psi_{dr}^{e} \end{bmatrix} + \frac{\psi_{dr}^{e*}}{\tau_r L_m^*} \begin{bmatrix} w_{slip}\tau_r^* & -w_{slip}L_m^* \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta L_m \\ \Delta\tau_r \end{bmatrix} \quad (6.4)$$

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_m + L_{lr}}\left[(\psi_{dr}^{e*} + \Delta\psi_{dr}^{e}).i_{qs}^{e} + \Delta\psi_{qr}^{e}.i_{ds}^{e}\right] \quad (6.5)$$

Dada uma variação para a indutância mútua (ΔL_m) e para a constante de tempo do rotor ($\Delta\tau_r$), as equações (6.4) e (6.5) fornecem o valor efetivo de conjugado eletromagnético (T_e), o qual pode ser comparado com o conjugado comandado (T_e^*), expresso na forma:

$$T_e^* = \frac{3}{2}\frac{P}{2}\frac{L_m^*}{L_m^* + L_{lr}}\psi_{dr}^{e*}.i_{qs}^{e} \quad (6.6)$$

6.1.2. Modelo estático do efeito da variação de parâmetros

O modelo estático da parte eletromagnética da máquina de indução pode ser obtido anulando-se os termos em derivada que aparecem na equação (6.1). Além disso, assumindo a hipótese de alinhamento entre o eixo direto do sistema de referência e o vetor fluxo de rotor, então $\psi_{qr}^{e*}=0$, resultando nas seguintes expressões algébricas[4]:

$$w_{slip}^* = \frac{L_m^*}{\tau_r^*}\frac{i_{qs}^{e}}{\psi_{dr}^{e*}} \quad (6.7)$$

$$\psi_{dr}^{e*} = L_m^* i_{ds}^{e} \quad (6.8)$$

$$T_e^* = \frac{3}{2}\frac{P}{2}\frac{L_m^*}{L_m^* + L_{lr}}\psi_{dr}^{e*}.i_{qs}^{e} \quad (6.9)$$

Substituindo a equação (6.8) nas equações (6.7) e (6.9), obtém-se:

[4] O asterisco nos parâmetros indica valores nominais e nas variáveis indica valores calculados a partir dos valores nominais dos parâmetros.

$$w^*_{slip} = \frac{1}{\tau^*_r}\frac{i^e_{qs}}{i^e_{ds}} \Rightarrow \frac{i^e_{qs}}{i^e_{ds}} = w^*_{slip}.\tau^*_r \tag{6.10}$$

$$T_e^* = \frac{3}{2}\frac{P}{2}\frac{\left(L^*_m\right)^2}{L^*_m + L_{lr}} i^e_{ds}.i^e_{qs} \tag{6.11}$$

Na figura 6.2, é apresentado o posicionamento espacial mantido entre os vetores de corrente de estator ($\bar{I}^e_s = i^e_{ds}.\bar{d}^e + i^e_{qs}.\bar{q}^e$) e de fluxo de rotor ($\bar{\psi}^{e^*}_r$), onde é possível definir o ângulo de conjugado (θ^*) (Krishnan&Doran-87)[36].

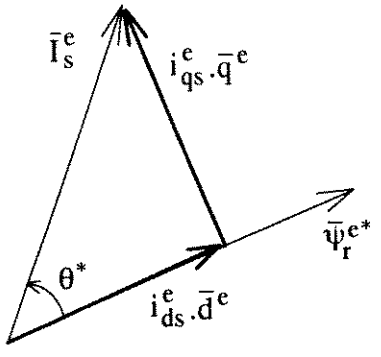


Figura 6.2 - Posicionamento espacial dos vetores de corrente de estator e fluxo de rotor

A partir da configuração da figura 6.2, pode-se obter as seguintes relações em termos do ângulo de conjugado:

$$\tan(\theta^*) = \frac{i^e_{qs}}{i^e_{ds}} \tag{6.12}$$

$$i^e_{qs} = \left\|\bar{I}^e_s\right\|.\mathrm{sen}(\theta^*) \tag{6.13}$$

$$i^e_{ds} = \left\|\bar{I}^e_s\right\|.\cos(\theta^*) \tag{6.14}$$

Substituindo a equação (6.10) na equação (6.12), obtém-se:

$$\tan(\theta^*) = w^*_{slip}\tau^*_r \Rightarrow \begin{cases} \mathrm{sen}(\theta^*) = \dfrac{w^*_{slip}\tau^*_r}{\sqrt{1+(w^*_{slip}\tau^*_r)^2}} \\ \cos(\theta^*) = \dfrac{1}{\sqrt{1+(w^*_{slip}\tau^*_r)^2}} \end{cases} \tag{6.15}$$

Substituindo as equações (6.13) e (6.14) na equação (6.11), obtém-se:

$$T_e^* = \frac{3}{2}\frac{P}{2}\frac{\left(L_m^*\right)^2}{L_m^* + L_{lr}}\left\|\bar{I}_s^e\right\|^2.\cos(\theta^*).\mathrm{sen}(\theta^*) \tag{6.16}$$

Caso os parâmetros L_m e τ_r não se encontrem em seus respectivos valores nominais, as relações envolvendo o ângulo de conjugado são dadas por:

$$\tan(\theta) = w_{slip}^*\tau_r \Rightarrow \begin{cases} \mathrm{sen}(\theta) = \dfrac{w_{slip}^*\tau_r}{\sqrt{1+(w_{slip}^*\tau_r)^2}} \\ \cos(\theta) = \dfrac{1}{\sqrt{1+(w_{slip}^*\tau_r)^2}} \end{cases} \tag{6.17}$$

e a expressão para o conjugado eletromagnético efetivamente produzido na máquina é dada na forma:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{\left(L_m\right)^2}{L_m + L_{lr}}\left\|\bar{I}_s^e\right\|^2.\cos(\theta).\mathrm{sen}(\theta) \tag{6.18}$$

Substituindo as equações (6.15) e (6.17) nas equações (6.16) e (6.18), é possível expressar a razão entre o conjugado efetivo (T_e) e o conjugado comandado (T_e^*) na forma (Krishnan&Doran-87)[36]:

$$\frac{T_e}{T_e^*} = \left[\frac{L_m}{L_m^*}\right]^2 \cdot \left[\frac{L_m^* + L_{lr}}{L_m + L_{lr}}\right] \cdot \left[\frac{\tau_r}{\tau_r^*}\right] \cdot \left[\frac{1+\left(w_{slip}^*\tau_r^*\right)^2}{1+\left(w_{slip}^*\tau_r\right)^2}\right] \tag{6.19}$$

6.1.3. Comparação entre os modelos estático e dinâmico

Com relação à produção de conjugado eletromagnético, as equações (6.4), (6.5) e (6.6) descrevem o efeito da variação de parâmetros tanto em regime transitório como em regime estacionário. Já a equação (6.19) descreve o efeito da variação de parâmetros apenas em regime estacionário. É interessante, portanto, comparar o desempenho dos dois conjuntos de equações na representação da variação de parâmetros, principalmente em termos da participação do regime transitório no comportamento da variação. Certamente, quanto menor a participação do regime transitório, mais precisa é a representação fornecida pela equação (6.19).

A seguir são apresentados resultados de simulação computacional onde são definidos perfis de variação para os parâmetros L_m e R_r assumindo valores dentro do intervalo de variação considerado típico para cada parâmetro. É utilizado um comando de conjugado (T_e^*) de valor constante e nominal.

Na simulação cujo resultado é apresentado na figura 6.3, foram consideradas variações abruptas nos parâmetros L_m e R_r, com o objetivo de forçar a ocorrência de transitórios significativos. A evolução temporal do conjugado efetivo T_e fornecido pelos modelos estático (T_e^{est}) e dinâmico (T_e^{din}), quando comparados com o conjugado comandado (T_e^*), mostra uma variação significativa e qualitativamente análoga ao perfil de variação dos parâmetros. Além disso, mesmo para variações abruptas de parâmetros, o regime transitório apresenta uma baixa constante de tempo e oscilações bem amortecidas. A variação percentual entre os valores fornecidos pelos dois modelos, dada por $(T_e^{din}-T_e^{est})/T_e^*$, permite concluir que, o valor de convergência do modelo dinâmico é idêntico ao valor fornecido pelo modelo estático.

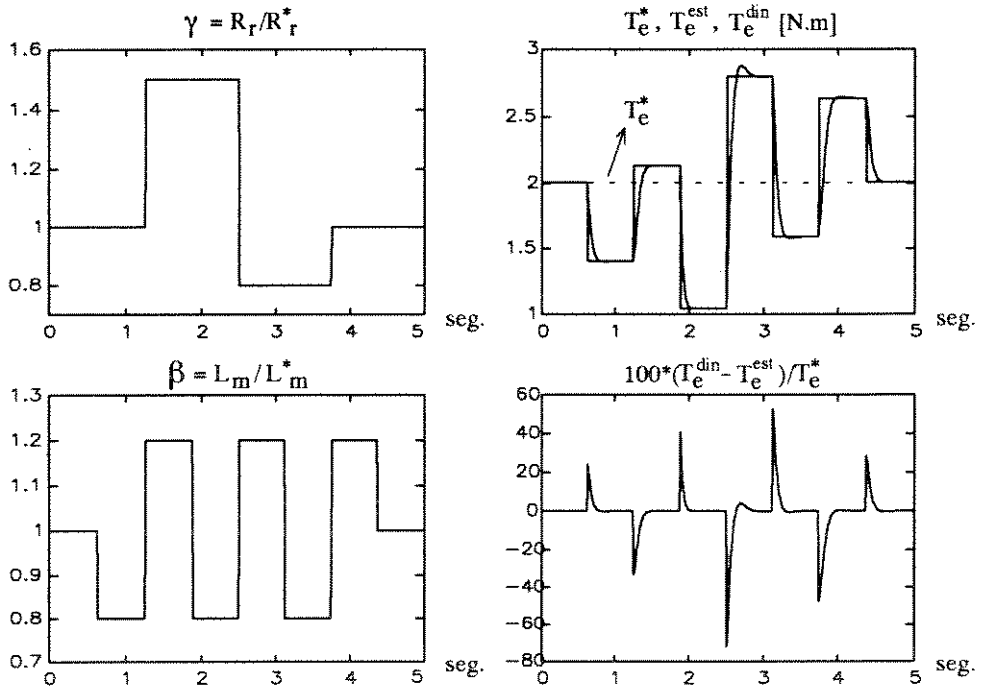


Figura 6.3 - Comparação entre os modelos estático e dinâmico para variações abruptas de parâmetros e comando de conjugado (T_e^*) constante

Quando o nível de saturação da máquina é mantido constante (comando de fluxo constante), as variações no parâmetro L_m podem ser desprezadas (Bose-86)[11]; (Sul-89)[64]. Dessa forma, na simulação cujo resultado é apresentado na figura 6.4, foram consideradas apenas variações no parâmetro R_r. Neste caso, é assumido um perfil de variação suave, sendo que os gráficos da primeira coluna correspondem aos resultados de uma variação senoidal de dois períodos em apenas 6 segundos, enquanto que os gráficos da segunda coluna correspondem aos resultados de uma variação senoidal de dois períodos em 600 segundos. Pode-se verificar que quanto mais lenta a variação mais próximos são os valores fornecidos pelos modelos estático e dinâmico.

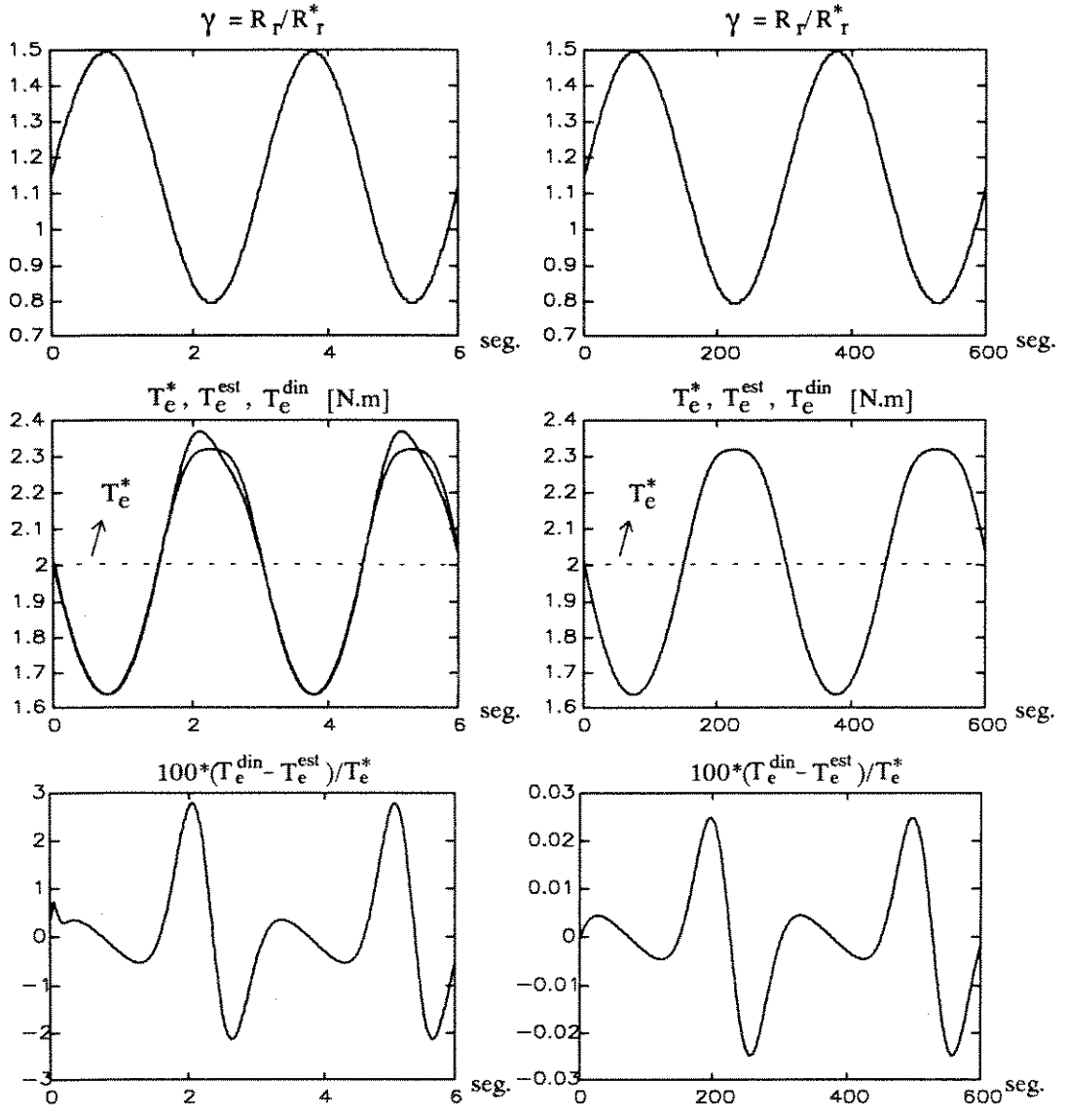


Figura 6.4 - Comparação entre os modelos estático e dinâmico para variações suaves de parâmetros

Considerando-se que, na prática, a constante de tempo de variação da resistência do rotor é dada em minutos, pode-se concluir que os modelos estático e dinâmico que descrevem o efeito da variação de parâmetros na produção de conjugado vão produzir resultados equivalentes. Com base na maior simplicidade do modelo estático dado pela equação (6.19), ele será adotado no desenvolvimento de procedimentos de estimação de parâmetros a serem apresentados nas próximas seções.

6.2. Análise do desempenho do método indireto de controle vetorial quando sujeito à variação de parâmetros

No Capítulo 5, por ocasião das simulações computacionais referentes à implementação do método indireto de controle vetorial, não foi considerado qualquer tipo de variação de parâmetros, ou seja, a todo instante os parâmetros do controlador correspondiam exatamente aos parâmetros da máquina de indução. Sendo assim, o controlador mostrou-se capaz de garantir a manutenção do desacoplamento entre as variáveis de controle, produzindo as correntes de alimentação necessárias para conduzir a máquina ao atendimento preciso dos comandos de referência de conjugado e fluxo.

Estando a máquina de indução sujeita à variação de parâmetros, é interessante verificar o desempenho do método indireto de controle vetorial quando os parâmetros correspondentes do controlador são ou não atualizados. Nos resultados de simulação apresentados a seguir, o comando de fluxo é fixado em seu valor nominal e são considerados três perfis de comando de conjugado eletromagnético a serem atendidos pela máquina de indução. Com base no estudo de variação de parâmetros apresentado nas seções anteriores, a fixação do fluxo em seu valor nominal permite considerar a indutância mútua (L_m) constante. Portanto, o único parâmetro passível de variação é a resistência do rotor (R_r). Neste caso, é assumido um perfil senoidal de variação para a resistência do rotor de forma a cobrir todo o intervalo típico de variação.

As figuras 6.5, 6.6 e 6.7 apresentam resultados de simulação para os três perfis de conjugado considerados. Para cada simulação é apresentado um conjunto de quatro gráficos, definidos da seguinte forma:

- Gráfico ψ_{qr}^{e} , ψ_{qr}^{e*}: apresenta o comportamento no tempo das componentes em quadratura do fluxo de rotor. A componente de referência ψ_{qr}^{e*} é definida como sendo identicamente nula (condição suficiente para garantir o alinhamento entre o eixo direto do sistema de referência síncrono e o vetor fluxo de rotor), enquanto que a componente ψ_{qr}^{e} corresponde ao valor efetivo produzido na máquina pelo ação de controle. Se os valores dos parâmetros do controlador se mantiverem idênticos aos parâmetros correspondentes da máquina, a componente ψ_{qr}^{e} deve seguir exatamente sua referência ψ_{qr}^{e*}, indicando a manutenção do alinhamento.

- Gráfico ψ_{r}^{e} , ψ_{dr}^{e} , ψ_{dr}^{e*}: apresenta o comportamento no tempo do módulo do vetor fluxo de rotor ψ_{r}^{e} e das componentes de eixo direto ψ_{dr}^{e} e ψ_{dr}^{e*}, onde a componente de referência de eixo direto ψ_{dr}^{e*} é definida como sendo idêntica ao comando de fluxo de referência (a componente de referência em quadratura não participa na representação do fluxo) e a componente ψ_{dr}^{e} corresponde ao valor efetivo produzido na máquina pela ação de

controle. Se os valores dos parâmetros do controlador se mantiverem idênticos aos parâmetros correspondentes da máquina, a componente ψ_{dr}^{e} deve seguir exatamente sua referência ψ_{dr}^{e*} e deve ser igual ao módulo do fluxo de rotor ψ_{r}^{e}, indicando o atendimento do comando de fluxo de referência e a existência de alinhamento (Caminhas *et al.*-90)[13].

- Gráfico T_e , T_e^*: apresenta o comportamento no tempo do conjugado de referência T_e^* e do conjugado T_e efetivamente produzido na máquina pela ação do controle vetorial. Em cada figura é utilizado um perfil de conjugado definido na forma:

Figura 6.5: Perfil de conjugado constante e igual ao valor nominal - utilizado para verificar o comportamento no ponto de operação nominal.

Figura 6.6: Perfil de conjugado senoidal assumindo valores entre os conjugados nominais de aceleração e frenagem - utilizado para verificar o comportamento em toda a região de operação.

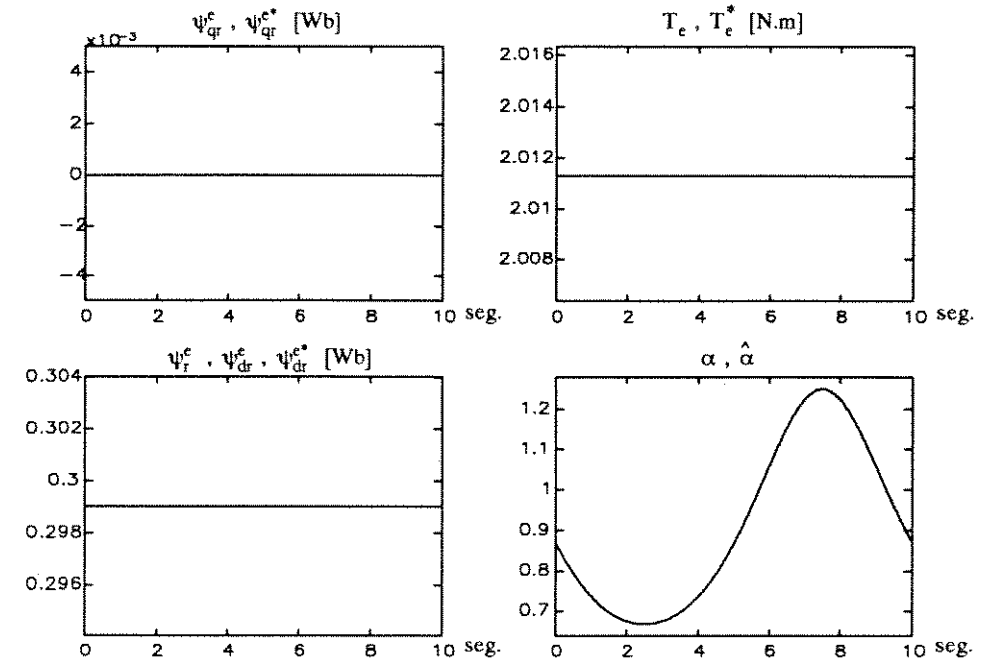
Figura 6.7: Perfil de conjugado tipo onda quadrada assumindo valores entre os conjugados nominais de aceleração e frenagem - utilizado para verificar o comportamento em vários pontos da região de operação para variações abruptas no comando de conjugado.

- Gráfico α , $\hat{\alpha}$: apresenta o comportamento no tempo do valor da constante de tempo do rotor relativamente ao seu valor nominal ($\alpha=\tau_r/\tau_r^*$), sendo que α corresponde ao parâmetro da máquina[5] e $\hat{\alpha}$ ao parâmetro do controlador. Para cada perfil de conjugado são consideradas duas situações, correspondentes aos itens a e b das figuras:

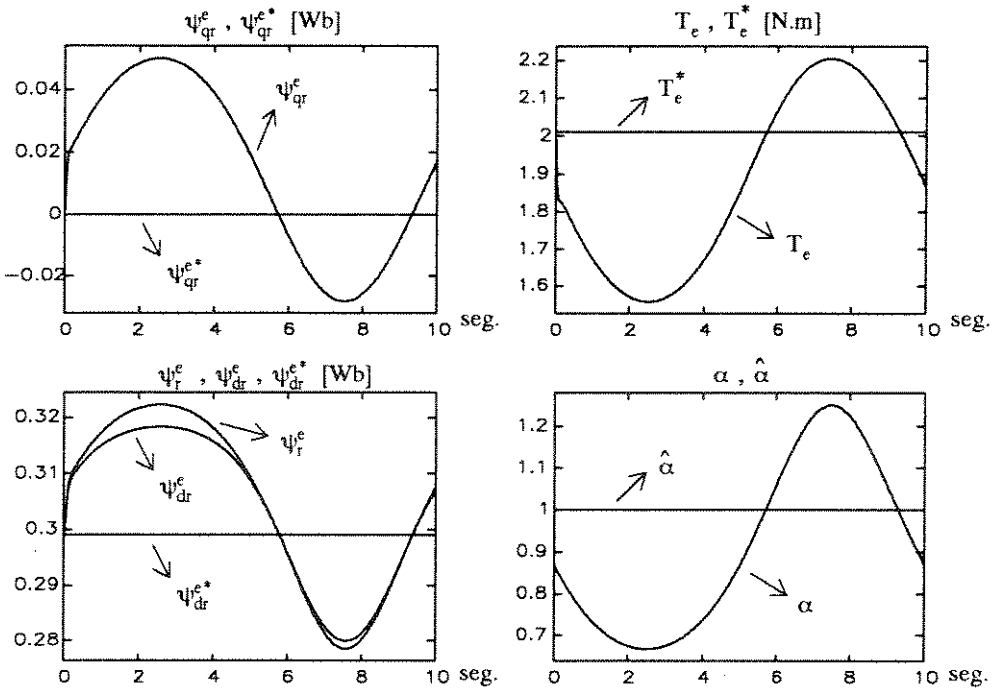
Item a: o valor do parâmetro $\hat{\alpha}$ do controlador acompanha exatamente o valor do parâmetro α da máquina.

Item b: não importando a variação do parâmetro α da máquina, o parâmetro $\hat{\alpha}$ do controlador é mantido constante no valor nominal.

[5]Pode-se observar que α é inversamente proporcional a R_r, ou seja, o perfil de α é o inverso do perfil senoidal de R_r.

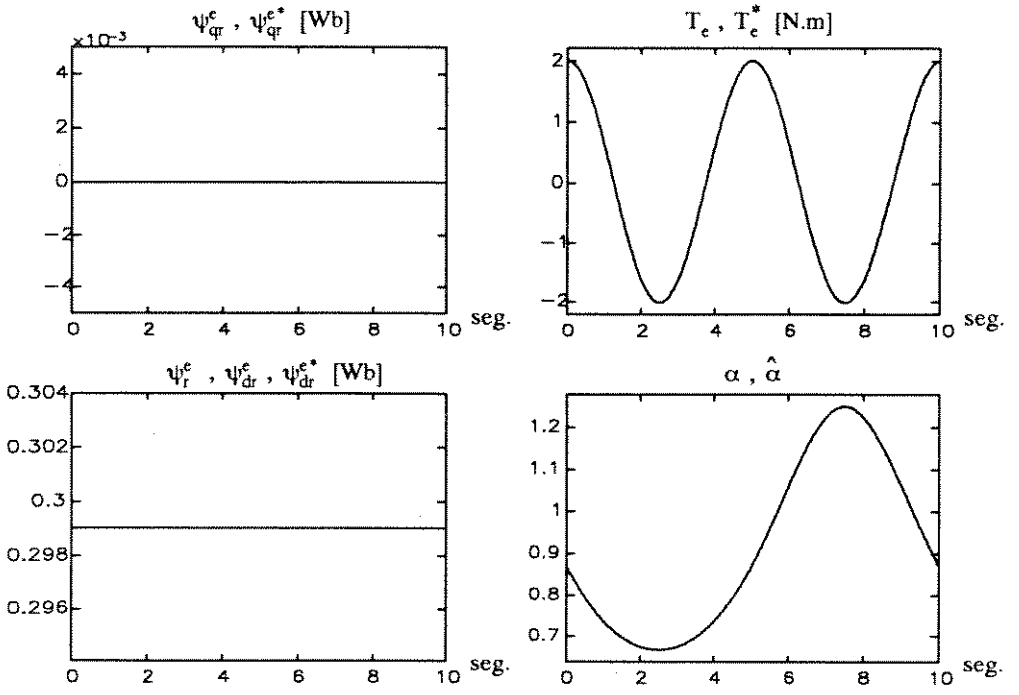


(a)

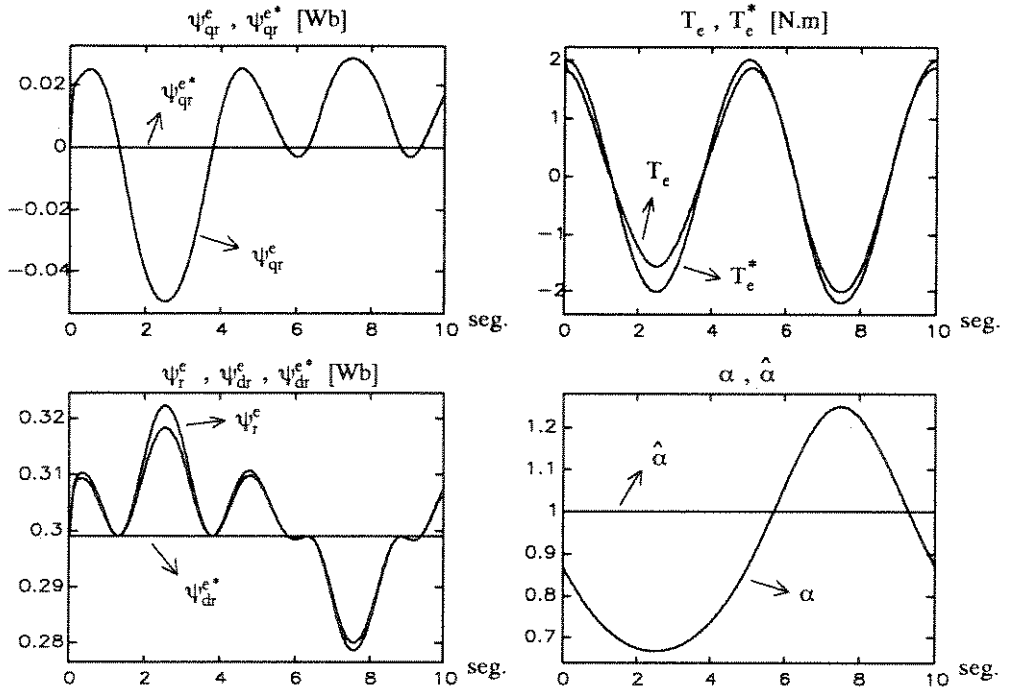


(b)

Figura 6.5 - Atuação do método indireto de controle vetorial para comando de conjugado constante e parâmetro α variante no tempo (**item a**: com correção do parâmetro $\hat{\alpha}$ do controlador; **item b**: sem correção)

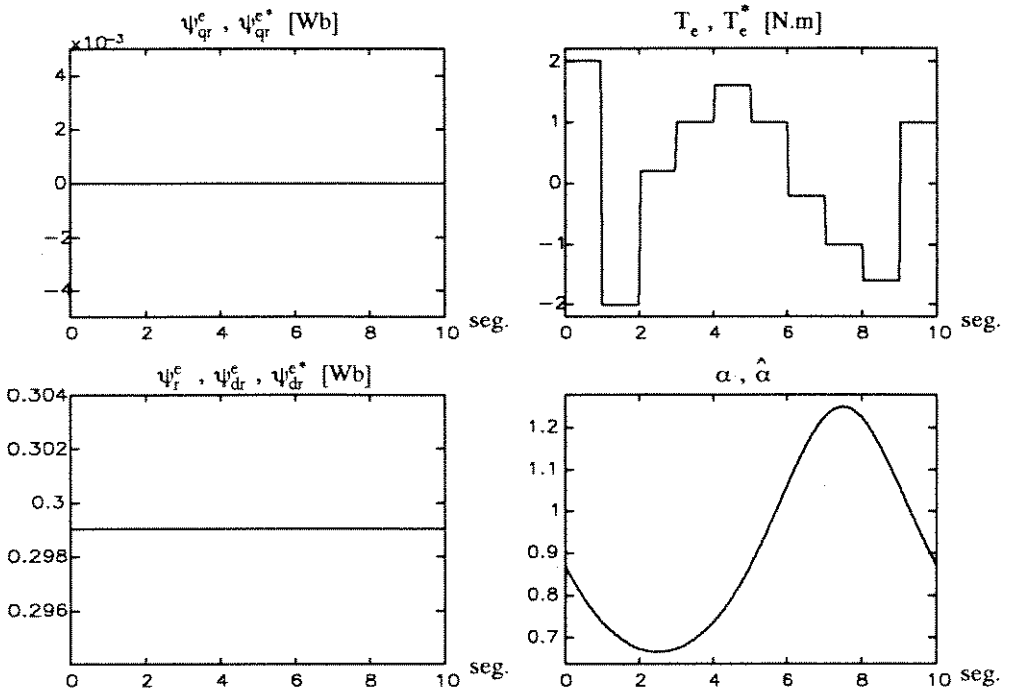


(a)

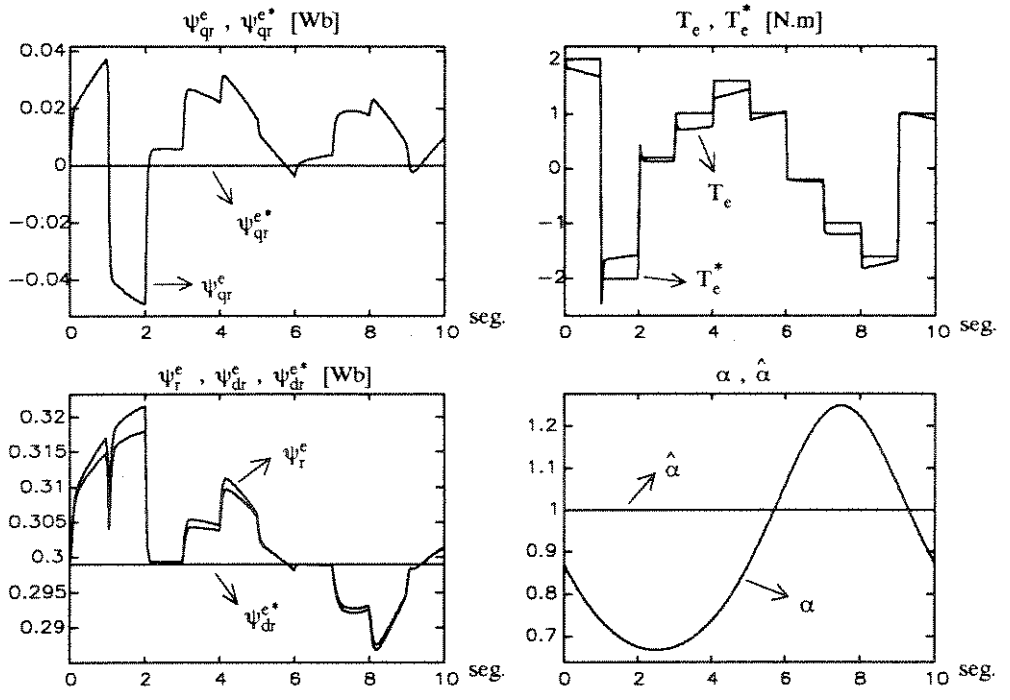


(b)

Figura 6.6 - Atuação do método indireto de controle vetorial para comando de conjugado tipo senoidal e parâmetro α variante no tempo (**item a**: com correção do parâmetro $\hat{\alpha}$ do controlador; **item b**: sem correção)



(a)



(b)

Figura 6.7 - Atuação do método indireto de controle vetorial para comando de conjugado tipo onda quadrada e parâmetro α variante no tempo (**item a**: com correção do parâmetro $\hat{\alpha}$ do controlador; **item b**: sem correção)

Os resultados de simulação apresentados nas figuras 6.5, 6.6 e 6.7 mostram que, independente do perfil de conjugado, a não coincidência de valores entre os parâmetros da máquina e do controlador (neste caso, representado apenas pelo parâmetro α) conduz invariavelmente à perda da condição de alinhamento entre o eixo direto do sistema de

referência e o vetor fluxo de rotor. Como conseqüência, pelo fato dos comandos de controle serem definidos com base neste alinhamento, os comandos de fluxo e conjugado acabam não sendo obedecidos pela máquina, provocando uma degradação de desempenho significativa.

Por outro lado, mesmo com parâmetros da máquina variantes no tempo, se os parâmetros correspondentes do controlador forem adequadamente atualizados, o desempenho do processo de controle se mantém inalterado. No entanto, por motivos de construção da máquina de indução com rotor tipo gaiola de esquilo, o parâmetro R_r não pode ser medido diretamente. Portanto, em situações práticas, a questão que surge é como detectar a variação deste parâmetro da máquina e atuar no sentido de corrigir o parâmetro correspondente do controlador.

Ainda neste capítulo, soluções para este problema de estimação de parâmetros serão propostas e implementadas computacionalmente.

6.3. Análise do desempenho do método por realimentação de estado quando sujeito à variação de parâmetros

Como mencionado por ocasião do desenvolvimento do método por realimentação de estado, seu desempenho está totalmente vinculado ao desempenho do observador de fluxo baseado em redes neurais. Portanto, é interessante avaliar primeiramente o desempenho do observador de fluxo quando a máquina de indução apresenta parâmetros variantes no tempo.

A seguir, são apresentados resultados de simulação onde é possível comparar as componentes de fluxo estimadas pelo observador de fluxo ($\hat{\psi}^e_{qr}$ e $\hat{\psi}^e_{dr}$) e as correspondentes componentes de fluxo da máquina de indução (ψ^e_{qr} e ψ^e_{dr}) em duas situações caracterizadas pela operação da máquina na condição de alinhamento entre o eixo direto do sistema d-q de referência e o vetor fluxo de rotor. Os seguintes critérios foram adotados no processo de implementação computacional:

- o fluxo de referência é mantido constante no valor nominal;
- o conjugado de referência é mantido constante no valor nominal;
- o parâmetro L_m é considerado constante;
- o parâmetro R_r é considerado variável, e esta variação é indicada em termos do parâmetro α, que representa a variação ocorrida na constante de tempo do rotor em relação ao seu valor nominal. São escolhidos dois perfis de variação para o parâmetro R_r, ambos assumindo valores no intervalo típico de variação.

A figura 6.8 apresenta os comandos de fluxo e conjugado e os dois perfis de variação de R_r expressos em termos do parâmetro α.

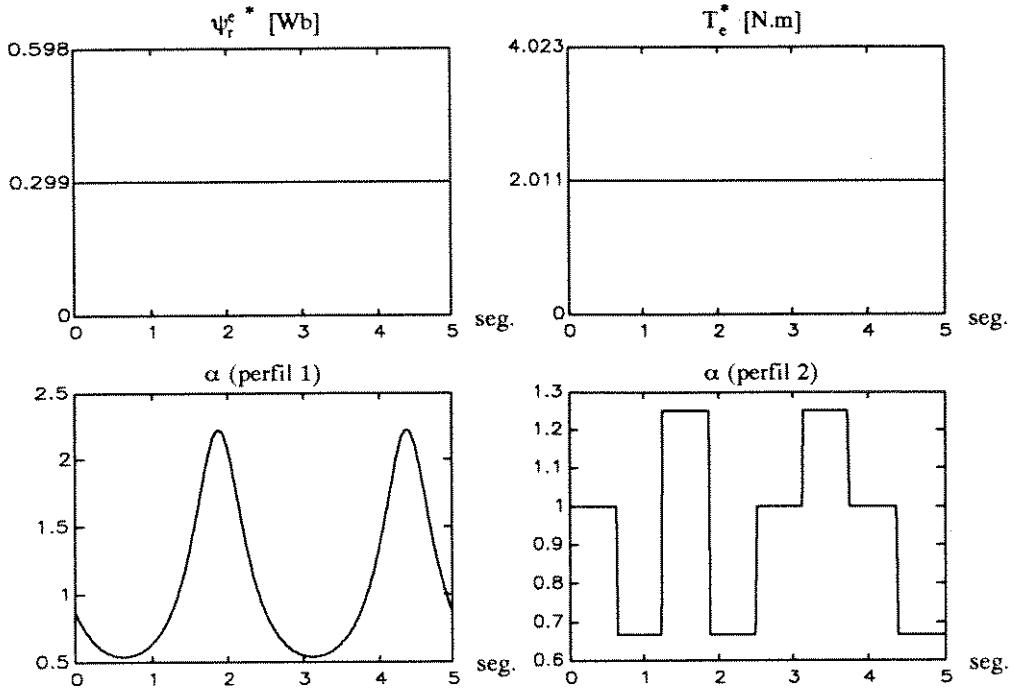


Figura 6.8 - Evolução temporal dos comandos de controle e das variações assumidas para o parâmetro α

As figuras 6.9 e 6.10 apresentam os resultados de simulação referentes ao desempenho do observador de fluxo baseado em redes neurais.

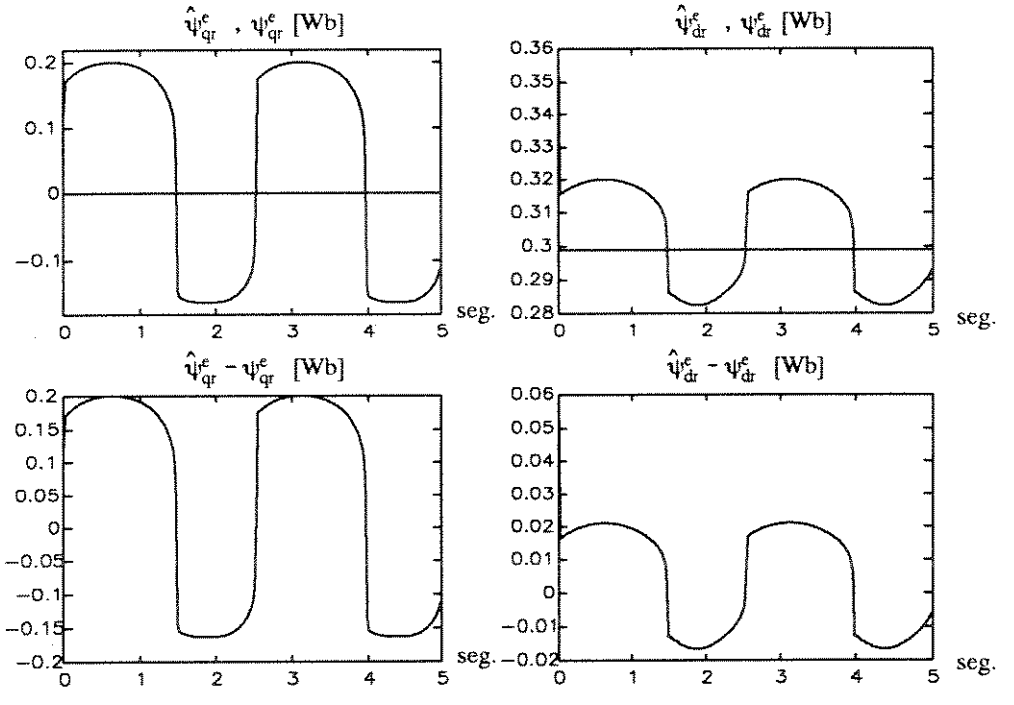


Figura 6.9 - Desempenho do observador de fluxo (implementado considerando-se parâmetros constantes) quando a máquina opera na condição de alinhamento e apresenta o parâmetro α variante no tempo (perfil 1 - figura 6.8)

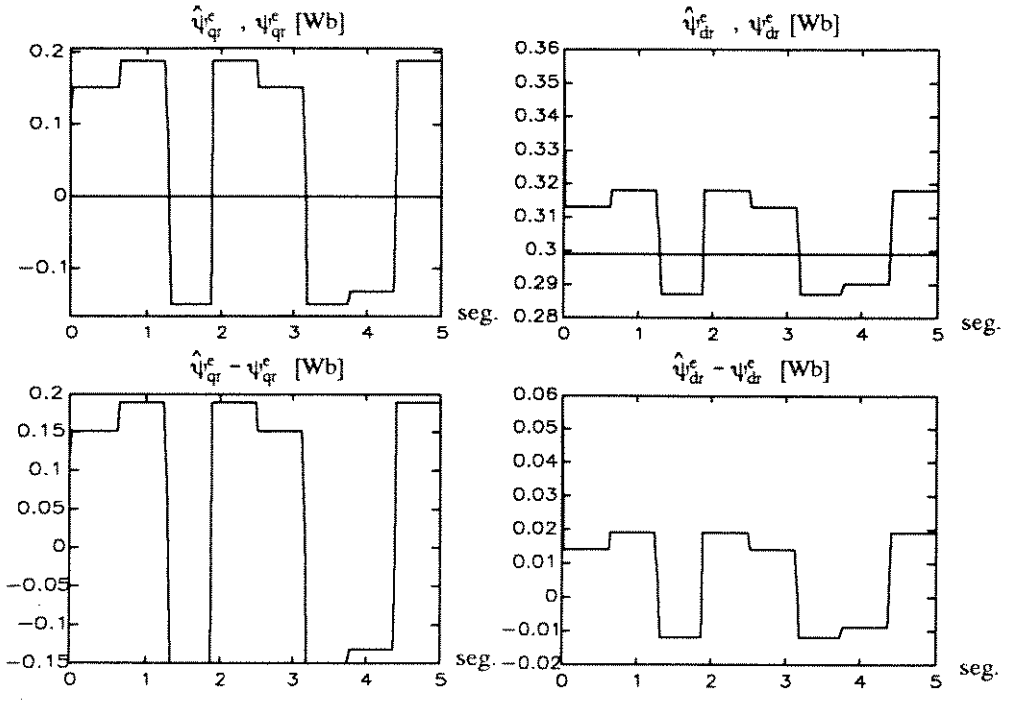


Figura 6.10 - Desempenho do observador de fluxo (implementado considerando-se parâmetros constantes) quando a máquina opera na condição de alinhamento e apresenta o parâmetro α variante no tempo (perfil 2 - figura 6.8)

Apesar de considerarem apenas a situação onde o comando de conjugado é constante, os resultados de simulação apresentados nas figuras 6.9 e 6.10, quando comparados com o resultado apresentado na figura 5.13, Capítulo 5, já indicam uma degradação significativa de desempenho do observador de fluxo. Este resultado já era esperado, visto que o observador foi treinado utilizando a máquina com parâmetros invariantes no tempo.

Considerando o parâmetro α da máquina de indução variando de acordo com o perfil 1 (figura 6.8) e mantendo a referência de fluxo no valor nominal, dois exemplos de simulação utilizando o observador de fluxo no esquema de controle por realimentação de estado (veja figura 5.8, Capítulo 5) são apresentados na seqüência, diferindo com relação à referência de conjugado (T_e^*) no seguinte aspecto:

Figura 6.11: manutenção da referência de conjugado no valor nominal;

Figura 6.12: perfil de conjugado do tipo senoidal, percorrendo todo o intervalo entre aceleração e frenagem nominal;

A operação do controlador é descrita pelas equações (5.44), (5.45) e (5.46) (Capítulo 5), sendo que os coeficientes de ganho dos controladores PI são dados por:

$$K_{Pq} = K_{Pd} = 1;\ K_{Iq} = K_{Id} = 2$$

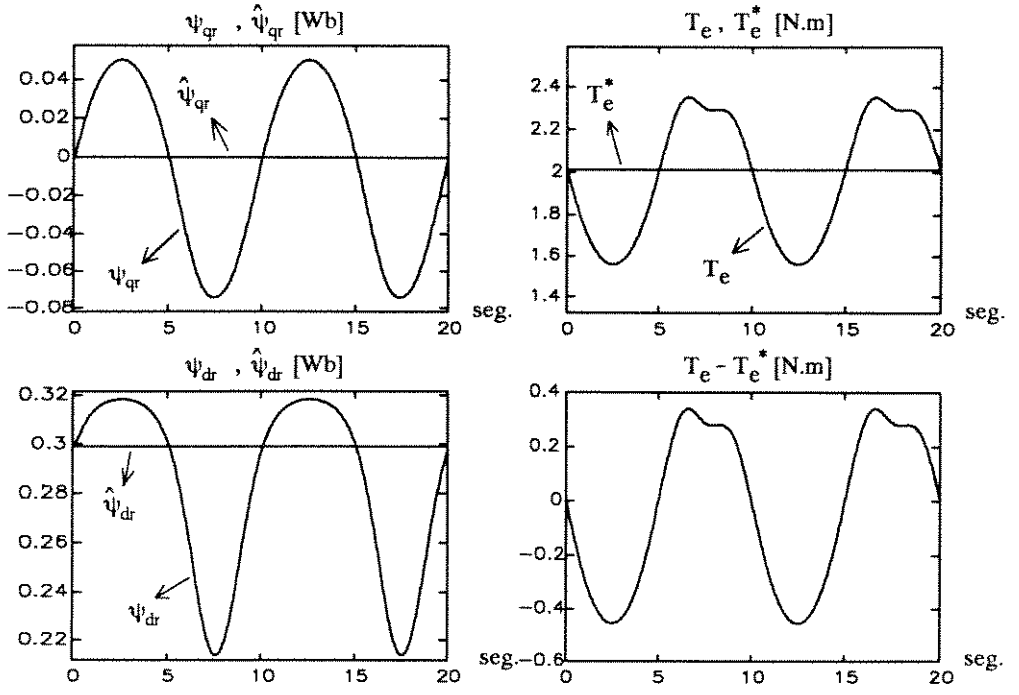


Figura 6.11 - Desempenho do método de controle por realimentação de estado quando a máquina de indução apresenta o parâmetro α variante no tempo (perfil 1 - figura 6.8) e o comando de conjugado (T_e^*) é constante

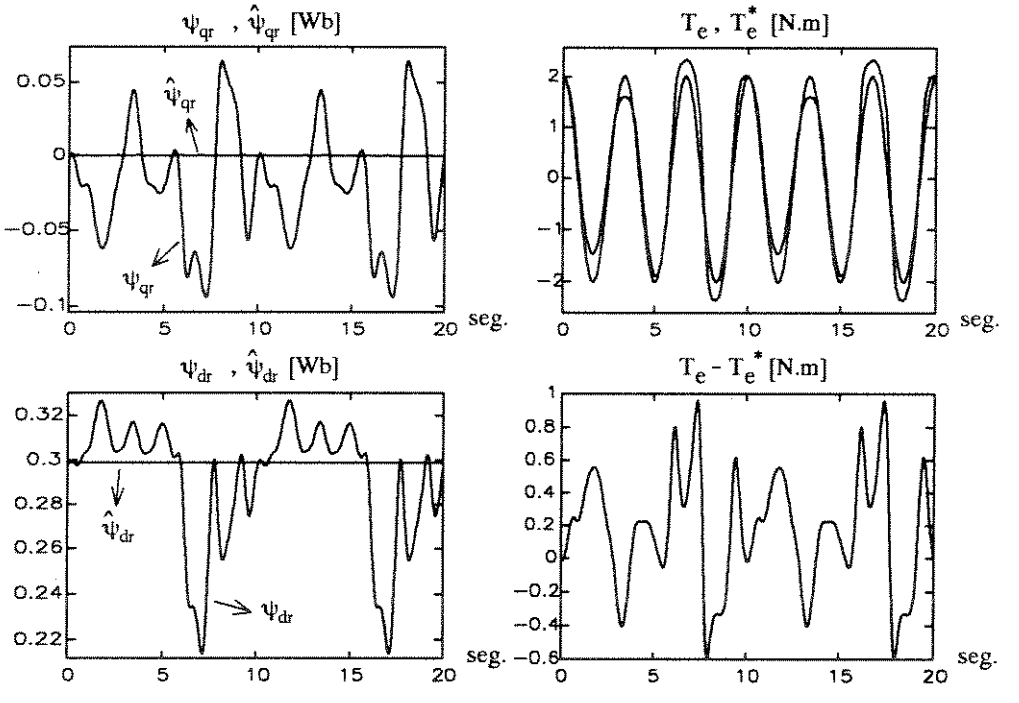


Figura 6.12 - Desempenho do método de controle por realimentação de estado quando a máquina de indução apresenta o parâmetro α variante no tempo (perfil 1 - figura 6.8) e o comando de conjugado (T_e^*) varia senoidalmente

Como pode ser observado nas figuras 6.11 e 6.12, os controladores PI são responsáveis por conduzir os valores estimados de fluxo para as respectivas referências. No entanto, como o desempenho do observador está bastante degradado pelo efeito da variação do parâmetro α, o valor real das componentes de fluxo não acompanham os correspondentes valores estimados. Com isso, o conjugado efetivamente produzido na máquina (T_e) não acompanha o comando de conjugado (T_e^*).

Portanto, chega-se à mesma conclusão extraída dos resultados de simulação apresentados na seção anterior, ou seja, para a manutenção do nível de desempenho do processo de controle, é necessário utilizar métodos de estimação para detectar possíveis variações de parâmetros da máquina e atuar no sentido de corrigir os correspondentes parâmetros do controlador.

6.3.1. O observador expandido de fluxo

Assumindo que o comportamento da variação do parâmetro α é conhecido, é preciso definir uma forma de se atualizar o observador de fluxo, até então desenvolvido com base na invariância dos parâmetros da máquina. A modificação a ser introduzida no projeto do observador deve buscar uma maneira de se representar os parâmetros variantes explicitamente, permitindo a aplicação direta do processo de ajuste. Uma solução imediata é dada pela introdução dos parâmetros variantes como entradas adicionais do observador de fluxo.

Considerando apenas a variação do parâmetro α, o observador de fluxo vai, então, adquirir uma forma expandida, conforme apresentado na figura 6.13.

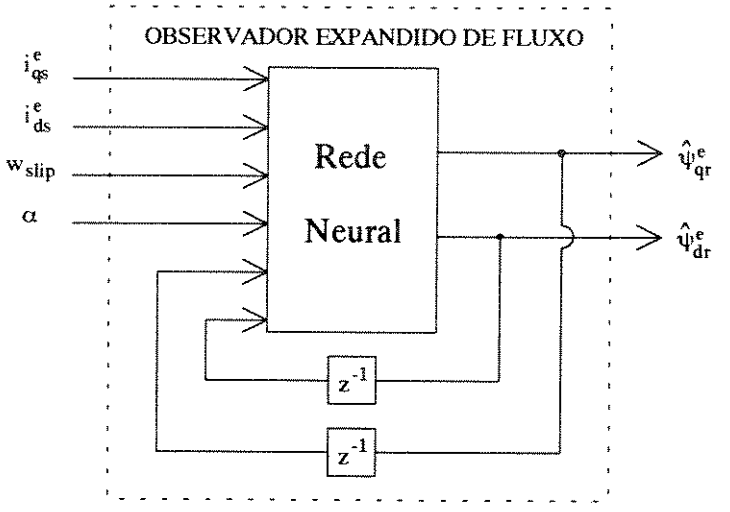


Figura 6.13 - Observador expandido de fluxo

Utilizando o mesmo procedimento apresentado no Capítulo 5 para realizar o treinamento da rede neural (os parâmetros de treinamento são apresentados no Apêndice F,

seção 4), o observador de fluxo foi, então, testado para as mesmas condições de operação definidas na geração dos resultados de simulação das figuras 6.9 e 6.10, produzindo os resultados apresentados nas figuras 6.14 e 6.15.

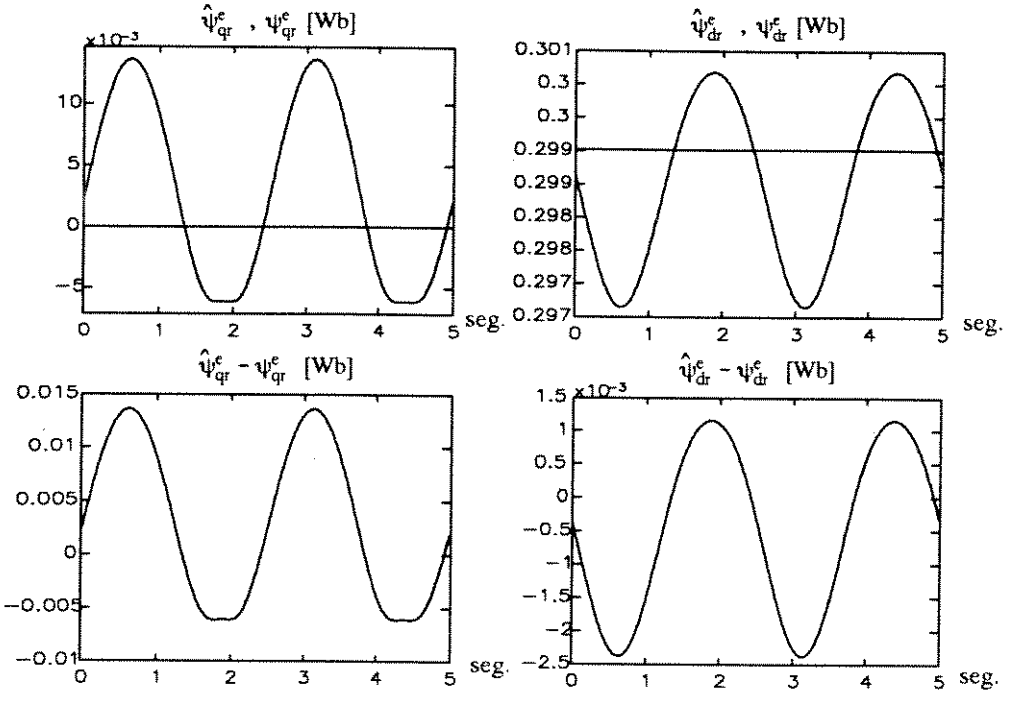


Figura 6.14 - Desempenho do observador expandido de fluxo quando a máquina opera na condição de alinhamento e apresenta o parâmetro α variando continuamente no tempo (perfil 1 - figura 6.8)

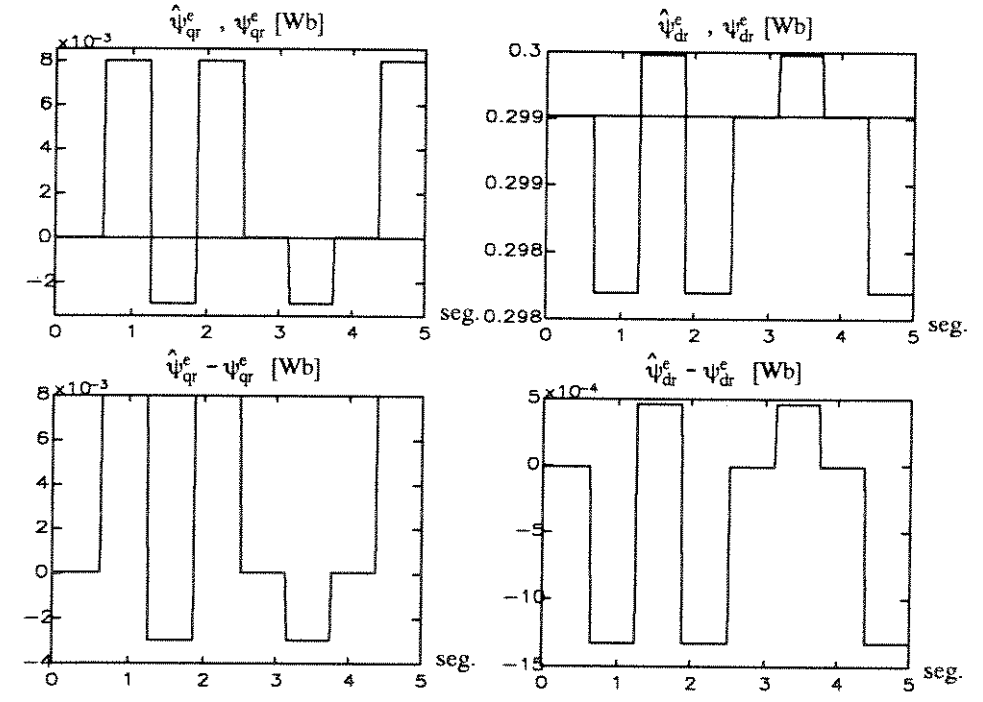


Figura 6.15 - Desempenho do observador expandido de fluxo quando a máquina opera na condição de alinhamento e apresenta o parâmetro α variando em degrau (perfil 2-figura 6.8)

Comparando-se os resultados de simulação apresentados nas figuras 6.14 e 6.15 com aqueles das figuras 6.9 e 6.10, pode-se verificar uma acentuada redução do erro de estimação. É importante verificar também a influência da condição inicial no desempenho do observador de fluxo. Assumindo uma condição inicial que não corresponde aos valores iniciais das componentes reais de fluxo, as figuras 6.16 e 6.17 apresentam o comportamento de convergência do observador.

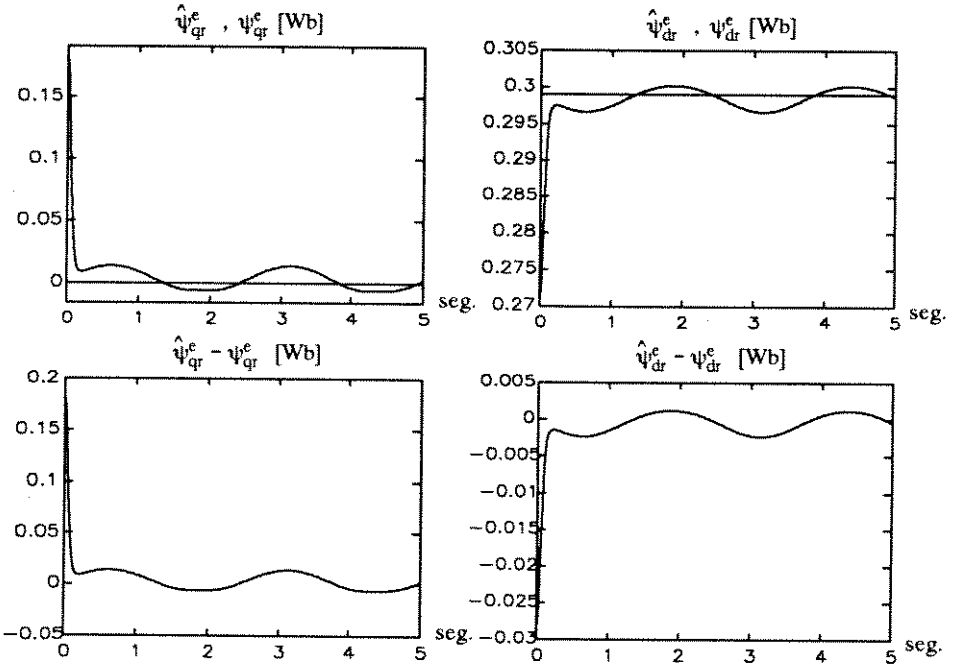


Figura 6.16 - Desempenho do observador expandido de fluxo para condição inicial arbitrária, para a máquina operando na condição de alinhamento e apresentando o parâmetro α variando continuamente no tempo (perfil 1 - figura 6.8)

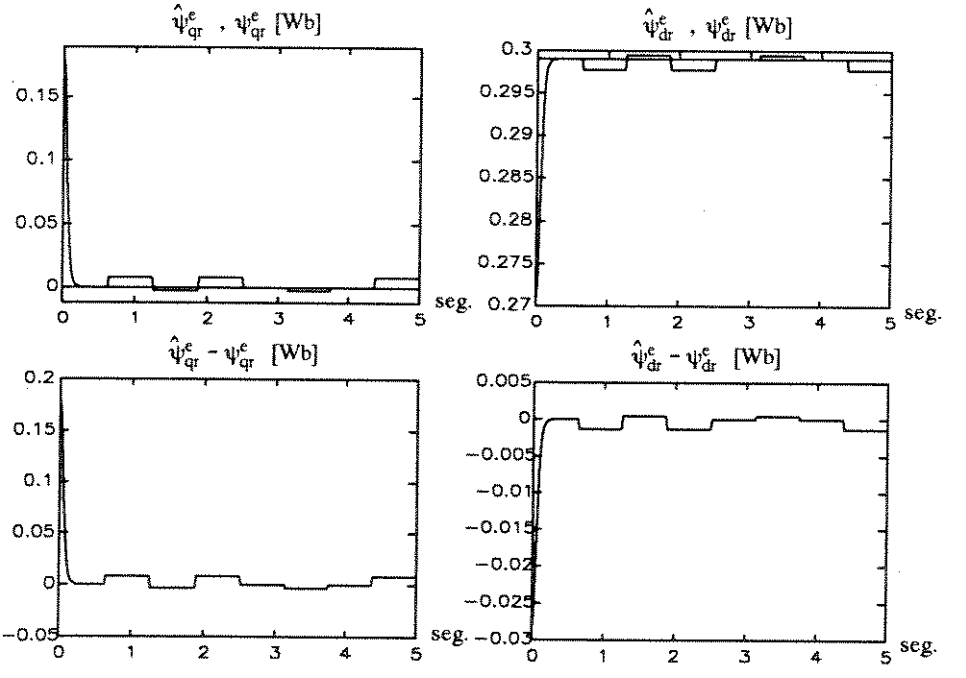


Figura 6.17 - Desempenho do observador expandido de fluxo para condição inicial arbitrária, para a máquina operando na condição de alinhamento e apresentando o parâmetro α variando em degrau (perfil 2 - figura 6.8)

Com base nas figuras 6.16 e 6.17, pode-se verificar que, mesmo para condições iniciais inadequadas, as saídas do observador de fluxo convergem para valores próximos aos valores reais de fluxo, indicando que seu desempenho não depende da condição inicial.

Introduzindo, agora, o observador expandido de fluxo no esquema de controle por realimentação de estado e testando para as mesmas condições de operação definidas na geração dos resultados de simulação das figuras 6.11 e 6.12, obtém-se os resultados apresentados nas figuras 6.18 e 6.19.

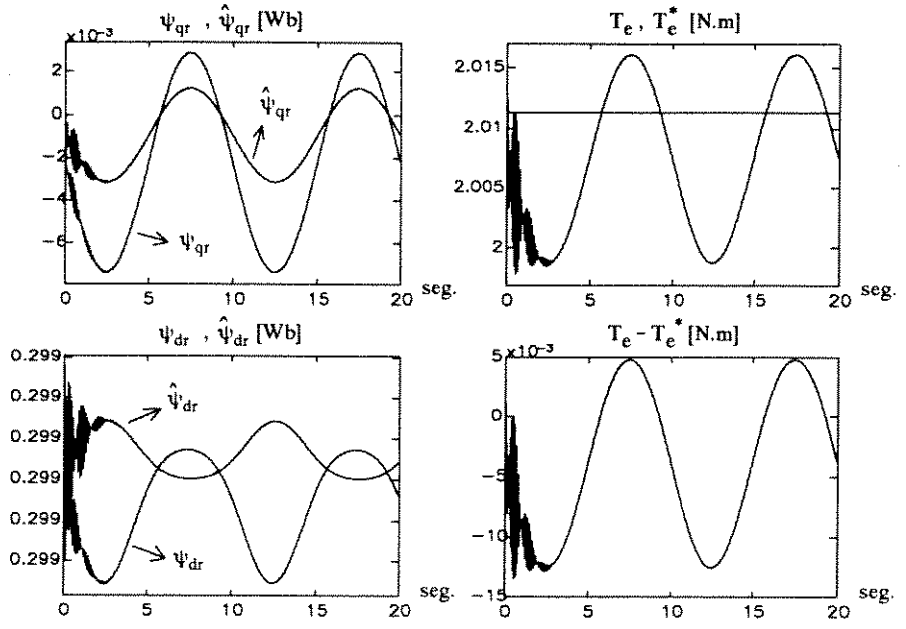


Figura 6.18 - Desempenho do método de controle por realimentação de estado utilizando observador expandido de fluxo quando a máquina de indução apresenta o parâmetro α variante no tempo e o comando de conjugado é constante

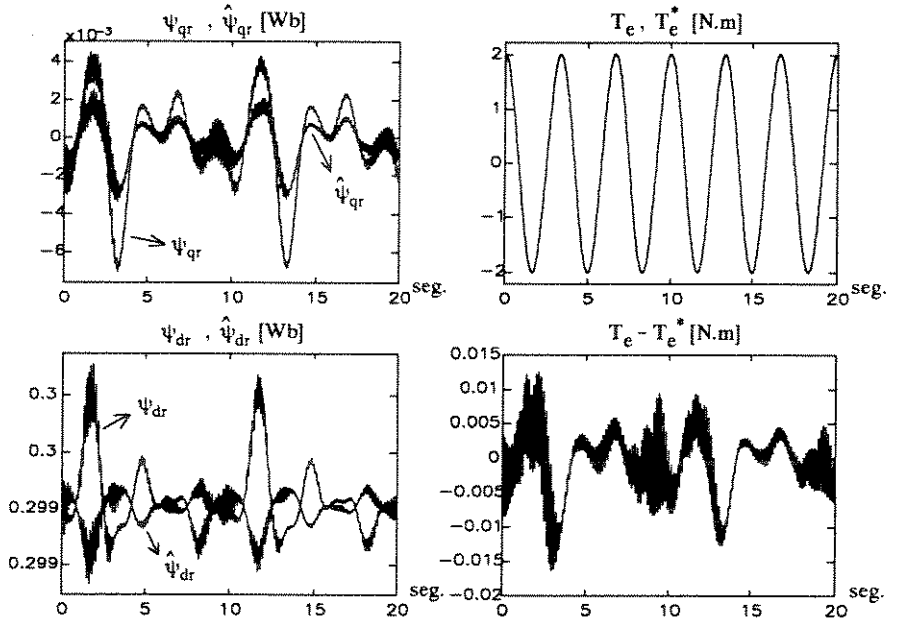


Figura 6.19 - Desempenho do método de controle por realimentação de estado utilizando observador expandido de fluxo quando a máquina de indução apresenta o parâmetro α variante no tempo e o comando de conjugado é senoidal

Devido à constante alteração da condição de operação o controlador já não consegue conduzir as componentes de fluxo estimadas para zero. No entanto, comparando-se com os resultados apresentados nas figuras 6.11 e 6.12, observa-se uma operação muito próxima da condição de alinhamento, viabilizando o atendimento do comando de conjugado.

A partir dos resultados apresentados, pode-se concluir que, na presença de parâmetros variantes no tempo, a atualização de parâmetros do controlador com a introdução de processos de estimação de parâmetros pode garantir o bom desempenho da malha de controle.

6.4. Estimação não-linear de parâmetros utilizando redes neurais

O procedimento de estimação de parâmetros a ser adotado no desenvolvimento que se segue está baseado na estrutura de controle adaptativo denominada regulador auto-ajustável, descrita no Capítulo 3. O esquema básico do regulador auto-ajustável é apresentado na figura 6.20.

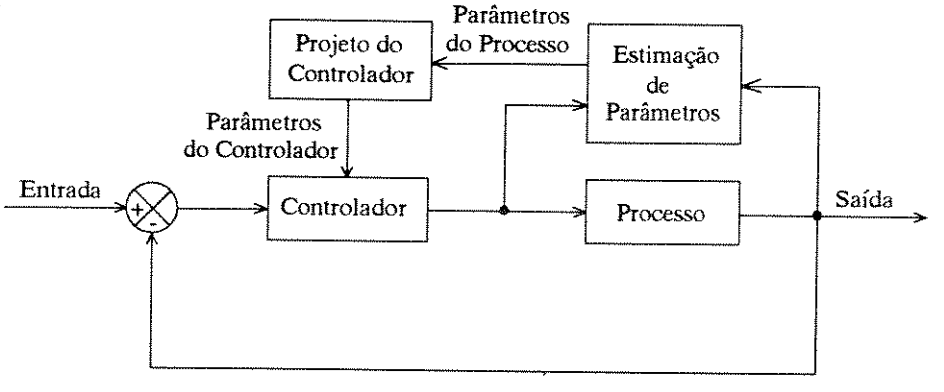


Figura 6.20 - Esquema básico do regulador auto-ajustável

Neste esquema, os parâmetros desconhecidos do processo são estimados durante a operação do sistema, sendo que os valores estimados são utilizados no ajuste dos parâmetros do controlador. O que geralmente se utiliza no bloco de estimação de parâmetros é um modelo linear explícito do processo na forma de equações de estado. No entanto, é possível reparametrizar o processo de tal forma que ele possa ser expresso em termos de parâmetros do controlador. Desta forma, os parâmetros do controlador podem ser ajustados diretamente, com base em um modelo implícito do processo, sendo que o controlador passa a ser denominado regulador auto-ajustável implícito.

No caso do problema de controle vetorial da máquina de indução, um modelo implícito do processo é dado pela equação (6.19), uma equação não-linear envolvendo os parâmetros passíveis de variação L_m e R_r, a velocidade de escorregamento calculada pelo

controlador w_s^*, o comando de conjugado T_e^* e o conjugado efetivamente produzido T_e. Portanto, o estimador de parâmetros deve realizar um mapeamento não-linear.

6.4.1. O procedimento de solução do problema

Com relação ao problema de controle, as seguintes hipóteses devem ser consideradas:

1. os parâmetros da máquina de indução a ser utilizada nas simulações computacionais são apresentados no Apêndice E, incluindo os valores nominais para a velocidade de escorregamento (WSnom) e conjugado eletromagnético (TEnom);
2. os comandos de referência para o controlador, correspondentes ao sinal de entrada da figura 6.20, são dados pelo fluxo de referência (ψ_r^{e*}) e pelo conjugado eletromagnético de referência (T_e^*);
3. o comando de fluxo de referência é assumido constante e de valor nominal (ψ_r^{e*}=Fnom). Com isso, como discutido em seções anteriores, torna-se desnecessário considerar a variação na indutância mútua (L_m) caso o processo de controle garanta a operação da máquina na condição de alinhamento;
4. o controlador da figura 6.20 está baseado na técnica de controle vetorial, sendo que, tanto no método indireto como no método por realimentação de estado, é utilizado o conhecimento da dinâmica inversa da máquina na geração dos comandos de controle. Portanto, caso o processo de estimação de parâmetros seja bem sucedido, a condição de desacoplamento entre as variáveis de controle é sempre mantida;
5. o inversor fonte de corrente utilizado para alimentar a máquina é considerado ideal;

Com base nestas hipóteses, as seguintes expressões passam a ser válidas:

- fluxo de referência constante:

$$L_m = L_m^* \tag{6.20}$$

- resistência de rotor variável (expressa em termos do parâmetro α):

$$\tau_r = \alpha\tau_r^* \Rightarrow (L_m^*+L_{lr})/R_r = \alpha(L_m^*+L_{lr})/R_r^* \Rightarrow R_r^* = \alpha R_r \tag{6.21}$$

- com a hipótese de alinhamento entre eixo direto do sistema de referência girante e o vetor fluxo de rotor, o fluxo nominal de referência (Fnom) pode ser expresso na forma:

$$Fnom = \sqrt{\frac{2}{3}\frac{2}{P}R_r^*\frac{TEnom}{WSnom}} \tag{6.22}$$

- a velocidade de escorregamento dada pelo método indireto de controle vetorial pode ser descrita em termos dos comandos de controle na forma:

$$w_{slip}^* = \frac{2}{3}\frac{2}{P}R_r\frac{T_e^*}{Fnom^2} \tag{6.23}$$

Substituindo as equações (6.21) e (6.22) na equação (6.23), obtém-se:

$$w_{slip}^{*} = \frac{1}{\alpha} WSnom \frac{T_e^{*}}{TEnom} \tag{6.24}$$

Multiplicando a equação (6.24) por τ_r^{*} e τ_r, obtém-se:

$$w_{slip}^{*} \tau_r^{*} = \frac{1}{\alpha} K \frac{T_e^{*}}{TEnom} \tag{6.25}$$

$$w_{slip}^{*} \tau_r = K \frac{T_e^{*}}{TEnom} \tag{6.26}$$

com $K = \tau_r^{*} . WSnom$.

Utilizando as equações (6.20), (6.21), (6.25) e (6.26), é possível reescrever a equação (6.19) na forma:

$$\frac{T_e}{T_e^{*}} = \frac{1}{\alpha} \frac{\alpha^2 + \left(K \frac{T_e^{*}}{TEnom} \right)^2}{1 + \left(K \frac{T_e^{*}}{TEnom} \right)^2} \tag{6.27}$$

Substituindo na equação (6.27) os valores de α (representando a variação da constante de tempo do rotor) e de T_e^{*} (representando o comando de conjugado), é possível obter o valor da razão T_e/T_e^{*}, representando uma comparação entre o conjugado efetivamente produzido na máquina e o conjugado comandado. No entanto, o mapeamento a ser realizado pelo estimador de parâmetros deve fornecer o valor de α e não recebê-lo como informação de entrada. Devido à característica não-linear da equação (6.27), não é possível expressar α em função de T_e^{*} e T_e. Para solucionar este problema, pode-se explorar a capacidade de aproximação universal de redes neurais artificiais (veja capítulo 2 e Apêndice A) na aproximação do mapeamento realizado pela equação (6.27). Por ser um mapeamento estático, a utilização de uma rede neural não-recorrente é recomendável. A figura 6.21 apresenta o esquema utilizado para treinar a rede neural (os parâmetros de treinamento são apresentados no Apêndice F, seção 5)[6].

[6] Observe que T_e^{*} pode ser tomado em módulo, pois só aparece em termos elevados à potência 2.

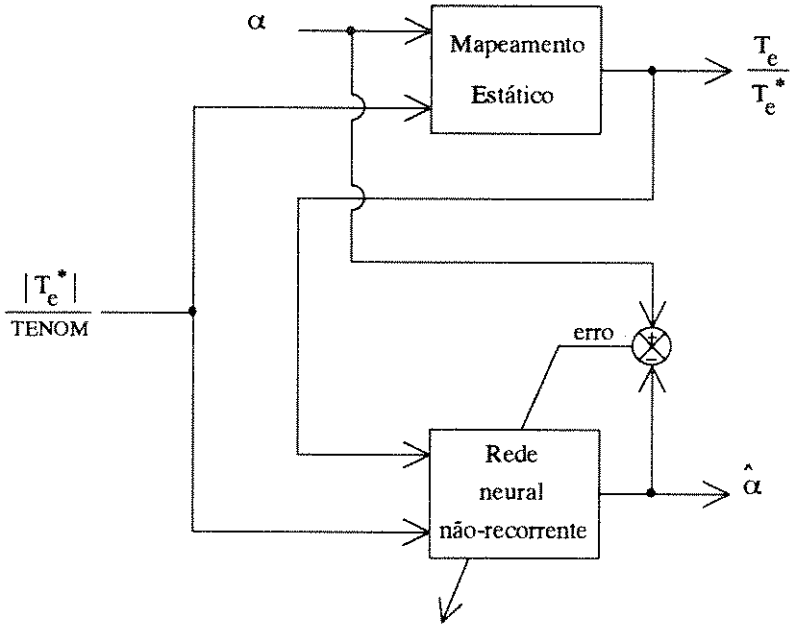


Figura 6.21 - Esquema de treinamento da rede neural

De acordo com a configuração da figura 6.21, a rede neural deve aproximar um mapeamento parcialmente inverso em relação àquele realizado pela equação (6.27). Como discutido no Capítulo 4, por ocasião da apresentação de esquemas de identificação da dinâmica inversa de sistemas, se o mapeamento direto não for bem comportado, o mapeamento inverso não pode ser obtido exatamente. Considere, por exemplo, os dois conjuntos fictícios de pares de entrada-saída do mapeamento direto:

	α	T_e^*		T_e
Conjunto 1	1,5	2	→	3
	0,8	2	→	1,6
Conjunto 2	1,5	2	→	3
	0,8	2	→	3

O conjunto 1 é dito ser bem comportado, pois permite obter o conjunto 1' representando o mapeamento parcialmente inverso na forma:

	T_e	T_e^*		α
Conjunto 1'	3	2	→	1,5
	1,6	2	→	0,8

Já o conjunto 2 é dito ser mal comportado, pois o conjunto 2' representando o seu mapeamento parcialmente inverso é dado por:

	T_e	T_e^*		α
Conjunto	3	2	→	1,5
2'	3	2	→	0,8

ou seja, uma mesma entrada deve ser mapeada em duas saídas distintas.

O que geralmente se obtém no caso do conjunto 2' é um mapeamento aproximado que toma a média entre os dois valores de saída.

6.4.2. O esquema de ajuste de parâmetros

Em situações práticas, a rede neural, uma vez treinada, deve estimar a constante de tempo do rotor durante a operação da máquina submetida ao método de controle vetorial. Sempre que o conjugado efetivamente produzido na máquina não corresponder exatamente ao comando de conjugado, assume-se a existência de variação de parâmetros, no caso, variação da constante de tempo do rotor. Utilizando como entrada da rede neural as razões entre o conjugado comandado e o conjugado nominal e entre o conjugado efetivo e o conjugado comandado, a rede neural fornece como valor de saída a razão entre o valor estimado e o valor até então atribuído à constante de tempo do rotor. Esta razão dada por $\hat{\alpha}$ é imediatamente utilizada para ajustar os parâmetros do controlador, de forma a garantir que seus valores sejam idênticos aos valores dos parâmetros da máquina. O esquema geral de ajuste de parâmetros, válido tanto para o método indireto de determinação da posição do fluxo quanto para o método por realimentação de estado, é apresentado na figura 6.22.

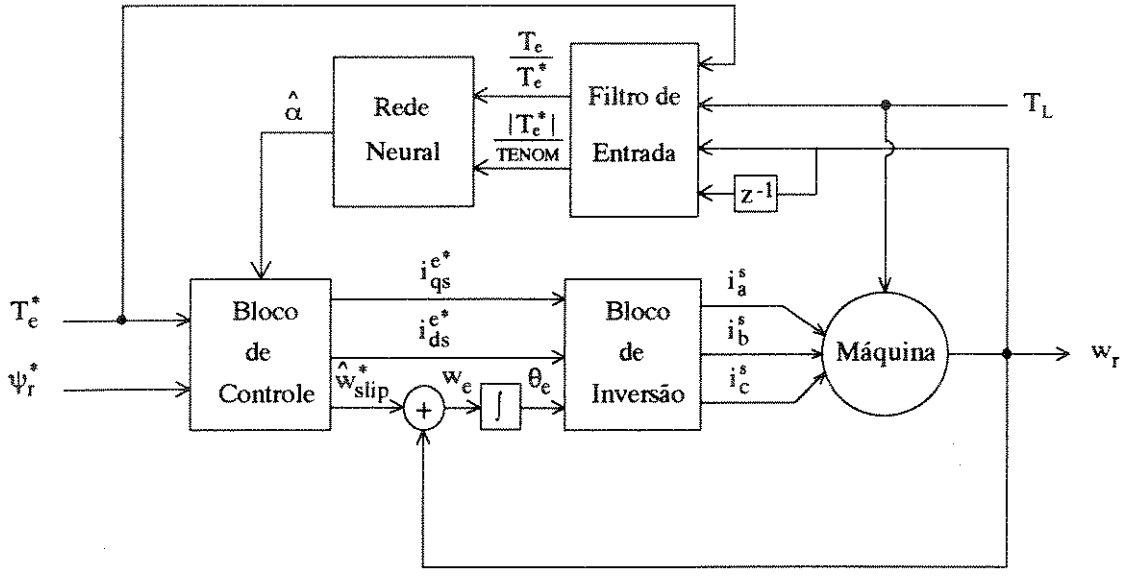


Figura 6.22 - Método de controle adaptativo utilizando redes neurais

O esquema do bloco de controle para os métodos indireto e por realimentação de estado são dados respectivamente pelas figuras 5.6 e 5.8 do Capítulo 5.

Observe que, para determinar o vetor de entrada para a rede neural é necessário conhecer-se o valor do conjugado de carga (T_L), pois discretizando a equação que representa a parte mecânica da máquina (equação (6.3)) com período de amostragem dado pelo intervalo entre duas estimativas de $\hat{\alpha}$ (T_α), obtém-se:

$$w_r(k) = e^{-\frac{K}{J}T_\alpha} w_r(k-1) + \frac{P}{2K}\left(1 - e^{-\frac{K}{J}T_\alpha}\right)\left(T_e(k) - T_L(k)\right)$$

Com isso, é possível obter $T_e(k)$ em função de $T_L(k)$, $w_r(k)$ e $w_r(k\text{-}1)$, na forma:

$$T_e(k) = T_L(k) + \frac{2}{P}\frac{K}{1 - e^{-\frac{K}{J}T_\alpha}}\left[w_r(k) - e^{-\frac{K}{J}T_\alpha} w_r(k-1)\right] \qquad (6.28)$$

O bloco denominado Filtro de Entrada calcula T_e com o auxílio da equação (6.28) e gera as entradas da rede neural a partir do conhecimento de T_e e T_e^*.

6.4.3. Implementação da solução proposta

Dado um instante de tempo discreto k, a atualização da constante de tempo do rotor (τ_r) utilizada pelo controlador é realizada com base no valor de saída da rede neural ($\hat{\alpha}$) na forma:

$$\tau_r(k) = \hat{\alpha}(k).\tau_r(k-1) \qquad (6.29)$$

Logo, a partir da constante de tempo inicial $\tau_r(0)$, o valor da constante de tempo no n-ésimo instante de tempo é dado por:

$$\tau_r(n) = \left[\prod_{i=1}^{n} \hat{\alpha}(i)\right].\tau_r(0) \qquad (6.30)$$

donde se conclui que os resultados do processo de estimação vão participando de forma cumulativa na determinação da constante de tempo do rotor.

Com o objetivo de tornar o mapeamento realizado pelo estimador de parâmetros baseado em redes neurais independente do período de tempo fixado para a geração de estimativas (T_α), adotou-se um passo adicional no processo de apresentação dos sinais de entrada da rede neural e obtenção do sinal de saída, conforme apresentado na figura 6.23.

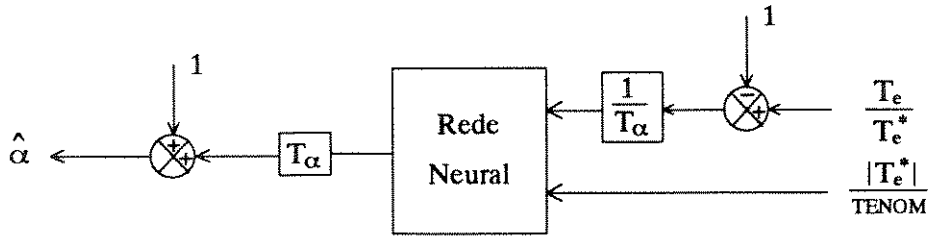


Figura 6.23 - Normalização das entradas e saídas da rede neural

Com base no esquema de treinamento da figura 6.21 e introduzindo a normalização da figura 6.23, foram gerados os padrões para treinamento, que são apresentados na figura 6.24, juntamente com o mapeamento a ser aproximado pela rede neural.

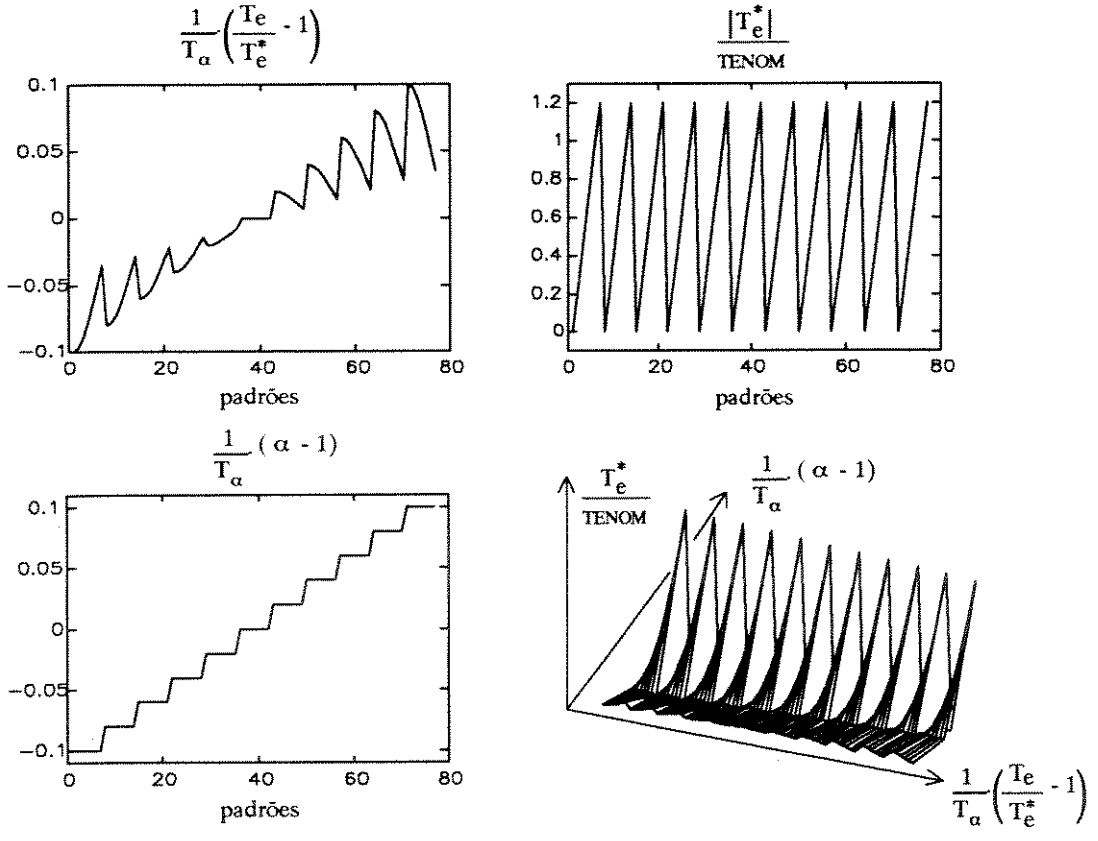
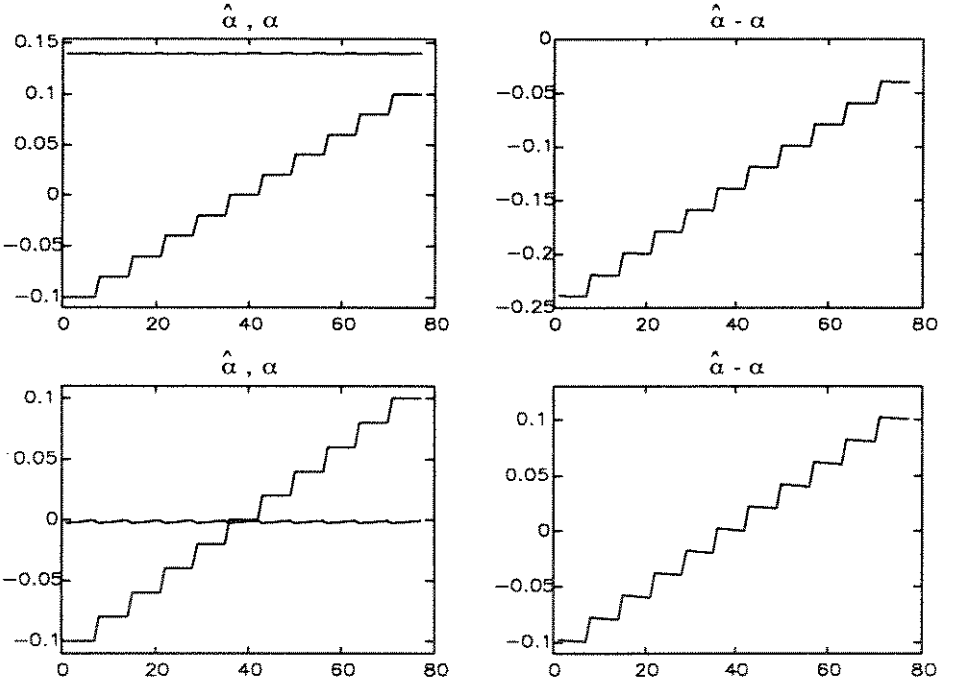


Figura 6.24 - Padrões de treinamento para a rede neural

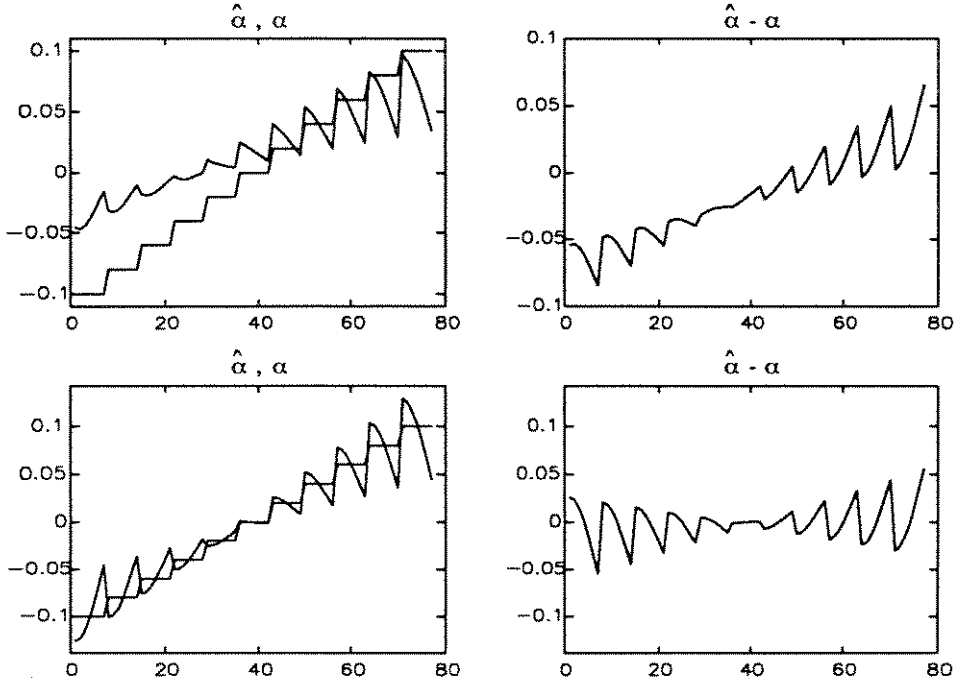
Apesar dos gráficos aparecerem como curvas contínuas, apenas 80 pontos igualmente espaçados foram utilizados como padrões de treinamento, de forma a abranger toda a região de mapeamento considerada. As amplitudes de $(1/T_\alpha).(T_e/T_e^*-1)$ e $(1/T_\alpha).(\alpha-1)$ representam o intervalo de variação considerado admissível entre duas estimativas consecutivas. Já no caso de $|T_e^*|$/TEnom, o intervalo considerado admite comandos de conjugado que assumam valores até 20% acima do conjugado nominal (TEnom).

6.4.4. A evolução do treinamento da rede neural

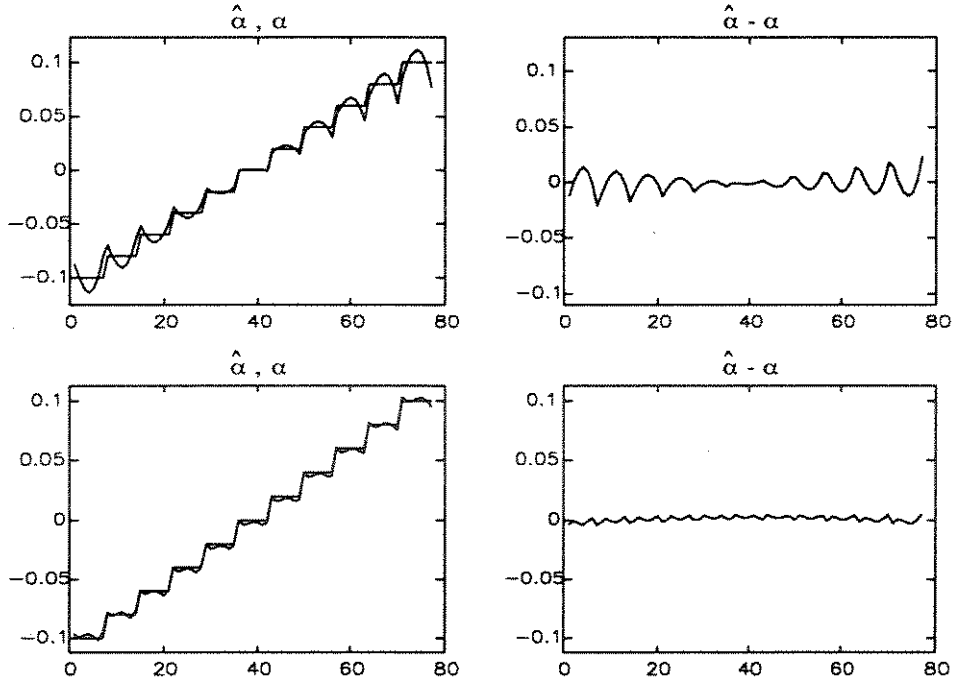
A seguir, é apresentada uma seqüência de resultados referentes à evolução do processo de treinamento da rede neural, implementado a partir do esquema da figura 6.21 (os parâmetros de treinamento são apresentados no Apêndice F, seção 5). Utilizando o conjunto de 80 padrões de treinamento, é desejado que a saída da rede neural aproxime os valores $(1/T_\alpha).(\alpha\text{-}1)$ apresentados na figura 6.24. Na figura 6.25, os dois gráficos de cada linha representam uma fase na evolução do treinamento, sendo que nos gráficos da 1ª coluna são mostrados os padrões de saída desejados (α) e a correspondente saída da rede neural ($\hat{\alpha}$), enquanto que nos gráficos da 2ª coluna é mostrada a diferença entre a saída da rede neural e a saída desejada.



(a) Condição inicial e anulação da média do erro



(b) Anulação da inclinação do erro



(c) Redução gradativa da variância do erro

Figura 6.25 - Evolução do treinamento da rede neural a ser utilizada no processo de estimação do parâmetro α da máquina de indução

Os gráficos mostram que a evolução do treinamento pode ser divida nas seguintes etapas[7]:

[7] O comportamento da evolução do treinamento depende do tipo de mapeamento a ser aproximado.

Etapa 1. Condição inicial: a rede neural filtra qualquer tipo de sinal de entrada, apresentando um nível de sinal de saída praticamente constante.

Etapa 2. Deslocamento da média do erro para zero;

Etapa 3. A curva de erro ganha inclinação zero;

Etapa 4. A variância do erro cai gradativamente.

6.5. A introdução do estimador de parâmetros baseado em redes neurais na malha de controle vetorial

Utilizando o esquema de controle adaptativo da figura 6.22, para efeito de simulação computacional, foram implementadas as técnicas de controle vetorial baseadas no método indireto e no método por realimentação de estado, utilizando o estimador de parâmetros baseado na rede neural cujo nível de treinamento foi apresentado na seção anterior.

As simulações computacionais a serem apresentadas procuram explorar boa parte dos pontos de operação admissíveis para a máquina de indução. As simulações consideram o comando de fluxo constante no valor nominal e vão diferir nos perfis escolhidos para o comando de conjugado eletromagnético e para a variação no parâmetro α. A figura 6.26, a seguir, apresenta os perfis a serem utilizados, definidos de forma a explorar condições de variação suaves e abruptas, abrangendo todo o intervalo típico de variação para α e valores de comando de conjugado até 20% acima do conjugado nominal (tanto em acelereção como frenagem).

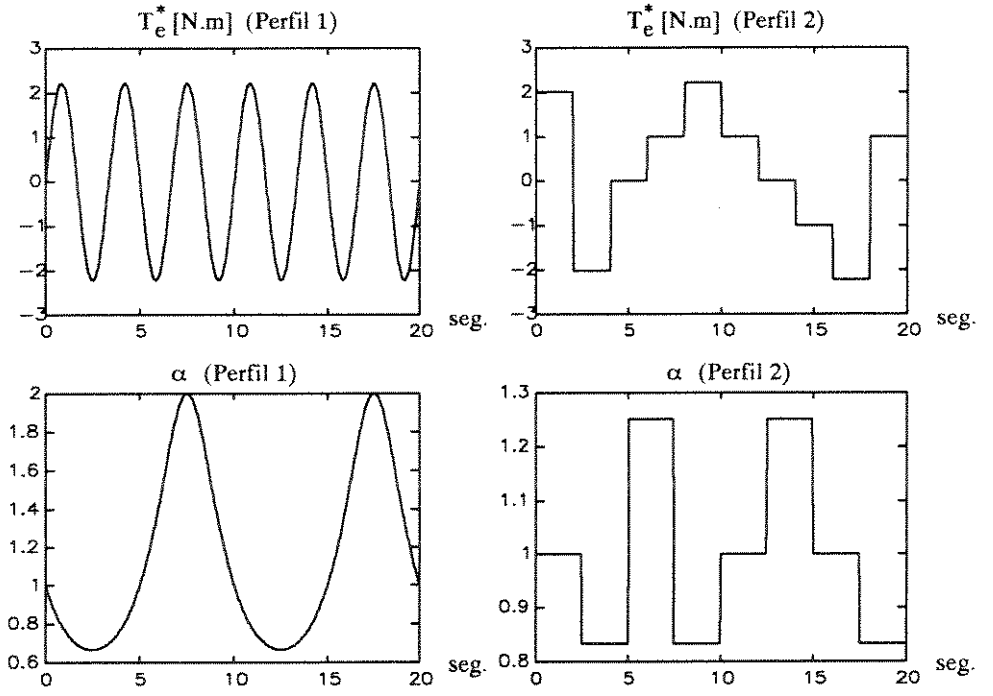


Figura 6.26 - Perfis de comando de conjugado eletromagnético e variação do parâmetro α a serem utilizados nas simulações

O objetivo das simulações é testar o desempenho do estimador de parâmetros baseado em redes neurais na estimação do parâmetro α para as quatro combinações possíveis entre os perfis apresentados na figura 6.26, e também analisar o desempenho dos métodos indireto e por realimentação de estado.

Os resultados de cada simulação serão apresentados em uma figura contendo os seguintes gráficos:

T_e , T_e^*: curvas do comando de conjugado T_e^* e do conjugado efetivamente produzido na máquina T_e;

T_e-T_e^*: diferença entre o comando de conjugado T_e^* e o conjugado efetivamente produzido na máquina T_e (permite verificar a eficiência do controlador no atendimento ao comando de conjugado);

α , $\hat{\alpha}$: curvas do valor real do parâmetro α e do valor estimado $\hat{\alpha}$;

α-$\hat{\alpha}$: diferença entre o valor real do parâmetro α e o valor estimado $\hat{\alpha}$ (permite verificar a eficiência do estimador);

ψ^e_{qr}-ψ^{e*}_{qr}: diferença entre o comando de fluxo em quadratura ψ^{e*}_{qr} e o fluxo em quadratura efetivamente produzido na máquina ψ^e_{qr} (permite verificar a condição de alinhamento);

ψ^e_{dr}-ψ^{e*}_{dr}: diferença entre o comando de fluxo de eixo direto ψ^{e*}_{dr} e o fluxo de eixo direto efetivamente produzido na máquina ψ^e_{dr} (permite verificar a eficiência do controlador no atendimento ao comando de fluxo);

6.5.1. Resultados de simulação para o método indireto

Utilizando-se o método indireto de determinação da posição do fluxo de rotor, o bloco de controle presente no esquema de controle adaptativo da figura 6.22 é implementado conforme apresentado na figura 5.6 (Capítulo 5). Como já verificado nas simulações cujos resultados são apresentados nas figuras 6.5, 6.6 e 6.7, caso o estimador de parâmetros seja capaz de detectar corretamente possíveis variações no parâmetro α, os sinais de controle devem ser capazes de manter a condição de alinhamento entre o eixo direto do sistema de referência e o vetor fluxo de rotor e de conduzir a máquina ao atendimento dos comandos de fluxo e conjugado.

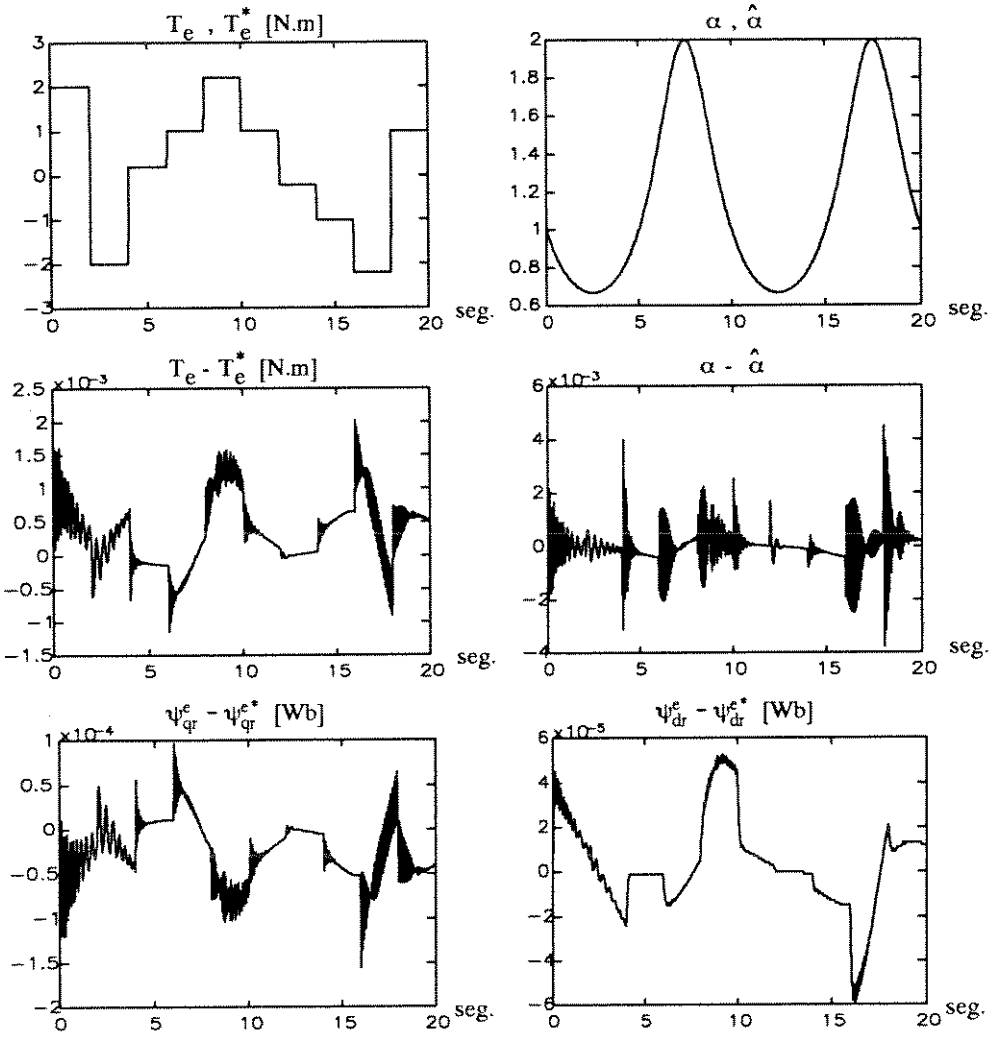


Figura 6.27 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação em degrau e uma variação contínua do parâmetro α da máquina

Os resultados de simulação apresentados na figura 6.27 permitem verificar que o estimador de parâmetros é capaz de ajustar o parâmetro $\hat{\alpha}$ do controlador de forma que este acompanhe a variação do parâmetro α com um erro máximo da ordem de 10^{-3}. Deste modo, a máquina mantém seu ponto de operação sempre muito próximo da condição de alinhamento, o que possibilita o atendimento dos comandos de fluxo e conjugado com pequenas variações em torno dos valores de referência.

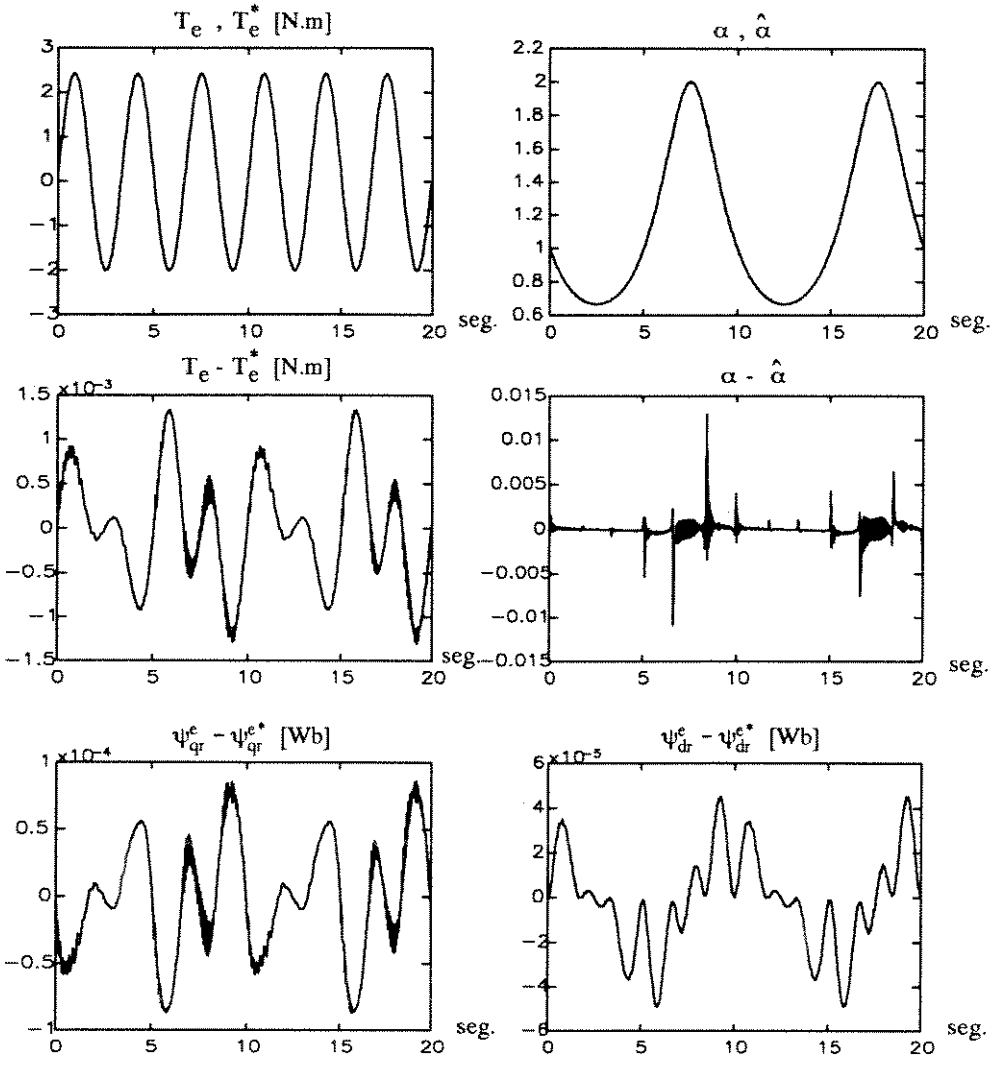


Figura 6.28 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação senoidal e uma variação contínua do parâmetro α da máquina

O resultado de simulação da figura 6.28 é semelhante ao apresentado na figura 6.27, a menos da existência de valores máximos mais elevados para o erro de estimação. Isto pode ser explicado pela variação contínua do ponto de operação da máquina, dificultando a detecção, por parte do estimador, de variações de comportamento da dinâmica da máquina. No caso da figura 6.27, a máquina apresenta pontos de operação constantes por partes, o que facilita a tarefa do estimador.

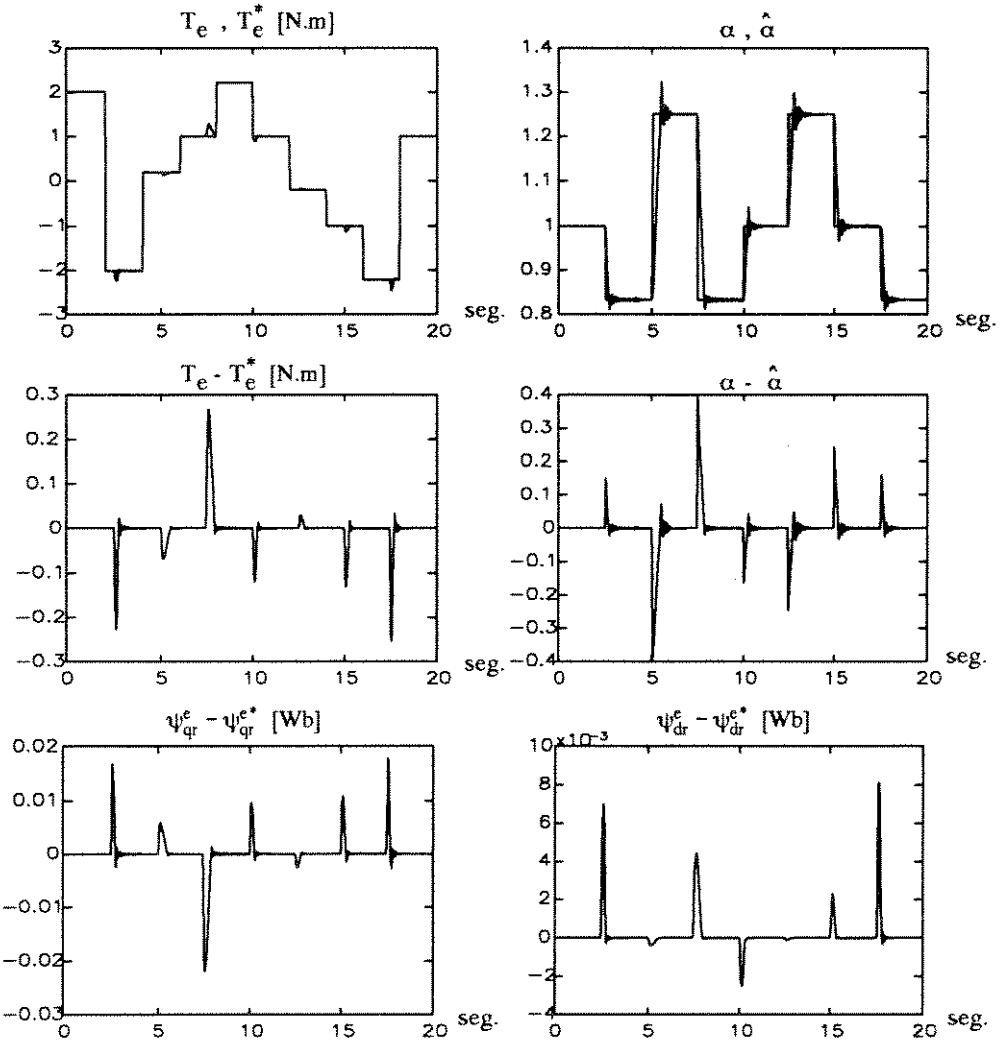


Figura 6.29 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação em degrau e uma variação em degrau do parâmetro α da máquina

Os resultados de simulação apresentados na figura 6.29 apresentam duas propriedades fundamentais do estimador de parâmetros:

1. se o parâmetro α da máquina permanece constante em qualquer valor dentro do intervalo típico de variação (o que ocorre entre dois instantes de variação em degrau), o erro de estimação cai assintoticamente para zero.

2. se o valor atual do parâmetro $\hat{\alpha}$ do controlador estiver muito distante do valor do parâmetro α da máquina (o que ocorre nos instantes de variação em degrau), o estimador é capaz de conduzir $\hat{\alpha}$ para α em poucas iterações.

A primeira propriedade é importante por demonstrar a eficiência do estimador em regime estacionário, enquanto a segunda propriedade permite afirmar que o desempenho do estimador de α não depende da condição inicial.

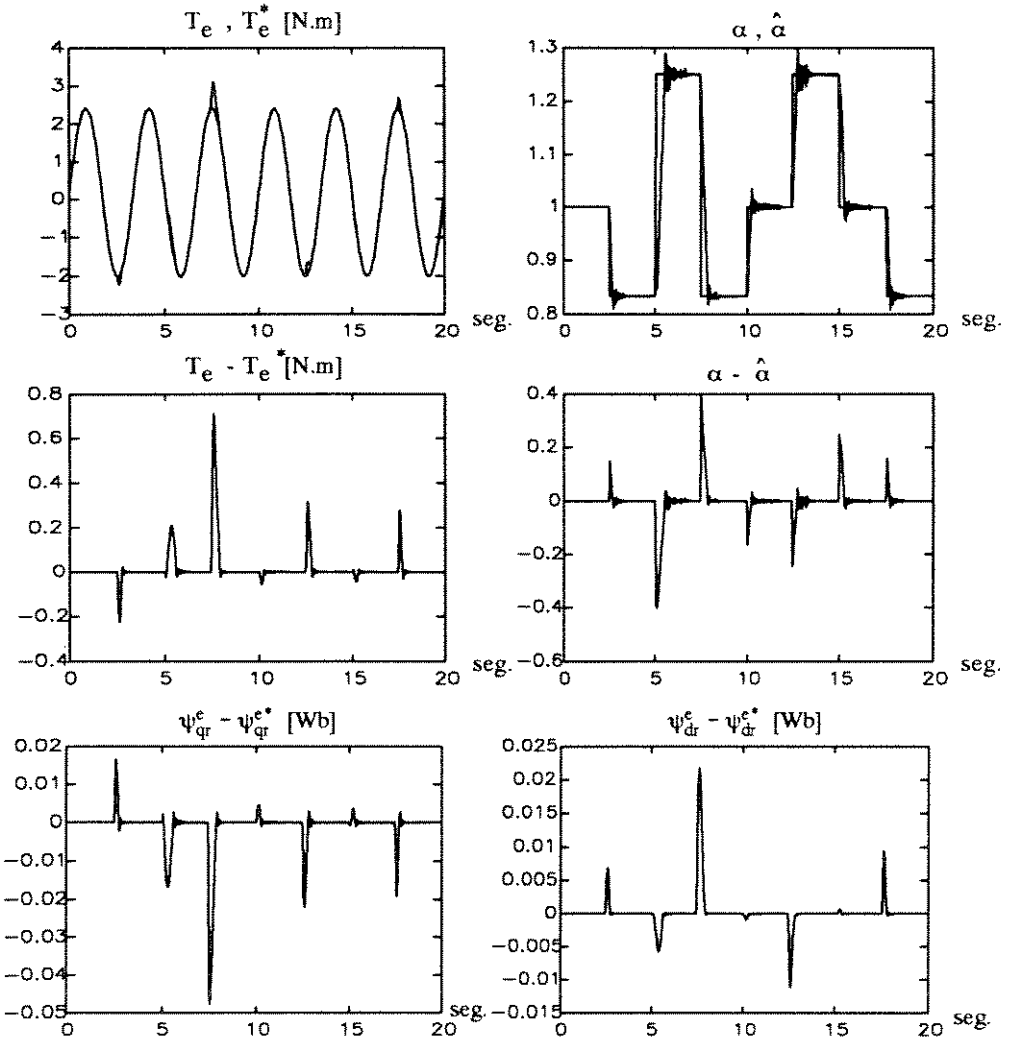


Figura 6.30 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação senoidal e uma variação em degrau do parâmetro α da máquina

Os resultados de simulação apresentados na figura 6.30 endossam as conclusões extraídas a partir dos resultados de simulação das figuras 6.27, 6.28 e 6.29.

Em situações práticas, a resistência do rotor e, conseqüentemente, o parâmetro α apresenta uma constante de tempo de variação muitas vezes maior que aquelas utilizadas nestas simulações. O objetivo da escolha, a nível de simulação, de constantes de tempo bem menores é submeter o estimador de parâmetros a situações garantidamente mais complexas que aquelas a serem encontradas em aplicações práticas.

Na figura 6.31 é apresentada a evolução temporal da saída do estimador de parâmetros. É importante salientar que os valores de α e $\hat{\alpha}$ apresentados nos gráficos das figuras 6.27, 6.28, 6.29 e 6.30 correspondem à relação entre a constante de tempo de rotor nos instantes atual e inicial, enquanto nos gráficos da figura 6.31 correspondem à relação entre a constante de tempo de rotor nos instantes atual e imediatamente anterior, ou seja:

$$\hat{\alpha} = \frac{\tau_r(k)}{\tau_r(0)} \text{ (figuras 6.27, 6.28, 6.29 e 6.30)} \qquad \hat{\alpha} = \frac{\tau_r(k)}{\tau_r(k-1)} \text{ (figura 6.31)}$$

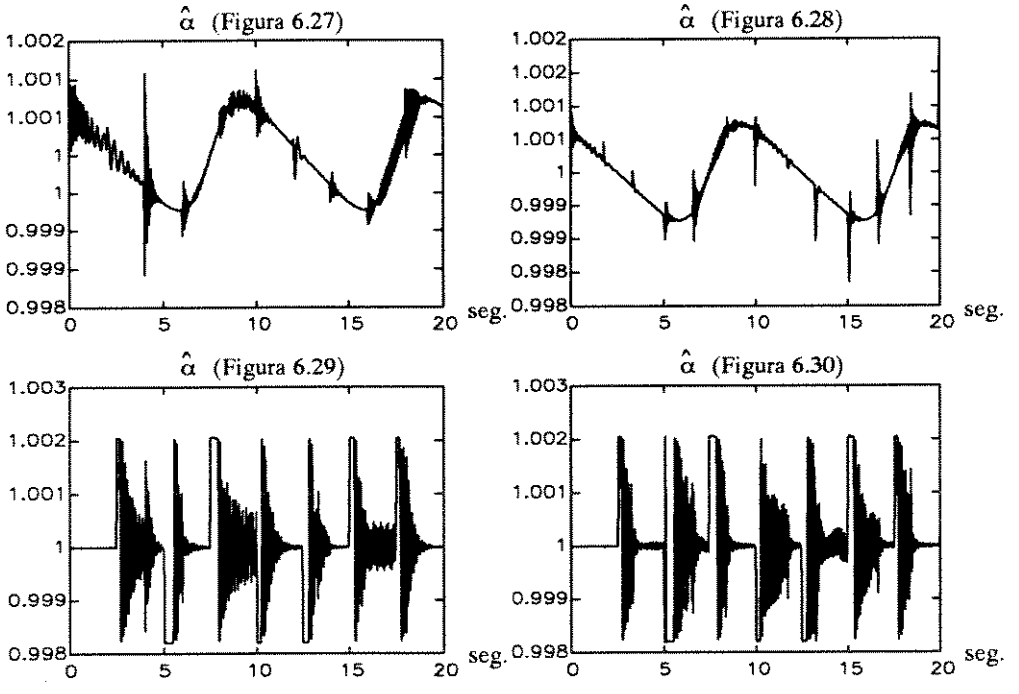


Figura 6.31 - Saída do estimador baseado em redes neurais para as simulações apresentadas nas figuras 6.27, 6.28, 6.29 e 6.30

6.5.2. Resultados de simulação para o método por realimentação de estado

Utilizando-se o método por realimentação de estado na determinação da posição do fluxo de rotor, o bloco de controle presente no esquema de controle adaptativo da figura 6.22 é implementado conforme apresentado na figura 5.8 (Capítulo 5). Como já verificado nas simulações cujos resultados são apresentados nas figuras 6.18 e 6.19, caso o estimador de parâmetros seja capaz de detectar corretamente possíveis variações no parâmetro α, os sinais de controle devem ser capazes de manter a operação da máquina muito próxima da condição de alinhamento entre o eixo direto do sistema de referência e o vetor fluxo de rotor, dessa forma, viabilizando o atendimento dos comandos de fluxo e conjugado.

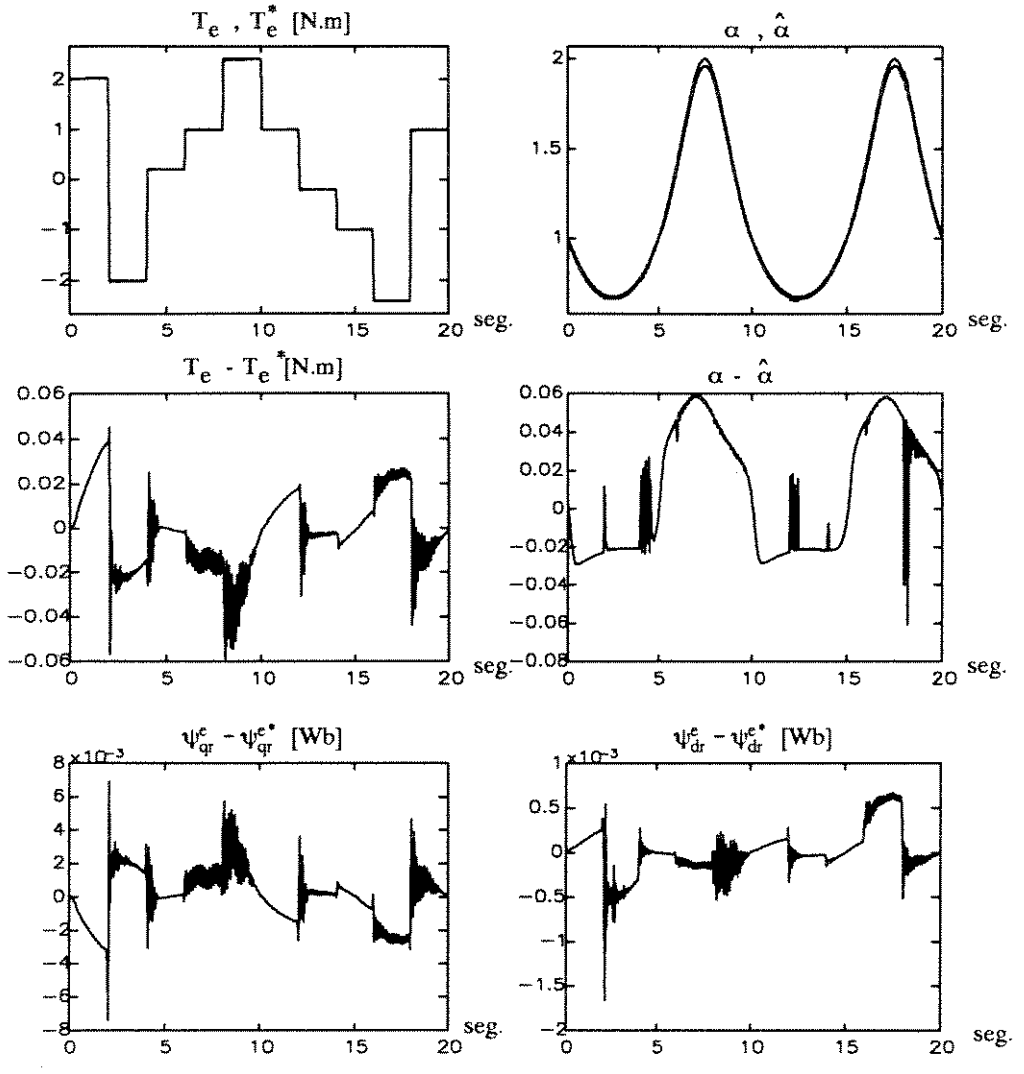


Figura 6.32 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação em degrau e uma variação contínua do parâmetro α da máquina

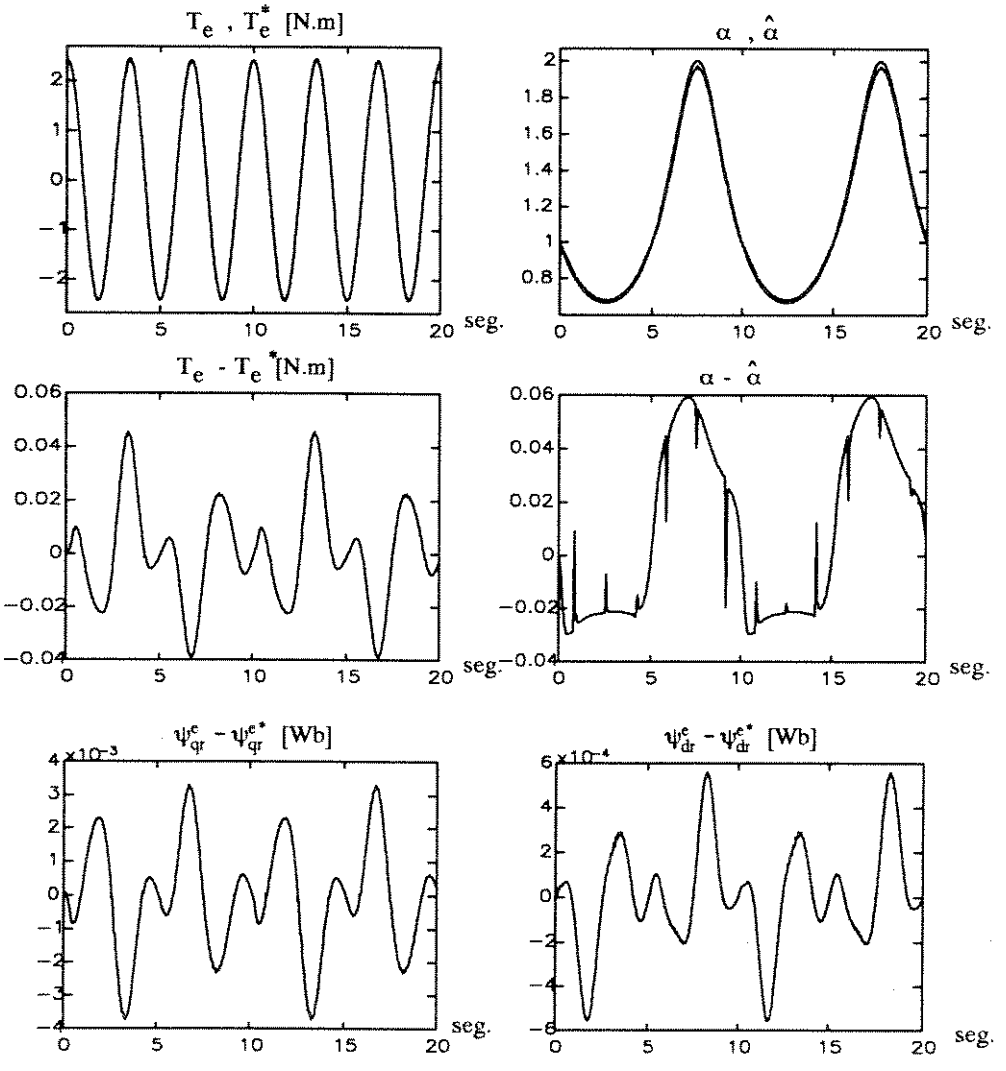


Figura 6.33 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação senoidal e uma variação contínua do parâmetro α da máquina

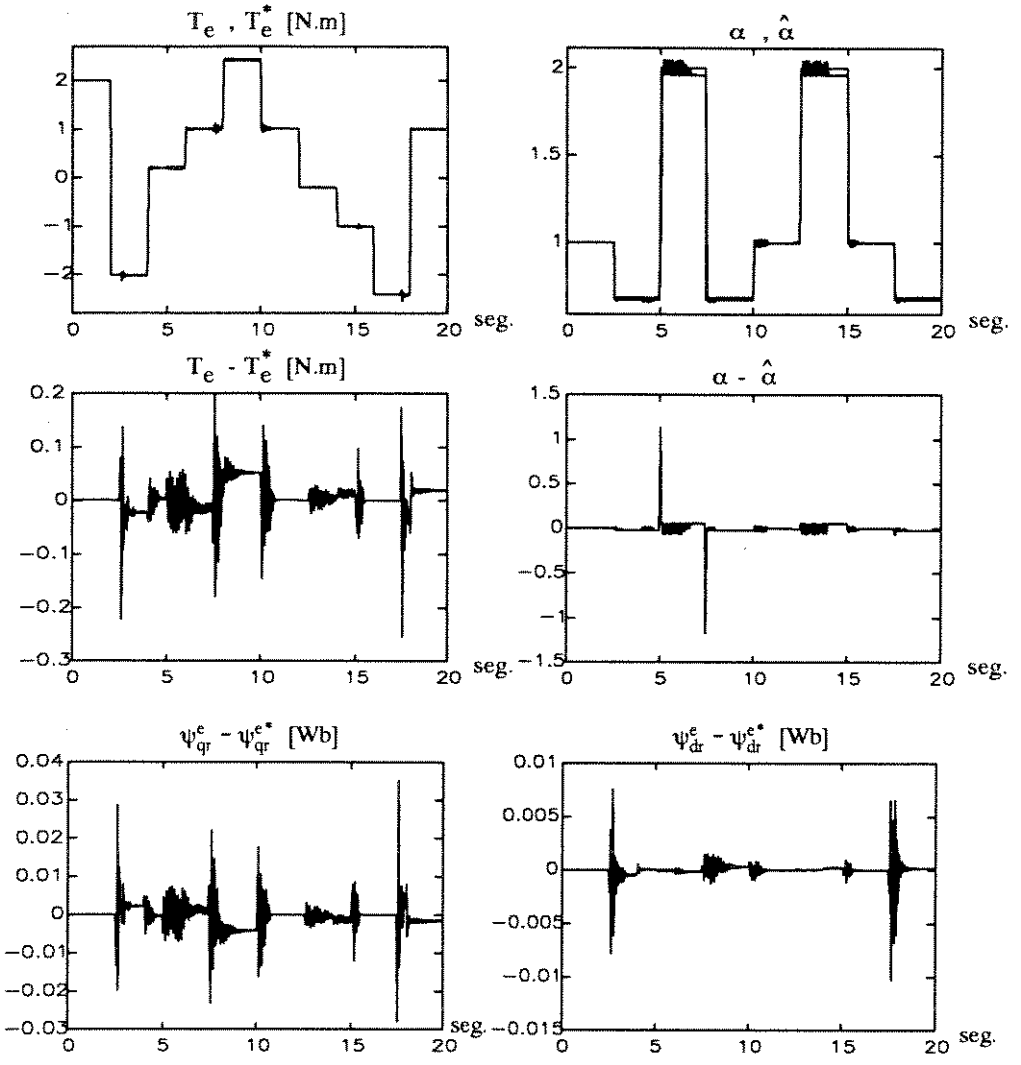


Figura 6.34 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado de variação em degrau e uma variação em degrau do parâmetro α da máquina

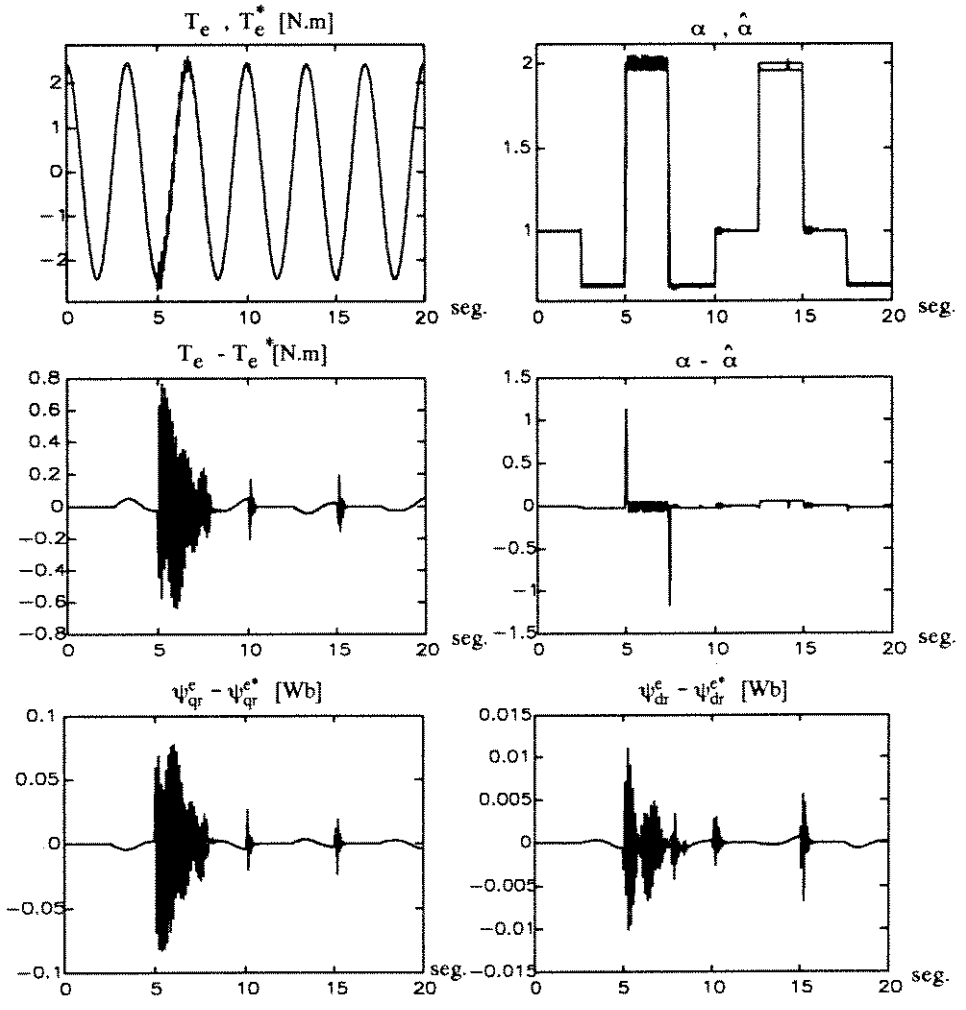


Figura 6.35 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado e um estimador de parâmetros baseado em redes neurais (figura 6.22), para um comando de conjugado T_e^* de variação senoidal e uma variação em degrau do parâmetro α da máquina

As simulações apresentadas nas figuras 6.32, 6.33, 6.34 e 6.35 ilustram a capacidade de aproximação do estimador de parâmetros baseado em redes neurais quando aplicado ao método por realimentação de estado e permitem avaliar o desempenho do processo de controle. As simulações referentes ao método por realimentação de estado apresentam um comportamento semelhante àquele obtido para o método indireto, o que deve conduzir às mesmas conclusões gerais.

Quando comparado com o método indireto, o método por realimentação de estado tem um desempenho inferior. A degradação de desempenho se deve a imprecisões na determinação dos valores das componentes de fluxo de rotor, fornecidas pelo observador de fluxo baseado em rede neurais. Os valores imprecisos conduzem à degradação no desempenho do estimador de parâmetros e, conseqüentemente, no desempenho do controlador.

Na figura 6.36 é apresentada a evolução temporal da saída do estimador de parâmetros para as quatro simulações (valem as mesmas observações consideradas por ocasião da apresentação da figura 6.31).

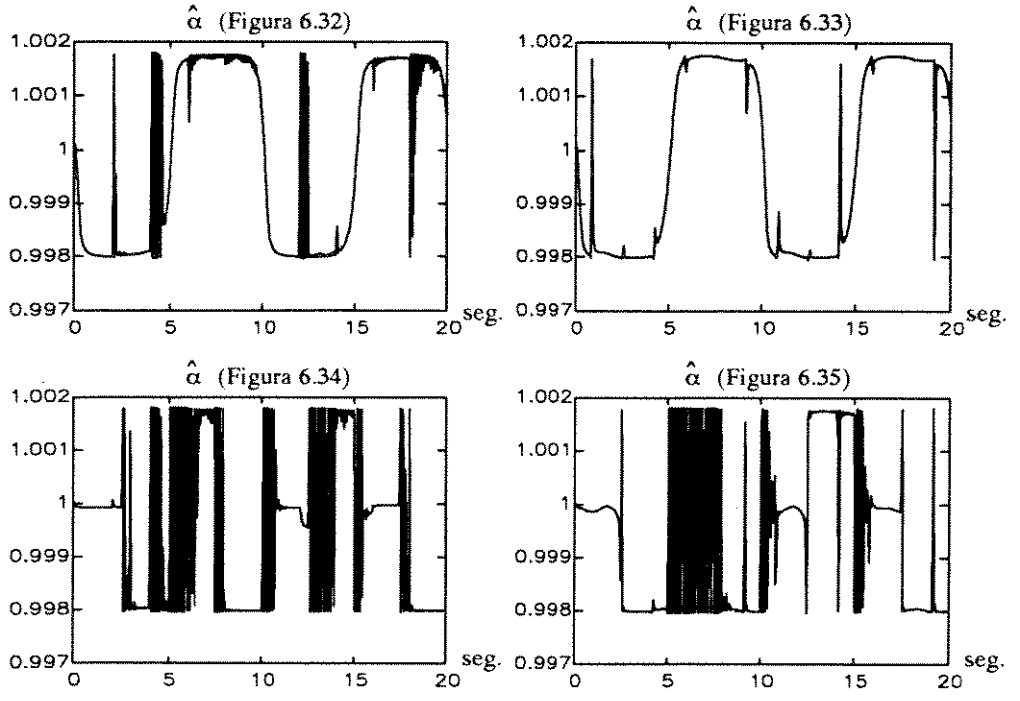


Figura 6.36 - Saída do estimador para as simulações apresentadas nas figuras 6.32, 6.33, 6.34 e 6.35

6.6. Utilização de um observador de fluxo na obtenção do conjugado efetivamente produzido na máquina

De acordo com o esquema de controle adaptativo da figura 6.22, no cálculo do valor do conjugado efetivamente produzido na máquina (T_e), a ser utilizado na definição dos sinais de entrada para o estimador de parâmetros baseado em redes neurais, é necessário o conhecimento do valor do conjugado de carga (T_L). Em aplicações voltadas, por exemplo, para tracionamento de veículos de transporte, o cálculo do conjugado de carga pode ser uma tarefa simples. No entanto, existem aplicações onde o valor do conjugado de carga não pode ser medido. Neste caso, é necessário desenvolver um método alternativo de obtenção do conjugado efetivamente produzido na máquina. Com base na equação (6.2), pode-se concluir que, com o conhecimento do valor das componentes de fluxo de rotor e corrente de estator definidas em um sistema de referência d-q síncrono, é possível obter T_e na forma:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_m + L_{lr}}[\psi^e_{dr} i^e_{qs} - \psi^e_{qr} i^e_{ds}] \qquad (6.31)$$

De fato, de acordo com o resultado de simulação da figura 6.37, os valores obtidos para o conjugado eletromagnético efetivamente produzido na máquina através das equações (6.28) (T_e^{mec}) e (6.31) (T_e^{elet}) são muito próximos entre si e também com relação ao conjugado eletromagnético real (T_e). O exemplo de simulação considera comando de fluxo (ψ_r^*) constante e estimação ($\hat{\alpha}$) ideal do parâmetro α, com a máquina submetida ao método indireto de controle vetorial.

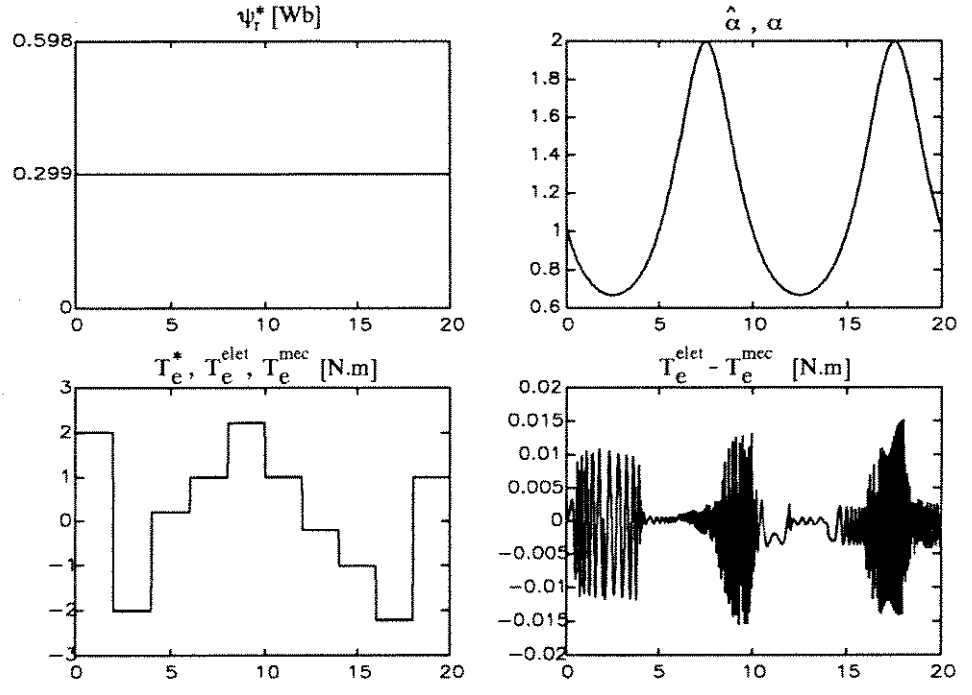


Figura 6.37 - Comparação entre os valores de conjugado eletromagnético produzidos pelas equações (6.28) (T_e^{mec}) e (6.31) (T_e^{elet})

Na equação (6.31), os valores das componentes de corrente de estator estão sempre disponíveis, pois são calculados pelo controlador. No entanto, como discutido no capítulo 5, a impossibilidade de medição direta das componentes de fluxo de rotor requer a utilização de um observador de fluxo. Portanto, utilizando o observador expandido de fluxo apresentado na figura 6.13 e já empregado na implementação do método por realimentção de estado, uma outra alternativa para a obtenção do conjugado efetivamente produzido na máquina pode ser implementada, sendo que o esquema de controle adaptativo da figura 6.22 assume a forma:

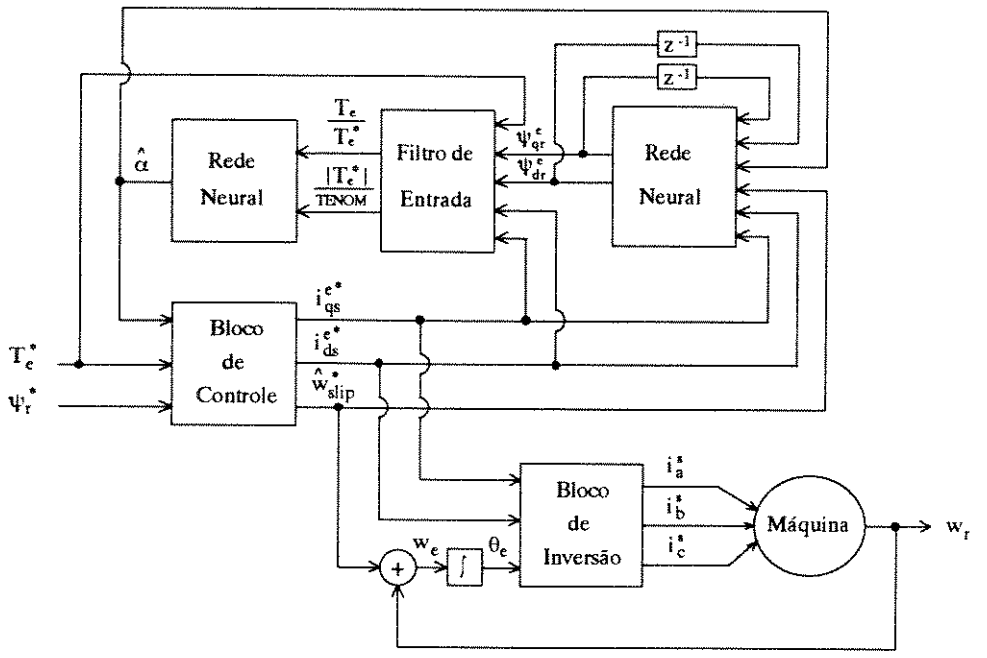


Figura 6.38 - Método de controle adaptativo utilizando redes neurais sem a necessidade do conhecimento do conjugado de carga

Utilizando o esquema de controle adaptativo da figura 6.38, para efeito de simulação computacional, foram implementadas as técnicas de controle vetorial baseadas no método indireto e no método por realimentação de estado, utilizando o estimador de parâmetros baseado na rede neural da figura 6.21 e o observador expandido de fluxo baseado na rede neural da figura 6.13. É importante observar que no caso do método por realimentação de estado, o mesmo observador de fluxo é utilizado para gerar as entradas do estimador e para gerar os comandos de controle.

As simulações computacionais a serem apresentadas procuram explorar boa parte dos pontos de operação admissíveis para a máquina de indução. As simulações consideram o comando de fluxo constante no valor nominal e vão diferir nos perfis escolhidos para o comando de conjugado eletromagnético e para a variação no parâmetro α. A figura 6.26, apresenta os perfis a serem utilizados, definidos de forma a explorar condições de variação suaves e abruptas, abrangendo todo o intervalo típico de variação para α e valores de comando de conjugado até 20% acima do conjugado nominal (tanto em aceleração como frenagem).

O objetivo destas simulações é testar o desempenho do estimador de parâmetros baseado em redes neurais na estimação do parâmetro α para as quatro combinações possíveis entre os perfis apresentados na figura 6.26, e também analisar o desempenho dos métodos indireto e por realimentação de estado. A diferença em relação às simulações já apresentadas nas figuras 6.27 a 6.31 (método indireto) e 6.32 a 6.36 (método por realimentação de estado) está na não utilização do valor do conjugado de carga (T_l) na geração do vetor de entrada do estimador.

6.6.1. Resultados de simulação para o método indireto

Utilizando-se o método indireto de determinação da posição do fluxo de rotor, o bloco de controle presente no esquema de controle adaptativo da figura 6.38 é implementado conforme apresentado na figura 5.6 (Capítulo 5).

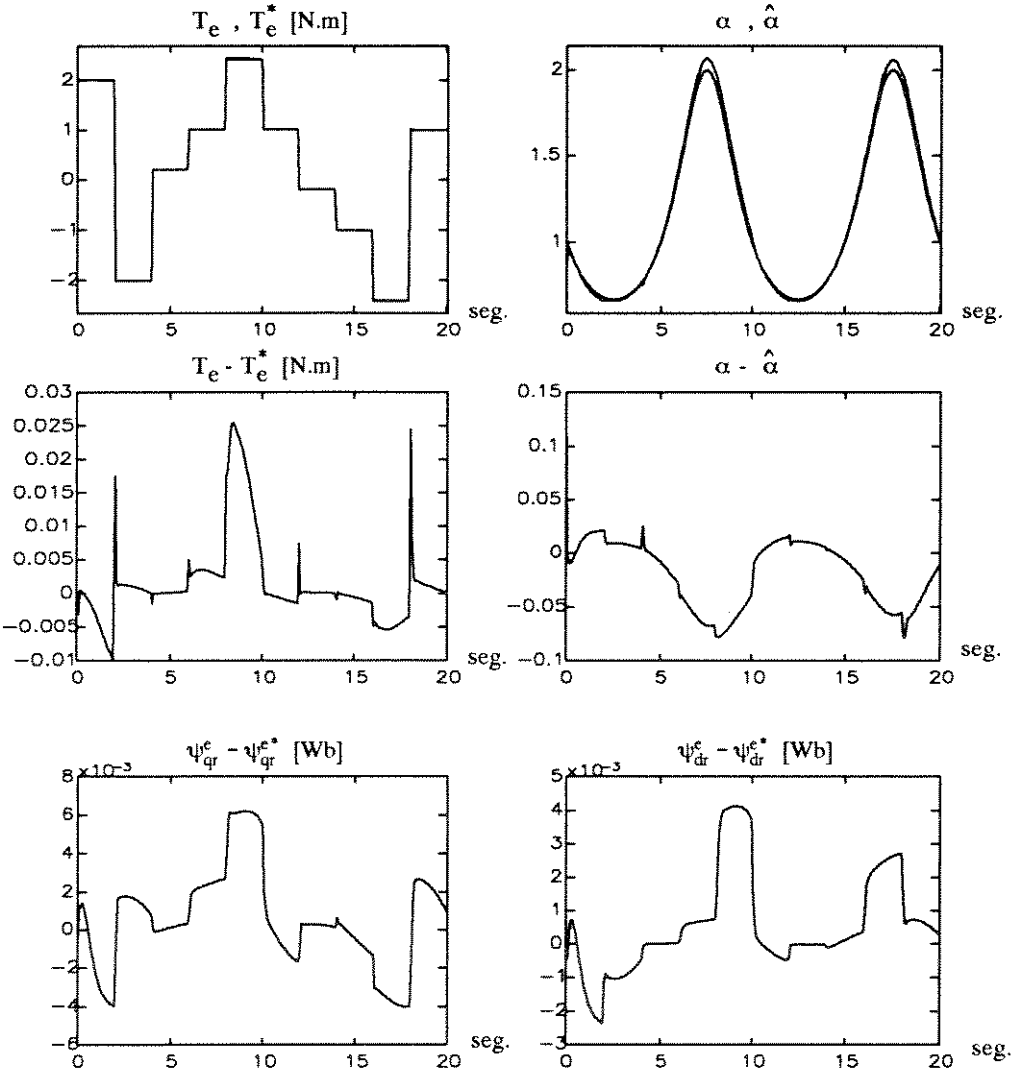


Figura 6.39 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação em degrau e variação contínua do parâmetro α da máquina

Os resultados de simulação apresentados na figura 6.39 permitem verificar que o estimador de parâmetros é capaz de ajustar o parâmetro $\hat{\alpha}$ do controlador de forma que este acompanhe a variação do parâmetro α com um erro máximo da ordem de 10^{-2}. Deste modo, a máquina mantém seu ponto de operação sempre muito próximo da condição de alinhamento, o que possibilita o atendimento dos comandos de fluxo e conjugado com pequenas variações em torno dos valores de referência.

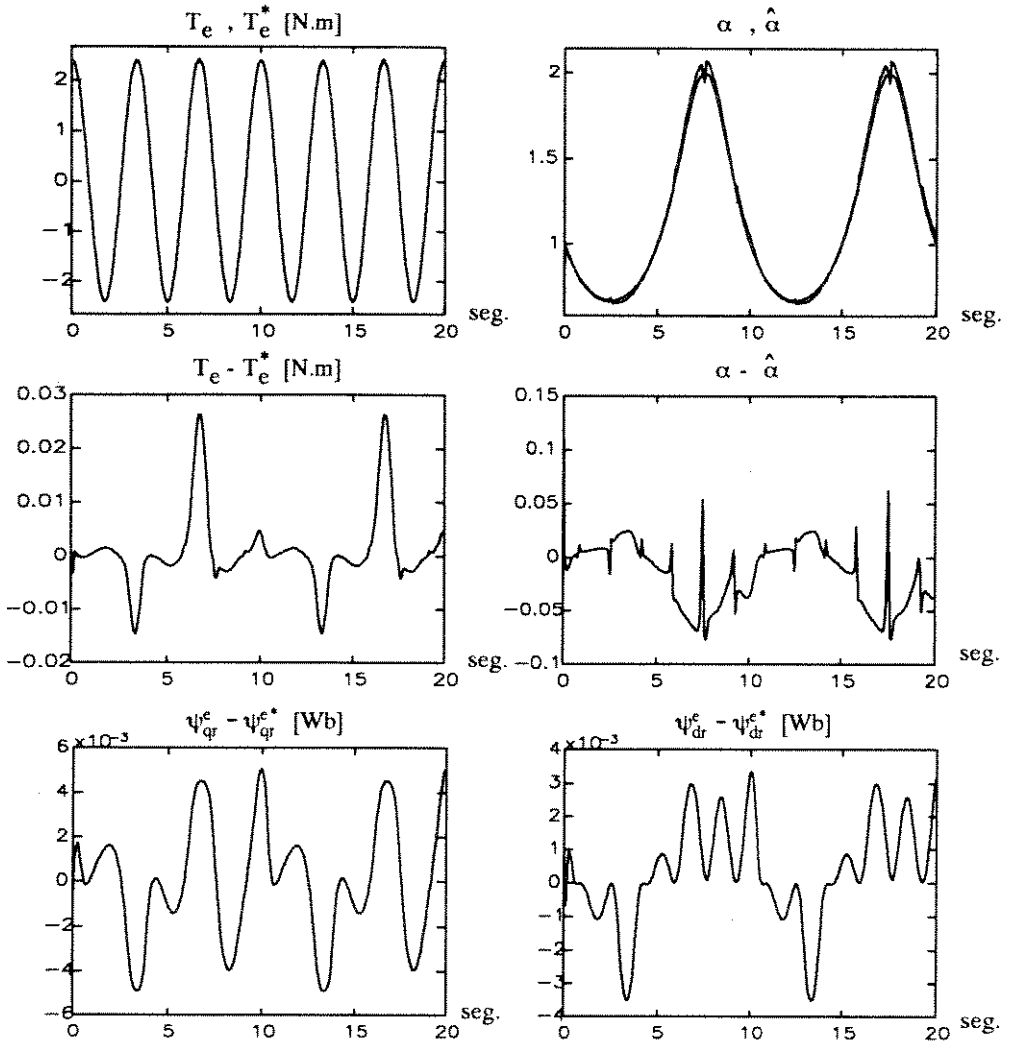


Figura 6.40 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação senoidal e variação contínua do parâmetro α da máquina

A simulação da figura 6.40 apresenta resultados semelhantes àqueles obtidos para a simulação da figura 6.39. A diferença básica entre estas duas simulações está no fato de que, no caso da figura 6.39, o ponto de operação da máquina é constante por partes, enquanto que, no caso da figura 6.40, o ponto de operação da máquina varia continuamente. Apesar das evoluções no tempo do erro de estimação e do erro de atendimento aos comandos de controle apresentarem perfis distintos, os desempenhos dos processos de estimação e de controle para os dois casos são muito semelhantes.

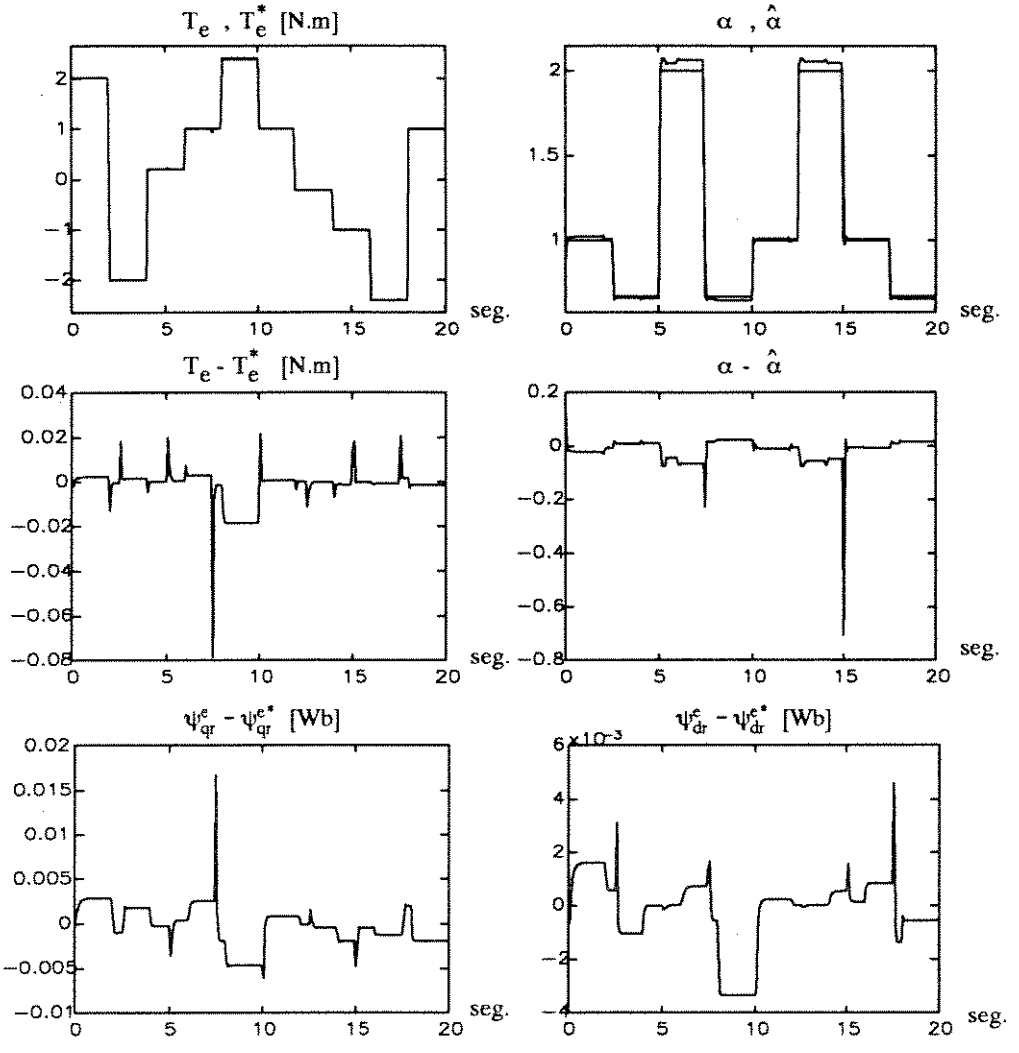


Figura 6.41 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação em degrau e variação em degrau do parâmetro α da máquina

A simulação da figura 6.41 apresenta resultados importantes quanto à eficiência do estimador em regime estacionário e quanto à independência do estimador em relação à condição inicial:

- o estimador apresenta um erro próximo de zero e praticamente constante em regime estacionário (operação no intervalo entre variações abruptas de conjugado);
- o estimador é capaz de responder corretamente a variações abruptas no parâmetro α, situação que provoca a ocorrência momentânea de diferenças expressivas entre os parâmetros α e $\hat{\alpha}$. Com isso, pode-se concluir que uma condição inicial arbitrária para o parâmetro $\hat{\alpha}$ não vai influenciar no desempenho global do estimador.

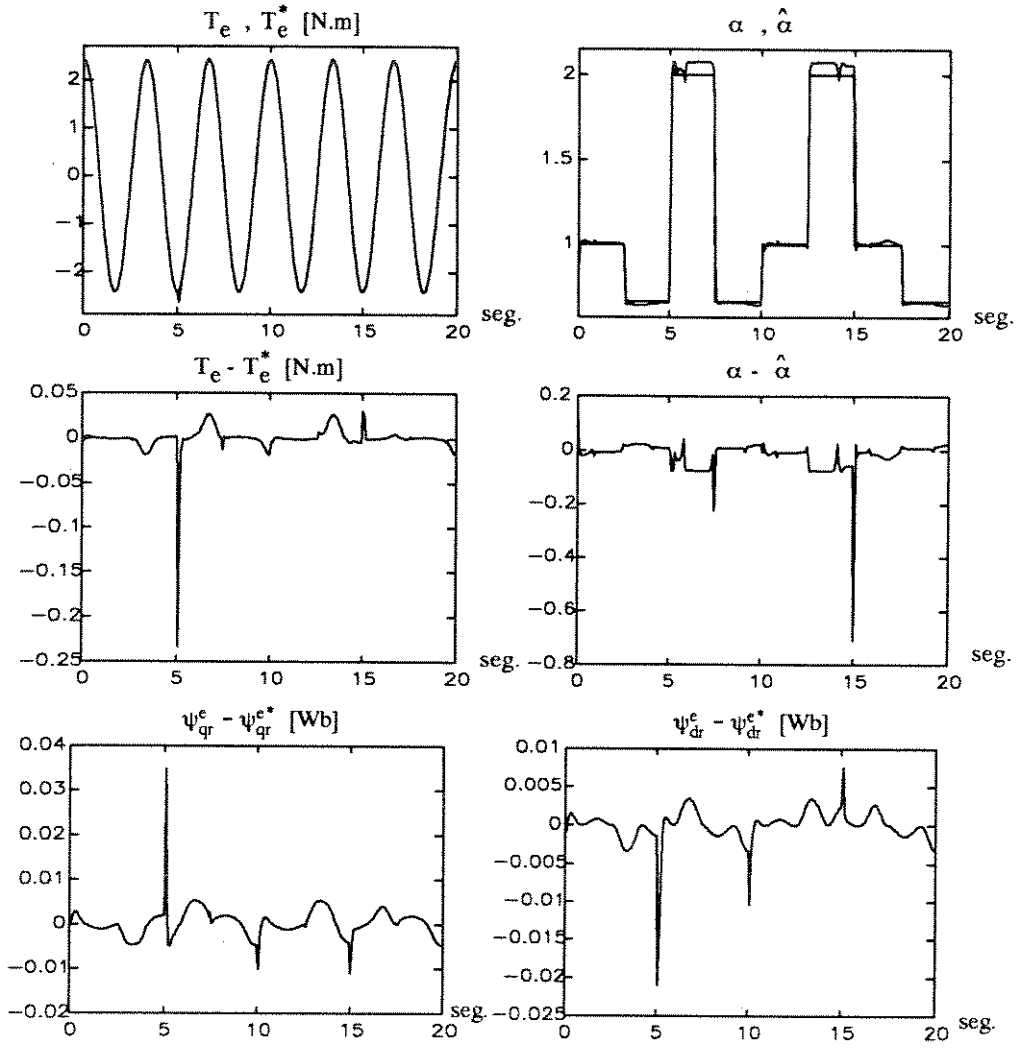


Figura 6.42 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método indireto, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação senoidal e variação em degrau do parâmetro α da máquina

Os resultados de simulação apresentados na figura 6.42 endossam as conclusões extraídas a partir dos resultados de simulação das figuras 6.39, 6.40, 6.41.

Em situações práticas, a resistência do rotor e, conseqüentemente, o parâmetro α apresentam uma constante de tempo de variação muitas vezes maior que aquelas utilizadas nestas simulações. O objetivo da escolha, a nível de simulação, de constantes de tempo bem menores é submeter o estimador de parâmetros a situações garantidamente mais complexas que aquelas a serem encontradas em aplicações práticas.

Na figura 6.43 é apresentada a evolução temporal da saída do estimador de parâmetros.

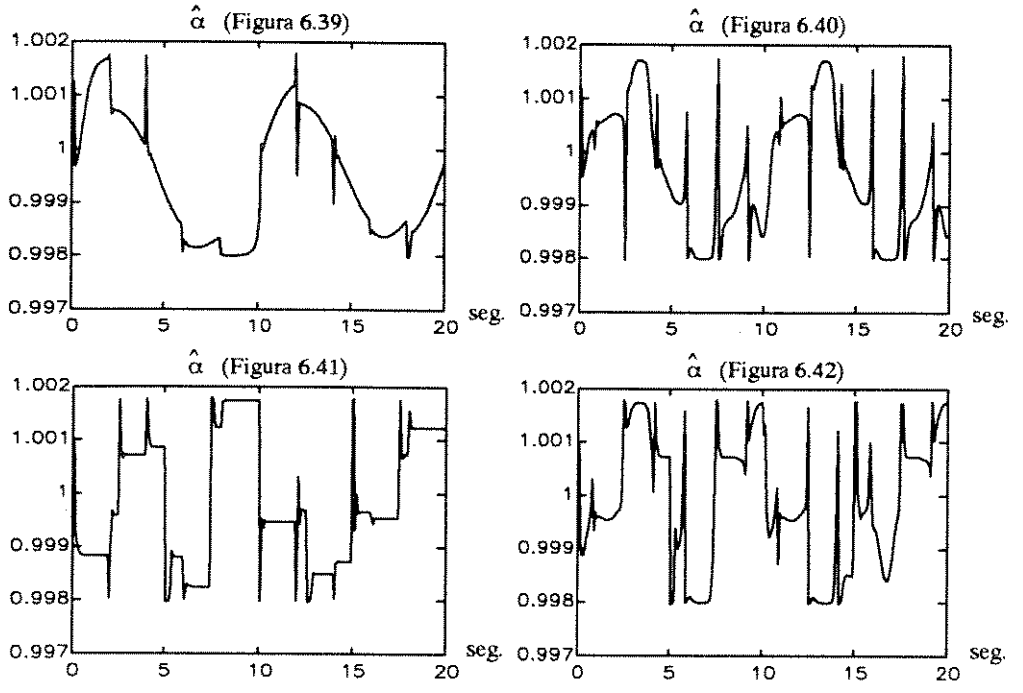


Figura 6.43 - Saída do estimador para as simulações apresentadas nas figuras 6.39, 6.40, 6.41 e 6.42

Quando comparados com os resultados apresentados nas figuras 6.27, 6.28, 6.29 e 6.30 (caracterizados pela utilização do valor do conjugado de carga na definição do vetor de entrada do estimador - figura 6.22), os resultados apresentados nas figuras 6.39, 6.40, 6.41 e 6.42 (caracterizados pela utilização do observador de fluxo na definição do vetor de entrada do estimador - figura 6.38) têm um desempenho inferior, mas ainda bastante satisfatório. A degradação de desempenho se deve a imprecisões na determinação dos valores das componentes de fluxo de rotor, fornecidas pelo observador de fluxo baseado em rede neurais. Os valores imprecisos conduzem à degradação no desempenho do estimador de parâmetros e, conseqüentemente, no desempenho do controlador. No entanto, do ponto de vista de aplicação do método indireto, o esquema de controle da figura 6.38 é geralmente mais interessante, pois não exige o conhecimento do valor do conjugado de carga.

6.6.2. Resultados de simulação para o método por realimentação de estado

Utilizando-se o método por realimentação de estado na determinação da posição do fluxo de rotor, o bloco de controle presente no esquema de controle adaptativo da figura 6.38 é implementado conforme apresentado na figura 5.8 (Capítulo 5).

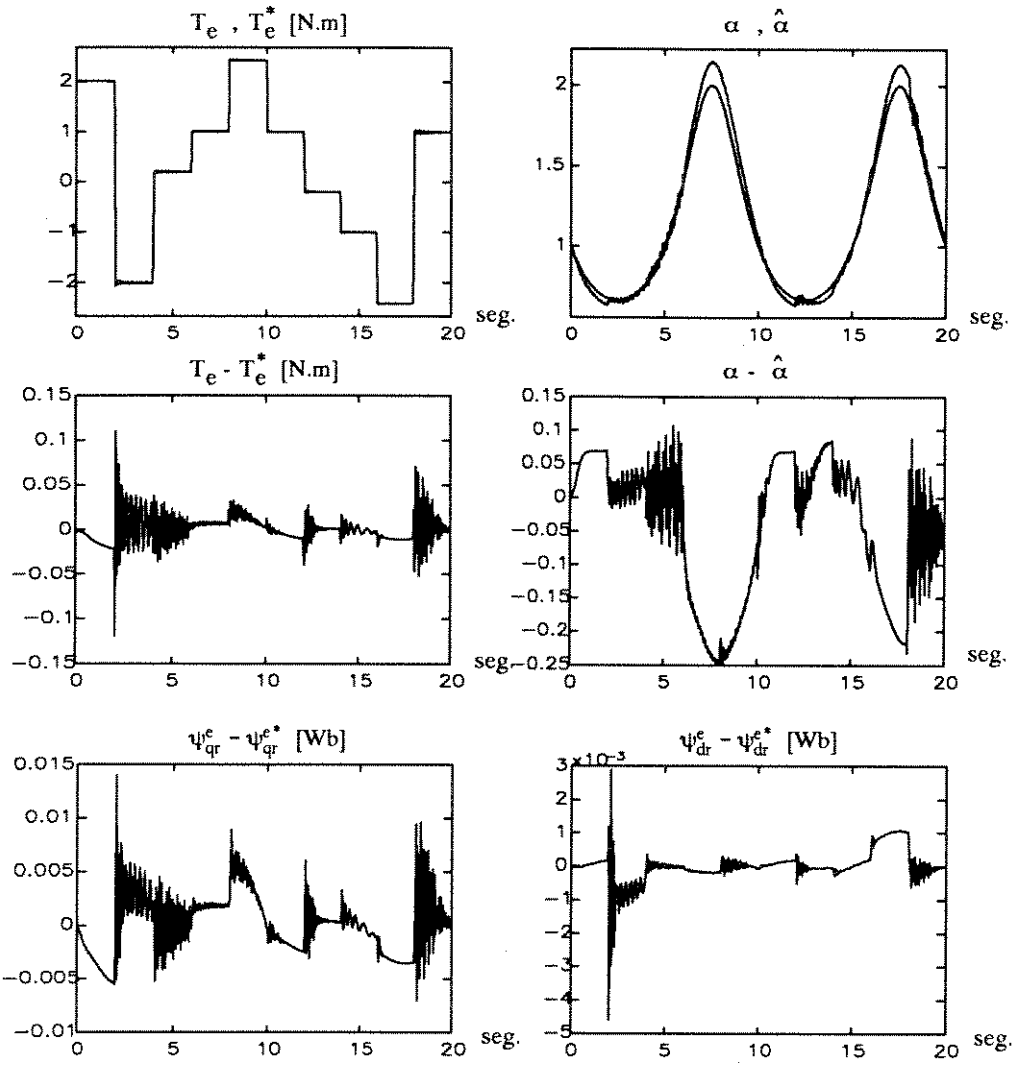


Figura 6.44 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação em degrau e variação contínua do parâmetro α da máquina

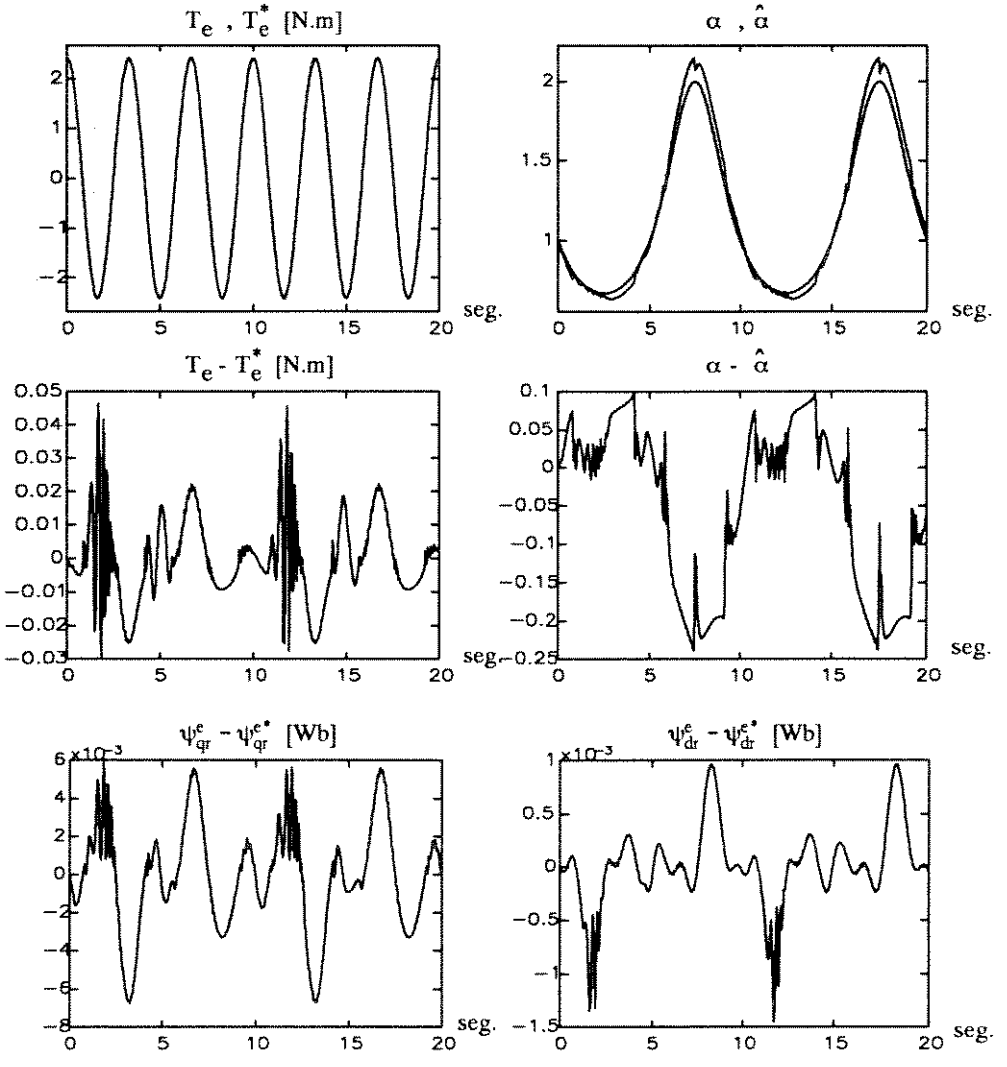


Figura 6.45 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação senoidal e variação contínua do parâmetro α da máquina

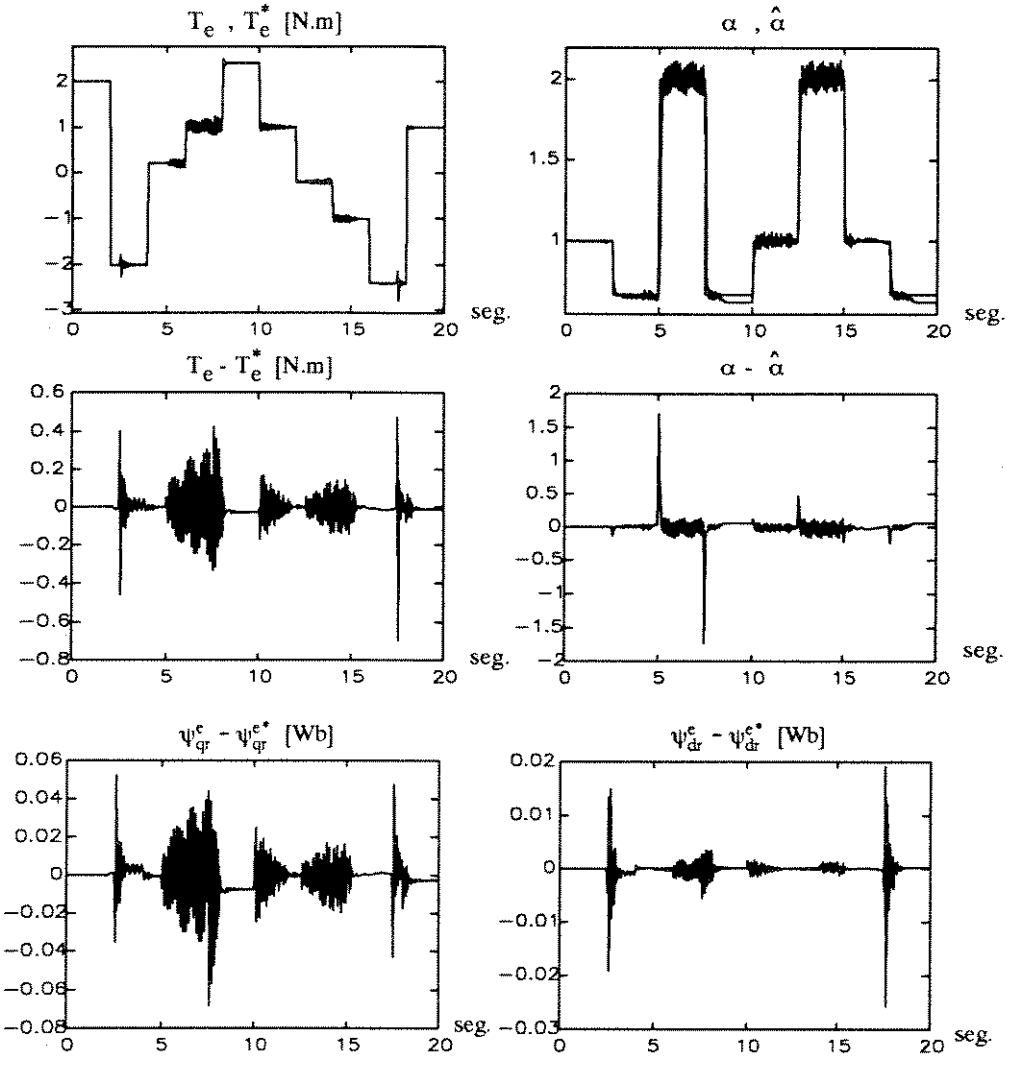


Figura 6.46 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação em degrau e variação em degrau do parâmetro α da máquina

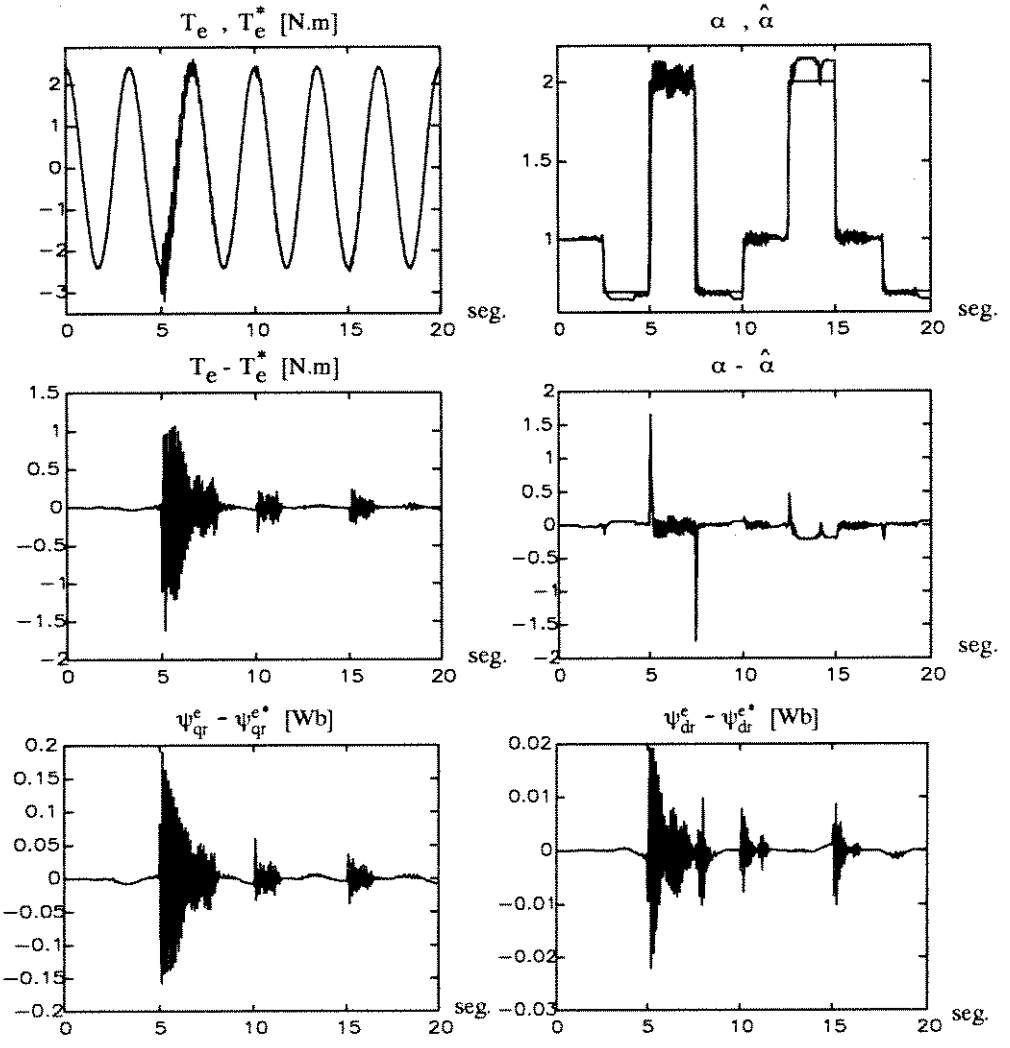


Figura 6.47 - Desempenho do sistema de controle adaptativo da máquina de indução, composto por um controlador baseado no método por realimentação de estado, um estimador de parâmetros e um observador de fluxo (figura 6.38), para um comando de conjugado com variação senoidal e variação em degrau do parâmetro α da máquina

As simulações apresentadas nas figuras 6.44, 6.45, 6.46 e 6.47 ilustram a capacidade de aproximação do estimador de parâmetros baseado em redes neurais quando aplicado ao método por realimentação de estado e permitem avaliar o desempenho do processo de controle. As simulações referentes ao método por realimentação de estado apresentam um comportamento semelhante àquele obtido para o método indireto, o que deve conduzir às mesmas conclusões gerais.

Quando comparado com o método indireto com e sem conhecimento do conjugado de carga e com o próprio método por realimentação de estado com conhecimento do conjugado de carga, o método por realimentação de estado sem utilizar o conhecimento do conjugado de carga tem um desempenho inferior. A degradação de

desempenho se deve a imprecisões na determinação dos valores das componentes de fluxo de rotor, fornecidas pelo observador de fluxo baseado em rede neurais (utilizado tanto no processo de realimentação de estado como no processo de definição das entradas do estimador). Os valores imprecisos conduzem à degradação no desempenho do estimador de parâmetros e, conseqüentemente, no desempenho do controlador. Não sendo necessário conhecer o conjugado de carga, o método é muito interessante do ponto de vista de aplicação.

Na figura 6.48 é apresentada a evolução temporal da saída do estimador de parâmetros para as quatro simulações.

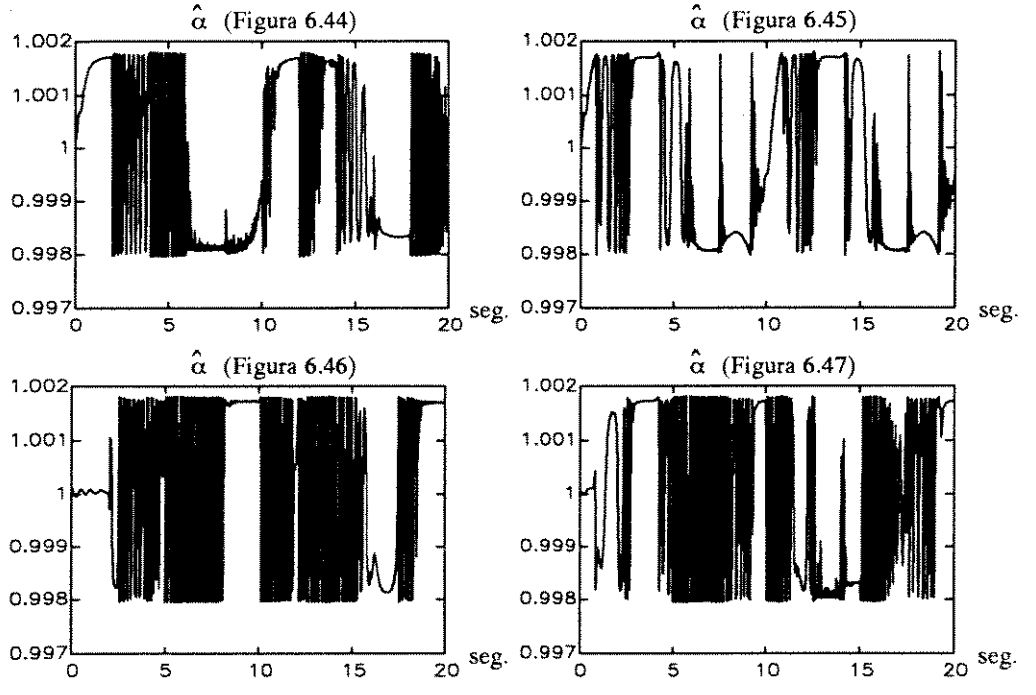


Figura 6.48 - Saída do estimador para as quatro simulações apresentadas nas figuras 6.44, 6.45, 6.46 e 6.47

Capítulo 7

Conclusões

A aplicação de redes neurais artificiais a problemas de controle de sistemas dinâmicos envolve uma série de etapas de desenvolvimento, iniciando com a definição da unidade processadora, passando pelo desenvolvimento de algoritmos de treinamento associados às arquiteturas de rede neural a serem utilizadas, e concluindo com a definição da procedência de informação para treinamento.

Após a realização destas etapas, procedimentos de implementação das estruturas de processamento resultantes devem ser obedecidos de forma a garantir o sucesso do processo de treinamento e, conseqüentemente, possibilitar a execução por parte da rede neural da tarefa especificada.

No caso dos conceitos básicos relacionados com os procedimentos mencionados acima, o fato de ter sido dedicada uma parcela considerável do texto (Capítulos 2, 3 e 4) ao seu desenvolvimento se justifica não apenas pela sua posterior utilização na fase de tratamento do problema de controle (Capítulos 5 e 6), mas também pela ausência, na literatura, de trabalhos dedicados exclusivamente ao estudo conjunto destes conceitos.

Neste caso, é importante afirmar que a diversidade de conceitos considerados no estudo está vinculada à iniciativa de se analisar inúmeros tópicos inter-relacionados e não à tentativa de se realizar uma análise aprofundada e abrangente de cada tópico. Na verdade, os conceitos desenvolvidos neste trabalho são caracterizados principalmente pela especificidade com relação ao tipo de aplicação que se propõe realizar.

Existem alguns tópicos, no entanto, que foram tratados de forma mais abrangente, como no caso dos métodos de controle adaptativo, onde apenas uma das três técnicas apresentadas foi efetivamente utilizada na fase de aplicação. A justificativa para este tratamento mais abrangente de alguns tópicos estudados é a necessidade de estabelecer comparações entre abordagens alternativas e detectar elos de ligação entre tópicos aparentemente distintos.

Na questão da solução do problema de controle, optou-se por dividir o estudo em dois capítulos, sendo que no Capítulo 5 foram desenvolvidas técnicas de controle considerando parâmetros invariantes no tempo, e no Capítulo 6, as técnicas de controle desenvolvidas no Capítulo 5 foram generalizadas de forma a considerar a variação de parâmetros.

Os principais resultados obtidos no desenvolvimento deste trabalho são enumerados a seguir:

1. estabelecimento e descrição detalhada de etapas básicas a serem seguidas na implementação de estruturas de identificação e controle de sistemas dinâmicos utilizando redes neurais;
2. desenvolvimento de um observador não-linear de fluxo baseado em redes neurais recorrentes;
3. desenvolvimento de um estimador não-linear de parâmetros baseado em redes neurais não-recorrentes;

Apesar de terem sido desenvolvidas para o atendimento específico dos requisitos do problema de aplicação, pode-se afirmar que, em linhas gerais, as etapas básicas (resultado 1) podem ser consideradas válidas para qualquer tipo de aplicação envolvendo redes neurais multi-camadas sujeitas à aplicação de algoritmos de treinamento supervisionado. Neste caso, está definida uma seqüência de procedimentos de implementação válida para grande parte das aplicações voltadas para o tratamento de sistemas dinâmicos utilizando redes neurais.

Da mesma forma, o observador não-linear de fluxo (resultado 2) é diretamente aplicável a outros problemas de identificação dinâmica de estados, requerendo apenas um ajuste da dimensão da rede neural e dos vetores de entrada e de saída. No entanto, devido à existência de outras técnicas de observação de estado, para efeito de implementação prática é necessário estabelecer níveis de comparação entre as técnicas disponíveis, de forma a considerar critérios como viabilidade e custo de implementação, ganho em eficiência, etc. Neste trabalho, o observador de fluxo baseado em redes neurais foi escolhido em razão de seu bom desempenho e de sua capacidade de explorar as características dinâmicas não-lineares apresentadas pela máquina de indução. Um procedimento interessante e que deve fazer parte de futuros trabalhos a serem desenvolvidos é a comparação com outras técnicas de observação de estado quando aplicadas ao mesmo problema.

Já no caso do estimador não-linear de parâmetros baseado em redes neurais não-recorrentes aplicado ao problema de estimação da constante de tempo do rotor (resultado 3), comparações com outros procedimentos já desenvolvidos para este propósito (Caminhas *et al.*-90)[13] ficam prejudicadas pela não coincidência do conjunto de informações consideradas no processo de estimação. O procedimento de implementação do estimador de parâmetros baseado em redes neurais pode não ser diretamente aplicável a outros problemas, mas o conceito básico de aproximação de mapeamentos não-lineares em regiões compactas de espaços multidimensionais representa a formulação básica de muitos problemas em diversas áreas. Uma extensão do estimador de parâmetros para estimar também possíveis variações da indutância mútua deve fazer parte de futuros trabalhos.

Apêndice A

A capacidade de aproximação universal de redes neurais artificiais

A.1. Motivação para o emprego de redes neurais em tarefas de aproximação

Após dois séculos de desenvolvimento, a teoria de aproximação pode ser considerada um ramo distinto da teoria de funções. A necessidade de aproximações surge por muitas razões. As leis quantitativas da natureza, por exemplo, são expressas em termos matemáticos, não com precisão absoluta, mas com graus variáveis de aproximação. Mesmo quando funções matemáticas são conhecidas precisamente, muitas vezes há a necessidade de aproximá-las por outras funções com precisão especificada de forma a possibilitar os cálculos em problemas práticos. O desenvolvimento da análise matemática tem conduzido ao descobrimento e estudo de importantes classes de funções de aproximação, que incluem polinômios, séries trigonométricas, funções ortogonais, "splines", etc. Neste contexto, as redes neurais artificiais de múltiplas camadas podem ser consideradas simplesmente como famílias específicas de funções para a aproximação de mapeamentos não-lineares arbitrários.

Uma função comumente utilizada para aproximar uma outra função f(u) dada é o polinômio algébrico:

$$P(u) = a_0 + a_1 u + ... + a_n u^n \qquad (A.1)$$

onde a_i (i = 0,1,...,n) são constantes.

O cálculo de P(u) envolve apenas as operações de adição, subtração e multiplicação e a simplicidade destas operações é uma das razões por que polinômios algébricos são as mais utilizadas funções de aproximação. Mas a simplicidade de construção de um polinômio não é razão suficiente para utilizá-lo. O que se procura é a possibilidade de se aproximar qualquer função contínua com um erro especificado arbitrariamente.

A atual teoria de aproximação de funções é o resultado do trabalho do grande matemático russo Tchebysheff, que introduziu o conceito da melhor aproximação. O próximo grande passo foi dado por Weierstrass que provou rigorosamente que uma função contínua arbitrária pode ser aproximada com qualquer grau desejado de precisão por um polinômio. Nos estágios iniciais, o problema usualmente consistia em aproximar funções individuais, mas gradualmente passou a considerar classes inteiras de funções. Bernstein, Borel, Lebesgue e Kolmogorov são alguns dos muitos nomes associados a este desenvolvimento.

Além dos polinômios algébricos, um outro importante método de aproximação de funções é baseado no uso de polinômios trigonométricos. Um polinômio trigonométrico de ordem *n* pode ser definido na forma:

$$T_n(u) = a_0 + \sum_{k=1}^{n} [a_k \cos(ku) + b_k \text{sen}(ku)] \qquad (A.2)$$

onde a_0, a_k e b_k são parâmetros constantes.

Séries trigonométricas para aproximação de funções foram investigadas a partir da metade do século 18, sendo que Euler e Bernoulli fizeram importantes contribuições nesta área. Eles também levantaram a questão fundamental a respeito da possibilidade de representar uma função arbitrária por uma série trigonométrica. Alguns dos maiores descobrimentos teóricos relacionados a estas séries foram realizados por Fourier durante a primeira metade do século 19, tanto que estas séries passaram a receber seu nome.

Muitos problemas práticos, no entanto, podem conduzir à expansão de funções em séries diferentes das séries trigonométricas. Por exemplo, apesar de manter uma similaridade matemática, as séries de funções ortogonais representam um outro tipo de aproximação dado na forma:

$$f(u) = \sum_{i=1}^{\infty} \alpha_i \phi_i(u) \qquad (A.3)$$

onde ϕ_i (i=1,2,...) formam um conjunto ortonormal. O parâmetro constante α_i pode ser obtido na forma:

$$\alpha_i = \int_a^b f(u)\phi_i(u)du \qquad (A.4)$$

onde [a,b] é o intervalo onde $\phi_i(u)$ (i = 1,2,...) são ortogonais. Na prática, um número finito de termos da série infinita é utilizado para se obter uma aproximação da função f. O grau de aproximação pode ser melhorado com o aumento no número de termos na expansão e a principal vantagem é que a adição de um $(n+1)$-ésimo termo na aproximação não afeta os valores calculados dos n primeiros termos.

Desta breve discussão, é possível concluir que cada método tem sua própria teoria e história. Condições sob as quais um método é preferível a outro são também conhecidas, mas nenhum método pode ser considerado superior a todos os outros. Além disso, esta discussão é muito relevante quando se analisam as potencialidades de aproximação de redes neurais artificiais. Já que polinômios, séries de Fourier, funções ortogonais, etc., podem aproximar funções em um espaço compacto $C(R^n,R^m)$, a questão que surge é por que redes neurais deveriam ser escolhidas em lugar de tais representações. Em particular as seguintes questões devem ser abordadas quando se está por decidir pelas redes neurais em problemas práticos de aproximação de funções:

- quanto ao número de parâmetros, as redes neurais são representações mais reduzidas de classes especiais de funções contínuas? Se forem, quais são as características destas funções?
- as redes neurais são mais fáceis de construir?
- o procedimento para se determinar os parâmetros das redes neurais são mais fáceis em algum sentido?

Estas questões e outras similares, que têm recebido considerável atenção nos últimos anos, possuem apenas respostas parciais até o momento (Narendra&Parthasarathy-91)[53]. Assim, até que uma justificativa teórica rigorosa esteja disponível, a escolha de redes neurais deve ser baseada em considerações práticas.

A.2. Análise do potencial de aproximação de redes neurais

Nesta seção é analisada a habilidade das redes neurais em aproximar funções contínuas arbitrárias em conjuntos compactos $C(R^n,R^m)$ com qualquer grau de precisão desejado. Foi mencionada na seção anterior a importância da contribuição de Weierstrass para a teoria de aproximação. O famoso teorema de Weierstrass pode ser apresentado da forma (Narendra&Parthasarathy-91)[53]:

Teorema de Aproximação de Weierstrass: Seja C[a,b] o espaço de funções contínuas a valores reais definidas no intervalo [a,b]. Se $f \in C[a,b]$, então existe um polinômio P com coeficientes reais tais que:

$$|f(u)-P(u)| < \varepsilon \qquad \text{para } \varepsilon > 0 \text{ e } u \in [a,b].$$

O teorema afirma que qualquer função em C[a,b] pode ser aproximada arbitrariamente por um polinômio. Dito de outra forma, o conjunto dos polinômios é denso em C[a,b].

Seguindo o impressionante sucesso das redes neurais de múltiplas camadas em uma variedade de aplicações, o que naturalmente se pergunta é se tais redes são também capazes de aproximar funções arbitrárias com qualquer grau desejado de precisão. A resposta para tal questão pode ser dada utilizando-se uma generalização do teorema de Weierstrass devida a Stone.

Seja Ω um espaço métrico. O conjunto de funções contínuas mapeando Ω no eixo real é denotado por $C[\Omega]$. Dado um conjunto Ω arbitrário, a família F de funções f que mapeia $\Omega \rightarrow \Re$ é uma álgebra se:

$$f_1, f_2 \in F \Rightarrow \begin{cases} \alpha f_1 + \beta f_2 \in F \\ f_1 . f_2 \in F \end{cases} \qquad \text{com } \alpha, \beta \in \Re$$

A família F_1 de funções é uma sub-álgebra da família F_2 se F_1 é uma álgebra e $F_1 \subset F_2$.

Teorema de Stone-Weierstrass: Seja Ω um espaço métrico compacto e F uma sub-álgebra de $C[\Omega]$. Se F separa pontos de Ω e não se anula em nenhum ponto de Ω, então F é densa em $C[\Omega]$.

Quando o Teorema de Stone-Weierstrass é utilizado para determinar se redes neurais podem aproximar funções arbitrárias, o esforço principal é direcionado na determinação de como as condições do teorema são satisfeitas. O teorema a seguir é o resultado deste esforço:

Teorema 1: Seja $\Omega \subset \Re^n$ um conjunto compacto e seja $C[\Omega]$ o conjunto de funções contínuas mapeando Ω no eixo real. Seja F a família de funções $f: \Omega \rightarrow \Re$ onde:

$$f(u) = \sum_{i=1}^{N} \beta_i \gamma\left(\alpha_i^T u + b_i\right)$$

com $u, \alpha_i \in \Re^n$, $b_i, \beta_i \in \Re$ e $\gamma(.)$ uma função sigmoidal não-decrescente satisfazendo:

$$\lim_{z \rightarrow -\infty} \gamma(z) = 0 \qquad \text{e} \qquad \lim_{z \rightarrow +\infty} \gamma(z) = 1.$$

Então F é densa em $C[\Omega]$.

O Teorema 1 estabelece rigorosamente as potencialidades de aproximação de uma rede neural de duas camadas. Em particular, redes neurais de múltiplas camadas com pelo menos uma camada oculta são capazes de aproximar qualquer função contínua em $C[\Omega]$, onde Ω é um conjunto compacto no R^n.

Este importante resultado foi obtido independentemente por vários pesquisadores, incluindo Cybenko (89)[15], Funahashi (89)[25], Gallant e White (88)[26] e Hornik,Stinchcombe e White (88)[32]. A demonstração do Teorema 1 está baseada no fato de que uma aproximação para f(u) via série de Fourier dada na forma:

$$f(u) = \sum_{i=1}^{N} \beta_i \cos\left(\alpha_i^T u + b_i\right)$$

satisfaz as condições do teorema de Stone-Weierstrass e que funções cossenoidais podem ser expressas na forma de uma combinação linear finita de funções sigmoidais adequadamente escalonadas e transladadas em relação ao eixo real.

A.3. Comparação com outros métodos de aproximação

Desde que uma única camada oculta na rede a torna capaz de representar qualquer função contínua definida em um universo compacto, então a mesma propriedade é também compartilhada por redes neurais com mais de uma camada oculta. Contudo, como no caso dos polinômios, das séries de Fourier e das funções ortogonais, os teoremas não

fornecem uma fórmula para a escolha do número de termos necessários para se alcançar a aproximação desejada, ou seja, o número de camadas e o número de neurônios em cada camada da rede. Apesar de já existem técnicas de análise desenvolvidas com o objetivo de se encontrar a dimensão da rede adequada para cada problema, o procedimento mais difundido ainda é a determinação da dimensão adequada por tentativa e erro. Neste caso, é importante considerar que, como no caso dos polinômios, redes com um número pequeno de parâmetros podem não ser capazes de aproximar com precisão suficiente os dados de entrada-saída observados, enquanto que redes com um grande número de parâmetros podem não ser capazes de generalizar (interpolar) com precisão suficiente.

Como polinômios, funções ortogonais e "splines" possuem também esta capacidade de aproximação, é necessário justificar por que redes neurais devem ser escolhidas. Amplos estudos computacionais realizados durante os últimos anos revelam que redes neurais de múltiplas camadas apresentam algumas vantagens sobre métodos convencionais de aproximação (Narendra&Parthasarathy-91)[53]:

- Aproximações polinomiais não apresentam boas propriedades escalares. O número de termos necessários aumenta muito rapidamente com a dimensão do espaço de entrada, bem como o grau do polinômio utilizado. Se o espaço de entrada é de dimensão n e um polinômio de grau d é utilizado, n^d parâmetros devem ser ajustados. Métodos polinomiais são difíceis de implementar em "hardware" devido ao efeito de saturação dos sinais dos circuitos analógicos. Além disso, devido ao fato de que polinômios de ordem maior que 1 não satisfazem a condição de Lipschitz (Kailath-80)[33], surgirão dificuldades teóricas quando utilizados em problemas de controle. Quando o polinômio utilizado é de grau elevado e a magnitude da entrada excede a unidade, a saída pode apresentar valores muito elevados. Uma outra desvantagem da utilização de polinômios na aproximação de funções é encontrada no processo de ajuste de parâmetros, onde o ajuste de um parâmetro para reduzir o erro correspondente a um valor específico da entrada pode causar alterações significativas na saída, isto para outros valores da entrada. Este fenômeno pode acabar resultando em instabilidade numérica.
- As funções básicas para expansão em séries ortogonais são usualmente complexas e freqüentemente são fornecidas na forma de tabelas. Logo, elas não são particularmente adequadas para aplicações em tempo real. Da mesma forma que polinômios, elas também são difíceis de implementar em "hardware".
- "Splines" são geralmente determinados "off-line", utilizando-se dados de entrada-saída disponíveis, não sendo adequados para aplicação "on-line". Particularmente, não são adequados para aplicação em sistemas de controle.

- Em contraste com os métodos acima, redes neurais, além de apresentarem robustez, são prontamente implementáveis em "hardware". Esforços têm sido direcionados na confecção de circuitos integrados de alta densidade (VLSI) baseados na arquitetura de redes neurais. Pode ser demonstrado que pequenas alterações no "hardware" e a presença de ruído não afetam significativamente a saída do sistema. Já que os vários neurônios estão densamente interconectados, cada neurônio é responsável por uma pequena parcela da informação. Assim, falhas eventuais em neurônios internos não resultam em alterações significativas no desempenho. Além disso, esta mesma propriedade faz com que níveis de sinal no sistema permanecam dentro de limites aceitáveis. Conseqüentemente, redes neurais podem ser muito úteis como componentes de projeto de sistemas complexos.

Apêndice B

Análise de Sistemas Dinâmicos por Espaço de Estados

A análise e projeto de sistemas dinâmicos no espaço de freqüência, denominada teoria convencional de sistemas, é aplicável apenas para sistemas lineares invariantes no tempo. A característica essencial da teoria convencional é que ela é baseada na relação de entrada-saída do sistema, ou seja, em sua função de transferência.

Em virtude da necessidade de satisfazer especificações cada vez mais rigorosas com relação ao desempenho de processos de controle e da existência de ferramentas computacionais mais poderosas, a análise e projeto de sistemas dinâmicos vem gradualmente adquirindo maior complexidade. No entanto, a descrição e análise de sistemas mais complexos que apresentam, por exemplo, parâmetros variantes no tempo e características não-lineares não podem ser adequadamente realizadas por meio da teoria convencional. Dessa forma, uma nova abordagem na análise e projeto de sistemas de controle foi desenvolvida baseada no conceito de estado de um sistema. Denominada teoria moderna de sistemas (TM), esta nova abordagem contrasta com a teoria convencional (TC) em vários aspectos, dentre os quais é possível destacar:

- A TM, como descrito acima, permite o tratamento de sistemas mais complexos. Isto se deve principalmente à redução de complexidade na descrição matemática do sistema de controle obtida via análise por espaço de estados.
- A TC é essencialmente uma abordagem no domínio da frequência complexa, enquanto que a TM é uma abordagem no domínio do tempo. Do ponto de vista computacional, a abordagem no domínio do tempo é mais interessante, pois os modelos desenvolvidos para análise podem ser utilizados diretamente nos processos de simulação e implementação computacional.
- O projeto na TM pode ser feito para uma classe de entradas. ao invés de uma função de entrada específica (função impulsiva, degrau, ou senoidal) como ocorre na TC.
- A TM permite a inclusão de condições iniciais no projeto, o que não ocorre na TC.

Nas próximas seções, é apresentada uma introdução aos tópicos da teoria moderna de sistemas que, de alguma forma, são utilizados no desenvolvimento deste trabalho.

B.1. Representação de Sistemas Lineares de Tempo Contínuo por Espaço de Estados

A teoria moderna de sistemas se baseia na descrição das equações do sistema em termos das relações entre suas variáveis de estado. Assim, antes de prosseguir, é importante definir estado, variáveis de estado, vetor de estados e espaço de estados (Ogata-82)[54]:

Estado. O estado de um sistema dinâmico é o menor conjunto de variáveis (chamadas variáveis de estado) tal que o conhecimento destas variáveis em $t=t_0$, juntamente com a entrada para $t \geq t_0$, determina completamente o comportamento do sistema para qualquer instante $t \geq t_0$. Neste caso, o estado de um sistema dinâmico num instante $t \geq t_0$ é independente do estado e da entrada antes de t_0.

Variáveis de Estado. As variáveis de estado de um sistema dinâmico são o menor conjunto de variáveis que determina o estado do sistema. Note que as variáveis de estado não precisam ser grandezas fisicamente mensuráveis ou observáveis. Em termos práticos, entretanto, é conveniente escolher para variáveis de estado grandezas facilmente mensuráveis porque boa parte das leis de controle vai requerer a realimentação dessas variáveis com ganhos adequados.

Vetor de Estados. Se n variáveis de estado são necessárias para descrever completamente o comportamento de um dado sistema, então estas n variáveis de estado podem ser consideradas como as n componentes de um vetor de estados $\mathbf{x}(t)$.

Espaço de Estados. O espaço n-dimensional cujos eixos de coordenadas representam cada uma das componentes do vetor de estados $\mathbf{x}(t)$ é chamado de espaço de estados. Dessa forma, qualquer estado pode ser representado por um ponto no espaço de estados.

B.1.1. Sistemas descritos por equações diferenciais lineares de ordem *n*

Um sistema dinâmico contínuo que consiste de um número finito de elementos concentrados pode ser descrito por equações diferenciais ordinárias em que o tempo é a variável independente. Utilizando notação matricial, uma equação diferencial de ordem n pode ser representada por uma equação matricial diferencial de primeira ordem. Se os n elementos do vetor de variáveis independentes formam um conjunto de variáveis de estado, então a equação matricial diferencial é chamada uma equação de estado.

Considere o seguinte sistema de ordem n:

$$y^{(n)}+a_1y^{(n-1)}+...+a_{n-1}y^{(1)}+a_ny = u \qquad (B.1)$$

Observando que o conhecimento de $y(0),y^{(1)}(0),...,y^{(n-1)}(0)$, juntamente com a entrada $u(t)$ para $t \geq 0$, determina completamente o futuro comportamento do sistema, pode-se considerar $y(t),y^{(1)}(t),...,y^{(n-1)}(t)$ como um conjunto de *n* variáveis de estado. Matematicamente, tal escolha é bastante conveniente. Na prática, no entanto, em virtude da presença de perturbações e ruídos inerentes ao sistema, a escolha de variáveis de estado correspondentes a termos que envolvam derivadas de ordem elevada pode não ser um procedimento adequado.

Definindo:

$$x_1 = y$$
$$x_2 = y^{(1)}$$
$$\cdots$$
$$x_n = y^{(n-1)}$$

é possível reescrever a equação (B.1) na forma:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_3$$
$$\dot{x}_{n-1} = x_n$$
$$\dot{x}_n = - a_nx_1 - ... - a_1x_n + u$$

ou equivalentemente:

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu \qquad (B.2)$$

onde A e B são denominadas respectivamente matriz de estado e matriz de entrada, assumindo a forma:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & . & . & 0 \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & . & . & -a_1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 0 \\ 1 \end{bmatrix}$$

A equação de saída assume a forma:

$$y = Cx = \begin{bmatrix} 1 & 0 & . & . & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_n \end{bmatrix} \qquad \text{(B.3)}$$

onde C é denominada matriz de saída.

A equação matricial diferencial de primeira ordem (B.2) é a equação de estado, e a equação algébrica (B.3) é a equação de saída.

Através de escolhas adequadas para as variáveis de estado, sistemas mais complexos podem também ser representados na forma de equações de estado. Seja, portanto, um sistema de ordem *n* com as seguintes propriedades:

- a entrada do sistema é um vetor de dimensão *r*
- a saída do sistema é um vetor de dimensão *p*

então, sua representação por espaço de estados assume a forma:

$$\begin{cases} \dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} = C\mathbf{x} + D\mathbf{u} \end{cases} \text{ com } \begin{matrix} A^{nXn}, B^{nXr} \\ C^{pXn}, D^{pXr} \end{matrix} \qquad \text{(B.4)}$$

B.1.2. A não-unicidade do conjunto de variáveis de estado

Seja um vetor de estados $\mathbf{x}$ com componentes $x_1, x_2,...,x_n$. É possível considerar como outro vetor de estados $\hat{\mathbf{x}}$ qualquer conjunto de valores $\hat{x}_1, \hat{x}_2,...,\hat{x}_n$ tal que:

$$\mathbf{x} = P\hat{\mathbf{x}} \qquad \text{para P não-singular}$$

Neste caso, baseando-se na equação (B.4), a nova representação por espaço de estados assume a forma:

$$\begin{cases} \dot{\hat{\mathbf{x}}} = P^{-1}AP\hat{\mathbf{x}} + P^{-1}B\mathbf{u} \\ \mathbf{y} = CP\hat{\mathbf{x}} + D\mathbf{u} \end{cases} \qquad \text{(B.5)}$$

As equações (B.4) e (B.5) fornecem a mesma informação sobre o comportamento do sistema, já que seus polinômios característicos são idênticos. Este fato se deve à invariância dos autovalores sob uma transformação linear, o que é demonstrado a seguir:

- tomando o polinômio característico da equação (B.5) e sabendo que o determinante de um produto é o produto dos determinantes, obtém-se:

$$\left|\lambda I - P^{-1}AP\right| = \left|\lambda P^{-1}P - P^{-1}AP\right| = \left|P^{-1}(\lambda I - A)P\right|$$

$$= \left|P^{-1}\right|.\left|\lambda I - A\right|.\left|P\right| = \left|P^{-1}\right|.\left|P\right|.\left|\lambda I - A\right|$$

$$= \left|P^{-1}P\right|.\left|\lambda I - A\right| = \left|\lambda I - A\right|$$

B.1.3. Equação de estado linear invariante no tempo

B.1.3.1. Solução da equação de estado homogênea

Para a obtenção da solução geral da equação de estado linear invariante no tempo, considere inicialmente o caso homogêneo e escalar:

$$\dot{x} = ax \tag{B.6}$$

Na tentativa de resolver esta equação é possível supor uma solução do tipo:

$$x(t) = b_0 + b_1 t + b_2 t^2 + \ldots + b_k t^k + \ldots \tag{B.7}$$

Substituindo esta suposta solução na equação (B.6), obtém-se:

$$b_1 + 2b_2 t + 3b_3 t^2 + \ldots + kb_k t^{k-1} + \ldots = a(b_0 + b_1 t + b_2 t^2 + \ldots + b_k t^k + \ldots)$$

Para que a solução seja verdadeira, ela deve valer para qualquer *t*. Portanto, equacionando os coeficientes de igual potência de *t*, é possível obter:

$$b_1 = ab_0;\ b_2 = \frac{1}{2}ab_1 = \frac{1}{2}a^2 b_0;\ b_3 = \frac{1}{3}ab_2 = \frac{1}{3.2}a^3 b_0;\ \ldots;\ b_k = \frac{1}{k!}a^k b_0;\ \ldots$$

O valor de b_0 é determinado substituindo-se t=0 na equação (B.7). Dessa forma, a solução x(t) pode ser escrita como:

$$x(t) = \left(1 + at + \frac{1}{2!}a^2 t^2 + \ldots + \frac{1}{k!}a^k t^k + \ldots\right)x(0) = e^{at}x(0) \tag{B.8}$$

Dada a equação diferencial matricial:

$$\dot{\mathbf{x}} = A\mathbf{x} \tag{B.9}$$

onde: $\mathbf{x}$ = vetor n-dimensional

A = matriz constante nXn

por analogia com o caso escalar, sua solução pode ser escrita na forma:

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) \qquad \text{com} \qquad e^{At} = \sum_{k=0}^{\infty} \frac{A^k t^k}{k!} \tag{B.10}$$

Um método alternativo para a obtenção da solução da equação de estado homogênea pode ser encontrado aplicando-se a transformada de Laplace a ambos os lados da equação (B.9), produzindo:

$$s\mathbf{X}(s) - \mathbf{x}(0) = A\mathbf{X}(s) \qquad \text{onde } \mathbf{X}(s) = \mathcal{L}[x]$$

Logo,

$$(sI-A)\mathbf{X}(s) = \mathbf{x}(0) \Rightarrow \mathbf{X}(s) = (sI - A)^{-1}\mathbf{x}(0)$$

e aplicando a transformada inversa de Laplace,

$$\mathbf{x}(t) = \mathcal{L}^{-1}[(sI - A)^{-1}]\mathbf{x}(0) \tag{B.11}$$

Note que:

$$(sI - A)^{-1} = \frac{I}{s} + \frac{A}{s^2} + \frac{A}{s^3} + \ldots$$

Portanto, a transformada inversa de Laplace de $(sI - A)^{-1}$ fornece:

$$\mathcal{L}^{-1}\left[(sI - A)^{-1}\right] = I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + \ldots = e^{At} \tag{B.12}$$

Das equações (B.11) e (B.12) obtém-se novamente a equação (B.10), solução do problema. A importância da equação (B.12) está no fato de que ela provê um meio conveniente de se achar a solução em forma fechada para a matriz exponencial.

B.1.3.2. Solução da equação de estado não-homogênea

Considere inicialmente o caso escalar:

$$\dot{x} = ax + bu \tag{B.13}$$

Reescrevendo a equação (B.13) na forma:

$$\dot{x} - ax = bu$$

e multiplicando ambos os lados desta equação por e^{-at}, obtém-se:

$$e^{-at}[\dot{x}(t) - ax(t)] = \frac{d}{dt}[e^{-at}x(t)] = e^{-at}bu(t)$$

Integrando esta equação entre 0 e t, resulta:

$$e^{-at}x(t) = x(0) + \int_0^t e^{-as}bu(s)ds$$

ou equivalentemente:

$$x(t) = e^{at}x(0) + \int_0^t e^{a(t-s)}bu(s)ds \qquad (B.14)$$

O primeiro termo do lado direito da equação (B.14) é a resposta às condições iniciais, e o segundo termo é a resposta à entrada u(t).

Considere, agora, o caso multidimensional, tomando uma equação de estado na forma:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \qquad (B.15)$$

onde

$\mathbf{x}$ $\Rightarrow$ vetor n-dimensional

$\mathbf{u}$ $\Rightarrow$ vetor r-dimensional

$\mathbf{A}$ $\Rightarrow$ matriz constante nXn

$\mathbf{B}$ $\Rightarrow$ matriz constante nXr

De forma análoga ao caso escalar (equação (B.14)), a solução da equação (B.15) é dada por:

$$\mathbf{x}(t) = e^{At}\mathbf{x}(0) + \int_0^t e^{A(t-s)}\mathbf{Bu}(s)ds$$

Até agora, o tempo inicial foi tomado como sendo zero. Entretanto, se o tempo inicial é dado por $t=t_0$ ao invés de $t=0$, então a solução deve ser modificada para

$$\mathbf{x}(t) = e^{A(t-t_0)}\mathbf{x}(t_0) + \int_{t_0}^t e^{A(t-s)}\mathbf{Bu}(s)ds \qquad (B.16)$$

B.1.3.3. Matriz de Transição de Estados

Pode-se escrever a solução da equação de estado homogênea (B.9) como:

$$\mathbf{x}(t) = \phi(t)\mathbf{x}(0) \qquad (B.17)$$

onde $\phi(t)$, uma matriz nXn, é a solução única de

$$\dot{\phi}(t) = A\phi(t), \quad \phi(0) = I$$

Das equações (B.10), (B.11) e (B.17), obtém-se:

$$\phi(t) = e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}]$$

Da equação (B.17) pode-se concluir que a solução da equação (B.9) é simplesmente uma transformação da condição inicial. Portanto, a matriz $\phi(t)$ é chamada de matriz de transição de estados.

Para o caso não-homogêneo dado pela equação (B.15), a solução utilizando o conceito de matriz de transição e tempo inicial t_0 pode ser apresentada na forma:

$$x(t) = \phi(t - t_0)x(t_0) + \int_{t_0}^{t} \phi(t - s)Bu(s)ds \qquad (B.18)$$

B.1.3.4. Matriz de Transferência

O conceito de matriz de transferência é uma extensão do conceito de função de transferência, e pode ser obtida aplicando-se a transformada de Laplace às equações de estado e de saída e expressando-as na forma de uma razão entre as transformadas de Laplace da saída e da entrada.

Considere a representação por espaço de estados de um sistema de ordem *n*, com *r* entradas e *p* saídas dada pela equação (B.4). Aplicando-se a transformada de Laplace obtém-se:

$$\begin{cases} s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \\ \mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) \end{cases} \qquad (B.19)$$

A matriz de transferência é, então, expressa na forma:

$$\frac{\mathbf{Y}(s)}{\mathbf{U}(s)} = \mathbf{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \qquad (B.20)$$

B.1.4. Equação de estado linear variante no tempo

Uma vantagem da representação de sistemas dinâmicos por espaço de estados é que ela pode ser estendida a sistemas lineares variantes no tempo, adquirindo a forma:

$$\begin{cases} \dot{\mathbf{x}} = A(t)\mathbf{x} + B(t)\mathbf{u} \\ \mathbf{y} = C(t)\mathbf{x} + D(t)\mathbf{u} \end{cases} \quad \text{com} \quad \begin{matrix} A^{nXn}, B^{nXr} \\ C^{pXn}, D^{pXr} \end{matrix} \qquad (B.21)$$

Os resultados obtidos nas seções anteriores valem para o caso variante no tempo, desde que as modificações a seguir sejam consideradas.

No caso variante no tempo, a matriz ϕ de transição de estados depende tanto de t como de t_0, e não da diferença $t-t_0$. Portanto, não é possível fazer sempre o tempo inicial igual a zero.

Também é importante compreender que, agora, ϕ pode não ser dada como uma matriz exponencial. Considere a equação de estado homogênea

$$\dot{\mathbf{x}} = A(t)\mathbf{x} \qquad \text{com} \quad A^{nXn}$$

sua solução é dada por

$$\mathbf{x}(t) = \phi(t,t_0)\mathbf{x}(t_0)$$

onde $\phi(t,t_0)$ é a matriz não-singular nXn que satisfaz a seguinte equação diferencial

$$\dot{\phi}(t) = \mathbf{A}(t)\phi(t,t_0), \qquad \phi(t_0,t_0) = \mathbf{I}$$

Dessa forma, a matriz de transição de estados $\phi(t,t_0)$ é dada por uma matriz exponencial se e somente se $\mathbf{A}(t)$ e $\int_{t_0}^{t}\mathbf{A}(s)ds$ comutam (Chen-84)[14]. Caso contrário, $\phi(t,t_0)$ não pode ser expressa analiticamente em uma forma fechada.

B.2. Representação de Sistemas Lineares de Tempo Discreto por Espaço de Estados

A abordagem via espaço de estados para a análise de sistemas dinâmicos pode ser estendida para o caso de tempo discreto. A forma discreta da representação por espaço de estados é bastante análoga à forma contínua.

A representação por espaço de estados mais geral para sistemas lineares de tempo discreto é dada na forma:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{G}(k)\mathbf{x}(k) + \mathbf{H}(k)\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{D}(k)\mathbf{u}(k) \end{cases} \text{ com } \begin{matrix} \mathbf{G}^{nXn}, \mathbf{H}^{nXr} \\ \mathbf{C}^{pXn}, \mathbf{D}^{pXr} \end{matrix} \qquad \text{(B.22)}$$

onde $\mathbf{x}(k)$ é o vetor de estados, $\mathbf{u}(k)$ é o vetor de entrada, e $\mathbf{y}(k)$ é o vetor de saída, cada um especificado em t = kT, onde k=0,1,2,..., e T é o período de amostragem. Note que $\mathbf{x}(k)$ denomina o vetor $\mathbf{x}(t)$ em t=kT.

Além da existência de processos inerentemente de tempo discreto, cuja representação na forma de espaço de estados é adequadamente descrita pela equação (B.22), o tratamento via computador digital de processos de tempo contínuo requer a conversão das equações de estado de tempo contínuo para tempo discreto (Ästrom&Wittenmark-90)[4]. Portanto, antes de analisar a equação (B.22), é interessante apresentar um método de discretização que permita a obtenção de um equivalente de tempo discreto para representações por espaço de estados de tempo contínuo.

B.2.1. Discretização de Equações de Estado de Tempo Contínuo

A aplicação de computadores digitais no tratamento de processos físicos de tempo contínuo requer apenas uma descrição do processo em instantes de amostragem. O computador vai receber informações e atuar sobre um processo sempre em instantes discretos de tempo. Assim, o objetivo é desenvolver um modelo do processo de tempo

contínuo que descreva seu comportamento apenas nos instantes de amostragem, sem considerar o comportamento entre duas amostras consecutivas. Este procedimento de análise pode simplificar muito o tratamento de processos de tempo contínuo via computador digital.

Um problema fundamental é como descrever um sistema de tempo contínuo conectado a um computador via conversores A-D e D-A. Considere o sistema apresentado na figura B.1:

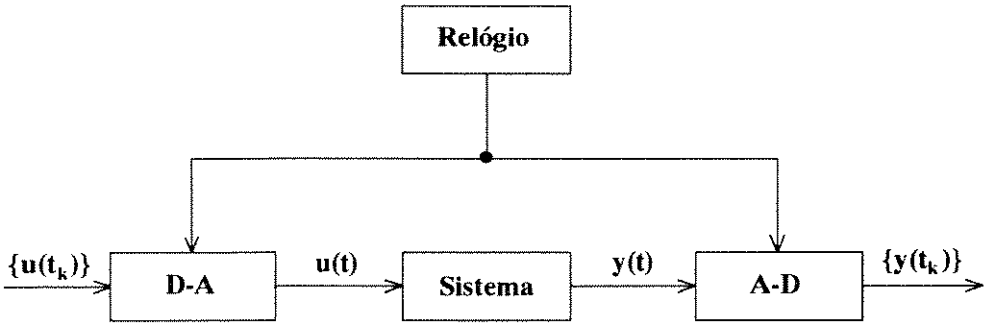


Figura B.1 - Sistema de tempo contínuo conectado a conversores A-D e D-A

Os sinais no computador correspondem às sequências $\{u(t_k)\}$ e $\{y(t_k)\}$. O problema básico está em encontrar uma relação entre estas sequências. O processo de obtenção de um equivalente de tempo discreto para um sistema de tempo contínuo é chamado amostragem de um sistema de tempo contínuo.

Dado o sistema de ordem n de tempo contínuo, é possível obter uma representação em tempo discreto que forneça valores exatos em t=kT, k=0,1,2,...

- Considere a equação de estado de tempo contínuo

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \qquad \text{(B.23)}$$

- A representação de tempo discreto da equação (B.23) vai assumir a forma (note que **G** e **H** dependem do período de amostragem T):

$$\mathbf{x}((k+1)T) = \mathbf{G}(T)\mathbf{x}(kT) + \mathbf{H}(T)\mathbf{u}(kT) \qquad \text{(B.24)}$$

- Para determinar-se **G**(T) e **H**(T), é preciso utilizar a solução da equação (B.23), dada pela equação (B.16), com t_0=0:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-s)}\mathbf{B}\mathbf{u}(s)ds \qquad \text{(B.25)}$$

- Supondo que todos os r componentes do vetor **u**(t) são constantes no intervalo entre quaisquer dois instantes consecutivos de amostragem, então:

$$\mathbf{u}(t)=\mathbf{u}(kT) \quad \text{para } kT \le t < (k+1)T$$

e da equação (B.25):

$$\mathbf{x}((k+1)T) = e^{A(k+1)T}\mathbf{x}(0) + e^{A(k+1)T}\int_0^{(k+1)T} e^{-As}\mathbf{B}\mathbf{u}(s)ds \qquad (B.26)$$

e

$$\mathbf{x}(kT) = e^{AkT}\mathbf{x}(0) + e^{AkT}\int_0^{kT} e^{-As}\mathbf{B}\mathbf{u}(s)ds \qquad (B.27)$$

- Multiplicando-se a equação (B.27) por e^{AT} e subtraindo da equação (B.26), obtém-se:

$$\begin{aligned}\mathbf{x}((k+1)T) &= e^{AT}\mathbf{x}(kT) + e^{A(k+1)T}\int_{kT}^{(k+1)T} e^{-As}\mathbf{B}\mathbf{u}(s)ds \\ &= e^{AT}\mathbf{x}(kT) + e^{AT}\int_0^{T} e^{-At}\mathbf{B}\mathbf{u}(kT)dt \\ &= e^{AT}\mathbf{x}(kT) + \int_0^{T} e^{A\lambda}\mathbf{B}\mathbf{u}(kT)d\lambda \text{ onde } \lambda = T - t \qquad (B.28)\end{aligned}$$

- Definindo:

$$\mathbf{G}(T) = e^{AT} \qquad (B.29)$$

$$\mathbf{H}(T) = \left(\int_0^{T} e^{At}dt\right)\mathbf{B} \qquad (B.30)$$

então a equação (B.28) assume a forma:

$$\mathbf{x}((k+1)T) = \mathbf{G}(T)\mathbf{x}(kT) + \mathbf{H}(T)\mathbf{u}(kT) \qquad (B.31)$$

que é a equação (B.24). Portanto, as equações (B.29) e (B.30) fornecem as matrizes $\mathbf{G}(T)$ e $\mathbf{H}(T)$ desejadas.

B.2.2. Solução da Equação de Estado, Controlabilidade e Observabilidade

Considere a equação (B.22) para o caso invariante no tempo:

$$\begin{cases}\mathbf{x}(k+1) = \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)\end{cases} \quad \text{com} \quad \begin{matrix}\mathbf{G}^{nXn}, \mathbf{H}^{nXr} \\ \mathbf{C}^{pXn}, \mathbf{D}^{pXr}\end{matrix} \qquad (B.32)$$

Os estados nos instantes (k+2), (k+3),...,(k+n) podem ser obtidos na forma:

$$\mathbf{x}(k+2) = \mathbf{G}^2\mathbf{x}(k) + \mathbf{G}\mathbf{H}\mathbf{u}(k) + \mathbf{H}\mathbf{u}(k+1)$$

... ...

$$\mathbf{x}(k+n) = \mathbf{G}^n\mathbf{x}(k) + \mathbf{G}^{n-1}\mathbf{H}\mathbf{u}(k) + \ldots + \mathbf{G}\mathbf{H}\mathbf{u}(k+n-2) + \mathbf{H}\mathbf{u}(k+n-1)$$

Se a matriz $[\mathbf{G}^{n-1}\mathbf{H} \quad \mathbf{G}^{n-2}\mathbf{H} \quad \ldots \quad \mathbf{GH} \quad \mathbf{H}]$ tem posto n, o sistema é dito ser completamente controlável, pois qualquer estado inicial $\mathbf{x}(0)=\mathbf{x}_I$ pode ser transferido para um estado final qualquer $\mathbf{x}_F$ em n passos.

A solução da equação de estado pode, então, ser descrita por:

$$\mathbf{x}(k) = \mathbf{G}^k\mathbf{x}(0) + \sum_{j=0}^{k-1}\mathbf{G}^{k-j-1}\mathbf{H}\mathbf{u}(j) \qquad k=1,2,3,\ldots \tag{B.33}$$

É possível verificar que $\mathbf{x}(k)$ consiste de duas partes, uma representando a contribuição do estado inicial $\mathbf{x}(0)$, e a outra a contribuição da entrada $\mathbf{u}(j)$, j=0,1,2,...,k-1.

Da equação (B.33), pode-se concluir que a matriz de transição de estados é dada por:

$$\phi(k) = \mathbf{G}^k \tag{B.34}$$

onde ϕ é uma matriz única satisfazendo: $\phi(k+1) = \mathbf{G}\phi(k), \quad \phi(0) = \mathbf{I}$

Em termos da matriz de transição de estados, a equação (B.33) pode ser escrita na forma:

$$\mathbf{x}(k) = \phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1}\phi(k-j-1)\mathbf{H}\mathbf{u}(j) \tag{B.35}$$

$$= \phi(k)\mathbf{x}(0) + \sum_{j=0}^{k-1}\phi(j)\mathbf{H}\mathbf{u}(k-j-1)$$

Com relação às saídas nos instantes (k+1), (k+2), ..., (k+n), é possível afirmar que:

$$\mathbf{y}(k+1) = \mathbf{C}[\mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k)]$$

$$\mathbf{y}(k+2) = \mathbf{C}[\mathbf{G}^2\mathbf{x}(k) + \mathbf{G}\mathbf{H}\mathbf{u}(k-1) + \mathbf{H}\mathbf{u}(k)]$$

$$\ldots \qquad\qquad \ldots$$

$$\mathbf{y}(k+n) = \mathbf{C}\mathbf{G}^n\mathbf{x}(k) + \sum_{l=0}^{n-r}\mathbf{C}\mathbf{G}^{k-l-1}\mathbf{H}\mathbf{u}(k+l) \tag{B.36}$$

Se as entradas e saídas são especificadas e a matriz $[\mathbf{C} \quad \mathbf{CG} \quad \ldots \quad \mathbf{CG}^{n-1}]$ é de posto n, o sistema descrito pela equação (B.32) é dito ser completamente observável, pois o estado do sistema pode ser determinado com o auxílio da equação (B.36).

B.2.3. Obtenção da função de entrada-saída: realizabilidade

Em um sistema descrito pela equação (B.32), se o estado inicial $\mathbf{x}_I$ no instante k é conhecido, o comportamento do sistema a partir deste estado, para qualquer seqüência de entrada, pode ser determinado, e $\mathbf{y}(k+n)$ pode ser descrito na forma:

$$\mathbf{y}(k+n) = \mathbf{F}[\mathbf{x}_I, \mathbf{u}(k+n-1), \mathbf{u}(k+n-2), \ldots, \mathbf{u}(k)] \tag{B.37}$$

onde **F** é denominado mapeamento de entrada-saída do sistema.

Para um sistema linear de ordem *n* que é controlável e observável, o seu estado inicial pode ser obtido a partir de *n* valores de saída e *n* valores de entrada. Isto implica que este sistema pode ser descrito por uma equação da forma:

$$\mathbf{y}(k+1) = \sum_{i=0}^{n-1} A_i \mathbf{y}(k-i) + \sum_{j=0}^{n-1} B_j \mathbf{u}(k-j) \tag{B.38}$$

onde A_i e B_j são matrizes pXp e pXr respectivamente.

Uma forma de se obter a equação (B.38) a partir da descrição de um sistema na forma da equação (B.32) é através da aplicação da transformada Z, produzindo:

$$\begin{cases} z\mathbf{X}(z) - z\mathbf{x}(0) = \mathbf{G}\mathbf{X}(z) + \mathbf{H}\mathbf{U}(z) \\ \mathbf{Y}(z) = \mathbf{C}\mathbf{X}(z) + \mathbf{D}\mathbf{U}(z) \end{cases} \tag{B.39}$$

Sua matriz de transferência fica:

$$\mathbf{G}(z) = \frac{\mathbf{Y}(z)}{\mathbf{U}(z)} = \mathbf{C}(z\mathbf{I} - \mathbf{G})^{-1}\mathbf{H} + \mathbf{D} \tag{B.40}$$

Aplicando-se, agora, a transformada Z inversa na equação (B.40) e expressando a saída mais recente como função das entradas e saídas anteriores, resulta na equação (B.38). Esta equação representa o modelo auto-regressivo e de média móvel (ARMA) para sistemas lineares. Ela é uma equação recorrente de entrada-saída e forma a base para boa parte da teoria moderna de identificação e análise de sistemas (Narendra&Parthasarathy-90)[52].

As equações de entrada-saída são de fundamental importância na teoria de sistemas, porque elas descrevem o comportamento do sistema quando este é observado por um observador externo, por exemplo, um controlador. Por outro lado, para permitir a aplicação de técnicas de equações diferenciais e de otimização, boa parte do desenvolvimento teórico em teoria de sistemas faz uso de representações por espaço de estados.

Com isso, uma questão básica é decidir quando um dado operador de entrada-saída admite uma representação na forma de vetor de estados, ou seja, quando que este

operador é dito ser realizável. Para sistemas lineares invariantes no tempo (equação (B.32)), sempre vai existir uma representação de entrada-saída na forma da equação (B.38). De forma análoga, dada uma representação na forma da equação (B.38), sempre vai existir uma representação por espaço de estados de dimensão finita equivalente e que não é única. Para sistemas não-lineares, no entanto, a realizabilidade não está garantida.

B.3. Representação de sistemas não-lineares por espaço de estados

A representação de sistemas dinâmicos por espaço de estados permite a extensão para o caso não-linear. No entanto, uma série de noções básicas válidas para o caso linear e discutidas nas seções anteriores podem não mais ser válidas no caso não-linear. Para tentar elucidar estas questões e outros conceitos envolvidos na representação de sistemas não-lineares, toda a análise a seguir vai se concentrar no caso discreto.

A representação por espaço de estados mais geral para sistemas não-lineares de tempo discreto é dada na forma:

$$\begin{cases} \mathbf{x}(k+1) = f[\mathbf{x}(k), \mathbf{u}(k)] \\ \mathbf{y}(k) = h[\mathbf{x}(k), \mathbf{u}(k)] \end{cases} \tag{B.41}$$

onde: $\mathbf{k} \in Z$ (conjunto dos inteiros)
$\mathbf{x} \in X$ (conjunto de estado de dimensão $\underline{n}$)
$\mathbf{u} \in U$ (conjunto de entrada de dimensão $\underline{r}$)
$\mathbf{y} \in Y$ (conjunto de saída de dimensão $\underline{p}$)

A função $f: X \times U \rightarrow X$ é a função de transição de estados de um passo à frente e $h: X \times U \rightarrow Y$ é a função de saída.

Pela aplicação repetida da função f, a função ϕ de transição de estados de múltiplos passos à frente pode ser obtida como segue:

$$\begin{aligned} \mathbf{x}(t+2) &= f[\mathbf{x}(t+1), \mathbf{u}(t+1)] \\ &= f[f[\mathbf{x}(t), \mathbf{u}(t)], \mathbf{u}(t+1)] \\ &= \phi[\mathbf{x}(t), \mathbf{u}(t+1), \mathbf{u}(t)] \end{aligned}$$

$$\begin{aligned} \mathbf{x}(t+3) &= f[f[f[\mathbf{x}(t), \mathbf{u}(t)], \mathbf{u}(t+1)], \mathbf{u}(t+2)] \\ &= \phi[\mathbf{x}(t), \mathbf{u}(t+2), \mathbf{u}(t+1), \mathbf{u}(t)] \end{aligned}$$

Denominando U^* o conjunto de todas as sequências de elementos do conjunto U, ou seja:

$$U^* = \{u_k, u_{k-1}, \ldots, u_1, u_0 / k \geq 0\} \quad \text{onde } u_j \in U \ (j=0,1,\ldots,k)$$

a função de transição de estados pode, agora, ser definida como ϕ: X x $U^* \rightarrow X$.

Se a sequência de entrada é $\{u_j\}$ (j=0,1,...,k-1) e $\mathbf{w}_t^k = u_{t+k-1}, u_{t+k-2}, \ldots, u_t$, então:

$$\mathbf{x}(t+k) = \phi[\mathbf{x}(t), \mathbf{w}_t^k] \tag{B.42}$$

Quando o sistema se encontra no estado $\mathbf{x}_0$, o mapeamento de entrada-saída do sistema é definido na forma:

$$F_{\mathbf{x}_0} : U^* \rightarrow Y$$

Assumindo que este mapeamento é dado, ele é dito ser finitamente realizável se e somente se existir uma representação por espaço de estados com o vetor de estados tendo dimensão finita.

Sontag (79)[63] obteve importantes resultados relacionados à função $F_{\mathbf{x}_0}$, os quais são apresentados a seguir:

Seja U_k^* o conjunto de todas as sequências de comprimento k. Para cada valor de k, é possível definir $F_{\mathbf{x}_0/k} : U_k^* \rightarrow Y$.

$F_{\mathbf{x}_0}$ é denominada função resposta polinomial se, para valores finitos de k, $F_{\mathbf{x}_0/k}$ é um polinômio de grau finito em todas as suas variáveis. O grau do polinômio pode tender a infinito quando k tende a infinito.

Uma função resposta polinomial é finitamente realizável se, e somente se, existirem dois polinômios P_1 e P_2 de grau finito tal que

$$\mathbf{y}(k) = \frac{P_1[\mathbf{y}(k-1), \ldots, \mathbf{y}(k-n), \mathbf{u}(k-1), \ldots, \mathbf{u}(k-n)]}{P_2[\mathbf{y}(k-1), \ldots, \mathbf{y}(k-n), \mathbf{u}(k-1), \ldots, \mathbf{u}(k-n)]} \tag{B.43}$$

Além disso, esta função admite uma representação similar à representação por espaço de estados na forma:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{G}[\mathbf{u}(k)]\mathbf{x}(k) + \mathbf{H}[\mathbf{u}(k)]\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}[\mathbf{u}(k)]\mathbf{x}(k) + \mathbf{D}[\mathbf{u}(k)]\mathbf{u}(k) \end{cases} \quad \text{com} \quad \begin{matrix} \mathbf{G}^{nXn}, \mathbf{H}^{nXr} \\ \mathbf{C}^{pXn}, \mathbf{D}^{pXr} \end{matrix} \tag{B.44}$$

onde $\mathbf{A}[.]$, $\mathbf{B}[.]$, $\mathbf{C}[.]$ e $\mathbf{D}[.]$ são matrizes polinomiais de grau finito.

Em um intervalo de tempo finito e com entradas limitadas, qualquer sistema de tempo discreto que dependa continuamente de suas entradas pode ser aproximado arbitrariamente bem dentro do conjunto de sistemas descritos pela equação (B.44).

Estes resultados são matematicamente atrativos e de considerável significância no contexto de sistemas não-lineares.

B.3.1. Observabilidade no caso não-linear

No caso de sistemas lineares de ordem n observáveis, foi discutido em seções anteriores que o estado do sistema $\mathbf{x}$ no instante k pode ser determinado unicamente realizando-se n medidas $\mathbf{y}(k)$, $\mathbf{y}(k+1)$,...,$\mathbf{y}(k+n-1)$ da saída. Neste caso, a saída no instante $k+n$ é unicamente determinada pelos n valores passados da saída e pelos n valores passados da entrada, garantindo a existência de equações de entrada-saída recursivas (equação (B.38)).

Dessa forma, a existência de equações de entrada-saída recursivas está intimamente relacionada com a habilidade de determinar-se unicamente o estado do sistema a partir de um número finito de dados de entrada-saída. Ou, de outro modo, o sistema será observável em tempo finito se equações de entrada-saída recursivas puderem ser obtidas.

Ao contrário do caso linear, restrições muito fortes devem ser consideradas para que uma solução no caso não-linear exista. Por exemplo, no caso linear, é fato que (teorema de Cayley-Hamilton) se um sistema não é observável em $m=n$ instantes de tempo, ele não será observável para $m>n$ (Chen-84)[14]. Também no caso linear, se um sistema é observável, ele é globalmente observável. Já no caso não-linear, mais do que n instantes de tempo podem ser necessários para se determinar o estado inicial. Além disso, é possível que o estado inicial tenha que ser restrito a alguma região para que a solução exista.

Apêndice C

Conjugado Eletromagnético em Máquinas Trifásicas de Corrente Alternada

Considere uma máquina rotativa trifásica de corrente alternada. Utilizando um sistema de referência bifásico composto por dois eixos ortogonais, denominados eixo direto (d) e em quadratura (q), girando a uma velocidade *w* arbitrária. é possível obter expressões para o conjugado eletromagnético em termos de componentes d-q de fluxo e corrente, referidas a estator e rotor.

A potência elétrica trifásica ativa fornecida pela fonte de alimentação pode ser apresentada em notação fasorial, na forma:

$$P_e = \frac{3}{2} Re[\hat{V}_s . \hat{I}_s + \hat{V}_r . \hat{I}_r] \qquad (C.1)$$

onde $\hat{V}_s$: fasor tensão de estator;

$\hat{V}_r$: fasor tensão de rotor;

$\hat{I}_s$: fasor corrente de estator;

$\hat{I}_r$: fasor corrente de rotor.

Em notação escalar, obtém-se:

$$P_e = \frac{3}{2} [v_{ds}.i_{ds} + v_{qs}.i_{qs} + v_{dr}.i_{dr} + v_{qr}.i_{qr}] \qquad (C.2)$$

onde os subíndices **d** e **q** referem-se, respectivamente, a componentes de eixo direto e eixo em quadratura.

Tomando como base o mesmo sistema de referência, é possível relacionar tensão (v), corrente (i) e fluxo (ψ) nos circuitos de estator e rotor, na forma:

Circuito de Estator

$$v_{ds} = R_s i_{ds} + \frac{d\psi_{ds}}{dt} - w\psi_{qs} \qquad (C.3)$$

$$v_{qs} = R_s i_{qs} + \frac{d\psi_{qs}}{dt} - w\psi_{ds} \qquad (C.4)$$

$$\psi_{ds} = L_{ls} i_{ds} + L_m(i_{ds}+i_{dr}) = L_{ls} i_{ds} + \psi_{dm} \qquad (C.5)$$

$$\psi_{qs} = L_{ls} i_{qs} + L_m(i_{qs}+i_{qr}) = L_{ls} i_{qs} + \psi_{qm} \qquad (C.6)$$

Circuito de Rotor

$$v_{dr} = R_r i_{dr} + \frac{d\psi_{dr}}{dt} - (w-w_r)\psi_{qr} \qquad (C.7)$$

$$v_{qr} = R_r i_{qr} + \frac{d\psi_{qr}}{dt} - (w-w_r)\psi_{dr} \qquad (C.8)$$

$$\psi_{dr} = L_{lr} i_{dr} + L_m(i_{ds}+i_{dr}) = L_{lr} i_{dr} + \psi_{dm} \qquad (C.9)$$

$$\psi_{qr} = L_{lr} i_{qr} + L_m(i_{qs}+i_{qr}) = L_{lr} i_{qr} + \psi_{qm} \qquad (C.10)$$

onde
- s : subíndice indicando componentes de estator;
- r : subíndice indicando componentes de rotor;
- d : subíndice indicando componentes de eixo direto;
- q : subíndice indicando componentes de eixo em quadratura;
- R : Resistência elétrica
- L_l : Indutância de dispersão
- L_m : Indutância mútua
- ψ_m : Fluxo de magnetização

Retornando à equação (C.2), é possível desenvolvê-la termo a termo, com base nas equações (C.3) a (C.10):

$$v_{ds}.i_{ds} = R_s i^2_{ds} + \frac{d\psi_{ds}}{dt} i_{ds} - w\psi_{qs} i_{ds}$$

$$= R_s i^2_{ds} + L_{ls}\frac{di_{ds}}{dt} i_{ds} + L_m \frac{d(i_{ds}+i_{dr})}{dt} i_{ds} - wL_{ls} i_{qs} i_{ds} - wL_m(i_{qs}+i_{qr}) i_{ds}$$

$$= R_s i^2_{ds} + \frac{1}{2} L_{ls}\frac{d(i^2_{ds})}{dt} + L_m \frac{d(i_{ds}+i_{dr})}{dt} i_{ds} - wi_{ds}[L_{ls} i_{qs} + L_m(i_{qs}+i_{qr})]$$

De maneira análoga, pode-se obter:

$$v_{qs}.i_{qs} = R_s i^2_{qs} + \frac{1}{2} L_{ls}\frac{d(i^2_{qs})}{dt} + L_m \frac{d(i_{qs}+i_{qr})}{dt} i_{qs} + wi_{qs}[L_{ls} i_{ds} + L_m(i_{ds}+i_{dr})]$$

$$v_{dr}.i_{dr} = R_r i^2_{dr} + \frac{1}{2} L_{lr}\frac{d(i^2_{dr})}{dt} + L_m \frac{d(i_{ds}+i_{dr})}{dt} i_{dr} - (w-w_r) i_{dr}[L_{lr} i_{qr} + L_m(i_{qs}+i_{qr})]$$

$$v_{qr}.i_{qr} = R_r i^2_{qr} + \frac{1}{2} L_{lr}\frac{d(i^2_{qr})}{dt} + L_m \frac{d(i_{qs}+i_{qr})}{dt} i_{qr} + (w-w_r) i_{qr}[L_{lr} i_{dr} + L_m(i_{ds}+i_{dr})]$$

Substituindo os termos desenvolvidos diretamente na equação (C.2), é possível obter uma expressão para a potência elétrica trifásica ativa P_e em termos de componentes d-q de corrente de estator e rotor. Antes da apresentação da expressão resultante, é interessante dividí-la em três parcelas de acordo com o tipo de conversão de potência elétrica envolvida:

P_{cobre} : parcela de perdas no cobre

$P_{d\lambda}$: parcela de perdas no ferro

P_{mec} : parcela efetivamente convertida em potência mecânica

Com isso, P_e assume a forma:

$$P_e = P_{cobre} + P_{d\lambda} + P_{mec} \tag{C.11}$$

onde:

$$P_{cobre} = \frac{3}{2}[R_s(i^2_{ds}+i^2_{qs}) + R_r(i^2_{dr}+i^2_{qr})] \tag{C.12}$$

$$P_{d\lambda} = \frac{3}{2}\left\{\frac{1}{2}\left[L_{ls}\frac{d(i^2_{ds}+i^2_{qs})}{dt} + L_{lr}\frac{d(i^2_{dr}+i^2_{qr})}{dt}\right] + L_m\left[\frac{d(i_{ds}+i_{dr})}{dt}(i_{ds}+i_{dr}) + \frac{d(i_{qs}+i_{qr})}{dt}(i_{qs}+i_{qr})\right]\right\} \tag{C.13}$$

$$P_{mec} = \frac{3}{2}w_r L_m(i_{qs}i_{dr} - i_{ds}i_{qr}) \tag{C.14}$$

Note que os termos em w foram cancelados, o que significa que o desenvolvimento não depende da velocidade adotada para o sistema de referência bifásico girante. Considere, agora, as seguintes relações:

$$w_r = \frac{P}{2}w_{mec} \tag{C.15}$$

$$P_{mec} = T_e w_{mec} \tag{C.16}$$

onde P : número de pólos da máquina;

w_{mec} : velocidade mecânica do rotor da máquina

T_e : conjugado eletromagnético produzido.

Substituindo as equações (C.15) e (C.16) na equação (C.14), é possível obter uma expressão para o conjugado eletromagnético na forma:

$$T_e = \frac{3}{2}\frac{P}{2}L_m(i_{dr}i_{qs} - i_{qr}i_{ds}) = \frac{3}{2}\frac{P}{2}L_m\left(\hat{I}_s \times \hat{I}_r\right) \tag{C.17}$$

onde $\hat{I}_s = \begin{bmatrix} i_{qs} & i_{ds} & 0 \end{bmatrix}$ e $\hat{I}_r = \begin{bmatrix} i_{qr} & i_{dr} & 0 \end{bmatrix}$

Substituindo convenientemente as equações (C.5), (C.6), (C.9) e (C.10) na equação (C.17), pode-se obter outras expressões para o conjugado, na forma:

$$T_e = \frac{3}{2}\frac{P}{2}(\psi_{dm} i_{qs} - \psi_{qm} i_{ds}) = \frac{3}{2}\frac{P}{2}\left(\hat{I}_s \times \hat{\psi}_m\right) \tag{C.18}$$

$$T_e = \frac{3}{2}\frac{P}{2}(\psi_{dm} i_{qr} - \psi_{qm} i_{dr}) = \frac{3}{2}\frac{P}{2}\left(\hat{I}_r \times \hat{\psi}_m\right) \tag{C.19}$$

$$T_e = \frac{3}{2}\frac{P}{2}(\psi_{ds} i_{qs} - \psi_{qs} i_{ds}) = \frac{3}{2}\frac{P}{2}\left(\hat{I}_s \times \hat{\psi}_s\right) \tag{C.20}$$

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}(\psi_{dr} i_{qs} - \psi_{qr} i_{ds}) = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\hat{I}_s \times \hat{\psi}_r\right) \tag{C.21}$$

No caso especial em que a velocidade w adotada para o sistema de referência bifásico girante é dada pela velocidade síncrona w_e, é comum utilizar um superíndice **e** para representar as componentes de corrente e fluxo. Por exemplo, no sistema de referência síncrono, a equação (C.21) assume a forma:

$$T_e = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\psi^e_{dr} i^e_{qs} - \psi^e_{qr} i^e_{ds}\right) = \frac{3}{2}\frac{P}{2}\frac{L_m}{L_r}\left(\hat{I}^e_s \times \hat{\psi}^e_r\right) \tag{C.22}$$

Apêndice D

Relações de transformação entre sistemas de referência trifásicos e bifásicos

As grandezas elétricas tensão, corrente e fluxo podem ser representadas com base em diferentes sistemas de referência. Dada uma representação baseada em um determinado sistema de referência, é possível definir-se relações de transformação que permitam obter a representação em um outro sistema de referência. Esta flexibilidade de representação é importante à medida que permite a adoção do sistema de referência mais adequado para cada aplicação.

D.1. Transformação de um sistema trifásico em um sistema bifásico equivalente

A equivalência entre sistemas trifásicos e bifásicos geralmente é dada em termos de potência. Considere o esquema apresentado na figura D.1, onde estão representadas as componentes de tensão nas referências estacionárias a^s-b^s-c^s e d^s-q^s.

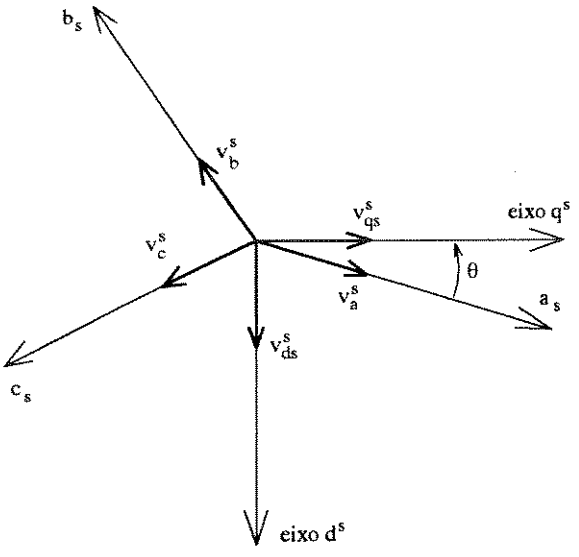


Figura D.1 - Componentes de tensão nas referências estacionárias a^s-b^s-c^s e d^s-q^s.

O ângulo θ entre os dois conjuntos de eixos é arbitrário, sendo conveniente considerar-se θ=0, de forma que o eixo q^s coincida com o eixo a^s. A representação das tensões v^s_a, v^s_b v^s_c em termos das tensões v^s_{ds} e v^s_{qs} podem ser expressas na forma matricial:

$$\begin{bmatrix} v^s_a \\ v^s_b \\ v^s_c \end{bmatrix} = A \begin{bmatrix} v^s_{qs} \\ v^s_{ds} \\ v^s_{os} \end{bmatrix}, \text{ com } A = \begin{bmatrix} \cos(\theta) & \text{sen}(\theta) & 1 \\ \cos(\theta-120°) & \text{sen}(\theta-120°) & 1 \\ \cos(\theta+120°) & \text{sen}(\theta+120°) & 1 \end{bmatrix} \qquad \text{(D.1)}$$

e a correspondente relação inversa é dada por:

$$\begin{bmatrix} v^s_{qs} \\ v^s_{ds} \\ v^s_{os} \end{bmatrix} = A^{-1} \begin{bmatrix} v^s_a \\ v^s_b \\ v^s_c \end{bmatrix}, \text{ com } A^{-1} = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos(\theta - 120°) & \cos(\theta + 120°) \\ \text{sen}(\theta) & \text{sen}(\theta - 120°) & \text{sen}(\theta + 120°) \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \quad \text{(D.2)}$$

onde v^s_{os} é a componente de seqüência zero, cuja consideração permite a obtenção de relações de transformação únicas. É importante salientar que a componente de seqüência zero é nula para condições trifásicas balanceadas.

As componentes de corrente e fluxo podem ser transformadas de maneira totalmente análoga, utilizando as matrizes A e A^{-1}.

D.2. Transformação de um sistema bifásico estacionário em um sistema bifásico girante

Com o auxílio da representação da figura D.2, as componentes de tensão na referência estacionária d^s-q^s podem ser convertidas em componentes de tensão na referência d^e-q^e, que gira à velocidade síncrona w_e, na forma:

$$\begin{bmatrix} v^e_{qs} \\ v^e_{ds} \end{bmatrix} = B \begin{bmatrix} v^s_{qs} \\ v^s_{ds} \end{bmatrix}, \qquad \text{com } B = \begin{bmatrix} \cos(w_e t) & -\text{sen}(w_e t) \\ \text{sen}(w_e t) & \cos(w_e t) \end{bmatrix} \quad \text{(D.3)}$$

e a correspondente relação inversa é dada por:

$$\begin{bmatrix} v^s_{qs} \\ v^s_{ds} \end{bmatrix} = B^{-1} \begin{bmatrix} v^e_{qs} \\ v^e_{ds} \end{bmatrix}, \qquad \text{com } B^{-1} = \begin{bmatrix} \cos(w_e t) & \text{sen}(w_e t) \\ -\text{sen}(w_e t) & \cos(w_e t) \end{bmatrix} \quad \text{(D.4)}$$

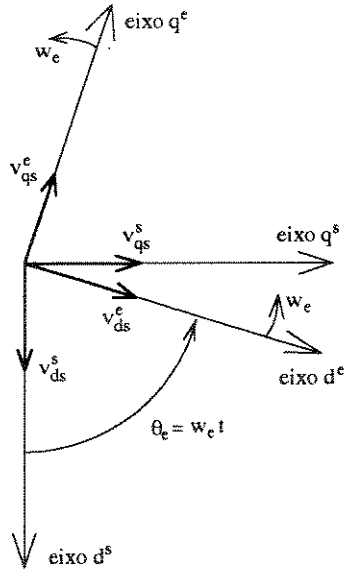


Figura D.2 - Componentes de tensão nas referências estacionária d^s-q^s e girante d^e-q^e

Apêndice E

Parâmetros e condições nominais da máquina de indução

Parâmetros

Número de pólos: P = 8

Freqüência elétrica: f = 60 Hz

Momento de inércia: J = 0,0016 kg.m^2

Resistência de estator: R_s = 4,9833 Ω

Resistência de rotor: R_r = 3,0167 Ω

Indutância de magnetização: L_m = 138.4648 mH

Indutância de dispersão de estator: L_{ls} = 10,4678 mH

Indutância de dispersão de rotor: L_{lr} = 15,7017 mH

Condições nominais

Corrente nominal: ISnom = 4,1 A

Tensão nominal: VSnom = 75 V

Potência nominal: PEnom = $\frac{1}{4}$ cv

Velocidade nominal de escorregamento (3%): WSnom = 2,8274 rad/s

Torque nominal: TEnom = 2,0113 N.m

Apêndice F

Parâmetros e resultados do processo de treinamento de redes neurais artificiais

Os processos de treinamento para as aplicações de redes neurais artificiais desenvolvidas neste trabalho, juntamente com os processos de controle da máquina de indução, foram implementados computacionalmente utilizando-se a linguagem de programação C e foram executados em computadores do tipo IBM-PC e em estações de trabalho RISC-IBM. Zuben e Netto [80] apresentam os programas básicos resultantes deste processo de implementação.

Os resultados de simulação apresentados no texto foram gerados utilizando-se as ferramentas gráficas do software MATLAB. O algoritmo de treinamento de 2ª ordem foi também implementado utilizando-se o MATLAB.

F.1. Aproximação de um período de senóide (Capítulo 4)

- $\alpha = 0,1$
- $\beta = 0,1$
- Dimensão da rede neural: camada de entrada: 10 unidades processadoras
  camada escondida: 5 unidades processadoras
  camada de saída: 1 unidade processadora
- Geração dos pesos iniciais: valores aleatórios distribuídos uniformemente no intervalo [-0,15;0,15].
- Número de padrões de treinamento: 100 padrões
- Ciclos de treinamento: 1000000 ciclos
- Políticas de apresentação dos padrões à rede neural:
  - aplicação do treinamento somente após a apresentação de todos os padrões à rede neural
  - aplicação do treinamento com base somente nos padrões que produzem um desempenho abaixo de um determinado limiar
  - apresentação aleatória dos padrões para treinamento
  - apresentação seqüêncial dos padrões para treinamento
- Algoritmo de treinamento de 1ª ordem

F.2. Aproximação de uma representação não-linear por espaço de estados (Capítulo 4)

- $\alpha = 0,1$
- $\beta = 0,1$
- Dimensão de R_1: camada de entrada: 20 unidades processadoras
  camada escondida: 10 unidades processadoras
  camada de saída: 1 unidade processadora
- Dimensão de R_2: camada de entrada: 20 unidades processadoras
  camada escondida: 10 unidades processadoras
  camada de saída: 1 unidade processadora
- Geração dos pesos iniciais: valores aleatórios distribuídos uniformemente no intervalo [-0,15;0,15].
- Ciclos de treinamento: 120000 ciclos
- Política de apresentação dos padrões à rede neural:
  - geração em tempo real de padrões de treinamento a partir de uma entrada u assumindo valores aleatórios distribuídos uniformemente no intervalo [−1;1] e aplicação do treinamento após a geração de cada padrão.
- Algoritmo de treinamento de 1ª ordem
- Erro quadrático médio resultante: $3,78 \times 10^{-2}$

F.3. Observador de fluxo de rotor (Capítulo 5)

- $\alpha = 0,01$
- $\beta = 0,1$
- Dimensão da rede neural: camada de entrada: 25 unidades processadoras
  camada escondida: 15 unidades processadoras
  camada de saída: 2 unidades processadoras
- Geração dos pesos iniciais: valores aleatórios distribuídos uniformemente no intervalo [-0,15;0,15].
- Número de padrões de treinamento: 400 padrões
- Ciclos de treinamento: 71000 ciclos
- Política de apresentação dos padrões à rede neural:
  - aplicação do treinamento com base somente nos padrões que produzem um desempenho abaixo de um determinado limiar e aplicação da técnica de "teaching forcing" (veja Capítulo 4)
- Algoritmo de treinamento de 1ª ordem
- Erro quadrático médio resultante: $1,16 \times 10^{-3}$

F.4. Observador expandido de fluxo de rotor (Capítulo 6)

- $\alpha = 0,01$ $\beta = 0,1$
- Dimensão da rede neural: camada de entrada: 30 unidades processadoras
  camada escondida: 20 unidades processadoras
  camada de saída: 2 unidades processadoras
- Geração dos pesos iniciais: valores aleatórios distribuídos uniformemente no intervalo [-0,15;0,15].
- Número de padrões de treinamento: 400 padrões
- Ciclos de treinamento: 234000 ciclos
- Política de apresentação dos padrões à rede neural:
  - aplicação do treinamento com base somente nos padrões que produzem um desempenho abaixo de um determinado limiar e aplicação da técnica de "teaching forcing" (veja Capítulo 4)
- Algoritmo de treinamento de 1ª ordem
- Erro quadrático médio resultante: $2,00 \times 10^{-3}$

F.5. Estimador de constante de tempo de rotor (Capítulo 6)

- $\alpha = 0,1 \rightarrow 0,001$ $\beta = 0,1$
- Dimensão da rede neural: camada de entrada: 20 unidades processadoras
  camada escondida: 10 unidades processadoras
  camada de saída: 1 unidade processadora
- Geração dos pesos iniciais: valores aleatórios distribuídos uniformemente no intervalo [-0,15;0,15].
- Número de padrões de treinamento: 80 padrões
- Ciclos de treinamento: 15000 ciclos para o algoritmo de 1ª ordem
  520 ciclos para o algoritmo de 2ª ordem
- Política de apresentação dos padrões à rede neural no caso do algoritmo de 1ª ordem:
  - aplicação do treinamento com base somente nos padrões que produzem um desempenho abaixo de um determinado limiar. O valor de α é reduzido conforme o treinamento evolui, até um ponto em que o algoritmo de 2ª ordem é aplicado.
- Política de apresentação dos padrões à rede neural no caso do algoritmo de 2ª ordem:
  - aplicação do treinamento somente após a apresentação de todos os padrões à rede neural.
- Erro quadrático médio resultante: $2,40 \times 10^{-3}$

Bibliografia

[1] Ästrom, K. J. Expert Control. *Automatica*, vol. 22, no. 3, p. 277-286, 1986.

[2] Ästrom, K. J. Theory and Applications of Adaptive Control-A Survey. *Automatica*, vol. 19, no. 5, p. 471-486, 1983.

[3] Ästrom, K. J., Wittenmark, B. *Adaptive Control*. Lund Institute of Technology, Addison-Wesley Publishing Company, 1989.

[4] Ästrom, K. J., Wittenmark, B. *Computer-Controlled Systems - Theory and Design*. Prentice-Hall, Inc., 1990.

[5] Barto, A. G. Connectionist Learning for Control: An Overview. *COINS Technical Report 89-89*, Department of Computer and Information Science, University of Massachusetts, Amherst MA, 1989.

[6] Bellini, A., Figalli, G., Ulivi, G. Analysis and Design of a Microcomputer-based Observer for an Induction Machine. *Automatica*, vol. 24, no. 4, p. 549-555, 1988.

[7] Bellman, R. *Dynamic Programming*. Princeton University Press, 1957.

[8] Bernard, J. A. Use of a Rule-Based System for Process Control. *IEEE Control Systems Magazine*, vol. 8, no. 5, p. 3-13, 1988.

[9] Black, H. S. Inventing the negative feedback amplifier. *IEEE Spectrum*, p. 54-60, 1977.

[10] Blaschke, F. The principle of field orientation as applied to the new transvector closed loop control system for rotating field machines. *Siemens Review*, vol. 39, no. 5, p. 213-220, 1972.

[11] Bose, B. K. *Power Electronics and AC Drives*. Prentice-Hall, Inc., 1986.

[12] Brod, D. M., Novotny, D. W. Current Control of VSI-PWM Inverters. *IEEE Transactions on Industry Applications*, vol. 21, no. 4, p. 562-570, 1985.

[13] Caminhas, W. M., Menezes, B. R., Tribuzi, A. R., Silva, S. R. *Influência, identificação e compensação das variações de parâmetros do motor de indução no controle vetorial indireto*. 8º CBA, UFPa, Belém, p. 936-941, 1990.

[14] Chen, C. *Linear System Theory and Design*. Holt, Rinehart and Winston, 1984.

[15] Cybenko, G. Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, p. 303-314, 1989.

[16] Doya, K. Bifurcation in the learning of recurrent neural networks. *Proceedings of 1992 IEEE International Symposium on Circuits and Systems*, vol 6, p. 2777-2780, 1992.

[17] Doya, K. Bifurcation of Recurrent Neural Network in Gradient Descent Learning. *IEEE Transactions on Neural Networks* (submitted), 1993.

[18] Elman, J. L. Finding structure in time. *Cognitive Science*, vol. 14, p. 179-211, 1990.

[19] Feldbaum, A. A. Dual Control Theory I. *Automatic & Remote Control*, vol. 21, p. 874, 1960.

[20] Feldbaum, A. A. Dual Control Theory II. *Automatic & Remote Control*, vol. 21, p. 1033, 1960.

[21] Feldbaum, A. A. Dual Control Theory III. *Automatic & Remote Control*, vol. 22, p. 1, 1961.

[22] Feldbaum, A. A. Dual Control Theory IV. *Automatic & Remote Control*, vol. 22, p. 109, 1961.

[23] Fitzgerald, A. E., Kingsley Jr., C., Kusko, A. *Máquinas Elétricas*. McGraw-Hill do Brasil, 3ª edição, 1975.

[24] Fletcher, R. *Practical Methods of Optimization*. John Wiley & Sons, 1975.

[25] Funahashi, K. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, vol. 2, p. 183-192, 1989.

[26] Gallant, A. R., White, H. There exists a neural network that does not make avoidable mistables. *IEEE Second International Conference on Neural Networks*, San Diego, p. 657-664, 1988.

[27] Garcés, L. J. Parameter Adaption for the Speed-Controlled Static AC Drive with a Squirrel-Cage Induction Motor. *IEEE Transactions on Industry Applications*, vol. 16, no. 2, p. 173-178, 1980.

[28] Hasse, K. *On the Dynamics of Speed Control of Static AC Drives with Squirrel Cage Induction Machines*. Ph.D. Dissertation, Technische Hochschule Darmstadt, Germany, 1969.

[29] Hebb, D. O. *Organization of Behaviour: A Neuropsychological Theory*. Wiley, New York, 1949.

[30] Heermann, P. D. *Neural Network Techniques for Stable Learning Control of Nonlinear Systems*. PhD Dissertation presented to the Faculty of the Graduate School of the University of Texas at Austin, May 1992.

[31] Hopfield, J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences 79*, p. 2554-2558, 1982.

[32] Hornik, K., Stinchcombe, M., White, H. Multi-layer feedforward networks are universal approximators. *Neural Networks*, vol. 2, p. 359-366, 1989.

[33] Kailath, T. *Linear Systems*. Stanford University, Prentice-Hall, Inc., 1980.

[34] Kalman, R. E. Design of a self-optimizing control system. *Trans ASME*, vol. 80, p. 468, 1958.

[35] Kosko, B. *Neural Networks and Fuzzy Systems - A dynamical systems approach to machine intelligence*. University of Southern California, Prentice-Hall International, Inc., 1992.

[36] Krishnan, R., Doran, F. C. Study of Parameter Sensitivity in High Performance Inverter-Fed Induction Motor Drive Systems. *IEEE Transactions on Industry Applications*, vol. 23, no. 4, p. 623-635, 1987.

[37] Kröse, B. J. A., van der Smagt, P. P. *An Introduction to Neural Networks*. Department of Computer Systems, University of Amsterdam, 1993.

[38] Kuo, B. C. *Automatic Control Systems*. 4ed, Prentice-Hall, Inc., 1982.

[39] Lapedes, A., Farber, R. Nonlinear Signal Processing Using Neural networks: Prediction and System Modelling. *Los Alamos National Laboratories Technical Report: LA-UR-87-2662*, 1987.

[40] Levis, A. H., Marcus, S. I., Perkins, W. R., Kokotovic, P., Athans, M., Brockett, R. W., Willsky, A. S. Challenges to Control: A Collective View - Report of the Workshop Held at the University of Santa Clara on September 18-19, 1986. *IEEE Transactions on Automatic Control*, vol. AC-32, no. 4, p. 275-285, 1987.

[41] Lima, E. P. *Acionamento de um motor de indução com comandos suaves de torque, utilizando a técnica de controle vetorial por campo orientado*. Dissertação de Mestrado, Faculdade de Engenharia Elétrica, Unicamp, 1993.

[42] Luenberger, D. G. *Optimization by vector space methods*. Stanford University, John Wiley & Sons, Inc., 1969.

[43] Matsuo, T., Lipo, T. A. A Rotor Parameter Identification Scheme for Vector-Controlled Induction Motor Drives. *IEEE Transactions on Industry Applications*, vol. 21, no. 4, p. 624-632, 1985.

[44] McClelland, J. L., Rumelhart, D. E., editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volume 1: Foundations*. M.I.T. Press, Cambridge, Massachusetts, 1986.

[45] McClelland, J. L., Rumelhart, D. E., editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volume 2: Psychological and Biological Models*. M.I.T. Press, Cambridge, Massachusetts, 1986.

[46] McCulloch, W. S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol 5, p. 115-133, 1943.

[47] Minsky, M. L., Papert, S. A. *Perceptrons: An introduction to Computational Geometry, Expanded Edition*. M.I.T. Press, Cambridge, Massachusetts, 292 pp., 1988.

[48] Minsky, M., Papert, S. *Perceptrons: An introduction to Computational Geometry*. M.I.T. Press, Cambridge, Massachusetts, 1969.

[49] Moller, M. F. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Preprint PB-339*, Computer Science Department, University of Aarhus, Denmark, 1990.

[50] Narendra, K. S., Mukhopadhyay, S. Intelligent Control Using Neural Networks. *IEEE Control Systems Magazine*, vol. 12, no. 2, p. 11-18, 1992.

[51] Narendra, K. S., Parthasarathy, K. Gradient Methods for the Optimization of Dynamical Systems Containing Neural Networks. *IEEE Transactions on Neural Networks*, vol. 2, no. 2, p. 252-262, 1991.

[52] Narendra, K. S., Parthasarathy, K. Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks*, vol. 1, no. 1, p. 4-27, 1990.

[53] Narendra, K. S., Parthasarathy, K. *Neural Networks in Control Systems*. American Control Conference, Boston, Massachusetts, 1991.

[54] Ogata, K. *Engenharia de Controle Moderno*. Prentice-Hall do Brasil, 1982.

[55] Packard, N. M., Crutchfield, J. P., Farmer, J. D., Shaw, R. S. Geometry from a Time-Series. *Physical Review Letters*, vol. 45, no. 9, p. 712-716, 1980.

[56] Parker, D. B. Learning Logic. *Technical Report TR-47*, Center for Comput. Res. Econ. and Management Science, M.I.T., Cambridge, Massachusetts, 1985.

[57] Pearlmutter, B. A. Dynamic Recurrent Neural Networks. *Technical Report CMU-CS-90-196*, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1990.

[58] Pham, D. T., Liu, X. Dynamic system modelling using partially recurrent neural networks. *Journal of Systems Engineering*, 1992.

[59] Pollack, J. B. *Book review*: Minsky, M. L., Papert, S. A. *Perceptrons: An introduction to Computational Geometry, Expanded Edition*. M.I.T. Press, Cambridge, Massachusetts, 292 pp., 1988.

[60] Rosenblatt, F. The Perceptron: A perceiving and recognizing automation. *Thecnical Report 85-460-1*, Cornell Aeronautical Laboratory, 1957.

[61] Rumelhart, D. E., Hinton, G. E., Williams, R. J. Learning internal representations by error propagation. Em Rumelhart, D. E., McClelland, J. L., editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volume 1: Foundations*. M.I.T. Press, Cambridge, Massachusetts, p. 318-362, 1986.

[62] Silvino, J. L. *Modelamento e observador discretos não-estacionários para máquina de indução*. Tese de Doutorado, Faculdade de Engenharia Elétrica, Unicamp, 1992.

[63] Sontag, E. D. *Polynomial Response Maps - Lecture notes in Control and Information Sciences*. Springer Verlag, Berlin, 1979.

[64] Sul, S. K. A Novel Technique of Rotor Resistance Estimation Considering Variation of Mutual Inductance. *IEEE Transactions on Industry Applications*, vol. 25, no. 4, p. 578-587, 1989.

[65] Thompson, J. M. T., Stewart, H. B. *Nonlinear Dynamics and Chaos - Geometrical Methods for Engineers and Scientists*. John Wiley and Sons, 1989.

[66] Tsypkin, Y. Z. *Adaptation and Learning in Automatic Systems*. Academic Press, New York, 1971.

[67] Verghese, G. C., Sanders, S. R. Observers for Flux Estimation in Induction Machines. *IEEE Transactions on Industrial Electronics*, vol. 35, no. 1, p. 85-94, 1988.

[68] Von Neumann, J. *The Computer and the Brain*. Yale University Press, New Haven, 1958.

[69] Werbos, P. J. Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, vol. 78, no. 10, p. 1550-1560, 1990.

[70] Werbos, P. J. *Beyond Regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, Cambridge, Massachusetts, 1974.

[71] Werbos, P. J. Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 1, p. 7-20, 1987.

[72] Whitaker, H. P., Yamron, J., Kezer, A. Design of model-reference adaptive control systems for aircraft. *Report R-164*, Instrumentation Laboratory, MIT, Cambridge, 1958.

[73] Widrow, B. Generalization and information storage in networks of adaline "neurons". Em Yovitz, M., Jacobi, G., Goldstein, G., editors. *Self-Organizing Systems*, p. 435-461, Spartan Books, Washington DC, 1962.

[74] Widrow, B., Hoff, M. Adaptive switching circuits. *1960 IRE WESCON Convention Record*, New York, p. 96-104, 1960.

[75] Widrow, B., Lehr, M. A. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *Proceedings of the IEEE*, vol. 78, no. 9, p. 1415-1442, 1990.

[76] Widrow, B., Stearns, S. D. *Adaptive Signal Processing*, Prentice-Hall, Inc., 1985.

[77] Widrow, B., Winter, R. Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition. *IEEE Computer*, p. 25-39, March 1988.

[78] Wiener, N. *Cybernetics*. M.I.T. Press, Cambridge, MA, 1948.

[79] Wittenmark, B., Ästrom, K. J. Pratical Issues in the Implementation of Self-tuning Control. *Automatica*, vol 20, p. 595-605, 1984.

[80] Zuben, F. J. V., Netto, M. L. A. Implementação computacional de algoritmos de treinamento e arquiteturas de controle utilizando redes neurais. *RT-DCA*, nº 14, Faculdade de Engenharia Elétrica, UNICAMP, 1993.