



Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Sistemas e Controle da Energia
Laboratório de Sistemas Modulares Robóticos



IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS USANDO REDES NEURAIS ARTIFICIAIS: UMA APLICAÇÃO A MANIPULADORES ROBÓTICOS

Autor: Prof. Jês de Jesus Fiais Cerqueira - UFBA
Orientador: Prof. Dr. Álvaro Geraldo Badan Palhares - UNICAMP
Co-Orientador: Prof. Dr. Marconi Kolm Madrid - UNICAMP

*Tese submetida à Faculdade de Engenharia Elétrica e de
Computação da Universidade Estadual de Campinas, para
preenchimento dos pré-requisitos parciais para obtenção do
título de*

DOUTOR EM ENGENHARIA ELÉTRICA

Banca Examinadora

Dr. Álvaro G. Badan Palhares (Presidente)
Dr. Fernando José Von Zuben - UNICAMP Dr. Ivan Nunes da Silva - UNESP
Dr. João Maurício Rosário - UNICAMP Dr. Maurício Fernandes Figueiredo - UEM

Campinas-SP, 31 de Agosto de 2001.

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

C355i Cerqueira, Jés de Jesus Fiais
Identificação de sistemas dinâmicos usando redes neurais artificiais: uma aplicação a manipuladores robóticos / Jés de Jesus Fiais Cerqueira - Campinas, SP:[s.n.], 2001

Orientadores: Álvaro Geraldo Badan Palhares, Marconi Kolm Madrid.

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

1. Modelos matemáticos. 2. Redes Neurais (Computação). 3. Liapunov, Funções de. 4. Robótica. 5. Sistema de controle ajustável. I. Badan Palhares, Álvaro Geraldo. II. Madrid, Marconi K. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Resumo

Este texto considera a análise da convergência do ajuste de parâmetros para identificadores *on-line* de modelos de sistemas dinâmicos no tempo, sendo os identificadores construídos com base na classe *multi-layer feedforward perceptron* de modelos de redes neurais artificiais.

É desenvolvida uma análise de convergência de ajuste de parâmetros baseado no método gradiente para modelos matemáticos não lineares genéricos. Esta análise é então aplicada à classe *multi-layer feedforward perceptron* de modelos de redes neurais artificiais, e proposto um algoritmo adaptativo para o ajuste de parâmetros nesta classe de rede. A análise de convergência é feita com base no segundo método de Lyapunov, e assegura boa performance ao algoritmo adaptativo proposto.

É apresentado o modelo de um identificador não linear multivariável usando a propriedade da observabilidade genérica de sistemas dinâmicos não lineares. O identificador é capaz de aproximar modelos entrada-saída multivariáveis e não lineares de sistemas que possuem a propriedade da observabilidade genérica, com seus parâmetros sendo ajustados como procedimento *on-line*. A identificação do modelo dinâmico de um sistema robótico simulado mostra a eficiência do paradigma.

Palavras Chave

Modelos Matemáticos; Redes Neurais (Computação); Liapunov, Funções de; Robótica; Sistemas de Controle Ajustável.

Abstract

This monograph considers the analysis of parameter estimation convergence for on-line identifiers for dynamical systems based on the multi-layer feedforward perceptron class of artificial neural networks.

A parameter estimation convergence analysis for generic nonlinear mathematical models is developed, based on the gradient method, and applied to the multi-layer feedforward perceptron. The analysis is based on Lyapunov's criteria, and assures good performance for a proposed adaptive algorithm.

It is presented a multivariable nonlinear identifier model based on the generic observability property of nonlinear systems. The identifier model is able to provide approximation of nonlinear multivariable input-output model of generically observable processes, with parameter adjusted by on-line procedure. The identification of the simulated robotic process shows the efficiency of the method.

Key Words

Mathematical Models; Neural Networks (Computation); Lyapunov Stability; Robotics; Adaptive Control Systems .

Agradecimentos

- À Fundação Escola Politécnica da Bahia pela ajuda financeira nos meses iniciais do curso de doutorado.
- Ao CNPq pela concessão de uma bolsa de estudos durante a realização do curso.
- À UNICAMP por ter proporcionado a oportunidade de realização do curso de doutorado, bem como um estágio de capacitação à docência de nível superior na disciplina Conversão Eletromecânica da Energia Elétrica.
- Ao Prof. Dr. Álvaro Geraldo Badan Palhares por ter aceito a minha proposta de trabalho, e pela excelente orientação.
- Ao Prof. Dr. Marconi K. Madrid pela co-orientação, pelos incentivos e pelas trocas de idéias que auxiliaram a elaboração de muitos pontos importantes do trabalho.
- Aos comentários e as sugestões dos membros da banca examinadora desta tese: Prof. Dr. Maurício Fernandes Figueiredo, da Universidade Estadual de Maringá; Prof. Dr. Ivan Nunes da Silva, da UNESP de Bauru; Prof. Dr. João Maurício Rosário, da FEM/UNICAMP; e o Prof. Dr. Fernando José Von Zuben, da FEEC/UNICAMP.
- Ao Prof. Dr. Ernesto Ruppert por ter sido meu orientador durante o Estágio de Capacitação Docência de Nível Superior.

- Aos revisores anônimos do 4^o Simpósio Brasileiro de Automação Inteligente, realizado em São Paulo em 1999, do 1st **IFAC** Conference on Mechatronic Systems, realizado em Darmstadt, Alemanha, em 2000, do IEEE-INNS-ENNS International Joint Conference on Neural Networks, realizado em Como, Itália, em 2000, e da revista *Automatica - A Journal of IFAC the International Federation of Automatic Control*, pelos comentários e sugestões que muito auxiliaram na elaboração deste trabalho.
- À Capes e à Funcamp por terem proporcionado apresentações de artigos em congressos nacionais e internacionais.
- Aos colegas que tive a oportunidade de conhecer durante a pós-graduação na UNICAMP, em especial os do Departamento de Sistemas e Controle de Energia, que com certeza proporcionaram muitos momentos de compreensão, encorajamento e alegria, e me ajudaram nos momentos difíceis.
- Aos muitos funcionários da UNICAMP que, quer diretamente quer indiretamente, funcionaram como suporte ao desenvolvimento deste trabalho.
- Aos colegas professores e funcionários da Universidade Federal da Bahia pelo apoio e incentivo.
- Aos muitos amigos e amigas que tive a oportunidade de conhecer em Campinas e que me deram muitos incentivos e alegrias.
- A toda a minha família, por terem suportado a minha ausência, em especial aos meus irmãos Wedem, William, Maria das Graças, e aos meus tios Miguel e Virgínia.
- A Noemi, meu grande amor e futura companheira, pela compreensão e pelo apoio a conclusão deste trabalho.
- Um profundo agradecimento aos meus pais Manuel e Angelina por tudo que me proporcionaram na vida e por sempre acreditarem e confiarem em mim.

- Por fim àquele ao qual, não apenas eu, mas todos nós, devemos ser gratos por todas as coisas, ao Único e Verdadeiro Deus cujo nome é Jeová.

Índice

Resumo	iii
Abstract	v
Agradecimentos	vii
Índice	xi
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Contribuições e Propostas da Tese	4

1.2	Estrutura do Texto	5
2	Preliminares Teóricas	7
2.1	Introdução	7
2.2	Representações de Sistemas	10
2.3	Nomenclatura Básica	13
2.4	Sistemas Não Lineares	13
2.4.1	Sobre Soluções de Equações de Estado	19
2.4.2	Sobre a Estabilidade de Sistemas Segundo Lyapunov	21
2.4.3	Sobre as Propriedades Genéricas	25
2.5	Observabilidade de Sistemas	25
2.6	Métodos de Otimização Não Linear	27
2.7	Aproximadores Universais	35
2.8	Transformada \mathcal{Z}	36
2.9	Sumário	38
3	Redes Neurais Artificiais do Tipo MLP	41
3.1	Introdução	41
3.2	O Método do Gradiente no Ajuste de Parâmetros	46
3.3	O Método do Gradiente Aplicado à Rede MLP	57
3.4	Algoritmo Adaptativo para Ajuste de Parâmetros na Rede MLP	63
3.4.1	Sobre as Condições Iniciais para a Rede MLP	67
3.4.2	Sobre a Escolha do Número de Neurônios na Primeira Camada	68
3.5	Sumário	69

4	Descrição de um Sistema Robótico	71
4.1	Introdução	71
4.2	Análise Cinemática de Manipuladores	73
4.3	Análise da Dinâmica de Manipuladores	77
4.3.1	Dinâmica do Atuador Elétrico	78
4.3.2	Dinâmica do Manipulador com Inclusão do Atuador Elétrico . .	82
4.3.3	Sensores para o Espaço de Juntas	83
4.4	Modelo de um Sistema Articulado	85
4.5	Sumário	88
5	Identificação Não Linear de Sistemas Dinâmicos	91
5.1	Introdução	91
5.2	Comparação Entre as Regiões de Densidade	94
5.3	O Modelo do Identificador Não Linear	98
5.4	Aplicação do Paradigma e Resultados de Simulações	102
5.5	Sumário	117
6	Considerações Finais	121
6.1	Sugestões de Futuros Trabalhos	122
	Referências Bibliográficas	125
	Índice Remissivo	137

Lista de Figuras

3.1	Ilustração do Procedimento de Ajuste de Parâmetros.	47
3.2	Identificação de Sistemas Dinâmicos.	48
3.3	Modelo da Rede <i>Multi-Layer Feedforward Perceptron</i>	58
3.4	Fluxograma do algoritmo de ajuste de parâmetros para a rede MLP	66
4.1	Localização dos eixos coordenados em uma cadeia articulada.	74
4.2	Modelo de campo elétrico do motor.	79
4.3	Modelo mecânico do motor elétrico.	79
4.4	Sistema robótico de dois GL	86
5.1	Regiões de densidade do sistema e do identificador.	96
5.2	Identificador rastreando o sistema.	97
5.3	Modelo do Identificador.	100

5.4	Sinais de Entrada.	105
5.5	Saídas do sistema e dos identificadores.	106
5.6	Evoluções das funções objetivo.	107
5.7	Evoluções dos η	108
5.8	Evoluções dos $w_{2_{\max}}$	109
5.9	Evoluções dos $u_{m_{\max}}$	110
5.10	Evoluções das funções objetivo na janela de tempo entre 7 e 9 segundos.	112
5.11	Simulação usando iniciação de parâmetros no intervalo $(-1; 1)$	113
5.12	Simulação usando iniciação de parâmetros no intervalo $(-0.01; 0.01)$	114
5.13	Simulação usando $N = 3$ e ruído nos sinais de saída.	116

Lista de Tabelas

4.1	Parâmetros dos elos e das juntas	75
5.1	Parâmetros do Manipulador de Yu e Lloyd (1995).	103
5.2	Parâmetros para as MLPs	104

Capítulo 1

Introdução

RECONHECIDAMENTE, é fundamental conhecer e compreender um sistema antes de manipulá-lo. Um sistema pode ser conhecido, estudado e compreendido a partir do seu modelo. A existência de um modelo pode ser considerado até mesmo como um pré-requisito para que uma estrutura de controle ou gerenciamento possa ser posta em operação (Unberhauen e Roa, 1998).

Um modelo é na prática uma forma de representação da realidade que é o sistema. Existem diversas formas ou tipos de modelos que podem ser usados ou selecionados dependendo do objetivo do seu uso. Um modelo pode ser proposto para descrever um sistema e seus parâmetros podem ser ajustados [i.e., os do modelo]. Os modelos identificados descrevem o sistema a partir de informações de seus sinais de entrada e saída, e pelo ajuste de alguma classe de modelos matemáticos previamente selecionados, sendo este modelo escolhido sem se fazer o uso de informações acuradas a respeito da estrutura ou organização do sistema. Nessa situação, o modelo proposto é muitas vezes chamado caixa-preta (*black-box*) (Sjöberg *et al.*, 1995).

Um dos objetivos dos modelos caixas-pretas é descrever de forma aproximada o comportamento do sistema dentro de determinada região ou faixa de operação, para uso, por exemplo, em sistemas de controle. Os modelos caixas-pretas podem ser de natureza linear ou não linear. Durante os anos de 70 e até meados dos anos 80, grande es-

forço da comunidade científica foi concentrado no estudo de modelos de identificadores de natureza linear, e como resultado esta teoria é atualmente bastante consolidada (Ljung, 1987; Unberhauen e Roa, 1998).

Desde segunda metade da década de 80 até o presente, muita atenção tem sido dispensada pela ao estudo de modelos de identificadores não lineares (Leontaritis e Billings, 1985*a*; Leontaritis e Billings, 1985*b*; Sjöberg *et al.*, 1995; Juditsky *et al.*, 1995). Os avanços concernentes à teoria de aproximadores universais não lineares, aliados à diminuição dos preços e facilidade de acesso a sistemas computacionais de alta capacidade de processamento, abriram novas perspectivas para a área.

Na teoria de aproximadores universais, as redes neurais artificiais (**RNAs**) tiveram um grande desenvolvimento teórico, bem como aplicações neste período (Cybenko, 1989; Hornik *et al.*, 1989; Mhaskar, 1992; Barron, 1993). Numerosos estudos foram feitos sobre a aplicação das **RNAs** na identificação não linear de sistemas (Narendra e Parthasarathy, 1990; Hunt *et al.*, 1992; Levin e Narendra, 1993; Levin e Narendra, 1995; Suykens *et al.*, 1995; Levin e Narendra, 1996; Jagannathan e Lewis, 1996; Chen *et al.*, 1997).

Levin e Narendra (1995) propuseram um identificador não linear usando **RNA** baseado na propriedade da observabilidade genérica. A propriedade da observabilidade genérica foi apresentada a comunidade científica por Aeyels (1981) como sendo uma propriedade particular de modelos matemáticos não lineares discretos no tempo, sendo possível a sua aplicação a modelos de tempo contínuo a partir de discretizações. Este identificador proposto por Levin e Narendra (1995) apresenta a vantagem de requerer uma quantidade mínima de informações a respeito do sistema para que a propriedade da observabilidade genérica seja garantida e o identificador implementado.

O trabalho de Levin e Narendra (1995) concentrou-se na capacidade de aproximação do modelo proposto, sem preocupação com a influência do método de ajuste de parâmetros na qualidade do processo de identificação. A literatura técnica apresenta diversos métodos de ajuste de parâmetros para **RNAs**, sendo o mais amplamente

conhecido o algoritmo *back-propagation* apresentado como aplicação às **RNAs** por Rumelhart *et al.* (1986).

Cerqueira *et al.* (2000a) apresentaram uma extensão ao algoritmo *back-propagation* que possibilita a determinação da taxa de aprendizagem, que é um parâmetro interno do algoritmo com influência na convergência do processo de ajuste de parâmetros. O método proposto por Cerqueira *et al.* (2000a) é apropriado para ajuste de parâmetros *on-line*, e foi aplicado na identificação de sistemas (Cerqueira e Badan Palhares, 1999; Cerqueira *et al.*, 2000b; Cerqueira *et al.*, 2000c).

Uma outra questão importante não abordada por Levin e Narendra (1995) é sobre que tamanho deveria ter uma **RNA** para proporcionar uma identificação eficiente [i.e., quantos neurônios deveria ter a **RNA** para permitir uma identificação eficiente]. Cerqueira *et al.* (2000c) apresentaram um estudo baseado na abordagem de regiões de densidade, e mostraram que uma **RNA** com pequena região de densidade pode ser muito eficiente.

Uma categoria especial de sistemas dinâmicos é a dos manipuladores robóticos, que são amplamente usados nas linhas de produção e manufatura dos dias atuais. Nessas aplicações os manipuladores possuem uma forma de operação autônoma [i.e., sem a intervenção humana]. Em adição, os manipuladores robóticos têm tido outras aplicações, muitas vezes teleoperadas [i.e., com intervenção humana], tais como execução de cirurgias por controle remoto e limpezas de áreas perigosas.

Os manipuladores robóticos apresentam uma dinâmica complexa, não linear e com parâmetros variantes no tempo, podendo ser multivariáveis e com acoplamento entre as variáveis (para os casos com mais de um grau de liberdade). A maioria dos manipuladores existentes usam apenas sensores de posição no espaço de juntas para a obtenção de informações sobre a saída do sistema. Nesta situação, a construção de um modelo para o manipulador robótico pode tornar-se difícil ou até mesmo impraticável (Klafter *et al.*, 1989; Cerqueira e Badan Palhares, 1999; Cerqueira *et al.*, 2000b).

As técnicas de controle de manipuladores robóticos usando apenas informações

sobre as posições no espaço de juntas são em geral baseadas no algoritmo **PID** e possuem desempenho limitado (Spong e Vidyasagar, 1989). As técnicas avançadas de controle fazem uso de informações sobre posições, velocidades e acelerações no espaço de juntas para a construção de um modelo dinâmico objetivando fazer-se um cancelamento da dinâmica do manipulador dentro de um bem elaborado algoritmo de controle (Spong e Vidyasagar, 1989; Ortega e Spong, 1989; Abdallah *et al.*, 1991; Sage *et al.*, 1999).

Cerqueira e Badan Palhares (1999), Cerqueira *et al.* (2000*b*), Cerqueira *et al.* (2000*c*) mostraram que a partir do modelo de identificador baseado na propriedade da observabilidade genérica apresentado por Levin e Narendra (1995) é possível a construção de um modelo dinâmico para um manipulador robótico usando apenas informações a respeito das posições no espaço de juntas. Este modelo pode ser usado por alguma estrutura de controle avançada para fazer-se o cancelamento da dinâmica do manipulador.

1.1 Contribuições e Propostas da Tese

As principais contribuições e propostas desta tese são as seguintes:

- (i) Apresenta uma contribuição ao algoritmo *back-propagation*, em que é feita a análise de convergência do processo de ajuste de parâmetros para **RNAs** do tipo *multi-layer feedforward perceptron* (**MLP**), e é fornecido um limitante superior para a taxa de aprendizagem ou passo do algoritmo (Cerqueira *et al.*, 2000*a*; Cerqueira *et al.*, 2000*c*).
- (ii) Especifica um modelo de identificador baseado na propriedade da observabilidade genérica para aplicações em sistemas robóticos e usando apenas informações sobre a posição no espaço de juntas (Cerqueira e Badan Palhares, 1999; Cerqueira *et al.*, 2000*b*; Cerqueira *et al.*, 2000*c*).

Como contribuições e propostas menores esta tese apresenta:

- (i) Um algoritmo de aprendizagem adaptativo para ajuste de parâmetros em **RNAs** do tipo **MLP**, onde é mantida a mesma simplicidade do algoritmo *back-propagation* e sendo necessário apenas a especificação de um parâmetro (o *momentum*) dentro de uma faixa de variação pré-definida para torná-lo convergente (Cerqueira *et al.*, 2000c).
- (ii) Um estudo sobre como o tamanho da região de densidade do identificador (neste caso uma **RNA** tipo **MLP**) em relação ao tamanho da região de densidade do sistema influencia a capacidade de aproximação do identificador (Cerqueira *et al.*, 2000c).

1.2 Estrutura do Texto

O capítulo 2 apresenta os conceitos teóricos preliminares necessários para a compreensão do conteúdo da tese, objetivando o entendimento ou vocabulário comum, sobre alguns conceitos e definições. Inicialmente, são apresentados conceitos sobre sistemas e sobre as formas de representá-los. Em seguida, são apresentados importantes tópicos sobre teoria de sistemas não lineares, otimização e aproximação de funções.

No capítulo 3, é feita uma descrição matemática do modelo *multi-layer feedforward perceptron* de rede neural artificial e apresentado um algoritmo de aprendizagem adaptativo para ajuste paramétrico na modelagem de sistemas dinâmicos. O algoritmo é baseado no método de otimização pelo gradiente e a análise de convergência local é feita através da função de *Lyapunov*. O algoritmo é estruturado de tal forma que o usuário necessita especificar apenas um parâmetro dentro de uma faixa de variação pré-definida para torná-lo convergente, e mantém, em adição, a simplicidade dos algoritmos baseados no método gradiente.

No capítulo 4, é feita a descrição matemática de um sistema robótico, onde são apresentadas uma análise cinemática e uma análise dinâmica com a inclusão do modelo de acionadores elétricos. São feitos também comentários a respeito dos principais

sensores usados para espaço de juntas.

O capítulo 5 apresenta a identificação de sistemas dinâmicos não lineares sob a óptica da propriedade da observabilidade genérica. São comentados alguns desenvolvimentos na área de identificação não linear e discutida a propriedade da observabilidade genérica. Um identificador para propósitos *on-line* é apresentado, usando como ferramenta de modelagem não linear o modelo *multi-layer feedforward perceptron* de redes neurais artificiais apresentado no capítulo 3. O identificador é aplicado por simulação ao modelo de manipulador robótico de dois graus de liberdade apresentado no capítulo 4.

O capítulo 6 apresenta algumas considerações finais, bem como sugestões para trabalhos futuros.

Capítulo 2

Preliminares Teóricas

Este capítulo apresenta os conceitos teóricos preliminares necessários para compreensão do conteúdo da tese, objetivando um entendimento ou vocabulário comum sobre alguns conceitos e definições. Inicialmente, são apresentados conceitos sobre sistemas e sobre as formas de representá-los. Em seguida, são apresentados importantes tópicos sobre teoria de sistemas não lineares, otimização e aproximação de funções.

2.1 Introdução

UM conceito fundamental para as áreas científica, tecnológica, produtiva e humana é o conceito sobre o que é um **sistema**. A palavra sistema é usada genérica e vagamente no dia a dia da maioria das pessoas, sendo aplicada a qualquer coisa complexa com alguma forma de organização, mas que é difícil de ser especificada com precisão. Em muitas discussões técnicas, o termo tende a ser usado quando um termo mais apropriado não é disponível (Dandy e Warner, 1989).

Existem numerosas definições e conceitos sobre o que é um sistema. Koenig *et al.* (1967) definem como sendo “uma coleção de componentes que interagem”. Dandy e Warner (1989) definem como sendo “qualquer coleção de componentes interrelaciona-

dos que interagem e trabalham juntos de forma organizada para atingir determinado objetivo”. Close e Frederick (1995) definem como sendo “qualquer coleção de componentes em que os seus elementos interagem para que existam relações de causa-e-efeito entre eles ou entre suas variáveis”.

Estas definições são necessariamente gerais, pois devem abranger um grande número de coisas as quais possamos chamar de sistema. O mais importante nestas definições é que devem existir as interações dos componentes ou variáveis que compõem o sistema.

Em muitas situações, os diversos componentes de um sistema também podem ser considerados como um sistema ou apresentar as características de um sistema. Em tais situações, estes componentes de sistemas podem ser chamados de **subsistema**, e o sistema maior chamado de **metasistema**.

Em um sistema, os subsistemas podem se interrelacionar de diversas formas. Uma forma muito comum e importante é a hierárquica, com os diversos subsistemas sendo conectados de cima para baixo [estrutura *top-down*], formando níveis e subníveis de comando, controle e decisão. Tais sistemas são chamados de sistemas **hierárquicos**.

Em alguns sistemas, as suas variáveis, ou seus componentes, ou seus parâmetros, são dependentes de ocorrências no tempo passado. Tais sistemas são chamados de sistemas **dinâmicos**. Caso contrário, os sistemas são considerados como **estáticos**, ou seja, não existem mudanças de comportamento relacionadas com o tempo passado.

Sistemas que possuem parâmetros que possam ser ajustados por mecanismos externos para atingir determinado objetivo são chamados de sistemas **ajustáveis**. Sistemas que possuem a capacidade de auto-ajustar ou auto-adaptar seus parâmetros para atingir determinado objetivo são chamados de sistemas **adaptativos**. Um sistema que possui a capacidade de suportar e de manter-se em funcionamento mesmo com grandes perturbações externas e variações de sua dinâmica interna são chamados de sistemas **robustos**.

Um sistema é dito **físico** se ele está construído fisicamente. Um sistema é dito

computacional se ele está construído ou implementado em computadores. Adicionalmente, ainda podem ser biológicos, químicos, econômicos, sociais, políticos, etc.

Um sistema é dito de **controle** se ele possui a capacidade de intervir, ou controlar, ou modificar algum outro sistema com a finalidade de fazer com que este outro sistema atinja determinado objetivo ou tenha determinado desempenho. Um sistema é dito de **monitoramento** ou de **supervisão** se possui a capacidade de monitorar, ou supervisionar, ou acompanhar o funcionamento ou desempenho de algum outro sistema. Um sistema é dito **inteligente** se ele possui a capacidade de tomar decisões por si próprio, ou modificar-se, ou adaptar-se, até podendo interferir em em outros sistemas como se fosse dotado de algum tipo de inteligência¹.

Um outro termo, **processo**, também é de uso genérico e vago tanto na literatura técnica quanto na não técnica. Em um contexto genérico, o termo é aplicado a qualquer atividade ou procedimento que é executado progressivamente no tempo (Dandy e Warner, 1989). Entretanto, é habitual o uso deste termo como referência a sistemas químicos.

De uma maneira genérica, a categoria de sistemas de interesse principal dentro da engenharia é a dos sistemas **realizáveis**. Um sistema é dito realizável² se ele pode ser realizado ou implementado de alguma forma (por exemplo, sistemas físicos e químicos).

Deve-se notar que um mesmo sistema pode apresentar características de mais de um dos tipos de sistemas citados acima. Adicionalmente, existem outros inúmeros tipos e classificações para sistemas que deixam de ser abordados aqui por estarem fora do escopo do texto.

¹Muitos autores na literatura técnica classificam como sistemas inteligentes apenas aqueles que são construídos com base em algum modelo lógico, ou de pensamento humano, ou de estrutura neurofisiológica dos seres vivos, e tendo alguma forma de implementação computacional.

²Sistemas que podem ser implementados em contextos sociais também são considerados como sendo realizáveis, como por exemplo sistemas de saúde, sistemas de segurança pública, sistemas financeiro, mas de um modo geral estão fora do interesse da engenharia.

2.2 Representações de Sistemas

Os sistemas podem ser representados por **modelos**. Rosenberg e Karnopp (1983) definem modelos como sendo “simplificações da realidade”. Os modelos podem ser usados com diversos objetivos. Podem ser usados para se prever o comportamento de um sistema real, e portanto determinar como o mesmo irá funcionar em determinadas condições de operação (Dandy e Warner, 1989). Podem ser usados para que pessoas [usuários, projetistas, etc] ou outros sistemas [sistemas de controle, monitoramento, etc] controlem, ou monitorem, ou aprendam, ou se comuniquem, ou sejam treinados com base no modelo do sistema (Krick, 1969).

Um modelo quanto à forma de construção pode assumir duas denominações: **concreto**, quando é uma forma de representação implementada, como por exemplo protótipos; e **abstrato** quando é uma forma de representação não implementada, como por exemplo modelos matemáticos ou modelos descritivos.³

Os modelos físicos podem ser chamados de **icônicos** quando possuem alguma aparência similar ou equivalência de comportamento, podendo ser de dimensões reduzidas e menos detalhado que o sistema original ou desejado⁴ (Dandy e Warner, 1989). São chamados de **protótipos** quando possuem dimensões e níveis de detalhamento similares ao sistema original ou desejado, sendo geralmente usados para testes finais.

Os modelos abstratos podem assumir diversas formas. São **descritivos** quando expressos por meio de palavras ou termos rotulares que descrevam a sua forma construtiva ou funcional. São **gráficos** quando esboçados por alguma forma de desenho⁵. São **diagramáticos** quando são esboços gráficos da forma estrutural, ou comportamental, ou funcional do sistema⁶. São **matemáticos** quando esboçados equações

³Em geral, modelos computacionais são considerados como sendo abstratos.

⁴Considere como exemplo, a maquete de uma casa ou de uma unidade industrial. As aparências dessas maquetes são similares aos dos sistemas desejados ou implementados, mas possuem dimensões reduzidas e com uma quantidade de detalhes menor.

⁵Considere como exemplo, a planta de uma casa ou o esboço de um circuito elétrico.

⁶Considere como exemplo, o diagrama da estrutura administrativa de uma empresa qualquer.

matemáticas, sendo habitual na literatura técnica a aplicação do termo sistema aos modelos matemáticos.

Os modelos matemáticos, quanto à origem, podem ser **fenomenológicos**, quando oriundos de descrições e relações fenomenológicas existentes entre os componentes e variáveis que compõem o sistema, ou podem ser **identificados**, quando oriundos de ajustes paramétricos em modelos matemáticos propostos para descrever o sistema com base em algum critério de aproximação e especificação de tolerância a erros, e utilizando as variáveis de entrada e saída do sistema como informações para proceder com tais ajustes.

Quanto ao tipo, os modelos matemáticos podem assumir diversas formas em função de qual tipo de abordagem matemática é a mais apropriada para descrever um determinado sistema. Podem ser **formais** quando, por meio do modelo, é possível obter conhecimento ou informações a respeito do sistema, sem a necessidade de injeção de sinais de entrada e observação do comportamento da saída⁷, ou podem ser **não formais** em caso contrário.

A depender da natureza do sistema, existirá uma classe de modelos que melhor se adapta a modelagem do mesmo. Sistemas em que as mudanças de comportamento são observadas em instantes discretos de tempo, sistemas a **eventos discretos**, podem ser modelados por **redes de Petri**, ou por **álgebra de dioides**, ou por **automatos formais**, etc (Cassandras, 1993). Sistemas em que as mudanças de comportamento podem ser observadas continuamente no tempo, sistemas dinâmicos **contínuos**, podem ser modelados com base em modelos lineares variantes ou invariantes no tempo ou com base em modelos não lineares, podendo todos estes ainda serem de tempo contínuo ou discreto. Outros sistemas que reúnem as características de eventos discretos e dinâmica temporal são modelados por modelos **híbridos** (Labinaz *et al.*, 1996). Estas classes de modelos podem apropriadamente ser chamadas de formais.

⁷Considere como exemplo, modelos fenomenológicos, bem como modelos baseados na transformada de Laplace.

Ainda temos os modelos **determinísticos**, quando para um mesmo sinal injetado na entrada em instantes diferentes, as saídas se repetem, e os modelos **estocásticos** ou **probabilísticos**, quando a repetibilidade não existe.

Muitos sistemas apresentam níveis elevados de complexidade, ou então uma dinâmica de intensa não linearidade, que tornam os modelos formais lineares difíceis de serem usados. Alguns deles, como é o caso de muitos sistemas robóticos, possuem inclusive modelos fenomenológicos desenvolvidos, mas para algumas condições de operação estes se tornam imprecisos devido às elevadas variações paramétricas. Para estas situações, modelos não formais, tais como **redes neurais artificiais**, *wavelet* e *fuzzy*, estão sendo muito usados atualmente (Sjöberg *et al.*, 1995; Juditsky *et al.*, 1995).

Uma discussão completa a respeito de representações de sistemas por meio de modelos é extensa e complexa. A escolha de um modelo adequado para determinado sistema depende de diversos fatores, dentre os quais podemos citar a natureza do sistema, sua característica dinâmica ou estática, os objetivos para o uso do modelo, ferramentas de análise disponíveis, componentes disponíveis para implementação, erros de modelagem que podem ser admitidos, etc. Algo que deve ser destacado é que modelos são descrições simplificadas da realidade, e não é conveniente serem confundidos com o próprio sistema [a realidade], algo que ocorre com certa frequência⁸.

Este texto está limitado à discussão de modelos matemáticos determinísticos lineares e não lineares. A menos que referência contrária seja dada, devido ao uso habitual na literatura técnica, na seqüência deste capítulo o termo sistema será aplicado a modelos matemáticos lineares e não lineares.

⁸Considere como exemplo, algum modelo linear apresentado para representar um reator químico qualquer [sistema com comportamento altamente não linear]. Alguém ao analisar o modelo pode chegar a conclusão de que o sistema [o reator] é linear. Na realidade o sistema permanece apresentando um comportamento não linear, sendo linear apenas o modelo usado para representá-lo. Este modelo linear pode ser usado apenas para determinadas condições e regiões de operação.

2.3 Nomenclatura Básica

Na nomenclatura básica usada neste e nos demais capítulos, \mathbb{R} denota o conjunto dos números reais, \mathbb{R}_+^* é o conjunto dos reais não negativos, \mathbb{R}^n denota vetores de números reais, $\mathbb{R}^{n \times m}$ denota matrizes de número reais, com n e m pertencentes a \mathbb{N} , conjunto dos números naturais, e \mathbb{C} denota números complexos. A norma Euclidiana para vetores é denotada por $\|\mathbf{x}\|^2 := \mathbf{x}'\mathbf{x} = \sum_i x_i^2$ e a norma de *Frobenius* para matrizes é denotada por $\|\mathbf{A}\|^2 := \text{tr}(\mathbf{A}\mathbf{A}') = \sum_{i,j} a_{i,j}^2$. Adicionalmente, $x_{\max} = \max |x_i|$, e $a_{\max} = \max |a_{i,j}|$ denotam máximos, com $x_i \in \mathcal{X} \subset \mathbb{R}$ e $a_{i,j} \in \mathcal{A} \subset \mathbb{R}$, e $|\cdot|$ é o valor absoluto de x_i ou de $a_{i,j}$. É também usado $k \in \mathbb{Z}^+$ para representar instantes de tempo discretos. $f : \mathcal{U} \mapsto \mathcal{Y}$ denota um mapeamento [ou função] desde um conjunto [espaço ou subespaço] \mathcal{U} para um conjunto \mathcal{Y} . Um mapeamento composto [i.e., $g(f)$] será denotado por $g \circ f : \mathcal{U} \mapsto \mathcal{E} \mapsto \mathcal{Y}$, satisfazendo a $f : \mathcal{U} \mapsto \mathcal{E}$, $g : \mathcal{E} \mapsto \mathcal{Y}$.

2.4 Sistemas Não Lineares

A análise de sistemas não lineares geralmente envolve conceitos matemáticos mais avançados e complexos em detalhes que as abordagens por sistemas lineares. A menos que referência contrária seja fornecida, o conteúdo apresentado nesta seção pode ser encontrado em Vidyasagar (1993), Guckenheimer e Holmes (1983), Ruelle (1989), Parker e Chua (1989), Álvarez (1994).

Uma ampla variedade de sistemas físicos pode ser modelada com base em equações diferenciais ordinárias do tipo

$$\frac{d^n y(t)}{dt^n} = h \left(t, y(t), \dot{y}(t), \dots, \frac{d^{n-1} y(t)}{dt^{n-1}}, u(t) \right) \quad (2.1)$$

onde $h(\cdot)$ é uma função analítica, $t \in \mathbb{R}_+$ é o parâmetro tempo, $u(t) \in \mathbb{R}$ é o sinal ou variável de entrada ou simplesmente entrada, e $y(t) \in \mathbb{R}$ é o sinal ou variável de saída ou simplesmente saída.

Esta equação pode ser reescrita em forma de **representação de estado**. A representação de estado é um conjunto de n equações diferenciais ordinárias de ordem um, que descreve univocamente o comportamento das n variáveis linearmente independentes, escolhidas como relevantes [**variáveis de estados**]. As saídas podem ser obtidas em função das variáveis de estado, uma vez conhecida a entrada em qualquer momento e as condições das variáveis de estado no instante inicial.

O conjunto das variáveis de estado é chamado de **vetor de estado**. A este vetor está associado um **espaço de estado**, usualmente o \mathbb{R}^n ou um subespaço dele, onde n é o número de variáveis de estado.

Uma das formas de reescrita é a partir da definição de funções auxiliares na forma

$$\begin{aligned} x_1(t) &= y(t) \\ x_2(t) &= \dot{y}(t) \\ &\vdots \\ x_n(t) &= \frac{d^{n-1}y(t)}{dt^{n-1}}. \end{aligned}$$

Então a equação (2.1) pode ser reescrita na forma

$$\begin{aligned} \dot{x}_1(t) &= y(t) \\ \dot{x}_2(t) &= \dot{y}(t) \\ &\vdots \\ \dot{x}_n(t) &= h \left(t, y(t), \dot{y}(t), \dots, \frac{d^{n-1}y(t)}{dt^{n-1}}, u(t) \right). \end{aligned}$$

Definindo-se um vetor de estados $\mathbf{x}(t) \in \mathbb{R}^n$ e um mapeamento $f : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R} \mapsto$

\mathbb{R}^n , a representação de estado⁹ ganha o formato:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), u(t)), & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y}(t) = C \mathbf{x}(t) \end{cases} \quad (2.2)$$

para sistemas não autônomos¹⁰, onde $C \in \mathbb{R}^{1 \times n}$ mapeia \mathbf{x} para \mathbf{y} , ou ainda na forma:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), u(t)), & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y}(t) = C \mathbf{x}(t) \end{cases} \quad (2.3)$$

para sistemas autônomos.

Uma expressão mais genérica é encontrada na literatura técnica com o formato:

$$\begin{cases} \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), u(t)), & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y}(t) = g(\mathbf{x}(t)) \end{cases} \quad (2.4)$$

onde $g : \mathbb{R}^n \mapsto \mathbb{R}$ é uma relação algébrica entre \mathbf{x} e \mathbf{y} . Na maioria dos sistemas $g = C \mathbf{x}(t)$, e note que nesta equação esta implícito o mapeamento composto $g \circ f : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n \mapsto \mathbb{R}$. Note que todo o comportamento dinâmico do sistema pode ser analisado a partir do componente $\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), u(t))$ da equação (2.2), ou $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), u(t))$ da equação (2.3).

Definição 1 (Modelo Dinâmico Discreto no Tempo). Modelo [sistema] dinâmico discreto no tempo é uma representação matemática que permite descrever a evolução das variáveis de estado de um sistema numa seqüência de instantes discretos de tempo.

⁹Esta representação de estado foi feita para um sistema de única entrada e única saída, contudo o mesmo procedimento e forma de representação pode ser usado para sistemas com múltiplas entradas e múltiplas saídas.

¹⁰Em um sistema não autônomo, a variação do vetor de estados com o tempo é explícita. Em um sistema autônomo, ou invariante com o tempo, a variação do vetor de estado com o tempo é implícita (veja a definição 3).

Neste caso, o tempo não é considerado uma variável do domínio contínuo, mas sim uma seqüência de inteiros $k \in \mathbb{Z}^+ [k = 0, 1, 2, \dots]$ chamados de **tempos discretos**. A representação de estado discreta, também conhecida como recorrência ou transformação pontual, pode corresponder a um modelo para um sistema intrinsecamente discreto, ou à amostragem de um sistema contínuo no tempo. Esta representação discreta pode assumir a forma

$$\begin{cases} \mathbf{x}(k+1) = f(k, \mathbf{x}(k), u(k)), & \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(k) = C \mathbf{x}(k) \end{cases} \quad (2.5)$$

para sistemas não autônomos, ou ainda a forma

$$\begin{cases} \mathbf{x}(k+1) = f(\mathbf{x}(k), u(k)), & \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(k) = C \mathbf{x}(k) \end{cases} \quad (2.6)$$

para sistemas autônomos, onde $\mathbf{x}(k) \in \mathbb{R}^n$ e tem-se um mapeamento $f : \mathbb{Z}^+ \times \mathbb{R}^n \times \mathbb{R} \mapsto \mathbb{R}^n$.

O formato genérico encontrado na literatura técnica é

$$\begin{cases} \mathbf{x}(k+1) = f(k, \mathbf{x}(k), u(k)), & \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(k) = g(\mathbf{x}(k)) \end{cases} \quad (2.7)$$

onde $g : \mathbb{R}^n \mapsto \mathbb{R}$.

Definição 2 (Sistema Forçado e Sistema não Forçado). Um sistema é dito forçado se é explicitado a variável de entrada [veja a equação (2.2) ou a equação (2.4)], e dito não forçado no caso contrário [veja a equação (2.3) ou a equação (2.5)]

Em linhas gerais, na análise de sistemas pode-se supor que o sistema esteja sendo forçado por uma entrada conhecida e assim considerar o sistema como sendo não forçado.

Definição 3 (Sistema Autônomo e Sistema não Autônomo). Um sistema dinâmico contínuo [ou discreto] é dito autônomo se $\mathbf{x}(t) = f(t, \mathbf{x}(t))$ [ou $\mathbf{x}(k+1) = f(k, \mathbf{x}(k))$] é independente de t [ou k], e é dito não autônomo no caso contrário.

Definição 4 (Trajetória, Órbita e Fluxo). Trajetória é a evolução do vetor de estados no tempo contínuo. Órbita é a evolução do vetor de estados no tempo discreto. Fluxo é o conjunto de todas as trajetórias [ou órbitas para o caso do sistema ser discreto] possíveis a partir de diferentes condições iniciais, sendo denotado por $\phi_t : \mathcal{U} \mapsto \mathbb{R}^n$ [$\phi_t = \phi(t, \mathbf{x}(t))$ ou $\phi_t = \phi(\mathbf{x}(t))$] e sendo um mapeamento definido $\forall \mathbf{x} \in \mathcal{U}$ em algum intervalo de tempo $(a, b) \subseteq \mathbb{R}$, e ϕ satisfazendo a

$$\left. \frac{d}{dt}(\phi(t, \mathbf{x}(t))) \right|_{t=\tau} = f(\phi(\mathbf{x}(t), \tau))$$

para $\forall \mathbf{x} \in \mathcal{U}$ e $\forall \tau \in (a, b)$. Se o sistema é discreto temos que $\phi_k : \mathcal{U} \mapsto \mathbb{R}^n$ [$\phi_k = \phi(k, \mathbf{x}(k))$ ou $\phi = \phi_k(\mathbf{x}(k))$], e ϕ satisfazendo a

$$\left. \frac{d}{dk}(k, \phi(\mathbf{x}(k))) \right|_{k=\tau} = f(\phi(\mathbf{x}(k), \tau)).$$

Definição 5 (Conjunto Aberto e Conjunto Compacto). Um conjunto $\mathcal{U} \in \mathbb{R}^n$ é dito aberto se ele é limitado, e é dito compacto se ele é limitado e fechado.

Definição 6 (Função de Classe C^r). Uma função $h : \mathcal{U} \mapsto \mathbb{R}$, com $\mathcal{U} \subseteq \mathbb{R}$ e \mathcal{U} aberto, é dita de classe C^r se ela é r -vezes continuamente diferenciável, com $1 \leq r \leq \infty$.

Observação 1. Se uma função é de classe C^r , com $1 \leq r \leq \infty$, diz-se que ela é diferenciável. Se de classe C^0 , diz-se que ela é contínua; se de classe C^∞ , diz-se que ela é suave.

Definição 7 (Mapeamento de Classe C^r). Seja $f : \mathcal{U} \mapsto \mathcal{Y}$, com $\mathcal{U} \subseteq \mathbb{R}^n$ e $\mathcal{Y} \subseteq \mathbb{R}^m$, um mapeamento com $i = 1, \dots, m$ componentes de funções f_i , então o mapeamento f é dito de classe C^r se todos os f_i são de classe C^r , com $1 \leq r \leq \infty$.

Observação 2. Se um mapeamento é de classe C^r , com $1 \leq r \leq \infty$, diz-se que ele é diferenciável. Se de classe C^0 , diz-se que ele é contínuo; se de classe C^∞ , diz-se que ele é suave.

Definição 8 (Mapeamentos Injetores, Sobrejetores e Inversíveis). Um mapeamento [ou função] $f : \mathcal{U} \mapsto \mathcal{Y}$ é dito injetor se não existem dois pontos pertencentes a \mathcal{U} mapeados para um mesmo ponto de \mathcal{Y} . Um mapeamento $f : \mathcal{U} \mapsto \mathcal{Y}$ é dito sobrejetor se para cada ponto $\mathbf{y} \in \mathcal{Y}$, existe no mínimo um ponto em \mathcal{U} que é mapeado para \mathbf{y} por f . Um mapeamento $f : \mathcal{U} \mapsto \mathcal{Y}$ é inversível, com sua inversa denotada por $f^{-1} : \mathcal{Y} \mapsto \mathcal{U}$, se ele é injetor e sobrejetor ao mesmo tempo.

Definição 9 (Difeomorfismo). Um mapeamento $f : \mathcal{U} \mapsto \mathcal{Y}$ é um difeomorfismo de classe C^r se tanto f quanto f^{-1} são mapeamentos de classe C^r .

Observação 3. Se um mapeamento é um difeomorfismo de classe C^1 , ele é chamado simplesmente de difeomorfismo. Se um mapeamento é um difeomorfismo de classe C^0 , ele é chamado de homeomorfismo.

Definição 10 (Vizinhança). Uma vizinhança de um ponto \mathbf{x} é um conjunto aberto \mathcal{U} que contém \mathbf{x} no seu interior.

Definição 11 (Conjunto Denso). Um conjunto $\mathcal{X} \subset \mathcal{Y} \subset \mathbb{R}^n$ é dito denso se para cada $\mathbf{y} \in \mathcal{Y}$ e cada vizinhança aberta \mathcal{U} de \mathbf{y} é tipo que $\mathcal{U} \cap \mathcal{X}$ é não vazia para.

Definição 12 (Região de Densidade). Seja $\phi_t : \mathcal{U} \mapsto \mathbb{R}^n$ um fluxo definido $\forall \mathbf{x} \in \mathcal{U}$. A região \mathcal{U} é dita região de densidade se é limitada no espaço e totalmente preenchida pelo fluxo.

Definição 13 (Manifold ou Variedade). Uma superfície \mathcal{M} é chamada de manifold ou variedade [ou ainda superfície localmente euclidiana] de dimensão n , denotado por Mfold^n se ela é definida de tal modo que $\forall \mathbf{x} \in \mathcal{M} \exists$ uma vizinhança aberta \mathcal{U} de \mathbf{x} para a qual existe um difeomorfismo.

Em linhas gerais, um manifold ou superfície localmente euclidiana é uma região onde localmente estão associadas propriedades lineares e não lineares, bem como sub-espacos (manifolds) estáveis e instáveis (Guckenheimer e Holmes, 1983; Bacciotti, 1992; Glendinning, 1994).

Definição 14 (Constante de Lipschitz e Continuidade Lipschitz Local). Um mapeamento $f : \mathcal{U} \mapsto \mathcal{Y}$, onde $\mathcal{U} \subset \mathbb{R}^n$ é um subespaço euclidiano aberto ou uma superfície diferenciável \mathcal{M} , tem continuidade Lipschitz local em relação a um \mathbf{x}^* , se é satisfeita à condição

$$\|f(x) - f(x^*)\| \leq k \|\mathbf{x} - \mathbf{x}^*\|$$

para cada $\mathbf{x} \in \mathcal{U}$ tal que $\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon$, para algum $\varepsilon > 0$ e para algum $0 < k < \infty$, onde $\|\cdot\|$ é a norma euclidiana para vetores. Diz-se que k é a constante de Lipschitz para f .

Definição 15 (Matriz Jacobiana). Seja $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ um mapeamento continuamente diferenciável com respeito a cada um de seus argumentos, e seja definida em uma vizinhança $\mathcal{U} \subset \mathbb{R}^n$ de raio $r > 0$, sendo que \mathcal{U} é \mathbf{Mfold}^n . Adicionalmente, seja $\xi = \mathbf{x} - \mathbf{x}^*$, com $\xi \in \mathcal{U}$. Chama-se de matriz jacobiana, ou simplesmente jacobiano, a matriz $n \times n$ no formato:

$$\mathbf{D}_f(\mathbf{x}^*) = \left[\frac{\partial f}{\partial \mathbf{x}} \right]_{\mathbf{x}=\mathbf{x}^*}$$

tal que:

$$\dot{\xi} = \mathbf{D}_f(\mathbf{x}^*) \xi$$

sendo então dito que \mathbf{D}_f é a linearização de f em torno de \mathbf{x}^* .

Teorema 1 (Teorema da Função Inversa). *Seja $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ continuamente diferenciável em $\mathbf{x}^* \in \mathbb{R}^n$, e seja $\mathbf{y}^* = f(\mathbf{x}^*)$. Adicionalmente seja $\mathbf{D}_f(\mathbf{x}^*)$ não singular. Então existem conjuntos abertos $\mathcal{U} \subseteq \mathbb{R}^n$ contendo \mathbf{x}^* e $\mathcal{V} \subseteq \mathbb{R}^n$ contendo \mathbf{y}^* , tal que f é um difeomorfismo de \mathcal{U} para \mathcal{V} .*

2.4.1 Sobre Soluções de Equações de Estado

Teorema 2 (Teorema da Existência e Unicidade). *Seja $\mathcal{U} \subset \mathbb{R}^n$ aberto, seja um mapeamento $f : \mathcal{U} \mapsto \mathbb{R}^n$ com continuidade Lipschitz local em relação a $\forall \mathbf{x} \in \mathcal{U}$, e seja*

$\mathbf{x}_0 \in \mathcal{U}$. Então existe alguma constante $c > 0$, e uma única solução $\phi(\mathbf{x}_0, \cdot) : (-c, c) \mapsto \mathcal{U}$ [ϕ é um fluxo] satisfazendo a equação $\dot{\mathbf{x}} = f(\mathbf{x})$ com condição inicial $\mathbf{x}(0) = \mathbf{x}_0$.

Definição 16 (Ponto de Equilíbrio). Um ponto \mathbf{x}^* é dito de equilíbrio se ele satisfaz a condição:

$$f(t, \mathbf{x}^*) = 0 \quad \forall t.$$

Se o sistema é discreto, \mathbf{x}^* é um ponto de equilíbrio se satisfaz a condição:

$$f(k, \mathbf{x}^*) = 0 \quad \forall k$$

(Vidyasagar, 1993).

Definição 17 (Ponto de Equilíbrio Isolado). Um ponto de equilíbrio \mathbf{x}^* é dito isolado se existe uma vizinhança \mathcal{U} de \mathbf{x}^* que não contém nenhum outro ponto de equilíbrio, ou seja existe algum $\delta > 0$ tal que $f(\mathbf{x}) \neq 0$ onde quer que exista $0 < |\mathbf{x} - \mathbf{x}^*| < \delta$ (Vidyasagar, 1993).

Definição 18 (Mapeamento de Morse). Um mapeamento [ou uma função] é chamado de mapeamento de Morse se ele possui apenas pontos de equilíbrios isolados (Golubitsky e Guillemin, 1973).

Observação 4. Todo mapeamento de Morse de classe C^r é aberto e denso em C^r (Golubitsky e Guillemin, 1973).

Definição 19 (Ponto de Equilíbrio Atrativo). Um ponto de equilíbrio \mathbf{x}^* é dito atrativo se, para cada $t_0 \in \mathbb{R}_+$, existe um $\eta(t_0) > 0$ e uma função $s(t_0 + t, t_0, \|\mathbf{x}(t) - \mathbf{x}^*\|)$ tal que

$$\|\mathbf{x}^*\| < \eta(t_0) \Rightarrow s(t_0 + t, t_0, \|\mathbf{x} - \mathbf{x}^*\|) \rightarrow 0$$

quando $t \rightarrow \infty$. Se o sistema é discreto, um ponto de equilíbrio \mathbf{x}^* é dito atrativo se, para cada $k_0 \in \mathbb{Z}_+$, existe um $\eta(k_0) > 0$ e uma função $s(k_0 + k, k_0, \|\mathbf{x}(k) - \mathbf{x}^*\|)$ tal que

$$\|\mathbf{x}^*\| < \eta(k_0) \Rightarrow s(k_0 + k, k_0, \|\mathbf{x}(k) - \mathbf{x}^*\|) \rightarrow 0$$

quando $k \rightarrow \infty$ (Vidyasagar, 1993).

Adicionalmente, existem outros conceitos concernentes à soluções de equações de estado, tais como: pontos e soluções periódicas, quase periódicas, conjuntos limites, atratores estranhos, etc. Estes conceitos não serão abordados aqui por não estarem no contexto direto do trabalho.

2.4.2 Sobre a Estabilidade de Sistemas Segundo Lyapunov

A estabilidade é um tópico de elevada importância em análises de sistemas dinâmicos, pois determina qual a condição dinâmica [ou comportamental] de um ponto de equilíbrio. Do ponto de vista de engenharia, a estabilidade é importante porque nos sistemas reais podem existir perturbações externas aos sistemas que podem levá-los a situações de instabilidade, bem como condições de operações [ou pontos de operações] que apresentam comportamentos instáveis, periódicos, quase-periódicos, ou até mesmo caóticos.

A teoria de Lyapunov para estabilidade de sistemas dinâmicos pode ser resumida em três categorias: estabilidade; estabilidade assintótica; e estabilidade assintótica global (Lyapunov, 1892). O método apresentado na seqüência é conhecido como **Segundo Método de Lyapunov** ou **Método Direto de Lyapunov**.

Do ponto de vista matemático, em um sistema dinâmico não linear há a possibilidade de existência de vários pontos de equilíbrios, bem como a inexistência dos mesmos. Assim no estudo da estabilidade desses sistemas é permitida a sua análise qualitativa e quantitativa, mas isso não é sempre possível de ser feito de forma global.

Definição 20 (Pontos de Equilíbrio Estáveis e Instáveis). Um ponto de equilíbrio \mathbf{x}^* é dito estável em t_0 se, para cada $\varepsilon > 0$, existe um $\delta(t_0, \varepsilon) > 0$ tal que

$$\|\mathbf{x}(t_0) - \mathbf{x}^*\| < \delta(t_0, \varepsilon) \Rightarrow \|\mathbf{x}(t) - \mathbf{x}^*\| < \varepsilon \quad \forall t \geq t_0$$

onde $\|\cdot\|$ é a norma euclidiana para vetores. É dito ponto de equilíbrio instável no caso contrário. Se sistema é discreto, um ponto de equilíbrio \mathbf{x}^* é dito estável em k_0

se, para cada $\varepsilon > 0$, existe um $\delta(k_0, \varepsilon) > 0$ tal que

$$\|\mathbf{x}(k_0) - \mathbf{x}^*\| < \delta(k_0, \varepsilon) \Rightarrow \|\mathbf{x}(k) - \mathbf{x}^*\| < \varepsilon \quad \forall k \geq k_0.$$

(Vidyasagar, 1993).

Definição 21 (Ponto de Equilíbrio Assintoticamente Estável). Um ponto de equilíbrio \mathbf{x}^* é dito assintoticamente estável em t_0 se:

- (i) É estável em t_0 ;
- (ii) É atrativo;
- (iii) Existe um número $\delta_1(t_0)$ tal que

$$\|\mathbf{x}(t_0) - \mathbf{x}^*\| < \delta_1(t_0) \Rightarrow \lim_{t \rightarrow \infty} \|\mathbf{x}(t) - \mathbf{x}^*\| = 0.$$

Se o sistema é discreto, um ponto de equilíbrio \mathbf{x}^* é dito assintoticamente estável em k_0 se:

- (i) É estável em k_0 ;
- (ii) É atrativo;
- (iii) Existe um número $\delta_1(k_0)$ tal que

$$\|\mathbf{x}(k_0) - \mathbf{x}^*\| < \delta_1(k_0) \Rightarrow \lim_{k \rightarrow \infty} \|\mathbf{x}(k) - \mathbf{x}^*\| = 0.$$

Ponto de equilíbrio assintoticamente estável é definição dada por Vidyasagar (1993). Glendinning (1994) chama este tipo de ponto de equilíbrio de **base de atração** e Marino e Tomei (1995) chama-o de **domínio de atração**.

Os conceitos de estabilidade introduzidos até aqui são de natureza local. Eles permitem apenas analisar o comportamento da trajetórias de um sistema para condições iniciais próximas ao ponto de equilíbrio.

Definição 22 (Função Definida Positiva). Uma função $\mathcal{V} : \mathbb{R}_+ \times \mathbb{R}^n \mapsto \mathbb{R}$ é chamada de função definida positiva se

- (i) $\mathcal{V}(\mathbf{x})$ é de classe C^1 .
- (ii) $\mathcal{V}(0) = 0$.
- (iii) $\mathcal{V}(\mathbf{x}) > 0 \forall \mathbf{x} \in \mathcal{U} \subseteq \mathbb{R}^n$ tal que $\mathbf{x} \neq 0$.

É dita definida positiva e **decrecente** se existe uma função α tal que:

- (iv) $\mathcal{V}(\mathbf{x}) \leq \alpha(\|\mathbf{x}\|) \forall \mathbf{x} \in \mathcal{U} \subseteq \mathbb{R}^n$

(Vidyasagar, 1993).

Definição 23 (Função de Lyapunov). Uma função \mathcal{V} é dita de Lyapunov se ela é uma função definida positiva e sua primeira derivada é não positiva ($\dot{\mathcal{V}} \leq 0$ para sistemas contínuos ou $\Delta\mathcal{V} = \mathcal{V}(k+1) - \mathcal{V}(k) \leq 0$ para sistemas discretos) em uma vizinhança $\mathcal{U} \subseteq \mathbb{R}^n$ (Vidyasagar, 1993).

Visto ser \mathcal{V} uma função escalar associado a um vetor de estado, ela pode ser, por exemplo, uma norma, ou ter uma interpretação física, tal como o nível de energia de um sistema dinâmico. Intuitivamente, uma diminuição do nível de energia quando a trajetória aproxima-se de um ponto de equilíbrio corresponde a um comportamento estável ($\dot{\mathcal{V}} \leq 0$) ou assintoticamente estável ($\dot{\mathcal{V}} < 0$) (Álvarez, 1994).

Teorema 3 (Teorema da Estabilidade Assintótica). *Um ponto de equilíbrio \mathbf{x}^* é assintoticamente estável se existe uma função de Lyapunov decrescente \mathcal{V} definida em uma vizinhança $\mathcal{U} \subseteq \mathbb{R}^n$ de tal ponto, tal que $-\dot{\mathcal{V}}$ também é definida positiva e não decrescente, onde $\dot{\mathcal{V}} < 0$ é avaliada ao longo das trajetórias do sistema. Se o sistema é discreto, $-\Delta\mathcal{V}$ deve ser definida positiva e não decrescente (Vidyasagar, 1993).*

O método de análise de estabilidade de Lyapunov foi pela primeira vez aplicado em controle de sistemas por Kalman e Bertram (1960*a*; 1960*b*). O método de Lyapunov não requer que a solução das equações do sistema seja conhecida, o que o torna bastante útil principalmente para aplicações em sistemas não lineares, onde as soluções são em geral não conhecidas.

Uma real dificuldade no uso do método é encontrar uma função de Lyapunov apropriada [dita função candidata de Lyapunov]. A abordagem mais comum é o uso de uma função de energia para o sistema.

Uma questão que deve ser considerada é que os sistemas físicos são modelados de forma simplificada, existindo portanto erros de modelagem. Além disso, os mesmos podem estar expostos a estímulos ou perturbações externas não previstas na modelagem. Assim, é necessário a consideração da estabilidade de sistemas sobre situações de perturbações externas.

Considere agora um sistema sujeito a perturbações externas ou a erros de modelagem na forma:

$$\dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t)) + e(t), \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (2.8)$$

onde a função erro $e(t)$ possui a mesma dimensão de $f(t, \mathbf{x}(t))$, e representa as perturbações externas e/ou os erros de modelagem. Neste modelo, é considerado que o erro é pequeno, e que qualitativamente os comportamentos do sistema original e do sistema perturbado sejam similares.

Definição 24 (Estabilidade Sob Perturbação). Seja \mathbf{x}^* um ponto de equilíbrio de um sistema não perturbado [i.e., uma solução da equação (2.8) para $e(t) = 0$ ou para $e(k) = 0$ se o sistema é discreto], e \mathbf{x}_0 uma condição inicial qualquer. O ponto \mathbf{x}^* é dito estável sob perturbação se $\forall \epsilon > 0$ existe um $\delta_1(\epsilon)$ e um $\delta_2(\epsilon)$, tal que $\|\mathbf{x}^*\| < \delta_1$ e $\|e(t)\| < \delta_2, \forall t > t_0$, implicando que $\|\mathbf{x}(t)\| < \epsilon \forall t > t_0$ (Seibert, 1963).

Teorema 4 (Teorema da Estabilidade Sob Perturbação). *Se f possui continuidade Lipschitz local em relação a um ponto de equilíbrio \mathbf{x}^* , então f é estável sob perturbação se, e somente se, \mathbf{x}^* é assintoticamente estável (Seibert, 1963).*

2.4.3 Sobre as Propriedades Genéricas

Em sistemas físicos, apenas propriedades fisicamente significativas permanecem válidas quando o sistema é levemente deformado ou perturbado, sendo assim uma propriedade estável. Informalmente, dado um conjunto \mathcal{Y} , P é uma propriedade genérica se quase todos os membros de \mathcal{Y} exibem a propriedade P .

Definição 25 (Propriedade Genérica). Uma propriedade P referente aos membros de um conjunto \mathcal{Y} é uma propriedade genérica se o subconjunto de \mathcal{Y} cujos membros exibem a propriedade P é denso e aberto (Parker e Chua, 1989).

Um exemplo de propriedade genérica é a inversibilidade de uma matriz $\mathbf{A} \in \mathbb{R}^{n \times n}$. A matriz \mathbf{A} é inversível se $\det(\mathbf{A}) \neq 0$, assim a inversibilidade é uma propriedade genérica e estável para o subconjunto de matrizes $\mathbf{A} \in \mathbb{R}^{n \times n}$ que podem ser invertidas.

2.5 Observabilidade de Sistemas

A observabilidade é uma das questões fundamentais em teoria de Controle e foi apresentada sob um ponto de vista linear no início dos anos 60 (Kalman, 1960). A observabilidade é uma suposição necessária em problemas de reconstrução de sinais [i.e., o problema de se reconstruir a trajetória do vetor de estado em correspondência à trajetória do vetor de saída]. A observabilidade é importante também no estudo de turbulência e caos (Aeyels, 1981).

Considere o sistema não linear Σ_c a seguir,

$$\Sigma_c : \begin{cases} \dot{\mathbf{x}}(t) = f(t, \mathbf{x}(t), \mathbf{u}(t)), & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) \end{cases} \quad (2.9)$$

onde $f : \mathbb{R}_+ \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$, e $\mathbf{C} \in \mathbb{R}^{s \times n}$. Considere ainda $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$, $\mathbf{y} \in \mathcal{Y} \subset \mathbb{R}^s$ e f continuamente diferenciável.

A variável de controle \mathbf{u} representa o sinal de controle aplicado externamente, e a variável de saída \mathbf{y} representa os componentes do vetor de estado do sistema que são observados ou medidos na saída. O vetor de estado \mathbf{x} é usado para representar a memória do sistema, e pode não ter todos os seus componentes medidos ou mensuráveis.

Definição 26 (Estados Distinguíveis). Considere um sistema descrito pela equação (2.9). Dois estados $\mathbf{x}_0 \in \mathcal{X}$ e $\mathbf{x}_1 \in \mathcal{X}$ são ditos como sendo distinguíveis se existe uma entrada $\mathbf{u}(\cdot) \in \mathcal{U}$ tal que $\mathbf{y}(\cdot, \mathbf{x}_0, \mathbf{u}) \neq \mathbf{y}(\cdot, \mathbf{x}_1, \mathbf{u})$ (Vidyasagar, 1993).

De acordo com Hermann e Krener (1977), o espaço de estado \mathcal{X} pode ser deficiente de duas maneiras distintas:

- (i) O espaço de estado \mathcal{X} pode ser muito pequeno para adequadamente representar totalmente a variedade de estados de memória do sistema [i.e, pode falhar em distinguir diferentes estados correspondentes a diferentes entradas aplicadas, as quais exigem a observação de diferentes efeitos no sistema], e neste caso, o modelo matemático do sistema não representa adequadamente o sistema real.
- (ii) O espaço de estado \mathcal{X} pode ser muito grande, e neste caso o sistema pode não ser controlável [i.e., se Σ_c é estava em um dado estado $\mathbf{x}_0 \in \mathcal{X}$ em algum tempo t_0 , ele pode estar em um outro estado $\mathbf{x} \in \mathcal{X}$ tal que não seria possível o sistema ter sido conduzido para este estado a partir do conjunto de entrada \mathcal{U}], ou ainda pode ser que estados distintos sejam não distinguíveis do ponto de vista entrada-saída [i.e, se uma mesma entrada é aplicada a qualquer um desses estados ter-se-á como resultado a mesma saída].

Definição 27 (Sistema Observável). O sistema dinâmico Σ_c é dito observável se, para dois estados iniciais distintos $\mathbf{x}_0 \in \mathcal{X}$ e $\mathbf{x}_1 \in \mathcal{X}$, existe um $t_1 > t_0$ tal que conhecido um perfil de entrada $\mathbf{u}_{[t_0, t_1]} \in \mathcal{U}$, as saídas são distintas para os dois estados iniciais [i.e., $\mathbf{y}_{[t_0, t_1]}(\mathbf{x}_0) \neq \mathbf{y}_{[t_0, t_1]}(\mathbf{x}_1)$] (Levin e Narendra, 1995).

A observabilidade é uma propriedade do modelo matemático, e não do sistema físico, sendo invariante em relação ao perfil de sinal de entrada aplicado. Para um

modelo linear observável de ordem n , qualquer perfil de entrada $\mathbf{u}_{[t_0, t_1]}$ fará com que dois estados iniciais diferentes sejam distinguíveis um do outro. Se dois estados não são distinguíveis um do outro para um perfil de entrada escolhido aleatoriamente, então eles não podem ser distinguíveis para nenhum outro perfil de entrada.

Neste caso, o comportamento entrada-saída do sistema físico pode ser descrito por um modelo matemático observável de menor dimensão, através do uso de transformação de equivalência, onde cada estado no modelo equivalente representa uma classe equivalente correspondente ao conjunto de estados que poderiam ser distinguíveis no modelo original.

Ao contrário do conceito de observabilidade para modelos lineares, a definição de observabilidade para modelos não lineares é mais envolvente e menos restritiva. Note pela definição 27, que a observabilidade é um conceito global para modelos matemáticos, e ela permite viajar por um longo período nos sinais do sistema até que se obtenha a distinção dos estados sobre \mathcal{X} . No caso de modelos lineares esta viagem é restrita à dimensão da memória do sistema [i.e., à ordem do sistema].

Aeyels (1981) mostrou que particularmente os modelos matemáticos não lineares discretos apresentam a propriedade da observabilidade como sendo uma propriedade genérica (veja a seção 2.4.3). Neste caso, o sistema torna-se observável se é permitida a obtenção de $2n + 1$ amostras da órbita de saída, e não menos que isso, onde n é a ordem no sistema. Levin e Narendra (1995) usaram os resultados de Aeyels (1981) para a elaboração de identificadores baseados na propriedade da observabilidade genérica, o que será discutido com mais detalhes em capítulo posterior.

2.6 Métodos de Otimização Não Linear

A concepção de otimização é atualmente bem arraigada em áreas de ciências e engenharia como uma ferramenta básica para análise de problemas complexos, auxílio à tomada de decisão e alocação de recursos. A idéia da otimização é a seleção adequada

de valores para um conjunto de variáveis tendo como base um objetivo a ser alcançado, tal que este objetivo especifique um desempenho para o sistema. Dependendo da formulação, o objetivo pode ser minimizado ou maximizado, e ainda estar sujeito a restrições.

Nas últimas décadas, diversos métodos ou algoritmos de otimização têm sido propostos e desenvolvidos. Alguns desses métodos são baseados em teorias de matemática formal, como por exemplo os métodos baseados no gradiente e os baseados na matriz hessiana. Outros métodos são baseados em teorias matemáticas não formais, em métodos heurísticos de buscas e em algoritmos genéticos.

Os métodos formais possibilitam em geral o encontro da solução desejada de uma forma estável e segura do ponto de vista matemático. Os métodos não formais em geral não asseguram uma convergência estável para a solução desejada. No entanto, na prática, muitas vezes solucionam de forma mais rápida os problemas. Aqui nos concentraremos aos métodos formais.

Considere o problema de minimizar uma função $f(\mathbf{x})$ tal que $f : \mathbb{R}^n \mapsto \mathbb{R}$ e $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$. \mathcal{X} é comumente chamado de conjunto factível.

Definição 28 (Matriz Definida Positiva). Uma matriz simétrica \mathbf{A} é dita definida positiva se $\mathbf{x}' \mathbf{A} \mathbf{x} > 0 \quad \forall \mathbf{x} \neq 0$. É dita **semidefinida positiva** se $\mathbf{x}' \mathbf{A} \mathbf{x} \geq 0 \quad \forall \mathbf{x} \neq 0$. É dita **definida negativa** se $\mathbf{x}' \mathbf{A} \mathbf{x} < 0 \quad \forall \mathbf{x} \neq 0$. É dita **semidefinida negativa** se $\mathbf{x}' \mathbf{A} \mathbf{x} \leq 0 \quad \forall \mathbf{x} \neq 0$. É dita indefinida se $\mathbf{x}' \mathbf{A} \mathbf{x}$ é positiva para alguns valores de \mathbf{x} e negativa para outros (Luenberger, 1984).

Definição 29 (Vetor Gradiente). Seja $f \in C^1$. Chama-se vetor gradiente, denotado por $\nabla_{\mathbf{x}} f$, ao vetor construído pelas primeiras derivadas parciais de f em relação a cada

um dos componentes de \mathbf{x} no formato

$$\nabla_{\mathbf{x}}f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}.$$

Definição 30 (Matriz Hessiana). Seja $f \in C^2$. Chama-se de matriz hessiana, denotada por $\nabla_{\mathbf{x}}^2f$, a matriz simétrica $n \times n$ construída pelas primeiras derivadas parciais de $\nabla_{\mathbf{x}}f$ em relação a cada um dos componentes de \mathbf{x} no formato

$$\nabla_{\mathbf{x}}^2f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_n^2} \end{bmatrix}.$$

Definição 31 (Ponto Interior). Um ponto $\mathbf{x}^* \in \mathcal{X}$ é dito interior se existe um número $\epsilon > 0$ tal que $\|\mathbf{x} - \mathbf{x}^*\| < \epsilon \quad \forall \mathbf{x} \in \mathcal{X}$ (Luenberger, 1984).

Definição 32 (Mínimo Local). Um ponto \mathbf{x}^* é dito mínimo local de f em \mathcal{X} se, e somente se, existe um $\epsilon > 0$ tal que $f(\mathbf{x}) \geq f(\mathbf{x}^*) \quad \forall \mathbf{x} \in \mathcal{X}$ com $\|\mathbf{x} - \mathbf{x}^*\| < \epsilon$. Se $f(\mathbf{x}) > f(\mathbf{x}^*) \quad \forall \mathbf{x} \in \mathcal{X}$ tal que $\mathbf{x} \neq \mathbf{x}^*$ e $\|\mathbf{x} - \mathbf{x}^*\| < \epsilon$, é dito que \mathbf{x}^* é um mínimo local estrito [i.e, isolado] em \mathcal{X} (Luenberger, 1984).

Definição 33 (Mínimo Global). Um ponto \mathbf{x}^* é dito mínimo global de f em \mathcal{X} se, e somente se, $\forall \mathbf{x} \in \mathcal{X}$ tem-se que $f(\mathbf{x}) \geq f(\mathbf{x}^*)$. Se $f(\mathbf{x}) > f(\mathbf{x}^*) \forall \mathbf{x} \in \mathcal{X}$ tal que $\mathbf{x} \neq \mathbf{x}^*$, é dito que \mathbf{x}^* é um mínimo global estrito [i.e, isolado] em \mathcal{X} (Luenberger, 1984).

De forma análoga, podem ser definidos pontos de máximo local e global. Deve ser observado que maximizar $f(\mathbf{x})$ é equivalente à minimizar $-f(\mathbf{x})$. Assim, genericamente será usado o termo minimizar.

Para se atribuir a um ponto o rótulo de mínimo, seja ele local ou global, é necessário que certas condições sejam satisfeitas. As proposições a seguir estabelecem essas condições. A idéia é considerar o movimento de um ponto \mathbf{x} na direção de \mathbf{x}^* .

Proposição 1 (Condições Necessárias de Primeira Ordem). *Seja $f \in C^1$. Se \mathbf{x}^* é um mínimo local de f em \mathcal{X} , então para qualquer $\delta\mathbf{x} \in \mathcal{X}$, tal que $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}^*$ e que \mathbf{x} esteja se movimentando na direção de \mathbf{x}^* , é tido que $\delta\mathbf{x}' \nabla_{\mathbf{x}} f(\mathbf{x}) \geq 0$ (Luenberger, 1984).*

Corolário 1 (Caso Sem Restrições). *Seja $f \in C^1$. Se \mathbf{x}^* é um mínimo local de f em \mathcal{X} , então $\nabla_{\mathbf{x}} f(\mathbf{x}^*) = 0$ (Luenberger, 1984).*

Proposição 2 (Condições Necessárias de Segunda Ordem). *Seja $f \in C^2$. Se \mathbf{x}^* é um mínimo local de f em \mathcal{X} , então para qualquer $\delta\mathbf{x} \in \mathcal{X}$, tal que $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}^*$ e que \mathbf{x} esteja se movimentando na direção de \mathbf{x}^* , é tido que (Luenberger, 1984)*

$$(i) \delta\mathbf{x}' \nabla_{\mathbf{x}} f(\mathbf{x}) \geq 0.$$

$$(ii) \text{ Se } \delta\mathbf{x}' \nabla_{\mathbf{x}} f(\mathbf{x}) = 0, \text{ então } \delta\mathbf{x}' \nabla_{\mathbf{x}}^2 f(\mathbf{x}) \delta\mathbf{x} \geq 0.$$

Proposição 3 (Condições Necessárias de Segunda Ordem - Caso Sem Restrições). *Se \mathbf{x}^* é um ponto interior de \mathcal{X} , e supondo que \mathbf{x}^* é um ponto de mínimo local de $f \in C^2$ em \mathcal{X} , então (Luenberger, 1984)*

$$(i) \delta\mathbf{x}' \nabla_{\mathbf{x}} f(\mathbf{x}^*) = 0.$$

$$(ii) \delta \mathbf{x}' \nabla_{\mathbf{x}}^2 f(\mathbf{x}^*) \delta \mathbf{x} \geq 0 \quad \forall \delta \mathbf{x} \in \mathcal{X}.$$

Proposição 4 (Condições Suficientes de Segunda Ordem - Caso Sem Restrições). *Seja $f \in C^2$ uma função definida em uma região onde \mathbf{x}^* é um ponto interior de \mathcal{X} . Supondo em adição que*

$$(i) \nabla_{\mathbf{x}} f(\mathbf{x}^*) = 0.$$

$$(ii) \nabla_{\mathbf{x}}^2 f(\mathbf{x}^*) \text{ é definida positiva.}$$

Então, \mathbf{x}^* é um mínimo local estrito de f (Luenberger, 1984).

Determinar se um ponto é um mínimo local ou global é uma tarefa árdua. Em alguns casos, esta determinação é possível se a função objetivo possui convexidade. Este conceito é introduzido na seqüência.

Definição 34 (Conjunto Convexo). Um conjunto $\mathcal{X} \subset \mathbb{R}^n$ é dito convexo se para todo $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, existe um número real $\lambda \in [0, 1]$ que verifica $\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2 \in \mathcal{X}$ (Luenberger, 1984).

Proposição 5. *Conjuntos convexo em \mathbb{R}^n satisfazem as seguintes relações (Luenberger, 1984):*

(i) *Se $\mathcal{X} \subset \mathbb{R}^n$ é um conjunto convexo e β é um número real, o conjunto*

$$\beta \mathcal{X} = \{\mathbf{x} : \mathbf{x} = \beta \mathbf{x}_1, \mathbf{x}_1 \in \mathcal{X}\}$$

é convexo;

(ii) *Se \mathcal{X} e \mathcal{Y} são conjuntos convexos, então o conjunto*

$$\mathcal{X} + \mathcal{Y} = \{\mathbf{x} : \mathbf{x} = \mathbf{x} + \mathbf{y}, \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$$

é convexo;

(iii) A intersecção de qualquer coleção de conjuntos convexos é convexa.

Definição 35 (função Convexa). Uma função definida sobre um conjunto convexo $\mathcal{X} \subset \mathbb{R}^n$ é dita convexa se, e somente se, para todo $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, existe um número real $\lambda \in [0, 1]$ que verifica

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2).$$

Se para todo $\lambda \in [0, 1]$ e $\mathbf{x}_1 \neq \mathbf{x}_2$ é válido

$$f(\lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2) < \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2)$$

é dito que f é estritamente convexa (Luenberger, 1984).

Proposição 6. Funções convexas sobre algum conjunto $\mathcal{X} \subset \mathbb{R}^n$ satisfazem as seguintes relações (Luenberger, 1984):

- (i) Se f é uma função convexa sobre \mathcal{X} e $\beta \in \mathbb{R}^*$, a função βf é convexa;
- (ii) Se f_1 e f_2 são funções convexas sobre \mathcal{X} , então a função $f_1 + f_2$ é convexa;
- (iii) Se f é uma função convexa sobre \mathcal{X} , então o conjunto

$$\mathcal{X}_c = \{\mathbf{x} : \mathbf{x} \in \mathcal{X}, f(\mathbf{x}) \leq c\}$$

é convexo qualquer que seja o $c \in \mathbb{R}$.

Teorema 5. Seja f uma função convexa sobre \mathcal{X} . Então o conjunto \mathcal{X}^* onde f encontra seu mínimo local, \mathbf{x}^* , é convexo, e qualquer mínimo relativo de f é um mínimo global (Luenberger, 1984).

Teorema 6. Seja $f \in \mathcal{C}^2$ uma função convexa sobre \mathcal{X} . Se existe um ponto $\mathbf{x}^* \in \mathcal{X}$ tal que, para todo $\mathbf{x} \in \mathcal{X}$, $\nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \geq 0$, então \mathbf{x}^* é um mínimo global de f sobre \mathcal{X} (Luenberger, 1984).

Os métodos de otimização de primeira ordem são em geral muito simples e amplamente conhecidos. Destes, o mais referenciado é o **método do gradiente**, que é definido na forma

$$\Delta \mathbf{x}(k) = -\eta \nabla_{\mathbf{x}} f(\mathbf{x}(k)) \quad (2.10)$$

onde $\Delta \mathbf{x}(k) = \mathbf{x}(k+1) - \mathbf{x}(k)$ é chamado de **vetor de diferenças**, e $\eta \in \mathbb{R}_+^*$ é um parâmetro chamado de **passo** e determina a velocidade e a estabilidade do processo de busca da solução. A escolha de um valor elevado para η pode fazer com que o método inicialmente tenha uma convergência rápida. No entanto, após algumas interações o método pode divergir. A escolha de um valor muito pequeno pode afastar a possibilidade de divergência. No entanto, abre a possibilidade de uma convergência muito lenta.

Um outro método de primeira ordem bastante conhecido é o **método do gradiente recursivo**, uma variação mais veloz e eficiente do método do gradiente. Este método é definido na forma:

$$\Delta \mathbf{x}(k+1) - \alpha \Delta \mathbf{x}(k) = -\eta \nabla_{\mathbf{x}} f(\mathbf{x}(k)) \quad (2.11)$$

onde $|\alpha| \in [0; 1) \subset \mathbb{R}^+$ é um parâmetro chamado de *momentum*, que também tem influência na convergência.

Dos métodos de segunda ordem o mais geral e conhecido é o **método de Newton**, que tem o formato

$$\Delta \mathbf{x}(k) = - [\nabla_{\mathbf{x}}^2 f(\mathbf{x}(k))]^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}(k)). \quad (2.12)$$

Note que o método, além de apresentar todas as operações do método do gradiente, apresenta a inversão da matriz hessiana, que é uma matriz $n \times n$. Logo, o método apresenta um custo computacional muito maior que o método do gradiente. O método pode convergir em um número de interações muito menor que o método do gradiente, mas a depender das dimensões do problema a solucionar, a dimensão da

matriz hessiana pode ser muito grande e sua inversão muito demorada, mesmo com o uso de sistemas computacionais potentes e caros. Adicionalmente, pode haver problemas de singularidade, pseudo-singularidade e instabilidade na matriz ou mesmo erros numéricos, o que pode tornar a inversão infactível. Na literatura técnica, são apresentados diversas alternativas ou métodos alternativos para solucionar ou contornar estes problemas, no entanto estas alternativas sempre acarretam em um aumento da complexidade do método.

Ambos os métodos citados acima, o método do gradiente e o método de Newton, são métodos locais, ou seja, possibilitam apenas alcançar pontos de mínimo locais. A única possibilidade desses métodos alcançarem pontos de mínimo globais é se o problema é convexo [i.e., \mathcal{X} é um conjunto convexo e f é uma função convexa, como mostram os teoremas 5 e 6]. No entanto, para aplicações *on-line* a aplicação do método do gradiente pode ser bastante apropriada, pois a tarefa de busca de soluções pode ser feita em subespaços locais e atualizados a cada instante quando necessário [veja a seção 5.2, onde é feita uma aplicação à um identificador com ajuste de parâmetros *on-line*].

Um outro conceito importante na teoria de otimização é o conceito de ordem de convergência. Considere que $\mathbf{e}(k) = \mathbf{x}(k) - \mathbf{x}^*$ seja o erro cometido na aproximação pelo método de otimização. É óbvio que se $\lim_{k \rightarrow \infty} \mathbf{x}(k) = \mathbf{x}^*$, então $\lim_{k \rightarrow \infty} \mathbf{e}(k) = 0$, e assim, quanto mais rápido $\mathbf{e}(k)$ se aproximar de zero, melhor.

Uma medida da velocidade de convergência é comparação entre os erros cometidos entre dois ajustes sucessivos, $\mathbf{e}(k+1)$ e $\mathbf{e}(k)$. Assim, se $\|\mathbf{e}(k+1)\| \leq r \|\mathbf{e}(k)\|^p$, para algum $r \in [0; 1)$ e algum $p \in \mathbb{N}$, o processo de ajuste é convergente. Note que quanto menor for r mais rápida será a convergência, e que se $\|\mathbf{e}(k+1)\| < r \|\mathbf{e}(k)\|^p$ a convergência é assintótica.

Definição 36 (Ordem de Convergência). Considere que $\mathbf{x}(k)$ converge para \mathbf{x}^* . A ordem de convergência de $\mathbf{x}(k)$ é definida como o supremo dos números p que satisfazem a condição

$$0 \leq \lim_{k \rightarrow \infty} \frac{\|\mathbf{e}(k+1)\|}{\|\mathbf{e}(k)\|^p} < 1.$$

Adicionalmente, se a ordem de convergência de $\mathbf{x}(k)$ é p , e existe o limite

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{e}(k+1)\|}{\|\mathbf{e}(k)\|^p} = \beta,$$

então a convergência é assintótica.

A ordem de convergência de $\mathbf{x}(k)$ é dita linear se $p = 1$, e se

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{e}(k+1)\|}{\|\mathbf{e}(k)\|} = \beta,$$

e é dita super-linear se $p = 1$ e se

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{e}(k+1)\|}{\|\mathbf{e}(k)\|} = 0.$$

Note que uma convergência assintótica corresponde no mínimo a uma ordem de convergência linear.

Esta seção não objetiva um aprofundamento de conceitos e métodos de otimização não linear, bem como dos problemas envolvidos nas aplicações. Maiores detalhes sobre conceitos e métodos podem ser obtidos na literatura técnica apropriada [veja por exemplo Luenberger (1984) e Mangasarian (1994)].

2.7 Aproximadores Universais

Considere uma função qualquer $f(\mathbf{x}) \in C^{k_1}$, tal que $f: \mathbb{R}^n \mapsto \mathbb{R}$ e $k_1 > 0$, e uma outra função $\hat{f}(\mathbf{w}, \mathbf{x}) \in C^{k_2}$, tal que $\hat{f}: \mathbb{R}^{n_p} \times \mathbb{R}^n \mapsto \mathbb{R}$ e $k_2 \geq k_1$, onde $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^{n_p}$ é um vetor de parâmetros ajustáveis.

Suponha agora que a função $f(\mathbf{x})$ não seja perfeitamente conhecida ou até mesmo não conhecida, ou que seus parâmetros internos sejam não conhecidos ou variantes com o tempo. Suponha também que por meio de busca iterativa seja alcançado um vetor $\mathbf{w} \in \mathcal{W}$ tal que \hat{f} possa aproximar f dentro de uma região $\mathcal{X} \subset \mathbb{R}^n$ e com um erro arbitrariamente pequeno. Nesta situação é dito que \hat{f} é um aproximador de f .

Note que, para a aproximação ser possível, é necessário que $k_2 \geq k_1$. Em geral, \hat{f} é constituído pela somatória de funções pertencente a C^∞ e que sejam limitadas.

Definição 37 (Aproximador Universal). Seja $f(\mathbf{x}) \in C^k$, tal que $f : \mathbb{R}^n \mapsto \mathbb{R}$ e $k > 0$, e seja $\hat{f}(\mathbf{w}, \mathbf{x}) \in C^\infty$, tal que $\hat{f} : \mathbb{R}^{n_p} \times \mathbb{R}^n \mapsto \mathbb{R}$, onde $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^{n_p}$ é um vetor de parâmetros ajustáveis. É dito que \hat{f} é um aproximador universal se existem um n_p , um vetor $\mathbf{w}^* \in \mathcal{W}$ e um $\epsilon > 0$ tal que $\|f(\mathbf{x}) - \hat{f}(\mathbf{w}^*, \mathbf{x})\| < \epsilon \quad \forall \mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, qualquer que seja f .

O aproximador universal mais conhecido é a Série de Fourier. No entanto, atualmente existem diversos outros modelos de aproximadores universais tais como *spline* (Petrushev e Popov, 1985), *wavelet* (Mayer, 1990; Jaffard e Laurentçot, 1992), sistemas *fuzzy* (Wang, 1992; Castro, 1995; Castro e Delgado, 1996) e redes neurais artificiais (Cybenko, 1989; Hornik *et al.*, 1989; Mhaskar, 1992; Barron, 1993).

Uma particularidade da Série de Fourier é que a somatória de senos e cossenos em diferentes frequências forma uma base linear independente, sendo também linear a relação com os parâmetros ajustáveis, o que permite a obtenção de diversas informações sobre o sistema [bem como o uso de uma grande quantidade de ferramentas de análise e projeto existentes na literatura]. Os outros aproximadores universais acima citados não possuem toda esta flexibilidade devido a não linearidade paramétrica, sendo em geral possível apenas a aproximação da função. No entanto, estes outros aproximadores universais possibilitam a aproximação usando uma menor quantidade de parâmetros que a Série de Fourier [i.e., uma menor quantidade de parâmetros no vetor \mathbf{w}].

2.8 Transformada \mathcal{Z}

Em geral, os modelos matemáticos para sistemas físicos são apresentados em formato não linear. No entanto, a maior parte das ferramentas de análise e/ou projeto de sistemas existentes e desenvolvidas nas últimas décadas são baseada na teoria de sistemas lineares.

Uma ferramenta bastante difundida e eficiente é a transformada \mathcal{Z} , desenvolvida para aplicações em análise e projetos de sistemas discretos no tempo. A teoria da transformada \mathcal{Z} é geralmente abordada em cursos de graduação de sinais e sistemas ou então de análise de sistemas.

A transformada \mathcal{Z} é válida para regiões lineares ou manifolds. As análises são feitas a partir da transformação do modelo do sistema ou sinais do domínio do tempo para o domínio da frequência.

Após aplicada a transformada \mathcal{Z} a um sinal $x(k) \in \mathbb{R}$, ele passa a ser representado por

$$X(z) = \sum_{k=-\infty}^{\infty} x[k] z^{-k}$$

onde $z = \rho e^{j\Omega}$ é um número complexo e Ω é uma componente de frequência.

Um sistema pode ser representado a partir de uma função de transferência em z no formato

$$Y(z) = H(z) U(z)$$

onde $Y(z)$, $U(z)$ e $H(z)$ representam respectivamente o sinal de saída, o sinal de entrada e a função de transferência do sistema no domínio da transformada z .

A função de transferência $H(z)$ pode ser expressa por meio de uma função racional do tipo

$$H(z) = \frac{B(z)}{A(z)}$$

com o polinômio

$$B(z) = b_M z^M + b_{M-1} z^{M-1} + \dots + b_1 z + b_0$$

sendo chamado de numerador e o polinômio

$$A(z) = a_N z^N + a_{N-1} z^{N-1} + \dots + a_1 z + a_0$$

sendo chamado de denominador ou polinômio característico. Todos os a_i e b_i devem ser números reais. É suposto que o sistema é causal, e com isto $M \leq N$, e que não existem fatores comuns entre $B(z)$ e $A(z)$.

As raízes do polinômio característico são chamadas de pólos. Os pólos podem assumir tanto valores reais quanto complexos. Quando assumem valores complexos, os pólos sempre aparecem em pares conjugados.

Para um sistema ser estável pela abordagem da transformada \mathcal{Z} é necessário que todos os pólos estejam dentro de um círculo de raio igual a unidade centrado na origem dos eixos coordenados do plano \mathcal{Z} , círculo este chamado unitário. Maiores detalhes sobre a teoria da transformada \mathcal{Z} podem ser encontrados em Kamen e Heck (1999).

2.9 Sumário

Neste capítulo, foram apresentados conceitos importantes para a compreensão dos capítulos seguintes. Na seção 2.1, foram apresentadas definições concernentes a sistemas e descritos brevemente alguns tipos encontrados com mais freqüência em engenharia.

A seção 2.2 mostrou as diversas maneiras de como pode ser feita a representação de um sistema. Esta seção enfatizou que um modelo, ou representação, tem um objetivo, e para atingir este objetivo existe um tipo de modelo que melhor se ajusta. Destacou adicionalmente que os modelos são representações da realidade, e não a realidade propriamente dita [i.e., o sistema real propriamente dito]. Com isto, deve-se ter cautela ao atribuir aos sistemas propriedades que pertencem aos modelos, como por exemplo as propriedades da observabilidade e da controlabilidade.

A seção 2.4 objetivou trazer a atenção do leitor uma série de definições e conceitos sobre a teoria de sistemas não lineares com abordagem por equações de estado. Foram apresentadas definições e conceitos sobre estabilidade de sistemas do ponto de vista de Lyapunov, e em seguida conceituada a propriedade da observabilidade genérica para

sistemas não lineares.

Na seção 2.5, foi discutida a propriedade da observabilidade de sistemas, e pouco mais brevemente a propriedade da controlabilidade. Esta seção destacou que estas propriedades são do modelo matemático e não do sistema físico.

A seção 2.6 apresentou alguns conceitos sobre métodos de otimização e alguns algoritmos de primeira e segunda ordem. Na seção 2.7, foi feita a definição de aproximadores universais, que é uma ferramenta matemática muito usada na aproximação de funções e modelagem de sistemas. Na seção 2.7, foram feitos alguns comentários a respeito da teoria da transformada \mathcal{Z} .

Capítulo 3

Redes Neurais Artificiais do Tipo Multi-Layer Feedforward Perceptron

Neste capítulo, é feita uma descrição matemática do modelo multi-layer feedforward perceptron de rede neural artificial e apresentado um algoritmo de aprendizagem adaptativo para ajuste paramétrico na modelagem de sistemas dinâmicos. O algoritmo é baseado no método de otimização pelo gradiente e a análise de convergência local é feita através da função de Lyapunov. O algoritmo é estruturado de tal forma que o usuário necessita apenas um parâmetro dentro de uma faixa de variação pré-definida para torná-lo convergente, e mantém, em adição, a simplicidade dos algoritmos baseados no método do gradiente.

3.1 Introdução

NOS últimos anos, a modelagem matemática ou identificação de sistemas dinâmicos usando redes neurais artificiais (RNAs) é uma linha de pesquisa que tem provocado entusiasmo entre pesquisadores. O principal objetivo desta linha de pesquisa

é a obtenção de equivalentes artificiais dos sistemas cerebrais que controlam e gerenciam o corpo humano e de outros animais, para aplicações em sistemas diversos.

Vários modelos para células nervosas e suas formas de organização têm sido desenvolvidos. McCulloch e Pitts (1943) propuseram um modelo de **RNA**, concebido como uma soma de entradas modificadas por pesos e com a saída assumindo uma forma binária, dependendo da soma estar abaixo ou acima de algum valor de disparo (função de ativação). Este modelo foi posteriormente modificado com a saída passando a ser contínua, limitada, diferenciável (Rumelhart *et al.*, 1986). A estrutura adotada atualmente é em forma de rede e conhecida como *multi-layer feedforward perceptron* (**MLP**), sendo atualmente muito usada, principalmente após ter sido provada como sendo um modelo matemático aproximador universal para algumas configurações de rede (Cybenko, 1989; Hornik *et al.*, 1989; Mhaskar, 1992; Barron, 1993). Este modelo de rede possui parâmetros que podem ser ajustados com base em algum critério de desempenho para que a rede trabalhe com o comportamento desejado.

Existem vários métodos ou algoritmos de estimação de parâmetros que podem ser aplicados a modelos matemáticos. Usualmente estes modelos são baseados em algum método de otimização bem estabelecido conceitualmente. Existem duas linhas básicas para o desenvolvimento de tais métodos: um baseado no uso do vetor gradiente; e outro baseado no uso da matriz hessiana. Para problemas convexos, os dois métodos podem alcançar soluções globais [i.e., eles podem alcançar pontos de mínimos e máximos globais]. Se o problema não é convexo, ambos os métodos podem apenas atingir soluções locais. No caso do ajuste de parâmetros em modelos de **RNA**, não é possível garantir a convexidade pois os sistemas modelados ou controlados são tidos inicialmente como de modelos desconhecidos, não sendo possível qualquer prova de convexidade, tratando-se assim de problemas não convexos.

Os métodos baseados no vetor gradiente requerem um esforço computacional relativamente pequeno. Normalmente, os métodos baseados na matriz hessiana requerem uma elevada demanda computacional. A elevada demanda computacional imposta pelos

métodos baseados na matriz hessiana derivam da necessidade de manipular conjuntamente a inversão da matriz hessiana e todas as tarefas que envolvem o método do gradiente. Na prática os métodos baseados na matriz hessiana podem atingir uma solução em um menor número de interações que os métodos baseados no vetor gradiente, porém isto não significa que essa solução seja atingida em tempo menor (veja Robitaille *et al.* (1996) para mais detalhes sobre este tópico).

Para aplicações *on-line* [i.e., com ajustes de parâmetros em tempo real], a demanda computacional imposta pelos métodos baseados na matriz hessiana podem limitar ou mesmo inviabilizar o seu uso em sistemas em que as redes neurais possuem o método de ajuste de parâmetros incorporado e sendo executado em tempo real. Isto pode ocorrer quando o *hardware* não possui a capacidade de processamento necessária para executar todos os procedimentos demandados pelo algoritmo e em adição executar as outras tarefas demandadas pelo metassistema. Por exemplo, em sistemas de controle, além de modelagem, o metassistema pode necessitar executar tarefas tais como aquisição de dados, planejamento, supervisão, detecção de falhas, troca de informações com outros elementos do sistema, etc. Todos esses elementos juntos podem provocar problemas de alocação de recursos por parte do sistema computacional.

Em modelos de **RNAs**, os procedimentos de ajustes paramétricos são conhecidos como aprendizagem (*learning*). O algoritmo *back-propagation*, que produz em sua saída o vetor gradiente, aplicado inicialmente à rede **MLP** por Rumelhart *et al.* (1986) é o mais conhecido e usado até o momento.

Este algoritmo, pode ser lento quando comparado com algoritmos que utilizam informações da matriz hessiana¹, como os apresentados por Sjöberg e Ljung (1995) e Ro-

¹A questão da convergência lenta dos algoritmos baseados no método gradiente deve ser analisada com cautela. Robitaille *et al.* (1996) apresentam um estudo comparativo entre dois algoritmos baseados no vetor gradiente e outros algoritmos baseados em informações da matriz hessiana. É mostrado que em algumas situações um algoritmo baseado no método do gradiente recursivo converge com um menor número de interações quando comparado com alguns algoritmos que utilizam informações da matriz hessiana, e em outras situações a convergência é obtida em um tempo médio também inferior.

Robitaille *et al.* (1996). Contudo, os algoritmos de aprendizagem baseados no método de otimização pelo vetor gradiente possuem as vantagens de serem de fácil compreensão, e de fácil implementação por usuários sem bons conhecimentos de métodos de otimização ou técnicas de implementação computacional.

Na literatura técnica, poucos trabalhos abordam o problema da convergência do ajuste de parâmetros para métodos do gradiente para redes tipo **MLP** e, infelizmente, isto tem feito com que alguns usuários resolvam o problema de forma empírica². Kuan e Hornik (1991) investigaram a convergência do algoritmo para a condição de taxa de aprendizagem constante. Baldi (1995) apresentou uma visão geral dos principais algoritmos de aprendizagem baseados no método de otimização pelo gradiente, contudo não abordando a questão da convergência.

Em outra direção, Ku e Lee (1995) apresentaram uma análise de convergência em que a cada instante de tempo era ajustada uma camada da rede **MLP**. É importante observar que o gradiente de uma função escalar n -paramétrica em relação a seus n -parâmetros é um vetor n -dimensional. O uso de um procedimento de ajuste paramétrico onde os n -parâmetros são ajustados por etapas não garante que as soluções intermediárias estejam caminhando na direção do vetor gradiente n -dimensional do sistema. Isto torna-se mais evidente quando os sistemas aos quais deseja-se modelar são fortemente não lineares.

Golovko e Savitsky (1999) propuseram um método de convergência baseado em uma decomposição do algoritmo de ajuste por série de Taylor e na escolha empírica de um limitante superior para as derivadas das funções de ativação. Um pouco mais tarde, Golovko *et al.* (2000) propuseram um complicado método para determinação de

²Note, por exemplo, que Levin e Narendra (1996) usaram um valor fixo de 0.01 para a taxa de aprendizagem (parâmetro de projeto do algoritmo, geralmente fixado pelo usuário), provavelmente obtido de forma empírica. Ainda, Robitaille *et al.* (1996) usaram os algoritmos *back-propagation* e gradiente recursivo com valores fixos para a taxa de aprendizagem e o *momentum* (parâmetro de projeto do algoritmo, geralmente fixado pelo usuário) sem informações sobre o método utilizado para a escolha dos mesmos.

um valor mínimo aceitável para as derivadas das funções de ativação.

Magoulas *et al.* (2000) propuseram um método considerando que existe uma taxa de aprendizagem para cada parâmetro da rede, sendo esta taxa modificada a cada iteração, objetivando a manutenção da convergência. No entanto, o método desenvolvido por Magoulas *et al.* introduz um parâmetro de convergência no algoritmo de ajuste que tem seu valor escolhido empiricamente. Esta é a mesma situação do trabalho de Schraudolph (1998), um método onde há a introdução de um parâmetro escolhido empiricamente.

Outros pesquisadores optaram por seguir o caminho de métodos heurísticos para assegurar a convergência do ajuste de parâmetros, como por exemplo o método proposto por Shukla e Raghunath (1999). É importante citar que na literatura técnica existem vários algoritmos adaptativos para ajuste paramétrico em **RNA** do tipo **MLP**, sendo alguns deles heurísticos, nos quais os autores procuraram contornar o problema de convergência através de significativas modificações na estrutura dos algoritmos baseados no método gradiente. Nesta opção, é perdida toda a simplicidade intrínseca dos algoritmos baseados no método do gradiente, tornando-os tão ou mais complexo que os métodos baseados na matriz hessiana.

Brent (1991) propôs uma árvore de decisão para auxiliar o algoritmo *back-propagation*. Jin *et al.* (1995) apresentaram um algoritmo veloz baseado no filtro de Kalman e usando uma taxa de aprendizagem variável e ajustada adaptativamente para um valor ótimo a cada interação, mas a análise teórica da convergência do algoritmo não foi feita. Gori e Maggini (1996) provaram que a convergência *on-line* para uma solução ótima do algoritmo *back-propagation* é garantida para amostras linearmente separadas.

Recentemente, algoritmos de ajuste de parâmetros do tipo camada-por-camada (*layer-by-layer*) têm sido propostos, tal que as funções de ativação de cada camada da rede **MLP** são decompostas em duas partes, uma linear e outra não linear, e os parâmetros de cada camada ajustados separadamente, como por exemplo o método proposto por Wang e Chen (1996). Contudo, estes métodos têm sido criticados por

não serem adequados para aplicações em **MLPs** com número elevado de neurônios (Van Milligen *et al.*, 1998). Leone *et al.* (1998) propuseram uma variação do algoritmo *back-propagation* aplicando uma relaxação linear das equações da rede.

Cerqueira *et al.* (2000a) desenvolveram uma análise de convergência local para o algoritmo *back-propagation*, elaborada a partir da aplicação do Segundo Método de Lyapunov, e forneceram um limitante superior para o valor da taxa de aprendizagem sem a inclusão de qualquer parâmetro adicional para assegurar a convergência do ajuste de parâmetro. Em adição, Cerqueira *et al.* (2000c) propõem um algoritmo adaptativo para ajuste de parâmetros em redes **MLP** [um algoritmo *back-propagation* adaptativo] baseado nesta análise de convergência. Os conteúdos destes dois trabalhos serão apresentados a seguir.

3.2 O Método do Gradiente no Ajuste de Parâmetros de Modelos Não Lineares

Considere um modelo matemático como sendo um vetor de funções não lineares discretas $\phi : \mathbb{Z}^+ \times \mathbb{R}^{n_p} \times \mathbb{R}^{n_e} \rightarrow \mathbb{R}^{n_s}$, com $\phi \in C^q$ para algum $q \in \mathbb{N}$ suficientemente grande, na forma

$$\hat{\mathbf{y}}(k) = \phi(k, \hat{\mathbf{w}}, \mathbf{u}(k)). \quad (3.1)$$

onde $\hat{\mathbf{w}}(k) \in \mathcal{W} \subset \mathbb{R}^{n_p}$ é um vetor de parâmetros internos ajustáveis, $\mathbf{u}(k) \in \mathcal{U} \subset \mathbb{R}^{n_e}$ é um vetor de sinais de entrada, e $\hat{\mathbf{y}}(k) \in \mathcal{Y} \subset \mathbb{R}^{n_s}$ é o vetor com a aproximação da saída do sistema. Seja ainda $\mathbf{y}(k) \in \mathcal{Y}$ o vetor de saída de algum sistema, $n_e \in \mathbb{N}$, $n_s \in \mathbb{N}$, e \mathcal{W} , \mathcal{U} e \mathcal{Y} conjuntos limitados.

A figura 3.1 ilustra a modelagem matemática ou identificação de modelo para um sistema e o procedimento de ajuste de parâmetros do modelo matemático. O objetivo é a obtenção de uma aproximação do comportamento do sistema por meio de ajuste de parâmetros em um modelo matemático apropriado. Para isto, uma função objetivo

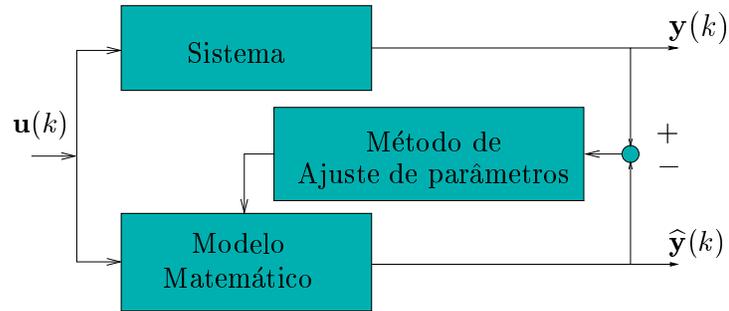


Figura 3.1: Ilustração do Procedimento de Ajuste de Parâmetros.

pode ser definida e minimizada (ou maximizada). Teoricamente, a aproximação pode ser feita localmente ou globalmente, a depender do problema ser convexo ou não. O sistema é tido *a priori* como sendo desconhecido, e neste caso não é possível estabelecer se o problema é convexo ou não, impossibilitando assim a obtenção de uma solução global (veja o capítulo 5).

O modelo representado pela expressão matemática (3.1) é discreto no tempo. No entanto, este mesmo modelo pode ser usado para representar sistemas contínuos no tempo, visto que modelos contínuos no tempo podem ser discretizados por diversos métodos (Kamen e Heck, 1999), como por exemplo, por amostragem.

Adicionalmente, o modelo expresso pela equação (3.1) é estático [i.e., não possui dinâmica temporal ou memória]. Assim, pela representação feita na figura 3.1 o modelo apenas poderia fazer a aproximação de um sistema sem dinâmica temporal. No entanto, este mesmo modelo pode ser usado para a modelagem ou identificação de sistema com dinâmica temporal por meio de recursividade [i.e., atrasos] dos sinais de entrada e saída do sistema como mostrado na figura 3.2, onde **TDL** é uma linha de retardo no tempo (*tapped delay line*).

Para esta situação o modelo matemático teria a forma

$$\hat{y}(k) = \phi(k, \hat{\mathbf{w}}, (\mathbf{Y}_{l_y}(k), \mathbf{U}_{l_u}(k))) \quad (3.2)$$

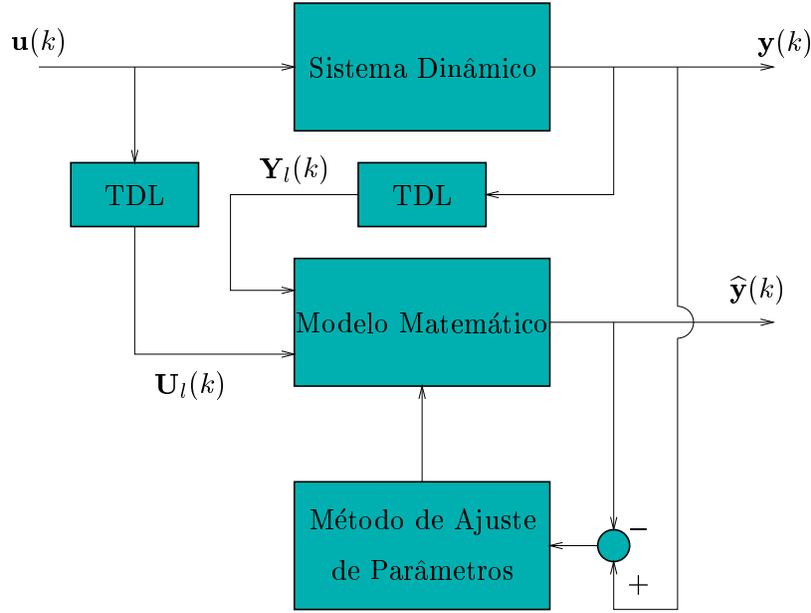


Figura 3.2: Identificação de Sistemas Dinâmicos.

onde $\phi : \mathbb{Z}^+ \times \mathbb{R}^{r_1} \times \mathbb{R}^{r_2} \mapsto \mathbb{R}^s$ é o vetor de funções não lineares, com $r_1 = (l_y \cdot s)$ and $r_2 = (l_u \cdot m)$. $\mathbf{Y}_{l_y}(k)$ e $\mathbf{U}_{l_u}(k)$ são seqüências no tempo de comprimento l_y e l_u , mantendo os valores assumidos pelas variáveis de entrada e saída do sistema, na forma

$$\mathbf{Y}_{l_y}(k) \triangleq \{\mathbf{y}(k-1), \mathbf{y}(k-2), \dots, \mathbf{y}(k-l_y)\}$$

$$\mathbf{U}_{l_u}(k) \triangleq \{\mathbf{u}(k-1), \mathbf{u}(k-2), \dots, \mathbf{u}(k-l_u)\}.$$

O modelo de identificação de sistemas dinâmicos representado pela equação (3.2) foi proposto por Narendra e Parthasarathy (1990), inicialmente atribuindo-se valores para l_y e l_u de forma empírica [i.e., sem qualquer sustentação matemática]. Posteriormente, vários estudos forneceram formalismo matemático para a equação, de modo que atualmente existem diversas opções teóricas para os parâmetros l_y e l_u (Levin e Narendra, 1995; Levin e Narendra, 1996; Hunt *et al.*, 1992; Suykens *et al.*, 1995; Cabrera e Narendra, 1999).

Do ponto de vista de modelo matemático, as equações (3.1) e (3.2) são equivalentes, visto que o par $(\mathbf{Y}_l(k), \mathbf{U}_l(k))$ da equação (3.2) equivale a $\mathbf{u}(k)$ na equação (3.1), sendo todos eles externos ao modelo matemático, não havendo assim recursividade entre a saída da rede e sua entrada. Sem perda de generalidade, os desenvolvimentos a seguir foram feitos em relação à equação (3.1).

Considere uma função objetivo a ser minimizada no formato

$$J(\widehat{\mathbf{w}}(k)) = \frac{1}{2} \mathbf{e}'(k) \mathbf{e}(k) \quad (3.3)$$

onde $J \in \mathbb{R}$ e $\mathbf{e}(k) = \mathbf{y}(k) - \widehat{\mathbf{y}}(k)$ é o erro entre a saída do sistema e o modelo matemático.

Definindo o método de ajuste de parâmetros como sendo o gradiente recursivo (veja a equação (2.11) na seção 2.6) na forma

$$\Delta \widehat{\mathbf{w}}(k+1) - \alpha \Delta \widehat{\mathbf{w}}(k) = -\eta \nabla_{\widehat{\mathbf{w}}} J(k) \quad (3.4)$$

onde $\Delta \widehat{\mathbf{w}}(k) = \widehat{\mathbf{w}}(k+1) - \widehat{\mathbf{w}}(k)$ é o vetor de diferenças de parâmetros ajustáveis, $\eta \in \mathbb{R}_+^*$ é um parâmetro chamado de passo na literatura técnica sobre otimização, e chamado de taxa de aprendizagem na literatura sobre **RNA**, $\|\alpha\| \in [0; 1) \subset \mathbb{R}^+$ é um parâmetro chamado de *momentum*. Ainda, $\nabla_{\widehat{\mathbf{w}}} J(k)$ é o gradiente da função objetivo em relação ao vetor de parâmetros ajustáveis $\widehat{\mathbf{w}}$, que deve ser expresso como

$$\begin{aligned} \nabla_{\widehat{\mathbf{w}}} J(k) &= \left[\frac{\partial \mathbf{e}(k)}{\partial \widehat{\mathbf{w}}} \right]' \mathbf{e}(k) \\ &= - \left[\frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}} \right]' \mathbf{e}(k) \end{aligned} \quad (3.5)$$

onde $\partial \widehat{\mathbf{y}}(k) / \partial \widehat{\mathbf{w}}$ é o jacobiano do modelo matemático em relação ao vetor de parâmetros ajustáveis.

A escolha apropriada dos parâmetros η e α influenciam na convergência do ajuste de parâmetros. A condição inicial é $\Delta \widehat{\mathbf{w}}(0) = \mathbf{0}_{n_p}$, onde $\mathbf{0}_{n_p}$ é um vetor nulo com dimensão n_p , e a condição de parada é alcançada quando $J(\cdot) \leq J_0$, em que J_0 é o máximo valor final aceitável para a função objetivo. Assim, quando a condição de

parada é satisfeita [i.e, a convergência é alcançada], tem-se:

$$\Delta \hat{\mathbf{w}}(k) = \Delta \hat{\mathbf{w}}(k + 1) = 0_{n_p}.$$

O procedimento de ajuste de parâmetros apresentado desta forma, com a função objetivo expressa pela equação (3.3) e a regra de ajuste expressa pela equação (3.4), é chamado de regra de ajuste incremental ou aprendizagem incremental. Os parâmetros são ajustados a cada instante, conforme a apresentação de um novo par $(\mathbf{y}(k), \mathbf{u}(k))$.

Uma alternativa é a regra de ajuste em grupo ou aprendizagem em *batch*, onde a função objetivo é expressa na forma:

$$J(\hat{\mathbf{w}}(k)) = \frac{1}{2} \sum_{l=1}^L \mathbf{e}_l'(k) \mathbf{e}_l(k)$$

onde $L \in \mathbb{N}$ é o número de pares $(\mathbf{y}(k), \mathbf{u}(k))$ apresentados para ajustes dos parâmetros, supondo L finito. Nesta regra, o ajuste dos parâmetros é feito apenas após todos os pares $(\mathbf{y}(k), \mathbf{u}(k))$ terem sido apresentados para serem processados.

Ambos os métodos conduzem a uma solução $\hat{\mathbf{w}}^*$ se α e η possuírem valores apropriados. Contudo, o método incremental apresenta duas vantagens sobre o método em grupo: (i) requer menor armazenagem de dados; (ii) apresenta melhores resultados em sistemas sujeitos a ruídos (para mais detalhes veja Hassoun (1995, pp. 202 and 203)). Assim, o método incremental será adotado neste trabalho em função das aplicações serem sistemas sujeitos a ruídos.

O parâmetro α é a raiz da equação característica³ associada com a equação recorrente (3.4), e portanto está diretamente associado com a estabilidade do procedimento de busca da solução. Abordando a equação recursiva (3.4) por meio de teoria de sistemas discretos, a recorrência é estável se $|\alpha| \in [0; 1) \subset \mathbb{R}^+$ e se η é escolhido apropriadamente. Por essa abordagem, a regra de ajuste comporta-se também como um

³De uma forma mais rigorosa, isto é verdade apenas quando o vetor de parâmetros encontra-se dentro de um manifold que contenha o vetor de parâmetros ótimo (veja a suposição 3).

filtro passa baixa que trabalha no sentido de diminuir os efeitos danosos de ruídos presentes nos sinais de entrada e saída do sistema.

Um alto valor de α [i.e., $\|\alpha\| \cong 1$] corresponde a uma baixa sensibilidade a distúrbios [i.e., um filtro com baixa frequência de corte]. Por outro lado, um baixo valor de α [i.e., $\|\alpha\| \cong 0$] corresponde a uma alta sensibilidade a distúrbios [i.e., um filtro com alta frequência de corte]. Se $\alpha = 0$, significa que o procedimento não apresenta rejeição a distúrbios. A escolha de um valor apropriado para α será sempre vinculada ao conhecimento a respeito dos distúrbios observados no sistema a ser identificado.

O procedimento de ajuste de parâmetro baseado na equação (3.4) não alcança necessariamente o mínimo ótimo da função objetivo (3.3), mas sim algum vetor na vizinhança ao vetor mínimo ótimo. Devido a imperfeições inerentes ao modelo matemático $\phi(\cdot)$ [i.e., sua capacidade de fazer a aproximação do modelo do sistema com a aproximação menor possível] é difícil e não realístico tratar este ótimo com precisão prática. Assim, o método apresentado aqui usa o conceito de zona-morta apresentado por Kreisselmeier e Anderson (1986) (veja também Chen e Khalil (1995)) como condição de parada.

Com a zona-morta, qualquer solução que esteja na vizinhança de um vetor ótimo (representado indiretamente por J_0) é considerado como sendo uma solução apropriada. Depois de ajustado (ou implementado em regime *on-line*) o modelo $\phi(\cdot)$ pode ser usado e a estabilidade do sistema que o usa ser provada com informações sobre limites de erros de modelagem sendo considerados *a priori* (Kreisselmeier e Anderson, 1986).

Suposição 1. *Os mapeamentos de ambos, do sistema e do modelo matemático, são de classes compatíveis [i.e., é possível a obtenção de uma aproximação do sistema através de ajustes paramétricos no modelo matemático].*

Suposição 2. *Δt , o tempo decorrido entre os instantes $k + 1$ e k , é constante e desprezível em relação à dinâmica do sistema.*

Suposição 3. *É suposto que $\hat{\mathbf{w}}^0 \in \mathcal{W}^0 \subset \mathbb{R}^{n_p}$ é um vetor de parâmetros especificado*

no início do processo de ajuste de parâmetros, e que $\widehat{\mathbf{w}}^* \in \mathcal{W}^* \subset \mathcal{W}^0$ é o vetor de parâmetros ótimos que minimiza a função objetivo (3.3), sendo \mathcal{W}^* um manifold.

Pela suposição 1, se os mapeamentos do sistema e do modelo matemático são compatíveis (e se o processo de ajuste de parâmetros é convergente), então a identificação ou modelagem pode ser realizada com sucesso. Na prática, isto quer dizer que a classe do mapeamento do modelo $\phi(\cdot)$ deve ser maior ou igual à classe do mapeamento do sistema (veja definição 7 na seção 2.4).

A suposição 3 estabelece condições para as superfícies em que estão contidos o vetor inicial, $\widehat{\mathbf{w}}^0$, e o vetor ótimo, $\widehat{\mathbf{w}}^*$. Note que uma superfície ser um manifold estabelece propriedades para ela, e não limites *a priori*, ou proximidade entre os vetores $\widehat{\mathbf{w}}^0$ e $\widehat{\mathbf{w}}^*$ (veja a definição 13 na seção 2.4).

Lema 1. *Considerando a suposição 2, existe um Δt pequeno o suficiente tal que o vetor de erro durante o processo de ajuste de parâmetros pode ser aproximado pelo primeiro termo da Série Taylor, na forma*

$$\Delta \mathbf{e}(k) = \frac{\partial \mathbf{e}(k)}{\partial \widehat{\mathbf{w}}} \Delta \widehat{\mathbf{w}}(k) \quad (3.6)$$

onde $\Delta \mathbf{e}(k) = \mathbf{e}(k+1) - \mathbf{e}(k)$.

Lema 2. *Durante o processo de ajuste de parâmetros, considerando a equação (3.4), quando $\widehat{\mathbf{w}}(k)$ é pertencente a \mathcal{W}^* e para um valor de k grande, a diferença entre o vetor de parâmetros ajustáveis é dado na forma*

$$\Delta \widehat{\mathbf{w}}(k) = -\eta \mathbf{f}_c \nabla_{\widehat{\mathbf{w}}} J(k) \quad (3.7)$$

onde \mathbf{f}_c é chamado de **fator de correção**, sendo dado por

$$\mathbf{f}_c = \frac{1}{1 - \alpha}. \quad (3.8)$$

Demonstração. Baseado na suposição 3, $\nabla_{\widehat{\mathbf{w}}} \mathbf{J}(k)$ é quase constante quando $\widehat{\mathbf{w}} \in \mathcal{W}^*$ (veja Hassoun (1995, pp. 213-214) and Hertz *et al.* (1991, pp. 123)). Logo, pode

ser feito $\nabla_{\widehat{\mathbf{w}}}\mathbf{J}(k) = \nabla_{\widehat{\mathbf{w}}}\mathbf{J}^*$, e a transformada \mathcal{Z} pode ser aplicada à equação (3.4), resultando em

$$(z - \alpha) \Delta \widehat{\mathbf{w}}(z) = -\eta \nabla_{\widehat{\mathbf{w}}}\mathbf{J}^* \frac{z}{z - 1}$$

ou

$$\Delta \widehat{\mathbf{w}}(z) = -\eta \nabla_{\widehat{\mathbf{w}}}\mathbf{J}^* \frac{z}{(z - \alpha)(z - 1)}$$

Agora pode ser aplicado o teorema do valor final. Isto irá resultar em

$$\begin{aligned} \lim_{k \rightarrow \infty} \Delta \widehat{\mathbf{w}}(k) &= \lim_{z \rightarrow 1} \left\{ \frac{z - 1}{z} \Delta \widehat{\mathbf{w}}(z) \right\} \\ &= \lim_{z \rightarrow 1} \left\{ \frac{z - 1}{z} \left(-\eta \nabla_{\widehat{\mathbf{w}}}\mathbf{J}^* \frac{z}{(z - \alpha)(z - 1)} \right) \right\} \\ &= -\frac{\eta}{1 - \alpha} \nabla_{\widehat{\mathbf{w}}}\mathbf{J}^* \end{aligned}$$

Assim, pode ser definido $\mathbf{f}_c = 1/(1 - \alpha)$, e $\Delta \widehat{\mathbf{w}}(k) = -\eta \mathbf{f}_c \nabla_{\widehat{\mathbf{w}}}\mathbf{J}(k)$ será válido \mathcal{W}^* .

□

Teorema 7. *Seja η a taxa de aprendizagem ou passo do método de ajuste de parâmetros baseado na equação (3.4), com função objetivo definida pela equação (3.3). O vetor de ajuste de parâmetros converge assintoticamente para $\widehat{\mathbf{w}}^*$ se η é escolhido dentro do intervalo dado por*

$$0 < \eta < \frac{2}{\mathbf{f}_c \left\| \frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}} \right\|^2}. \quad (3.9)$$

Demonstração. A demonstração é feita em duas etapas: (i) para $\widehat{\mathbf{w}} \in \mathcal{W}^*$ e um valor grande de k ; e (ii) para $\widehat{\mathbf{w}} \notin \mathcal{W}^*$ e um valor pequeno de k .

(i) para $\widehat{\mathbf{w}} \in \mathcal{W}^*$ e um valor grande de k .

Seja a função a seguir escolhida como função candidata de Lyapunov e aplicada à saída do sistema de ajuste de parâmetros:

$$\mathcal{V}(k) = \mathbf{e}'(k) \mathbf{e}(k).$$

Entre dois instantes de tempo, a diferença apresentada pela função de Lyapunov, devido ao processo de ajuste de parâmetros, é dada por

$$\begin{aligned} \Delta \mathcal{V}(k) &= \mathcal{V}(k+1) - \mathcal{V}(k) \\ &= \mathbf{e}'(k+1) \mathbf{e}(k+1) - \mathbf{e}'(k) \mathbf{e}(k). \end{aligned}$$

Denotando a matriz $\partial \widehat{\mathbf{y}}(k) / \partial \widehat{\mathbf{w}}$ por $\mathbf{A} \in \mathbb{R}^{n_s \times n_p}$, e aplicando-se as equações (3.4), (3.5) e (3.7) na equação (3.6), obtém-se

$$\Delta \mathbf{e}(k) = -\mathbf{f}_c \eta \mathbf{Q} \mathbf{e}(k)$$

onde $\mathbf{Q} \in \mathbb{R}^{n_s \times n_s}$ é simétrica, definida positiva, e dada por $\mathbf{Q} = \mathbf{A} \mathbf{A}'$. Além disso, $\mathbf{e}(k+1) = \Delta \mathbf{e}(k) + \mathbf{e}(k) = \{-\mathbf{f}_c \eta \mathbf{Q} + \mathbf{I}_{n_s}\} \mathbf{e}(k)$, com \mathbf{I}_{n_s} denotando uma matriz identidade com dimensão $\mathbb{R}^{n_s \times n_s}$. Logo

$$\begin{aligned} \Delta \mathcal{V}(k) &= \mathbf{e}'(k) \{-\mathbf{f}_c \eta \mathbf{Q} + \mathbf{I}_{n_s}\}^2 \mathbf{e}(k) - \mathbf{e}'(k) \mathbf{e}(k) \\ &= -\mathbf{f}_c \eta \mathbf{e}'(k) \mathbf{Q} \{2 \mathbf{I}_{n_s} - \mathbf{f}_c \eta \mathbf{Q}\} \mathbf{e}(k). \end{aligned}$$

Baseado no segundo método de Lyapunov para sistemas discretos, o método é assintoticamente estável [i.e., converge assintoticamente] se $\Delta \mathcal{V} < 0$. Isto requer que $(\mathbf{f}_c \eta) > 0$, como previamente definido [i.e., que η seja positivo], e também que

$$\begin{aligned} 2 \mathbf{I}_{n_s} - \mathbf{f}_c \eta \mathbf{Q} &> 0 \\ 2 - \mathbf{f}_c \eta \lambda_{\max}(\mathbf{Q}) &> 0 \end{aligned}$$

onde λ_{\max} corresponde ao autovalor máximo da matriz \mathbf{Q} , obtendo-se assim:

$$0 < \mathbf{f}_c \eta < \frac{2}{\lambda_{\max}(\mathbf{Q})} = \frac{2}{\left\| \frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}} \right\|^2}$$

ou

$$0 < \eta < \frac{2}{\mathbf{f}_c \left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \right\|^2}. \quad (3.10)$$

(ii) Para $\hat{\mathbf{w}} \notin \mathcal{W}^*$ e um valor pequeno de k .

O lema 2 é válido para $\hat{\mathbf{w}} \in \mathcal{W}^*$. Assim, a primeira etapa feita acima seria válida apenas dentro da região \mathcal{W}^* . Portanto, é necessário a análise da convergência fora desta região.

Para toda a região fora de \mathcal{W}^* , e para k pequeno, o efeito do termo que contém α é muito reduzido. Isto ocorre devido as acentuadas variações do gradiente, fazendo com que apenas o termo relativo a η seja significativo para a regra de ajuste de parâmetros (veja Hassoun (1995, pp. 213-214) e Hertz *et al.* (1991, pp. 123)). Com isto, o termo relativo ao *momentum*, α , será removido e considerado como uma perturbação.

Logo, a regra de ajuste de parâmetros torna-se

$$\Delta \hat{\mathbf{w}}(k) = -\eta \nabla_{\hat{\mathbf{w}}} \mathbf{J}(k)$$

e neste caso não é necessário o uso de lema 2. O lema 1 ainda é válido porque pode ser feita uma escolha apropriada para Δt . Seguindo o mesmo procedimento usado na primeira etapa da demonstração é obtido

$$0 < \eta < \frac{2}{\left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \right\|^2}. \quad (3.11)$$

Note que o intervalo expresso pela desigualdade (3.10) é mais restritivo que o intervalo expresso pela desigualdade (3.11), isto porque $\mathbf{f}_c > 1$. Assim,

$$\frac{2}{\mathbf{f}_c \left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \right\|^2} < \frac{2}{\left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \right\|^2}$$

O intervalo expresso pela desigualdade (3.10) já torna o ponto $\widehat{\mathbf{w}}$ atrativo e assintoticamente estável também para a região fora de \mathcal{W}^* , sendo isto suficiente para atrair o ponto $\widehat{\mathbf{w}}$ para dentro de \mathcal{W}^* . Seria muito difícil determinar o exato instante em que $\widehat{\mathbf{w}}$ entra em \mathcal{W}^* e fazer um chaveamento entre as desigualdades (3.10) e (3.11). Portanto, o intervalo expresso pela desigualdade (3.10) pode ser usado como critério para determinar um valor apropriado para η . Como o valor de η obtido a partir da desigualdade (3.10) torna a convergência assintoticamente estável, ele torna a convergência também estável sob perturbação, e por consequência estável em relação às perturbações oriundas do termo α consideradas no início da segunda parte da demonstração. Com isto, a demonstração do teorema é concluída.

□

Corolário 2. *Se o sistema for perturbado, e a perturbação for limitada e suficientemente pequena, o processo de ajuste de parâmetros é estável sob perturbação pela aplicação do teorema 7.*

Demonstração. Pela definição 24, o processo de ajuste de parâmetro seria estável sob perturbação se $\forall \epsilon > 0$ existe um $\delta_1(\epsilon)$ e um $\delta_2(\epsilon)$, tal que $\|\widehat{\mathbf{w}}^*\| < \delta_1$ e $\|e(t)\| < \delta_2$, $\forall t > t_0$, implicando que $\|\widehat{\mathbf{w}}(t)\| < \epsilon \quad \forall t > t_0$. O teorema 4 afirma que se $\|\widehat{\mathbf{w}}^*\|$ for assintoticamente estável, então o sistema é estável sob perturbação. A aplicação do teorema 7 faz com que $\|\widehat{\mathbf{w}}^*\|$ seja assintoticamente estável. □

Lema 3. *Seja α o parâmetro momentum e η a taxa de aprendizagem ou passo do método de ajuste de parâmetros definido pela equação (3.4). Seja ainda, $\|e(t)\| < \delta_1$. Então, existe um α e um η que podem tornar o ajuste de parâmetros estável sob perturbação, fazendo que $\|e(t)\| < \delta_2 \quad \forall t > t_0$, com $\delta_2 \leq \delta_1$, se η for obtido pelo teorema 7.*

Demonstração. O modelo de ajuste de parâmetros é também o modelo de um filtro de primeira ordem com ganho η e largura de faixa determinada inversamente por $(1 - \alpha)$.

Pelo teorema 7, o intervalo de aceitação de η é também inversamente proporcional a α . Conseqüentemente, pode-se escolher um valor para $(1 - \alpha)$ [i.e., a largura de faixa do filtro] e obter um valor de η pelo teorema 7 tal que o ajuste de parâmetros possa ser estável sob perturbação. \square

Observação 5. O teorema 7 foi provado sem a definição ou conhecimento do modelo matemático $\phi(\cdot)$, mas apenas supondo sua existência. Portanto, este teorema pode ser aplicado a qualquer modelo cujos parâmetros são ajustados pela regra de ajuste 3.4, considerando, é óbvio, que as suposições 1-3 são válidas.

Observação 6. Baseado no teorema 7, é suficiente apenas conhecer o comportamento do jacobiano de saída do modelo matemático [i.e., o jacobiano da saída do modelo em relação a seu vetor de parâmetros ajustáveis] para o intervalo de convergência de η ser encontrado.

3.3 O Método do Gradiente Aplicado à Rede *Multi-Layer Feedforward Perceptron*

O modelo perceptron apresentado por McCulloch e Pitts (1943) é da forma

$$y_i(k) = \phi_i \left(\sum_{i=0}^N w_i u_i \right)$$

onde w_i é um parâmetro ajustável, u_i uma variável de entrada, $\phi_i(\cdot)$ uma função de ativação do tipo binária (*threshold*), com u_0 sendo um termo independente e constante usualmente chamado de *bias*. Rosenblatt (1962) propôs a organização deste modelo em forma de camadas que são conhecidas atualmente como *Multi-Layer Feedforward Perceptron*. Minsky e Papert (1969) mostraram que para esta forma de organização, as redes de uma única camada juntamente com o método de ajuste de parâmetros existente na época poderiam apenas aproximar uma classe muito limitada de funções binárias.

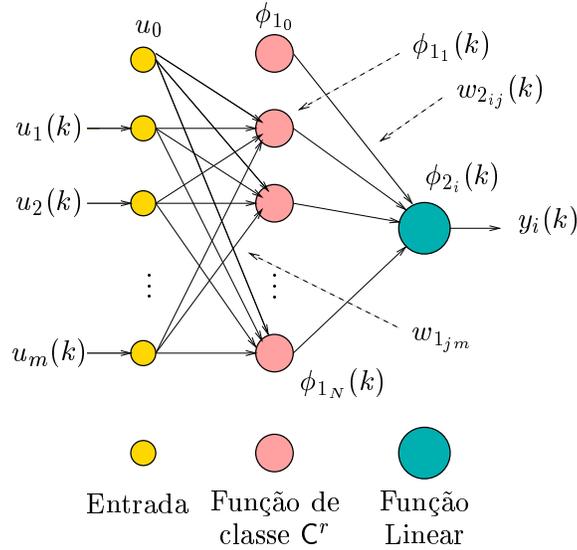


Figura 3.3: Modelo da Rede *Multi-Layer Feedforward Perceptron*.

Rumelhart *et al.* (1986) sugeriram modificações no tipo de função de ativação usado no modelo, de modo que essa função fosse diferenciável, ampliando assim as classes de funções ou mapeamentos capazes de serem aproximados pelo modelo **MLP**, e com isso a sua aplicabilidade.

Um modelo de **MLP** muito conhecido e usado é o de três camadas, como mostrado na figura 3.3 para uma rede de uma saída. A camada de entrada, ou simplesmente entrada, é responsável pela distribuição dos sinais de entrada da rede para a camada seguinte. Esta camada não faz nenhum tipo de processamento com os sinais de entrada por não apresentar neurônios processadores, apenas neurônios distribuidores. A camada intermediária de neurônios, também conhecida como camada escondida, possui neurônios de processamento com funções de ativação diferenciáveis e de classe C^r , com $r \geq 1$, como por exemplo a função sigmoideal ou a função de base radial (função *radial base*). Na última camada, ou camada de saída, os neurônios de processamento possuem funções de ativação lineares.

Devido a existência de apenas duas camadas de neurônios processadores, a ca-

mada intermediária será chamada no texto de primeira camada de neurônios processadores, ou simplesmente primeira camada. Em adição, a camada de saída será chamada de segunda camada de neurônios processadores, ou simplesmente segunda camada.

Considere o modelo de rede **MLP**⁴ expresso na forma (Cerqueira *et al.*, 2000a; Cerqueira *et al.*, 2000c)

$$\begin{cases} \mathbf{h}_1(k) &= \mathbf{W}_{1+} \mathbf{u}_+(k) \\ \hat{\mathbf{y}}_1(k) &= \phi_1(\mathbf{h}_1(k)) \\ \mathbf{h}_2(k) &= \mathbf{W}_{2+} \hat{\mathbf{y}}_{1+}(k) \\ \hat{\mathbf{y}}(k) &= \phi_2(\mathbf{h}_2(k)) \end{cases} \quad (3.12)$$

onde os subscritos referem-se às respectivas camadas de neurônios processadores, $\hat{\mathbf{y}}(k) \in \mathbb{R}^{n_s}$ é o vetor de saída da rede, $\phi_1(k) \in \mathbb{R}^N$ e $\phi_2(k) \in \mathbb{R}^{n_s}$ são os vetores de funções de ativação, $\mathbf{h}_1(k) \in \mathbb{R}^N$ e $\mathbf{h}_2(k) \in \mathbb{R}^{n_s}$ são vetores com as entradas ponderadas pelos parâmetros ajustáveis, $\mathbf{u}_+(k) \in \mathbb{R}^{n_e+1}$ e $\hat{\mathbf{y}}_{1+}(k) \in \mathbb{R}^{N+1}$ são os vetores de entrada da primeira e da segunda camada de neurônios processadores considerando o *bias*, $\hat{\mathbf{y}}_1(k) \in \mathbb{R}^N$ é o vetor de saída da primeira camada, $\mathbf{W}_{1+} \in \mathbb{R}^{N \times (n_e+1)}$ e $\mathbf{W}_{2+} \in \mathbb{R}^{n_s \times (N+1)}$ são matrizes de parâmetros ajustáveis, considerando o *bias*. Adicionalmente, sejam $\mathbf{W}_1 \in \mathbb{R}^{N \times n_e}$ e $\mathbf{W}_2 \in \mathbb{R}^{n_s \times N}$ matrizes de parâmetros ajustáveis, sem a inclusão do *bias*, e $\mathbf{u}(k) \in \mathbb{R}^{n_e}$ o vetor de entradas sem a inclusão do *bias*.

A matriz \mathbf{W}_{1+} é composta pelos parâmetros da primeira camada de neurônios

⁴Este modelo **MLP** é geralmente apresentado na literatura técnica em um formato de somatórias do tipo

$$y_i(k) = \phi_{2_i} \left(\sum_{j=1}^N w_{2_{ij}} \phi_{1_j} \left(\sum_{m=0}^{n_e} w_{1_{jm}} u_m(k) \right) - w_{2_{i0}} \phi_{1_0} \right).$$

Neste Trabalho foi dado um tratamento vetorial a esta equação.

processadores com o seguinte arranjo

$$\mathbf{W}_{1+} = \begin{bmatrix} w_{110} & w_{111} & \cdots & w_{11n_e} \\ w_{120} & w_{121} & \cdots & w_{12n_e} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1N0} & w_{1N1} & \cdots & w_{1Nn_e} \end{bmatrix}_{N \times (n_e+1)}$$

onde cada linha está associada com um neurônio artificial.

Os parâmetros da matriz \mathbf{W}_1 são distribuídos de maneira similar. No entanto, a primeira coluna relativa a \mathbf{W}_{1+} é removida, correspondendo à eliminação dos parâmetros de *bias*. A obtenção das outras matrizes seguem os mesmos procedimentos, observando, é claro, as suas respectivas dimensões.

Para a rede **MLP** expressa da forma (3.12), e usando o teorema 7, o vetor $\widehat{\mathbf{w}}$ é subdividido em dois subvetores, $\widehat{\mathbf{w}}_1 \in \mathbb{R}^{n_{p1}}$, com $n_{p1} = N(n_e + 1)$ parâmetros, e $\widehat{\mathbf{w}}_2 \in \mathbb{R}^{n_{p2}}$, com $n_{p2} = n_s(N + 1)$ parâmetros, arranjados como segue:

$$\widehat{\mathbf{w}}_1 = \begin{bmatrix} \mathbf{w}'_{11} \\ \mathbf{w}'_{12} \\ \vdots \\ \mathbf{w}'_{1N} \end{bmatrix}, \quad \widehat{\mathbf{w}}_2 = \begin{bmatrix} \mathbf{w}'_{21} \\ \mathbf{w}'_{22} \\ \vdots \\ \mathbf{w}'_{2n_s} \end{bmatrix}, \quad \widehat{\mathbf{w}} = \begin{bmatrix} \widehat{\mathbf{w}}_2 \\ \widehat{\mathbf{w}}_1 \end{bmatrix}$$

onde \mathbf{w}_{1_i} e \mathbf{w}_{2_i} correspondem às respectivas linhas de \mathbf{W}_{1+} e \mathbf{W}_{2+} .

O jacobiano da saída da rede **MLP** em relação ao subvetor $\widehat{\mathbf{w}}_2$ é dado por

$$\frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}_2} = \frac{\partial \phi_2(k)}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2(k)}{\partial \widehat{\mathbf{w}}_2} \quad (3.13)$$

onde $\partial \phi_2(k)/\partial \mathbf{h}_2$ é uma matriz diagonal com dimensão $\mathbb{R}^{n_s \times n_s}$, e

$$\frac{\partial \mathbf{h}_2(k)}{\partial \widehat{\mathbf{w}}_2} = \begin{bmatrix} \widehat{\mathbf{y}}'_{1+}(k) & 0_{(N+1)} & \cdots & 0_{(N+1)} \\ 0_{(N+1)} & \widehat{\mathbf{y}}'_{1+}(k) & \cdots & 0_{(N+1)} \\ 0_{(N+1)} & 0_{(N+1)} & \cdots & 0_{(N+1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{(N+1)} & 0_{(N+1)} & \cdots & \widehat{\mathbf{y}}'_{1+}(k) \end{bmatrix}_{n_s \times (n_s(N+1))} \quad (3.14)$$

onde $0_{(N+1)}$ é um vetor linha de elementos nulos com dimensão $N + 1$.

O jacobiano da saída da rede **MLP** em relação ao subvetor $\widehat{\mathbf{w}}_1$ é dado por

$$\frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}_1} = \frac{\partial \phi_2(k)}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2(k)}{\partial \phi_1} \frac{\partial \phi_1(k)}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1(k)}{\partial \widehat{\mathbf{w}}_1} \quad (3.15)$$

onde $\partial \mathbf{h}_2(k)/\partial \phi_1 = \mathbf{W}_2$, $\partial \phi_1(k)/\partial \mathbf{h}_1$ é diagonal com dimensão $\mathbb{R}^{N \times N}$, e

$$\frac{\partial \mathbf{h}_1(k)}{\partial \widehat{\mathbf{w}}_1} = \begin{bmatrix} \mathbf{u}'_+(k) & 0_{(n_e+1)} & \cdots & 0_{(n_e+1)} \\ 0_{(n_e+1)} & \mathbf{u}'_+(k) & \cdots & 0_{(n_e+1)} \\ 0_{(n_e+1)} & 0_{(n_e+1)} & \cdots & 0_{(n_e+1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{(n_e+1)} & 0_{(n_e+1)} & \cdots & \mathbf{u}'_+(k) \end{bmatrix}_{N \times (N(n_e+1))} \quad (3.16)$$

onde $0_{(n_e+1)}$ é um vetor linha de elementos nulos com dimensão $n_e + 1$.

Lema 4. *O jacobiano da saída da rede **MLP** em relação ao vetor de parâmetros ajustáveis $\widehat{\mathbf{w}}$ é*

$$\frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}} = \begin{bmatrix} \frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}_2} & \frac{\partial \widehat{\mathbf{y}}(k)}{\partial \widehat{\mathbf{w}}_1} \end{bmatrix}_{n_s \times n_p}$$

Lema 5. *Aplicando-se o método gradiente à rede **MLP**, a regra de ajuste de parâmetros pode ser definida como*

$$\Delta \widehat{\mathbf{w}}(k+1) = \eta \left[\frac{\partial \phi_2(k)}{\partial \mathbf{h}_2} \frac{\partial \mathbf{h}_2(k)}{\partial \widehat{\mathbf{w}}_2} \quad \frac{\partial \phi_2(k)}{\partial \mathbf{h}_2} W_2 \frac{\partial \phi_1(k)}{\partial \mathbf{h}_1} \frac{\partial \mathbf{h}_1(k)}{\partial \widehat{\mathbf{w}}_1} \right]' \mathbf{e}(k) + \alpha \Delta \widehat{\mathbf{w}}(k) \quad (3.17)$$

Demonstração. A partir do lema 4, e usando as equações (3.13), (3.15), (3.4) e (3.5) chega-se à equação (3.17). \square

Pelo teorema 7 e pelos lemas 4 e 5, o intervalo para o passo ou taxa de aprendizagem, η , poderia ser atualizado a cada instante de tempo ou iteração. Se fosse necessário recalcular o passo η a cada instante, existiria uma alta demanda computacional, o que poderia colocar em risco o uso do método em tempo real. Contudo, a

atualização do passo pode ser feita apenas quando estritamente necessário. Uma rotina de gerenciamento poderia ser usada para decidir sobre a aplicação do teorema 7 ou não. Esta rotina de gerenciamento é baseada no teorema 8, apresentado abaixo e explicada na seção 3.4. Devido às propriedades não lineares do modelo, será adotada a norma máxima do jacobiano da saída da rede **MLP** em relação seu vetor de parâmetros ajustáveis.

Teorema 8. *Dada uma rede **MLP** como apresentada pela equação (3.12), com a regra de ajuste de parâmetros expressa pela equação (3.17), com funções de ativação lineares para a segunda camada de neurônios processadores e funções de ativação sigmoidais para a primeira camada de neurônios processadores (veja a equação (3.21)), e respeitando-se as suposições 1-3, o processo de ajuste de parâmetros converge para $\hat{\mathbf{w}}^*$ quando η pertence ao intervalo*

$$0 < \eta < \frac{32}{\mathbf{f}_c \kappa_{\phi_1}^2 (16 n_s (N + 1) + n_s N (n_e + 1) w_{2\max}^2 u_{m\max}^2)} \quad (3.18)$$

onde $w_{2\max}$ e $u_{m\max}$ são os valores absolutos máximos aceitáveis para os parâmetros ajustáveis da segunda camada de neurônios processadores e para as componentes do vetor de entrada, respectivamente, e κ_{ϕ_1} é um parâmetro constante da função sigmoideal.

Demonstração. A partir do lema 4, obtém-se

$$\begin{aligned} \max_k \left\{ \left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \right\|^2 \right\} &= \max_k \left\{ tr \left(\begin{bmatrix} \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \\ \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \end{bmatrix} \begin{bmatrix} \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \\ \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}} \end{bmatrix}' \right) \right\} \\ &= \max_k \left\{ \left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}_2} \right\|^2 + \left\| \frac{\partial \hat{\mathbf{y}}(k)}{\partial \hat{\mathbf{w}}_1} \right\|^2 \right\}. \end{aligned} \quad (3.19)$$

Nas equações (3.13) e (3.15), observa-se que $\partial \phi_2(k)/\partial \mathbf{h}_2$ é uma matriz diagonal com dimensão $\mathbb{R}^{n_s \times n_s}$, $\partial \mathbf{h}_2(k)/\partial \hat{\mathbf{w}}_2$ é uma matriz pseudo-diagonal com o arranjo (3.14), $\partial \mathbf{h}_2(k)/\partial \phi_1 = W_2$, $\partial \phi_1(k)/\partial \mathbf{h}_1$ é uma matriz diagonal, e $\partial \mathbf{h}_1(k)/\partial \hat{\mathbf{w}}_1$ é uma matriz pseudo-diagonal com o arranjo (3.16). A equação (3.21) mostra as funções de ativação.

Para este conjunto de funções de ativação, $\partial\phi_2(k)/\partial\mathbf{h}_2$ torna-se uma matriz identidade com dimensão $\mathbb{R}^{n_s \times n_s}$.

Assim, a equação (3.13) pode ser calculada, e na seqüência o termo $\|\partial\hat{\mathbf{y}}(k)/\partial\hat{\mathbf{w}}_2\|^2$ é calculado e todos os elementos nulos são eliminados. Isto resulta em

$$\max_k \left\{ \left\| \frac{\partial\hat{\mathbf{y}}(k)}{\partial\hat{\mathbf{w}}_2} \right\|^2 \right\} = n_s (N + 1) \hat{y}_{1\max}^2.$$

A equação (3.15) e o termo $\|\partial\hat{\mathbf{y}}(k)/\partial\hat{\mathbf{w}}_1\|^2$ são então calculados. Isto resulta em

$$\max_k \left\{ \left\| \frac{\partial\hat{\mathbf{y}}(k)}{\partial\hat{\mathbf{w}}_1} \right\|^2 \right\} = n_s N (n_e + 1) \left| \frac{\partial\phi_{1jj}}{\partial h_{1jj}} \right|_{\max}^2 w_{2\max}^2 u_{m\max}^2.$$

Após as manipulações acima, a equação (3.19) pode ser obtida. Uma vez que $\hat{y}_{1\max} = \kappa_{\phi_1}$, $|\partial\phi_{1jj}/\partial h_{1jj}|_{\max} = 1/4$, e considerando $w_{2\max} \in \mathcal{W}_2$, $u_{m\max} \in \mathcal{U}_m$, com $\mathcal{W}_2 = [-w_{2\max}, w_{2\max}]$ e $\mathcal{U}_m = [-u_{m\max}, u_{m\max}]$, pode-se obter

$$\max_k \left\{ \left\| \frac{\partial\mathbf{y}(k)}{\partial\hat{\mathbf{w}}} \right\|^2 \right\} = \frac{\kappa_{\phi_1}^2 [16 n_s (N + 1) + n_s N (n_e + 1) w_{2\max}^2 u_{m\max}^2]}{16}. \quad (3.20)$$

A substituição da equação (3.20) na equação (3.9) resulta na equação (3.18). \square

3.4 Algoritmo Adaptativo para Ajuste de Parâmetros na Rede MLP

Na literatura técnica são apresentados resultados mostrando que a rede **MLP** do tipo⁵ $\phi : \mathbb{R}^{n_p} \times \mathbb{R}^{n_e} \mapsto \mathbb{R}$, possui a característica de aproximador universal se ϕ_{2_i} é linear e $\phi_{1,j}(\cdot)$ é sigmoidal, ou então de base radial, sendo essas duas últimas funções

⁵As redes **MLP**s são aproximadores universais para modelos do tipo $\phi : \mathbb{R}^{n_p} \times \mathbb{R}^{n_e} \mapsto \mathbb{R}$, com $N < \infty$ e $n_s = 1$, onde N é o número de neurônios na camada escondida. Este texto considera apenas o uso de **MLP** com $n_s = 1$.

do tipo suave (Cybenko, 1989; Hornik *et al.*, 1989; Mhaskar, 1992; Barron, 1993). O formato linear-*sigmoidal* é

$$\begin{cases} \phi_{2_i}(x) = x \\ \phi_{1_j}(x) = \frac{\kappa_{\phi_1}}{1 + e^{-x}} \end{cases} \quad (3.21)$$

onde $\kappa_{\phi_1} \in \mathbb{R}_+^*$ é um parâmetro constante da função sigmoidal. Usualmente $\kappa_{\phi_1} = 1$.

A capacidade deste tipo de **MLP** aproximar o modelo do sistema com determinada precisão ou erro está vinculada à sua região de densidade (veja a definição 12). Para a rede **MLP**, a região de densidade é definida pelo número de neurônios presentes na primeira camada de neurônios processadores ou camada escondida, e pela precisão desejada para a aproximação (Hornik *et al.*, 1989).

A escolha de η é uma decisão crítica para se assegurar a convergência do processo de ajuste de parâmetros. Se η é pequeno, a convergência será muito lenta devido ao grande número de iterações necessárias para alcançar um mínimo. Se η é grande, no começo a convergência será rápida, mas eventualmente o processo pode divergir e não alcançar o mínimo local. O Teorema 8 oferece um critério para conseguir um η que mantenha a convergência.

Analisando a equação (3.18), pode-se observar que η depende dos valores adotados para $\alpha \in [0; 1) \subset \mathbb{R}$ (considerando-se apenas valores positivos para α evita-se uma convergência oscilatória), $u_{m_{\max}} \in \mathbb{R}_+^*$ and $w_{2_{\max}} \in \mathbb{R}_+^*$, que devem ser pré-escolhidos. Desde que $w_{2_{\max}}$ é desconhecido *a priori*, uma superação deste valor pode ocorrer durante o processo de ajuste de parâmetros. $u_{m_{\max}}$ pode também ser superado devido a distúrbios no sinal de entrada. Assim, um procedimento adaptativo para a especificação do passo ou taxa de aprendizagem é necessário no sentido de estabelecer dinamicamente valores para $u_{m_{\max}}$ e $w_{2_{\max}}$, e evitar que o processo de ajuste de parâmetros torne-se instável.

O algoritmo 1 realiza tal procedimento, fazendo uma checagem para verificar se o

Algorithm 1 Algoritmo Adaptativo para a Rede **MLP**.

```

    definir variável global  $u_{m_{\max}}^0 \in \mathbb{R}^*$ 
    definir variável global  $w_{2_{\max}}^0 \in \mathbb{R}_+^*$ ;
    definir variável global  $\kappa \in (1; +\infty) \subset \mathbb{R}$ ;
    definir variável global  $\mathbf{J}_0 \in \mathbb{R}_+^*$ ;
    :
if ( $\mathbf{J} > \mathbf{J}_0$ ) then
    estimação de Parâmetros(...);
else if ( $\mathbf{J} \leq \mathbf{J}_0$ ) then
     $\Delta \hat{\mathbf{w}}(k) = 0_{n_p}$ ,  $\Delta \hat{\mathbf{w}}(k+1) = 0_{n_p}$ ;
end if
    :
função estimação de parâmetros(( $u(k) \in \mathbb{R}^{n_e}$ ,  $y(k) \in \mathbb{R}^{n_s}$ ,  $\hat{y}(k) \in \mathbb{R}^{n_s}$ )
    iniciar função
    definir variável  $u_{m_{\max}} \in \mathbb{R}$ ;
    definir variável  $w_{2_{\max}} \in \mathbb{R}$ ;
     $u_{m_{\max}} := \text{obter valor máximo}(u)$ ; {% A rotina de gerenciamento começa aqui.}
     $w_{2_{\max}} := \text{obter valor máximo}(W_2)$ ;
if ( $u_{m_{\max}} \geq u_{m_{\max}}^0$ ) ou ( $w_{2_{\max}} \geq w_{2_{\max}}^0$ ) then
    if ( $u_{m_{\max}} \geq u_{m_{\max}}^0$ ) then
         $u_{m_{\max}}^0 := \kappa \times u_{m_{\max}}$ ;
    end if
    if ( $w_{2_{\max}} \geq w_{2_{\max}}^0$ ) then
         $w_{2_{\max}}^0 := \kappa \times w_{2_{\max}}$ ;
    end if
    obter passo( $u_{m_{\max}}^0, w_{2_{\max}}^0$ ); {% Se necessário, um novo passo é calculado neste ponto.}
end if {% A rotina de gerenciamento termina neste ponto.}
    aplicar o método do gradiente( $\cdot$ ); {% O método do gradiente é aplicado neste ponto.}
end function

```

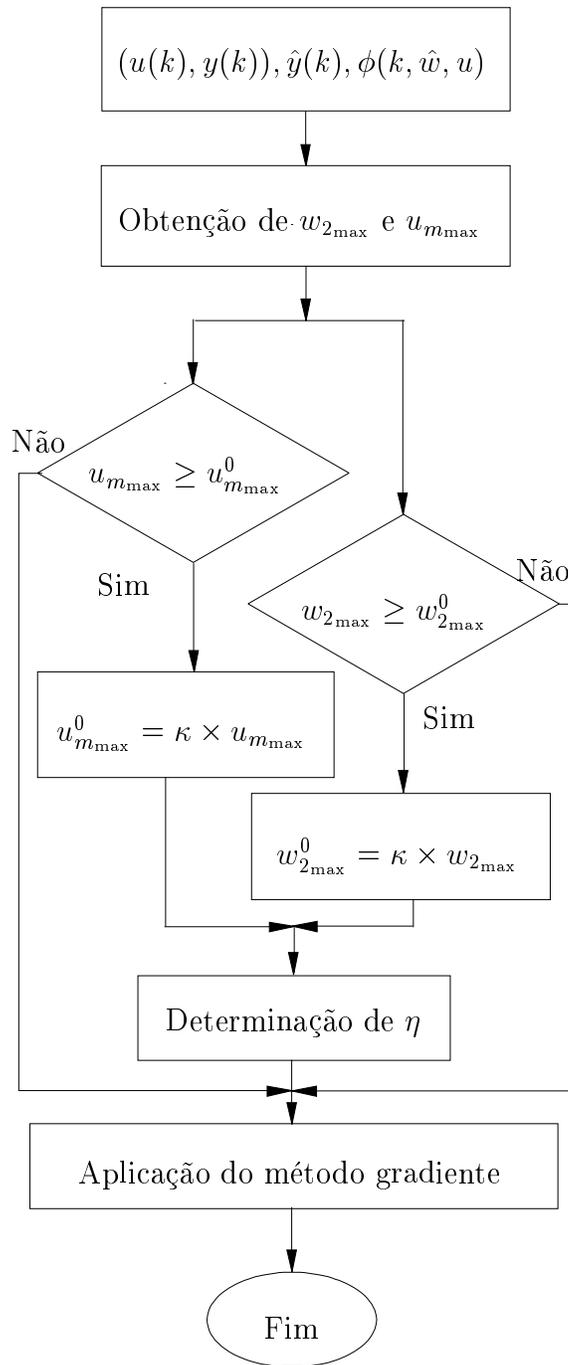


Figura 3.4: Fluxograma do algoritmo de ajuste de parâmetros para a rede **MLP**.

ajuste de parâmetros irá convergir. Este algoritmo é ilustrado pelo fluxograma da figura 3.4. O valor para $\alpha \in [0; 1) \subset \mathbb{R}$, e os valores iniciais para $u_{m_{\max}}$ e $w_{2_{\max}}$, chamados de $u_{m_{\max}}^0$ e $w_{2_{\max}}^0$, são todos pré-escolhidos. Então, um passo inicial η^0 é calculado usando a equação (3.18). A partir deste ponto, se $u_{m_{\max}} \geq u_{m_{\max}}^0$ para cada par $(u(k), \hat{y}(k))$ enviado para o processo de ajuste de parâmetros é verdadeiro, ou se $w_{2_{\max}} \geq w_{2_{\max}}^0$ é verdadeiro, então os valores de $u_{m_{\max}}^0$ e $w_{2_{\max}}^0$ são modificados para $\kappa \cdot u_{m_{\max}}$ e $\kappa \cdot w_{2_{\max}}$, respectivamente, onde $\kappa \in (1; +\infty) \subset \mathbb{R}$, e o passo é então recalculado.

3.4.1 Sobre as Condições Iniciais para a Rede MLP

Os métodos baseados no vetor gradiente são sensíveis às condições iniciais. Se o processo de ajuste de parâmetros é iniciado com $\hat{\mathbf{w}}^0$ próximo de $\hat{\mathbf{w}}^*$, então ele atingirá o objetivo rapidamente, mas se é iniciado distante de $\hat{\mathbf{w}}^*$, então ele atingirá o objetivo em um tempo mais longo. O problema principal neste caso é que $\hat{\mathbf{w}}^*$ é desconhecido e existem poucas informações a respeito do sistema a ser identificado ou modelado.

Neste modelo de **MLP**, as funções de ativação da primeira camada de neurônios processadores são limitadas e suaves, com soluções em \mathbb{R} para cada parâmetro da rede. Com isto, o subespaço \mathcal{W}^* contendo o vetor de parâmetros ótimos pode ser considerado o \mathbb{R}^{np} , e o subespaço \mathcal{W}^0 contendo o vetor de parâmetros iniciais pode ser considerado também o \mathbb{R}^{np} .

Contudo, existem alguns problemas relativos à saturação dos sinais de saída dos neurônios pertencentes à primeira camada, quando muitos parâmetros da rede assumem valores muito superiores a uma unidade [i.e., $\gg 1$] durante o processo de ajuste de parâmetros. Os problemas de saturação surgem em duas situações: (i) quando os parâmetros são iniciados com valores muito maiores que a unidade; (ii) quando o algoritmo de ajuste de parâmetros diverge. Para a primeira situação a consequência seria a baixa velocidade de convergência em direção a $\hat{\mathbf{w}}^*$. Logo, muitos pesquisadores têm sugerido que os parâmetros sejam iniciados e mantidos com valores próximo a uma unidade (veja por exemplo Hassoun (1995), Mehrotra *et al.* (1997) e Hertz *et al.*

(1991)). Para a segunda situação, o algoritmo proposto é sempre convergente porque a taxa de aprendizagem é escolhida com base no teorema 8 e existe uma rotina de gerenciamento para decidir sobre a aplicação do teorema. Assim, a rede torna-se livre do problema de saturação por divergência do ajuste de parâmetros.

O problema de determinar uma condição inicial para a rede **MLP** tem sido intensivamente pesquisado. Muitas técnicas têm sido sugeridas, mas não existe ainda uma solução geral para o problema. A literatura técnica apresenta duas abordagens básicas: o *easiest-path* ou caminho mais fácil, e o *shortest-path* ou caminho mais curto (Hassoun, 1995; Mehrotra *et al.*, 1997; Castro e Von Zuben, 1998; Redondo e Espinosa, 2000).

O paradigma *easiest-path* propõe a iniciação com os parâmetros distribuídos aleatoriamente e uniformemente sobre um intervalo $(-r; r)$, usualmente com $r \leq 1$, sem qualquer justificativa matemática.

Os paradigmas *shortest-path* são mais complexos. Eles levam em consideração um conhecimento *a priori* sobre os sinais do sistema, para usá-los, por exemplo, em algum método estatístico para a determinação de uma boa condição inicial. Ambos os paradigmas podem ser empregados no método proposto, mas está fora do escopo deste trabalho decidir sobre qual é o melhor. Aqui é adotado a abordagem *easiest-path*.

3.4.2 Sobre a Escolha do Número de Neurônios na Primeira Camada

A escolha do número adequado de neurônios para a primeira camada de neurônios processadores da rede **MLP** é também um problema difícil de resolver. A literatura técnica não apresenta um critério claro e objetivo para isto (Narendra e Mukhopadhyay, 1996; Mehrotra *et al.*, 1997). Barron (1993) apresenta um critério baseado na integral de um erro quadrático da aproximação desejada, mas o critério não é viável para propósitos práticos por demandar conhecimento sobre o sistema.

A melhor solução seria encontrada se a região de densidade da rede **MLP** pudesse ser maior que a do sistema. Isto é difícil de ser obtido em situações práticas, porque o sistema é assumido como sendo desconhecido ou não bem conhecido.

A pior situação ocorreria quando a região de densidade da rede **MLP** fosse menor que a do sistema. No entanto, como será discutido no capítulo 5, esta situação pode ser bastante útil na identificação de sistemas dinâmicos. Com o uso de um método de ajuste de parâmetros adaptativo, pode-se fazer a rede **MLP** viajar sobre todas as trajetórias ou órbitas do sistema [i.e., dentro de toda sua região de densidade]. Isto permitiria que o número de neurônios na primeira camada pudesse ser escolhido experimentalmente ou empiricamente. Isto será abordado na seção 5.2.

3.5 Sumário

Neste capítulo foi apresentado e descrito o modelo **MLP** de **RNA** e apresentado um método de ajuste de parâmetros para o modelo. A seção 3.1 apresentou uma breve discussão sobre diferentes métodos de ajuste de parâmetros para a rede **MLP** sob a óptica do esforço computacional para aplicações *on-line*, e enfatizou que métodos baseados no gradiente podem ser úteis para tais situações.

A seção 3.2 contribuiu com a análise de estabilidade ou convergência, com base no segundo método de Lyapunov, para o método do gradiente conjugado apresentado genericamente na seção 2.6. Chegou-se a um critério em que um intervalo de aceitação para o passo da regra de ajuste de parâmetros é obtido a partir do jacobiano do modelo da rede em relação aos seus parâmetros internos (veja o teorema 7) .

A seção 3.3 contribuiu com a aplicação dos procedimentos e resultados obtidos na seção 3.2 à rede **MLP**, e como consequência foi obtido uma expressão para um limitante superior da taxa de aprendizagem da rede **MLP** (veja o teorema 8).

A seção 3.4 contribuiu com a apresentação de um algoritmo adaptativo para a rede **MLP** usando os resultados da seção 3.4. Este algoritmo visa assegurar a con-

vergência da rede por meio do monitoramento dos parâmetros internos e dos sinais de entrada para, se necessário, alterar a taxa de aprendizagem. Nesta seção, também foram discutidos os aspectos de condição inicial e escolha do número de neurônios para a rede.

Capítulo 4

Descrição de um Sistema Robótico

Neste capítulo, é feita uma descrição matemática de um sistema robótico, apresentada uma análise cinemática e uma análise dinâmica, com a inclusão do modelo de acionadores elétricos. São feitos também comentários a respeito dos principais sensores usados na espaço de juntas.

4.1 Introdução

UM sistema robótico ou **manipulador robótico** é constituído de um sistema mecânico composto de **elos** (*links*) conectados por **juntas** em uma cadeia aberta ou fechada, controlada por um sistema hierárquico programável em vários níveis. Esta cadeia de elementos mecânicos tem por objetivo final o posicionamento e orientação da extremidade do último elo em um ponto do espaço, com ou sem controle de trajetórias pré-estabelecido.

As juntas são tipicamente classificadas como: **rotacionais** ou **revolutas**, que são as permitem movimentos de rotação entre dois elos; e **prismáticas**, que permitem movimentos de translação entre os elos. Define-se como **variáveis de juntas**, os deslocamentos relativos entre dois elos adjacentes, sendo denotadas por θ_i e chamadas

de **deslocamento angular** para as juntas rotacionais (**R**), e sendo denotada por d_i e chamadas de **deslocamento linear** para as juntas prismáticas (**P**). A posição e a orientação da extremidade do último elo são influenciadas por todas as outras variáveis de juntas. As juntas são acionadas pela aplicação de torques ou forças externas por meio de **atuadores**, que podem ser: hidráulicos (que permitem torques ou forças mais elevadas); pneumáticos (que permitem movimentos de precisão, porém com baixa potência); ou elétricos (com características intermediárias entre os outros dois, mas com facilidade de implementações, para controle, e com custos mais baixos).

A maioria dos sensores usados estão no nível de controle de juntas e são para as medidas das variáveis de juntas e para algumas variáveis dos atuadores, como corrente e tensão elétrica. Porém, em alguns sistemas são usados sensores de torque, de posicionamento espacial, de visão, de proximidade, de deslizamento, de força, etc (Klafter *et al.*, 1989).

Em geral, o número de juntas determina o número de graus de liberdade (**GL**) do mecanismo (*Degree-Of-Freedom*, **DOF**). Tipicamente, um robô deve possuir no mínimo seis **GL** independentes, três para o posicionamento e três para a orientação. Com menos de seis **GL** independentes não é possível atingir um posicionamento-orientação espacial completo em um espaço de trabalho pré-definido. Aplicações em que o ambiente de trabalho possui obstáculos aos movimentos do robô podem requerer mais de seis **GL**. Um robô com mais de seis **GL** é dito **cinematicamente redundante**.

O estudo ou análise de sistemas robóticos pode ser dividido em duas partes: a **análise cinemática** e a **análise dinâmica**. Na análise cinemática é feito o estudo das variáveis de movimento do sistema (posições, velocidades e acelerações) sem preocupação com suas causas. Já a análise dinâmica estuda os movimentos preocupando-se com as suas causas [i.e., os torques ou forças aplicados às juntas].

Usualmente, os manipuladores são classificados com base em sua estrutura cinemática nas três primeiras juntas, que é mais conhecido como braço, sendo o punho, ou garra, descrito em separado.

Em geral, os manipuladores podem ser classificados em quatro categorias ou configurações geométricas: **articulado (RRR)**, **esférico (RRP)**, **cilíndrico (RPP)**, ou **cartesiano (PPP)**.

Para facilitar a escrita das muitas equações que se seguem, adotaremos para a primeira, segunda e terceira derivadas no tempo de alguma variável x , como por exemplo, posição cartesiana ou então variáveis de junta, as seguintes notações respectivamente \dot{x} , \ddot{x} e $x^{(3)}$.

4.2 Análise Cinemática de Manipuladores: Convenção de Denavit e Hartenberg

Cada junta tem um **GL**, podendo esta ser rotacional ou prismática. Para um manipulador com n juntas, numeradas de 1 até n , existem $n + 1$ elos, numerados de 0 (zero) até n . O elo zero, geralmente fixo, é chamado de *elo base do sistema* e o elo n possui a extremidade final do sistema. Cada junta i , para $i = 1, 2, \dots, n$, conecta os elos i e $i - 1$.

Um elo pode ser considerado como um corpo rígido, podendo definir assim uma relação entre duas juntas vizinhas. Pode ser especificado através de dois parâmetros que definem um posicionamento em dois eixos coordenados espaciais: comprimento (*link length*) e posição angular (*link twist*).

Uma junta também pode ser especificada por dois parâmetros, a distância de um elo ao próximo ao longo do eixo da junta (*link offset*), e pelo ângulo da junta, que é a distância angular de um elo ao próximo sobre o eixo da junta (*joint angle*).

Para facilitar a localização de cada elo, são fixados eixos de coordenadas (xyz) nos mesmos, onde o eixo i está fixado no elo i . Denavit e Hartenberg (1955) propuseram um método matricial de alocação sistemática dos eixos coordenados para cada elo de uma cadeia articulada. No método, o eixo de cada junta i é alinhado com o eixo z do

sistema de coordenadas $i - 1$. O eixo x do sistema de coordenadas $i - 1$ é direcionado ao longo da normal entre os eixos z_{i-1} e z_i , como visto na figura 4.1. Os parâmetros dos elos das juntas estão resumidos na tab. 4.1.

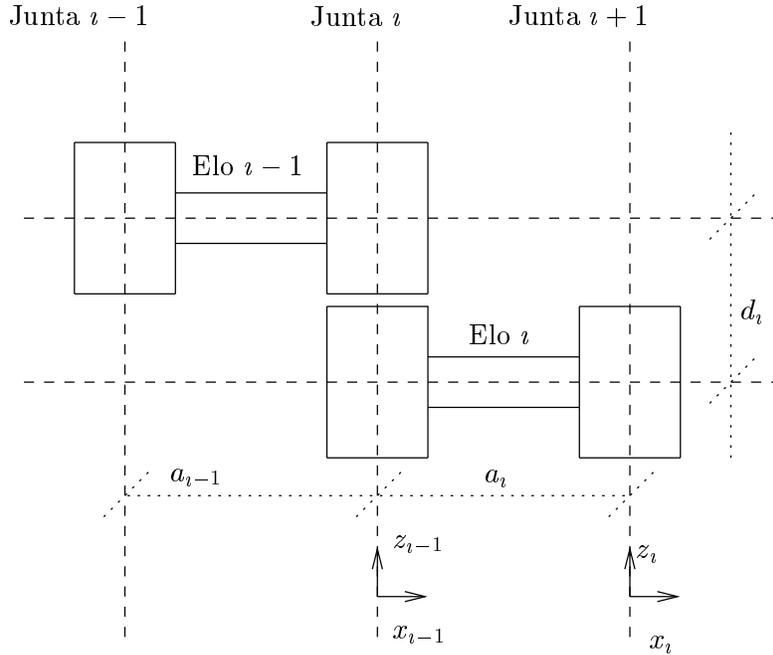


Figura 4.1: Localização dos eixos coordenados em uma cadeia articulada.

Para uma junta rotacional, θ_i é a variável de junta e d_i , *off-set*, é uma constante, enquanto que para uma junta prismática, d_i é a variável de junta e θ_i uma constante. É habitual generalizar-se as variáveis de junta por q_i , ditas **coordenadas generalizadas**.

A representação de Denavit e Hartenberg (**DH**) resulta em uma matriz de transformação de coordenadas entre elos adjacentes na forma (Spong e Vidyasagar, 1989)

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\text{sen} \theta_i \cos \alpha_i & \text{sen} \theta_i \text{sen} \alpha_i & a_i \cos \theta_i \\ \text{sen} \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \text{sen} \alpha_i & a_i \text{sen} \theta_i \\ 0 & \text{sen} \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Tabela 4.1: Parâmetros dos elos e das juntas

link length	a_i	distância entre os eixos z_{i-1} e z_i ao longo do eixo x_i
link twist	α_i	ângulo entre os eixos z_{i-1} e z_i sobre o eixo x_i
link offset	d_i	distância entre a origem do eixo coordenado $i - 1$ e o eixo x_i ao longo do eixo z_{i-1}
joint angle	θ_i	ângulo entre os eixos x_{i-1} e x_i sobre o eixo z_{i-1}

A representação das coordenadas de cada elo em relação ao sistema de coordenadas da base ($i = 0$) é dado por

$$\mathbf{T}_i^0 = \mathbf{T}_{i-1}^0 \mathbf{A}_i^{i-1} \quad (4.1)$$

onde \mathbf{T}_i^0 representa a transformação homogênea do sistema de coordenadas i em relação ao sistema de coordenadas-base.

Para um manipulador com n elos considerados rígidos, a **análise cinemática direta** é a determinação da posição e orientação da extremidade do último elo em relação ao sistema de coordenadas-base como função das variáveis de junta.

Este resultado pode ser obtido através da equação (4.1) na forma

$$\mathbf{T}_n^0 = \mathbf{A}_1^0 \mathbf{A}_2^1 \cdots \mathbf{A}_n^{n-1} \triangleq f(\mathbf{q}) \quad (4.2)$$

que é o produto das matrizes de transformação de cada elo. A transformação \mathbf{T}_n^0 será aqui denotada simplificada na forma \mathbf{T}_n . De forma equivalente, $f(\mathbf{q})$ é uma função de transformação de coordenadas, e \mathbf{q} um vetor de coordenadas generalizadas.

A **análise cinemática inversa** consiste na determinação das variáveis de juntas que possibilite um determinado posicionamento e/ou orientação da extremidade do último elo, tendo a forma

$$\mathbf{q} = f^{-1}(\mathbf{T}_n). \quad (4.3)$$

Este problema não possui uma única solução. Se não existir solução na equação (4.3) para uma determinada posição do manipulador esta é dita singular. As singularidades podem ocorrer devido a algum alinhamento de eixos de juntas, ou a reduzidos números de **GL**, ou ainda a um ponto fora do alcance do manipulador.

Define-se **trajetória** como um conjunto de pontos no espaço pelos quais o manipulador deverá posicionar-se seqüencialmente no tempo, continuamente ou discretamente. Em muitos casos, a trajetória é especificada com inclusão de velocidades cartesianas, que podem ser obtidas mediante diferenciação da equação (4.2), *equação cinemática direta*, na forma

$$\dot{\mathbf{x}} = \mathbf{D}_f(\mathbf{q}) \dot{\mathbf{q}}$$

onde $\mathbf{D}_f(\mathbf{q})$ é uma matriz jacobiana, e relaciona velocidades no espaço das juntas com a velocidade cartesiana da extremidade do último elo do manipulador.

Para posições de não singularidade em robôs com seis **GL** a matriz jacobiana é inversível, dando origem à chamada *matriz jacobiana inversa* na forma

$$\dot{\mathbf{q}} = \mathbf{D}_f^{-1}(\mathbf{q}) \dot{\mathbf{x}} \quad (4.4)$$

que relaciona velocidades no espaço cartesiano com velocidades no espaço das juntas.

Assim, especificada uma trajetória a ser seguida pelo manipulador, e estabelecendo-se restrições de velocidades cartesianas, $\dot{\mathbf{x}}$, pode-se especificar velocidades para o espaço de juntas de maneira a permitir ao manipulador sair de uma posição inicial a uma posição final, passando por todos os pontos da trajetória, de forma contínua. O planejamento de trajetória feito desta maneira, baseado na equação (4.4), é chamado de *Planejamento pelo Jacobiano Inverso*, o qual assume que a matriz jacobiana é inversível para cada ponto da trajetória especificada.

4.3 Análise da Dinâmica de Manipuladores

A análise da dinâmica de manipuladores pode ser feita baseada em equações de movimento dos manipuladores, sendo estes movimentos as respostas a torques externos aplicados, e considerando o sistema como sendo um corpo rígido. Existem dois problemas básicos relacionados com a dinâmica de manipuladores:

Dinâmica Direta Onde as equações de movimento são resolvidas para se encontrar as respostas a torques aplicados.

Dinâmica Inversa Onde as equações de movimento são resolvidas para a determinação dos torques necessários para um dado movimento.

Objetivando simplificações, muitos pesquisadores modelam os sistemas robóticos desconsiderando a dinâmica dos atuadores de cada junta. Para muitas aplicações, esta simplificação não satisfaz a realidade do sistema. Assim outros pesquisadores têm modelado os sistemas robóticos com a inclusão da dinâmica dos atuadores de cada junta e até mesmo alguns outros incluem a dinâmica dos amplificadores de potência elétrica usados no acionamento (Sage *et al.*, 1999).

O modelo dinâmico de um manipulador com n **GL** pode ser obtido a partir da equação de Euler-Lagrange do sistema (Spong e Vidyasagar, 1989). O resultado das manipulações da equação de Euler-Lagrange é uma equação diferencial não linear matricial de ordem $2 \times n$, no formato¹

$$\boldsymbol{\tau}_r(t) = \mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (4.5)$$

onde $\mathbf{q}(t) \in \mathbb{R}^n$ é o vetor de coordenadas generalizadas, $\mathbf{D}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ é a matriz de inércia dos *links*, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ é a matriz de torques (ou forças) de coriolis e centrípetas, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^n$ é o vetor de torques gravitacionais e $\boldsymbol{\tau}_r(t) \in \mathbb{R}^n$ é o vetor de torques aplicados nas juntas ou torques resistentes.

¹Um modelo mais completo que inclui atrito viscoso e atrito seco pode ser encontrado em (Sage *et al.*, 1999).

O modelo descrito pela equação (4.5) possui algumas propriedades fundamentais que podem ser exploradas para facilitar a análise e o projeto de sistemas de controle (Ortega e Spong, 1989).

Propriedade 1. *A matriz de inércia $\mathbf{D}(\mathbf{q})$ é simétrica, definida positiva, e ambos $\mathbf{D}(\mathbf{q})$ e $\mathbf{D}^{-1}(\mathbf{q})$ são uniformemente limitadas como uma função de $\mathbf{q}(t) \in \mathbb{R}^n$.*

Observação 7. Abordando o assunto de maneira mais rigorosa, para que a matriz de inércia seja limitada é necessário que todas as juntas sejam rotacionais.

Propriedade 2. *Existe uma entrada de controle independente para cada \mathbf{GL} do sistema.*

Propriedade 3. *Todos os parâmetros constantes de interesse, tais como massas dos links, momentos de inércias, etc., aparecem como coeficientes de funções conhecidas das coordenadas generalizadas.*

Propriedade 4. *Para o modelo expresso pela equação (4.5), se existe uma matriz $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{D}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$, então $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}})$ é uma matriz anti-simétrica.*

4.3.1 Dinâmica do Atuador Elétrico

Na maioria das aplicações, os manipuladores robóticos possuem suas juntas acionadas por motores elétricos de corrente contínua [*direct current (DC)*]. Na figura 4.2 temos o modelo do sistema elétrico e na figura 4.3 o modelo do sistema mecânico de um motor elétrico sendo a equação dinâmica do sistema elétrico dada por

$$L \frac{d i(t)}{d t} + R i(t) + e_{em}(t) = k_v u(t) \quad (4.6)$$

onde L é a indutância da armadura, R é a resistência elétrica da armadura do motor, $i(t)$ é a corrente elétrica do sistema, $e_{em}(t)$ é a força contra-eletromotriz, k_v é o ganho de voltagem do amplificador de potência que fornece energia ao motor, e $u(t)$ é a

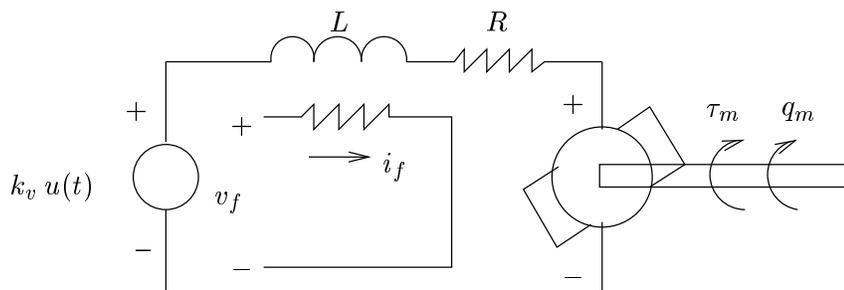


Figura 4.2: Modelo de campo elétrico do motor.

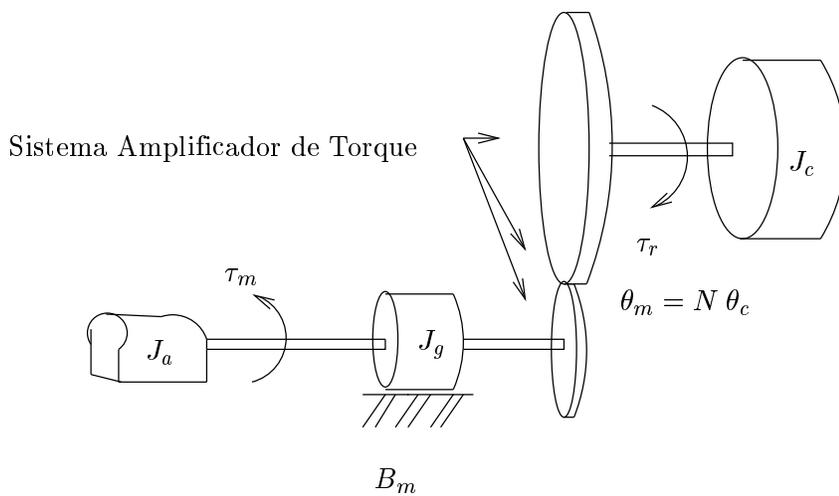


Figura 4.3: Modelo mecânico do motor elétrico.

voltagem elétrica aplicada na entrada do amplificador. O produto $k_v u(t)$ modela a voltagem elétrica aplicada na armadura do motor.

Para motores **DC** a campo constante [i.e., a corrente de campo constante, $i_f = i_{f_0} = cte.$], a força contra-eletromotriz é proporcional à velocidade angular do eixo do motor na forma:

$$e_{em}(t) = K_{em} \dot{q}_m \quad (4.7)$$

onde q_m é a posição angular do eixo do motor e K_{em} é uma constante de força contra-eletromotriz.

O torque produzido pelo motor é proporcional à corrente elétrica no formato

$$\tau_m(t) = K_t i(t) \quad (4.8)$$

onde τ_m é o torque motor e K_t é a constante de torque.

No caso de corpos rígidos, no acionamento de uma carga por um motor elétrico muitas vezes são utilizados sistemas redutores de movimento, ou amplificadores de torque, para adequar o movimento ou torque do motor elétrico ao movimento ou torque requerido pela carga. Assim, temos a seguinte relação entre os movimentos da carga e do motor elétrico:

$$N = \frac{q_m}{q} \quad (4.9)$$

onde N é o fator de redução e q é a posição angular da carga. A relação entre os torques é dada por

$$N = \frac{\tau}{\tau_m} \quad (4.10)$$

onde τ é o torque aplicado à carga.

A relação entre o torque motor e o torque resistente [veja a figura 4.3] possui a forma

$$\tau_m(t) - \frac{\tau_r(t)}{N} = J_m \ddot{q}_m + B_m \dot{q}_m \quad (4.11)$$

onde $J_m = J_a + J_g$ é o momento de inércia do atuador [i.e., a soma dos momentos de inércia do eixo do motor e do sistema redutor], e B_m é o coeficiente de atrito viscoso.

Substituindo a equação (4.9) na equação (4.7), a equação (4.10) na equação (4.8), e os resultados sendo substituídos na equação (4.6), é obtido:

$$\frac{L}{N K_t} \dot{\tau}(t) + \frac{R}{N K_t} \tau(t) + N K_{em} \dot{q} = k_v u(t). \quad (4.12)$$

Com a substituição das equações (4.10) e (4.9) na equação (4.11), é obtido

$$\tau(t) = \tau_r(t) + N^2 J_m \ddot{q} + N^2 B_m \dot{q}. \quad (4.13)$$

As equações (4.12) e (4.13) foram obtidas para atuação de um único motor. No caso de sistemas robóticos, considerando todas as juntas como sendo acionadas por motores elétricos de corrente contínua, temos um total de n motores elétricos [i.e., n motores corresponde a n **GL**]. Assim estas equações podem ser reescritas em formato vetorial como

$$\mathbf{L} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \dot{\boldsymbol{\tau}}(t) + \mathbf{R} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \boldsymbol{\tau}(t) + \mathbf{N} \mathbf{K}_{em} \dot{\mathbf{q}} = \mathbf{k}_v \mathbf{u}(t) \quad (4.14)$$

e

$$\boldsymbol{\tau}(t) = \boldsymbol{\tau}_r(t) + \mathbf{N}^2 \mathbf{J}_m \ddot{\mathbf{q}} + \mathbf{N}^2 \mathbf{B}_m \dot{\mathbf{q}} \quad (4.15)$$

onde $\boldsymbol{\tau}(t) \in \mathbb{R}^n$ é o vetor de torques aplicados às juntas, $\boldsymbol{\tau}_r(t) \in \mathbb{R}^n$ é o vetor de torques resistentes correspondentes à dinâmica do manipulador robótico, $\mathbf{u}(t) \in \mathbb{R}^n$ é o vetor de voltagens elétricas aplicadas nas armaduras dos motores, $\mathbf{q}(t) \in \mathbb{R}^n$ é o vetor de coordenadas generalizadas, e ainda \mathbf{L} , \mathbf{N} , \mathbf{K}_t , \mathbf{R} , \mathbf{K}_{em} , \mathbf{K}_v , \mathbf{J}_m , \mathbf{B}_m , todos pertencentes a $\mathbb{R}^{n \times n}$, são matrizes diagonais contendo respectivamente as indutâncias de armadura, os fatores de redução, as constantes de torque, as resistências de armadura, as constantes de forças contra-eletromotrizes, os ganhos de voltagens dos amplificadores de potência, os momentos de inércia dos atuadores, e os coeficientes de atrito viscoso, sendo todos relativos ao acionamento de cada junta do sistema independente.

4.3.2 Dinâmica do Manipulador com Inclusão do Atuador Elétrico

Para um sistema robótico, o torque resistente é dado pela equação (4.5). A substituição da equação (4.5) na equação (4.15) resulta em

$$\boldsymbol{\tau}(t) = \mathbf{D}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \mathbf{N}^2 \mathbf{J}_m \ddot{\mathbf{q}} + \mathbf{N}^2 \mathbf{B}_m \dot{\mathbf{q}}. \quad (4.16)$$

A derivada em relação ao tempo da equação (4.16) resulta em

$$\begin{aligned} \dot{\boldsymbol{\tau}}(t) = & \mathbf{D}(\mathbf{q}) \mathbf{q}^{(3)} + \dot{\mathbf{D}}(\mathbf{q}) \ddot{\mathbf{q}} + 2 \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \ddot{\mathbf{q}} + \dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \\ & + \dot{\mathbf{G}}(\mathbf{q}) + \mathbf{N}^2 \mathbf{J}_m \mathbf{q}^{(3)} + \mathbf{N}^2 \mathbf{B}_m \ddot{\mathbf{q}}. \end{aligned} \quad (4.17)$$

A substituição das equações (4.16) e (4.17) na equação (4.14) tem como resultado

$$\begin{aligned} \mathbf{k}_v \mathbf{u}(t) = & \mathbf{L} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \mathbf{M}(\mathbf{q}) \mathbf{q}^{(3)} + \mathbf{L} \mathbf{N}^{-1} \mathbf{K}_t^{-1} [\mathbf{F}(\mathbf{q}) + \mathbf{N}^2 \mathbf{B}_m] \ddot{\mathbf{q}} + \\ & + \mathbf{R} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{L} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \\ & + \mathbf{R} \mathbf{N}^{-1} \mathbf{K}_t^{-1} [\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{N}^2 \mathbf{B}_m] \dot{\mathbf{q}} + \\ & + \mathbf{N} \mathbf{K}_{em} \dot{\mathbf{q}} + \mathbf{L} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \dot{\mathbf{G}}(\mathbf{q}) + \mathbf{R} \mathbf{N}^{-1} \mathbf{K}_t^{-1} \mathbf{G}(\mathbf{q}) \end{aligned} \quad (4.18)$$

onde $\mathbf{M}(\mathbf{q}) = \mathbf{D}(\mathbf{q}) + \mathbf{N}^2 \mathbf{J}_m$ e $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{D}}(\mathbf{q}) + 2 \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$.

A equação (4.18) pode ser reescrita como uma equação de estado, para

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix}$$

no formato

$$\Upsilon : \begin{cases} \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \\ -[\mathbf{M}(\mathbf{q}(t))]^{-1} \bar{\mathbf{P}}(\dot{\mathbf{x}}(t)) \end{bmatrix} \\ + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ -[\mathbf{M}(\mathbf{q}(t))]^{-1} \mathbf{K}_t \mathbf{N} \mathbf{L}^{-1} \mathbf{K}_v \mathbf{u}(t) \end{bmatrix} \\ \mathbf{y}(t) = \mathbf{C} \mathbf{x}(t) \end{cases} \quad (4.19)$$

onde \mathbf{C} é a matriz que representa os ganhos dos sensores usados para obtenção das variáveis de saída, e

$$\begin{aligned} \bar{\mathbf{P}}(\dot{\mathbf{x}}(t)) = & \left[\mathbf{F}(\mathbf{q}) + \mathbf{N}^2 \mathbf{B}_m + \mathbf{R} \mathbf{L}^{-1} \mathbf{M}(\mathbf{q}) \right] \ddot{\mathbf{q}} + \\ & + \left[\dot{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{R} \mathbf{L}^{-1} \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{R} \mathbf{L}^{-1} \mathbf{N}^2 \mathbf{B}_m + \mathbf{L}^{-1} \mathbf{N}^2 \mathbf{K}_t \mathbf{K}_{em} \right] \dot{\mathbf{q}} + \\ & + \left[\dot{\mathbf{G}}(\mathbf{q}) + \mathbf{R} \mathbf{L}^{-1} \mathbf{G}(\mathbf{q}) \right]. \end{aligned} \quad (4.20)$$

Observação 8. Devido a existência de variáveis de posição, velocidade e aceleração para cada elo do sistema manipulador robótico, a ordem da equação de estado (4.19) é 3 vezes o número de \mathbf{GL} do sistema.

4.3.3 Sensores para o Espaço de Juntas

O vetor de estado da equação (4.19) é composto de três variáveis para as coordenadas generalizadas de cada junta: posição, velocidade e aceleração. Estas informações podem ser obtidas mediante o uso de sensores apropriados para cada componente do

vetor de estado [i.e., sensores de posição, velocidade e aceleração para todas as juntas], ou elas podem ser obtidas mediante o uso de sensores apropriados para posição e/ou velocidade de cada junta, e estimação dos outros componentes pelo uso de algum método de estimação de estados (Cerqueira e Badan Palhares, 1999; Cerqueira *et al.*, 2000b; Chen, 1970).

A análise de diversos fatores é necessária para decidir entre o uso de um sensor para cada componente do vetor de estados ou o uso de sensores para algumas componentes apenas e a estimação dos restantes (Klafter *et al.*, 1989). Usualmente, posição e velocidade são as variáveis mensuradas em cada junta.

Os mais comuns e eficientes sensores de posição são os *encoders* incrementais ópticos, que podem ser leves e baratos. Os *encoders* podem também ser usados como sensores de velocidade, mas a medição é feita de maneira indireta. Os mais comuns sensores de velocidade são os tacômetros, que são geralmente volumosos e pesados. Nas poucas vezes em que a aceleração é mensurada, são usados acelerômetros para a medição.

Fatores como espaço físico, custo, peso e outras questões técnicas justificam o fato de que em um grande número de manipuladores são usados apenas com sensores de posição para cada junta ou para algumas juntas. Por exemplo, o uso de tacômetros em juntas próximas da base ou na base pode não ser um problema, mas o uso de tacômetro em juntas da garra ou na garra pode tornar-se um problema. Adicionalmente, alguns fatores tecnológicos podem permitir apenas o uso de sensores de posição, como é o caso de alguns sistemas subaquáticos (Lane, 1998).

Informações sobre velocidade e aceleração poderiam ser obtidas mediante derivação ou por estimação de estado. Contudo, muitas vezes estas informações não podem ser obtidas mediante o uso destes métodos (Klafter *et al.*, 1989). Devido a problemas de quantização, o método de derivação é eficiente apenas dentro de uma faixa mediana de velocidade do sistema.

No caso do método de estimação de estados, o sistema pode tornar-se não ob-

servável quando analisado com a abordagem de sistemas lineares². Também os ganhos do observador podem amplificar ruídos medidos na saída do sistema e provocar uma performance geral insatisfatória para o sistema de controle (Spong e Vidyasagar, 1989). Adicionalmente, as variações paramétricas do modelo podem ser muito acentuadas, não permitindo assim uma eficiente estimação de estados pela abordagem linear. Fatores como estes têm tornado difícil o uso de algoritmos de controle adaptativos e/ou robustos em manipuladores construídos com apenas sensores de posição nas juntas.

As técnicas convencionais de controle por realimentação de estado, bem como maioria das técnicas avançadas de controle robusto e/ou adaptativo, fazem uso de informações sobre posições, velocidades e acelerações no espaço de juntas para a construção de um modelo dinâmico objetivando fazer-se um cancelamento da dinâmica do manipulador dentro de um bem elaborado algoritmo de controle (Chen, 1970; Spong e Vidyasagar, 1989; Ortega e Spong, 1989; Abdallah *et al.*, 1991; Sage *et al.*, 1999).

Com o objetivo de eliminar a necessidade do uso de um sensor para cada estado do sistema em algoritmos de controle avançado, ou de se realizar a estimação de estados, alguns estudos já estão surgindo na literatura especializada abordando este tema. Hsu e Lizarralde (1993; 1995) propuseram um esquema baseado em controladores adaptativos de modelo referência e leis de controle de estrutura variável usando apenas informações entrada-saída (Hsu, 1990). Outros trabalhos foram propostos usando um método baseado em filtragem não linear para suprimir a necessidade de medição da velocidade (Lizarralde e Wen, 1995; Lizarralde *et al.*, 1995; Lizarralde e Wen, 1996).

4.4 Modelo de um Sistema Articulado

Um sistema robótico articulado, também chamado de rotacional ou antropomórfico, possui suas juntas do braço do tipo rotacional (**RRR**). Uma característica importante nesta categoria de manipuladores é que a propriedade 1 é sempre válida

²Por exemplo, isto talvez ocorra quando apenas posições de juntas são mensuradas.

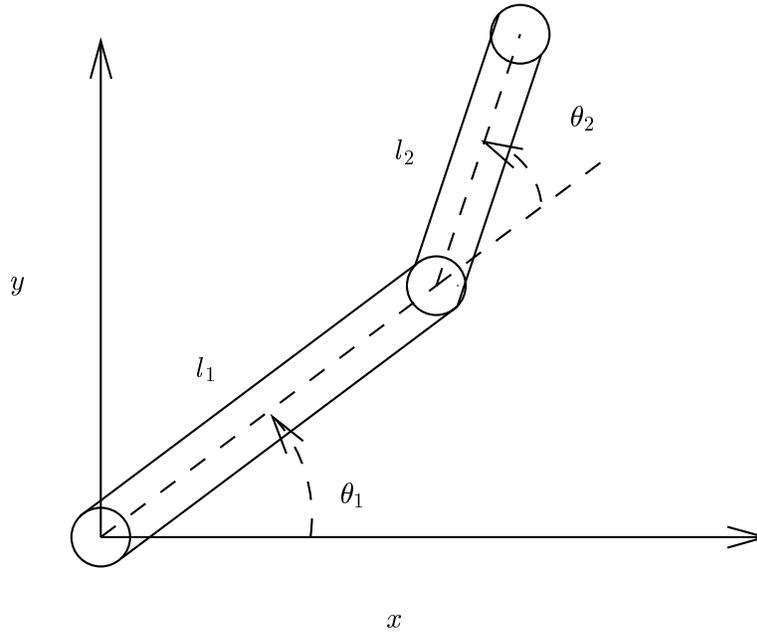


Figura 4.4: Sistema robótico de dois **GL**.

(veja a página 78). Nestes sistemas, para efeito de posicionamento do punho, devido a aspectos construtivos o controle da junta base não é freqüentemente problemático, nem mesmo a sua interação com o restante do mecanismo, ficando as maiores dificuldades para o controle do braço e ante-braço.

Aqui é apresentado um modelo de um sistema robótico articulado de dois **GL** (**RR**), usado em Cerqueira e Badan Palhares (1999), Cerqueira *et al.* (2000b) e Cerqueira *et al.* (2000c), correspondentes ao braço e ante-braço de um sistema de três **GL** e que pode ser visualizado pela figura 4.4. Detalhes sobre os significados físicos dos parâmetros podem ser encontrados em Spong e Vidyasagar (1989), Tarn *et al.* (1991) e Yu e Lloyd (1995). Modelos para sistemas de três e seis **GL** podem ser encontrados respectivamente em Leng e Chen (1997) e Neuman e Murray (1987).

Os termos $\mathbf{M}(\mathbf{q}(t)) \in \mathbb{R}^{2 \times 2}$ e $\bar{\mathbf{P}}(\dot{\mathbf{x}}(t)) = \left[\bar{P}_1(\dot{x}(t)) \quad \bar{P}_2(\dot{x}(t)) \right]' \in \mathbb{R}^{2 \times 1}$, respec-

tivamente, são da forma (veja as equações (4.18), (4.19) e (4.20))

$$M(\mathbf{q}(t)) = \begin{bmatrix} m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} C_2) + I_1 + I_2 + N_1^2 J_{m_1} & & \\ & m_2 (l_{c2}^2 + l_1 l_{c2} C_2) + I_2 & \\ & & m_2 (l_{c2}^2 + l_1 l_{c2} C_2) + I_2 \\ & & & m_2 l_{c2}^2 + I_2 + N_2^2 J_{m_2} \end{bmatrix}_{2 \times 2}$$

$$\begin{aligned} \bar{P}_1(\dot{\mathbf{x}}(t)) = & - m_2 l_1 l_{c2} S_2 (2 \omega_1 \alpha_2 + 4 \omega_2 \alpha_1 + 3 \omega_2 \alpha_2) + \\ & + \frac{R_1}{L_1} \{m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} C_2) + \\ & + I_1 + I_2 + N_1^2 J_{m_1}\} \alpha_1 + N_1^2 B_{m_1} \alpha_1 + \\ & + \frac{R_1}{L_1} \{m_2 (l_{c2}^2 + l_1 l_{c2} C_2) + I_2\} \alpha_2 + \\ & - m_2 l_1 l_{c2} C_2 (2 \omega_1 \omega_2^2 + \omega_2^3) + \\ & + \frac{R_1}{L_1} N_1^2 B_{m_1} \omega_1 + \\ & - \frac{R_1}{L_1} m_2 l_1 l_{c2} S_2 (2 \omega_1 \omega_2 + \omega_2^2) + \\ & + \frac{N_1^2 K_{t_1} K_{em_1}}{L_1} \omega_1 - (m_1 l_{c1} + m_2 l_1) g S_1 \omega_1 + \\ & - m_2 l_{c2} g S_{12} (\omega_1 + \omega_2) + \frac{R_1}{L_1} \{(m_1 l_{c1} + \\ & + m_2 l_1) g C_1 + m_2 l_{c2} g C_{12}\} \end{aligned}$$

$$\begin{aligned}
\overline{P}_2(\dot{\mathbf{x}}(t)) = & m_2 l_1 l_{c2} S_2 \alpha_1 (3 \omega_1 - \omega_2) + N_2^2 B_{m_2} \alpha_2 + \\
& + \frac{R_2}{L_2} [\{m_2 (l_{c2}^2 + l_1 l_{c2} C_2) + I_2\} \alpha_1 + \\
& + (m_2 l_{c2}^2 + I_2 + N_2^2 J_{m_2}) \alpha_2] + \frac{R_2}{L_2} N_2^2 B_{m_2} \omega_2 + \\
& + m_2 l_1 l_{c2} C_2 \omega_1^2 \omega_2 + \frac{R_2}{L_2} m_2 l_1 l_{c2} S_2 \omega_1^2 + \\
& + \frac{N_2^2 K_{t_2} K_{em_2}}{L_2} \omega_2 - m_2 l_{c2} g S_{12} (\omega_1 + \omega_2) + \\
& + \frac{R_2}{L_2} m_2 l_{c2} g C_{12}
\end{aligned}$$

onde $S_1 = \text{sen}(\theta_1)$, $C_1 = \text{cos}(\theta_1)$, $S_2 = \text{sen}(\theta_2)$, $C_2 = \text{cos}(\theta_2)$, $S_{12} = \text{sen}(\theta_1 + \theta_2)$ e $C_{12} = \text{cos}(\theta_1 + \theta_2)$. O sistema possui um ponto de equilíbrio estável em

$$p(\theta_1, \theta_2, \omega_1, \omega_2, \alpha_1, \alpha_2, u_1, u_2) = \left(\frac{3\pi}{2}, 0, 0, 0, 0, 0, 0, 0\right).$$

4.5 Sumário

Neste capítulo, foi feita uma descrição matemática de um sistema robótico. Na seção 4.1, foram introduzidos alguns termos freqüentes em sistemas robóticos. Na seção 4.2, foi apresentada a análise cinemática pela abordagem de Denavit e Hartenberg (1955) e comentado resumidamente sobre planejamento de trajetórias.

A seção 4.3 apresenta a análise dinâmica com a inclusão da dinâmica de atuadores elétricos [i.e., motores elétricos], com o modelo sendo descrito em forma de estados. Os desenvolvimentos matemáticos mostraram que o modelo dinâmico do sistema é não linear, de elevada complexidade e com interconectividade entre as variáveis de estado. As equações (4.18), (4.19) e (4.20) e o modelo de manipulador robótico articulado apresentado na seção 4.4 mostram que para os manipuladores os momentos de inércias variam em função das posições angulares e, conseqüentemente, são variantes no tempo. O modelo de manipulador articulado de dois graus de liberdade apresentado na seção 4.4 é usado em simulações apresentadas no capítulo 5.

Adicionalmente, uma breve apresentação dos principais sensores usados no espaço de juntas de manipuladores robóticos foi feita na seção 4.3.3. Um assunto importante discutido nesta seção foi sobre o uso ou não de um sensor para cada estado do sistema. Foi mostrado que diversos fatores têm contribuído para o uso de apenas sensores de posição para o espaço de juntas e estimação dos outros estados. No entanto, muitas vezes esta opção não é salutar para o uso de técnicas avançadas de controle, devido às características do sistema não permitirem uma estimação acurada dos outros estados.

Capítulo 5

Identificação Não Linear de Sistemas Dinâmicos

*Este capítulo apresenta a identificação de sistemas não lineares sob a óptica da propriedade da observabilidade genérica. São comentados alguns desenvolvimentos na área de identificação não linear e discutida a propriedade da observabilidade genérica. Um identificador para propósitos on-line é apresentado, usando como ferramenta de modelagem não linear o modelo **MLP** de **RNA** apresentado no capítulo 3. O identificador é aplicado por simulação ao modelo de manipulador robótico de dois **GL** apresentado no capítulo 4.*

5.1 Introdução

RECENTEMENTE, o desenvolvimento de paradigmas para a identificação de sistemas lineares tem recebido considerável atenção por parte de pesquisadores. Isto tem contribuído para solução de muitos problemas técnicos em engenharia de controle. O recente desenvolvimento de computadores de alta capacidade de processamento, bem como a facilidade de acesso, tem motivado vários pesquisadores a direcionar as suas atenções a paradigmas de identificação de sistemas não lineares (Leontaritis e

Billings, 1985*a*; Leontaritis e Billings, 1985*b*; Sjöberg *et al.*, 1995; Juditsky *et al.*, 1995). Neste contexto, **RNAs** tornaram-se uma ferramenta para modelagem matemática não linear (Narendra e Parthasarathy, 1990; Hunt *et al.*, 1992; Levin e Narendra, 1993; Levin e Narendra, 1995; Suykens *et al.*, 1995; Levin e Narendra, 1996; Jagannathan e Lewis, 1996; Chen *et al.*, 1997).

Alguns fundamentos sobre observabilidade e controlabilidade para sistemas não lineares foram lançados por Hermann e Krener (1977), que apresentaram as propriedades da observabilidade e da controlabilidade de sistemas não lineares com analogias a sistemas lineares, introduzindo conceitos tais como sistemas localmente controláveis e/ou observáveis, sistemas localmente fracamente observáveis e/ou controláveis, bem como a minimalidade de sistemas não lineares.

De um ponto de vista genérico, a observabilidade de sistemas não lineares discretos é discutida por Aeyels (1981), que expressou a propriedade da observabilidade em termos de condições de transversalidade topológica, mostrando que a intersecção de mapeamentos transversos produz sistemas genericamente observáveis. Para isso, seriam necessários um número de mínimo de $2n + 1$ amostras, onde n é a ordem do sistema.

As condições suficientes relativas à construção de modelos entrada-saída locais para sistemas dinâmicos não lineares, foram estabelecidas por Leontaritis e Billings (1985*a*; 1985*b*), onde foram introduzidos e definidos conceitos tais como a realização de Nerode, formas multiestruturais e alguns resultados originários da geometria diferencial que são usados para a obtenção de modelos de entrada-saída recursivos para sistemas não lineares multivariáveis determinísticos e estocásticos.

Sjöberg *et al.* (1995) apresentaram os aspectos comuns entre as diferentes abordagens de identificação de sistemas não lineares e os pontos relevantes a serem considerados para a escolha e o sucesso de alguma abordagem específica. Ainda, mostrou que as estruturas não lineares podem ser vistas como expansões de funções não lineares de base.

Juditsky *et al.* (1995) mostraram que a boa qualidade da identificação depende do número de parâmetros a ser estimado no modelo matemático. Quanto maior o número de parâmetros, mais flexível a identificação. No entanto, quanto maior o número de parâmetros maior o erro estocástico. Assim, fica estabelecida uma relação entre o número de parâmetros a ser estimado e a qualidade da identificação.

Um resumo dos principais métodos de identificação, tanto lineares quanto não lineares, com modelos contínuos foi apresentado por Unberhauen e Roa (1998).

Muitos trabalhos publicados tratam de identificadores baseado em modelos de **RNAs** e na capacidade de aproximação destas redes. Narendra e Parthasarathy (1990) apresentaram estruturas de identificadores não lineares baseados na literatura de sistemas adaptativo para identificação e controle de sistemas lineares. Suykens *et al.* (1995) apresentaram uma abordagem no espaço de estados e desenvolveram um algoritmo por previsão de erro para casos determinísticos e estocásticos.

Levin e Narendra (1995) apresentaram um identificador baseado na propriedade da observabilidade genérica discutida por Aeyels (1981) e um outro identificador baseado na abordagem de equivalência linear para aplicação em sistemas fortemente observáveis (Levin e Narendra, 1996).

Ng (1997) apresentou um estudo sobre o aprendizado supervisionado em redes **MLPs** em aplicações de controle *on-line* de sistemas não lineares de modelos desconhecidos. Entre as estratégias de controle estudadas estavam algumas que faziam o uso de modelos de identificados para a sistema físico, sendo analisados e estudada a convergência de dois algoritmos de ajuste de parâmetros para as **RNAs**: o algoritmo *backpropagation* com o termo *momentum*; e o algoritmo do mínimo quadrado recursivo. As análises de convergência foram feitas levando em consideração a ação conjunta do identificador e do controlador. Na análise do algoritmo do *backpropagation* com o termo *momentum* é suposto que o termo *momentum* é escolhido o suficientemente pequeno de modo a não afetar a estabilidade do ajuste de parâmetros, e assim não levado em consideração nas manipulações matemáticas (Ng, 1997, pp. 105) (veja também

Ballini e Von Zuben (2000)).

Chen *et al.* (1997) apresentaram um método usando **RNA** para a identificação de sistemas com comportamento caótico associando à **RNA** um filtro dinâmico de Wiener.

Neste capítulo é proposto um identificador não linear multivariável para sistemas genericamente observáveis que usa como base o identificador não recorrente apresentado Levin e Narendra (1995), além das demais referências citadas. O identificador proposto aqui pode aproximar em regime *on-line* modelos entrada-saída multivariáveis de sistemas genericamente observáveis. Em adição, o conteúdo apresentado neste capítulo pode ser consultado em Cerqueira e Badan Palhares (1999), Cerqueira *et al.* (2000*b*) e Cerqueira *et al.* (2000*c*).

5.2 Comparação Entre a Região de Densidade do Sistema e a Região de Densidade do Identificador

A questão da relação entre as regiões de densidade (veja a definição 12 na página 18) do sistema e do identificador é muito importante em um processo de identificação não linear (Cerqueira *et al.*, 2000*c*). Adicionalmente, esta questão está diretamente vinculada à escolha apropriada de um método de ajuste de parâmetros.

De um ponto de vista genérico, a região de densidade do identificador não linear fica delimitada a partir do instante em que ficam estabelecidos o sistema que ele deve aproximar e a tolerância ou precisão desejada para a aproximação. Como ilustrado pela figura 5.1, há três situações diferentes concernentes à capacidade de aproximação de um identificador:

- (i) Sua região de densidade é maior que a do sistema (figura 5.1(a)).

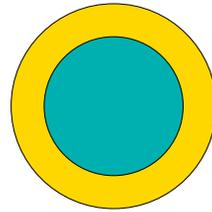
- (ii) Sua região de densidade é igual a do sistema (figura 5.1(b)).
- (iii) Sua região de densidade é menor que a do sistema (figura 5.1(c)).

No primeiro e no segundo caso, ilustrados pelas figuras 5.1(a) e 5.1(b), poderia ser obtida uma solução global [i.e., a região de densidade do identificador abrangeria toda a região de densidade do sistema e o ajuste de parâmetros seria realizado apenas uma vez] se a demanda computacional imposta pelo identificador e por outras tarefas exigidas pelo metasistema puderem ser executadas no período de amostragem do sistema dinâmico, (durante o procedimento de ajuste de parâmetros) e também se o sistema viajasse sobre quase todas as trajetórias dentro de sua região de densidade [i.e., sobre trajetórias que representassem significativamente toda a não linearidade da região de densidade, de modo a permitir que a **RNA** faça uma aproximação generalizada de toda a região] durante tal procedimento de ajuste de parâmetros. Neste caso, para implementação do identificador seria necessário um sistema computacional de alta capacidade de processamento, fazendo a relação custo/benefício ser altamente prejudicial.

A principal intenção da identificação é obter um modelo para um sistema supostamente não bem conhecido. Assim, é muito difícil especificar um identificador não linear capaz de modelar o sistema para o primeiro e segundo casos citados anteriormente e ilustrados pelas figuras 5.1(a) e 5.1(b), porque o tamanho e a forma da região de densidade do sistema não são conhecidos com exatidão. Algumas vezes, apenas pontos de equilíbrio e limites inferior e superior sobre algumas variáveis ou estados do sistema são conhecidos.

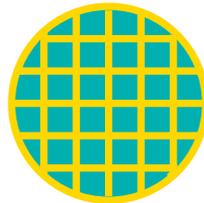
Para o terceiro caso, ilustrado pela figura 5.1(c), não seria alcançada uma solução global porque o identificador não seria capaz de aproximar o processo dentro de sua inteira região de densidade. Assim, sempre que o processo seguisse uma trajetória para uma região ainda não aproximada pelo ajuste de parâmetros, seria necessário fazer novos ajustes desses parâmetros. Para estes casos, podem ser aceitas soluções locais.

Se o problema é abordado em termos do terceiro caso acima, devido à falta de



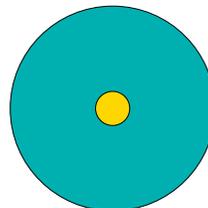
● ●
Sistema Identificador

(a) Caso (i).



● ●
Sistema Identificador

(b) Caso (ii).



● ●
Sistema Identificador

(c) Caso (iii).

Figura 5.1: Regiões de densidade do sistema e do identificador.

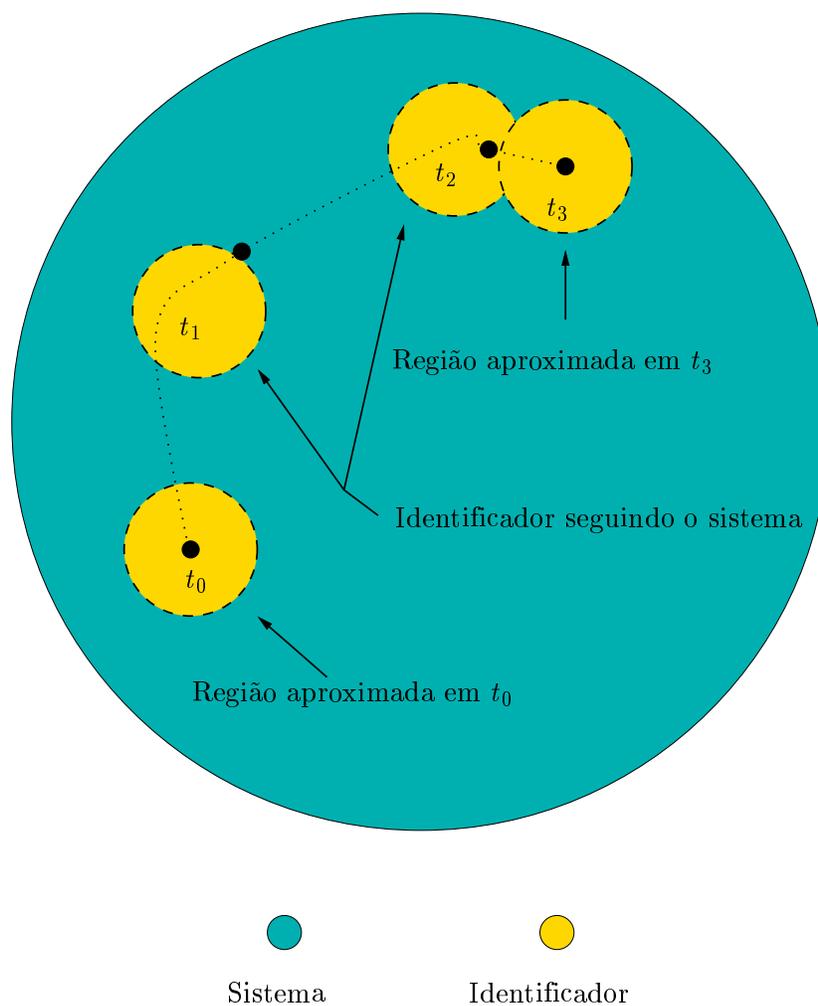


Figura 5.2: Identificador rastreando o sistema.

conhecimento a respeito do sistema a ser modelado, um identificador com região de densidade menor que a do sistema poderia ser especificado, e o modelo reajustado por algum método de ajuste de parâmetro adaptativo sempre que o sistema seguisse uma trajetória para uma região ainda não aproximada. A figura 5.2 ilustra o rastreamento do sistema pelo identificador. Esta solução, junto com o uso de um método baseado no vetor gradiente, reduziria as exigências para o sistema computacional e o identificador seria eficiente se o método de ajuste de parâmetros fosse sempre convergente. Métodos baseados em matriz Hessiana são intrinsecamente convergentes. Métodos baseados no vetor gradiente possuem a convergência dependente da escolha adequada do passo de busca da solução.

Em adição, se as informações a respeito de limitantes superiores para as variáveis medidas na entrada e na saída do sistema e a respeito de pontos de equilíbrios são usados, um identificador com região de densidade pequena pode tornar-se compatível dentro da região de densidade do sistema (veja o capítulo 3). Assim, soluções locais podem se tornar possíveis e satisfatórias do ponto de vista computacional.

5.3 O Modelo do Identificador Não Linear

Seja Σ a representação de um sistema dinâmico no tempo discreto como segue (veja a seção 2.4):

$$\Sigma : \begin{cases} \mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k) &= h(\mathbf{x}(k)) \end{cases} \quad (5.1)$$

onde $\mathbf{x}(k) \in \mathbb{R}^n$ é o vetor de estado, n é a ordem, $\mathbf{u}(k) \in \mathbb{R}^m$ é o vetor de entrada, $\mathbf{y}(k) \in \mathbb{R}^s$ é o vetor de saída, e $f : \mathbb{Z}^+ \times \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$ e $h : \mathbb{Z}^+ \times \mathbb{R}^n \mapsto \mathbb{R}^s$ são dois vetores de funções não lineares, com $f, h \in C^q$ para algum $q \in \mathbb{N}$.

Como o sistema real não é bem conhecido, algumas suposições a respeito de sua estrutura são necessárias para fazer o problema de identificação tratável.

Suposição 4. O mapeamento entrada-saída do sistema Σ é diferenciável e inversível [i.e., para um dado $\mathbf{u}(k)$, $f(\cdot)$ define um difeomorfismo sobre $\mathbf{y}(k)$, garantindo que ambos, o passado e o futuro dos sinais de saída sejam únicos].

Suposição 5. O sistema Σ é um mapeamento de Morse [i.e., possui apenas pontos de equilíbrio isolados (veja a definição 18)], existindo algum ponto $\Sigma(x_0, u_0)$ de equilíbrio estável no senso de Lyapunov e tomado como origem do sistema.

Suposição 6. O sistema Σ é genericamente observável.

Suposição 7. Para o procedimento de identificação $f(\cdot)$ and $h(\cdot)$ são desconhecidos. Apenas informações de sinais de entrada e saída, $\mathbf{u}(k)$ e $\mathbf{y}(k)$, e sobre um limitante superior da ordem do sistema são disponíveis.

Note que de acordo com a suposição 4, sistemas com histerese em $h(\cdot)$ estão fora do escopo [i.e., com histerese na saída], mas não existem restrições com respeito a sistemas com histerese apenas em $f(\cdot)$ [i.e., com histerese em alguns componentes do vetor de estado, desde que a histerese não seja observada nas variáveis de saída]. As suposições 4 e 5 garantem a propriedade da observabilidade genérica ao sistema Σ (Aeyels, 1981; Levin e Narendra, 1995).

Agora, considere o modelo de identificador da figura 5.3, descrito na forma

$$\hat{\mathbf{y}}(k) = \phi(k, \hat{\mathbf{w}}, (\mathbf{Y}_l(k), \mathbf{U}_l(k))) \quad (5.2)$$

onde $\hat{\mathbf{y}}(k) \in \mathbb{R}^s$ é o vetor de saída do identificador, $\hat{\mathbf{w}}(k) \in \mathcal{W} \subset \mathbb{R}^{n_p}$ é um vetor de parâmetros internos ajustáveis, $\phi : \mathbb{Z}^+ \times \mathbb{R}^{r_1} \times \mathbb{R}^{r_2} \mapsto \mathbb{R}^s$ é um vetor de funções não lineares capaz de aproximar o modelo do sistema Σ por meio do ajuste de seus parâmetros internos, com $r_1 = (l \cdot s)$ e $r_2 = (l \cdot m)$. $\mathbf{Y}_l(k)$ and $\mathbf{U}_l(k)$ são ocorrências no tempo de comprimento l mantendo os valores assumidos pelas variáveis de entrada e saída do sistema, na forma

$$\mathbf{Y}_l(k) \triangleq \{\mathbf{y}(k-1), \mathbf{y}(k-2), \dots, \mathbf{y}(k-l)\}$$

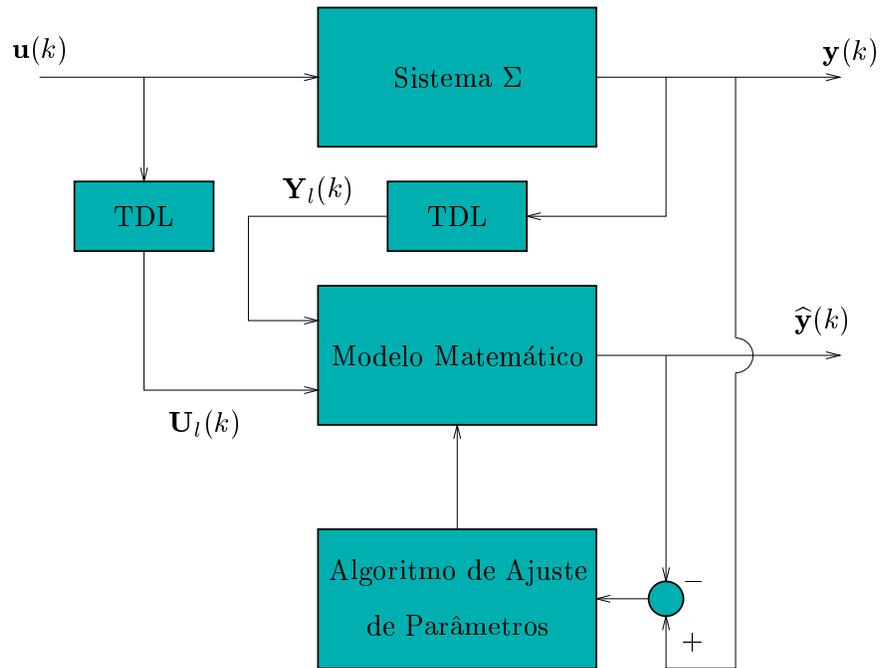


Figura 5.3: Modelo do Identificador.

$$\mathbf{U}_l(k) \triangleq \{\mathbf{u}(k-1), \mathbf{u}(k-2), \dots, \mathbf{u}(k-l)\}.$$

Baseados nos resultados sobre a propriedade da observabilidade genérica para sistemas não lineares obtidos por Aeyels (1981), Levin e Narendra (1995) mostraram que um identificador descrito pelo modelo (5.2) pode ser eficientemente especificado com base em tal propriedade, e aplicado a qualquer sistema genericamente observável, se no mínimo $l = 2n + 1$.

A partir da equação (5.1), o vetor de funções $h(\mathbf{x}(k))$ pode ser expresso na forma

$$h(\mathbf{x}(k)) = \begin{bmatrix} h_1(\mathbf{x}(k)) \\ h_2(\mathbf{x}(k)) \\ \vdots \\ h_s(\mathbf{x}(k)) \end{bmatrix}$$

onde os $h_i : \mathbb{Z}^+ \times \mathbb{R}^n \mapsto \mathbb{R}$ são as funções que compõem o vetor de funções $h : \mathbb{R}^n \mapsto \mathbb{R}^s$.

Com isto, o modelo do identificador $\phi : \mathbb{Z}^+ \times \mathbb{R}^{r_1} \times \mathbb{R}^{r_2} \mapsto \mathbb{R}^s$ pode ser expresso na forma

$$\phi(k, \hat{\mathbf{w}}, (\mathbf{Y}_l(k), \mathbf{U}_l(k))) = \begin{bmatrix} \phi_1(k, \hat{\mathbf{w}}, (\mathbf{Y}_l(k), \mathbf{U}_l(k))) \\ \phi_2(k, \hat{\mathbf{w}}, (\mathbf{Y}_l(k), \mathbf{U}_l(k))) \\ \vdots \\ \phi_s(k, \hat{\mathbf{w}}, (\mathbf{Y}_l(k), \mathbf{U}_l(k))) \end{bmatrix}$$

onde os $\phi_i : \mathbb{Z}^+ \times \mathbb{R}^{r_1} \times \mathbb{R}^{r_2} \mapsto \mathbb{R}$ são as aproximações dos h_i , obtidas a partir de algum aproximador universal.

Conseqüentemente, cada h_i pode ser aproximado por meio de uma rede neural **MLP**, via processos de ajuste de parâmetros independentes, usando o algoritmo apresentado na seção 3.4. Cada rede neural **MLP** pode ser ajustada usando uma função objetivo da forma

$$J(k, \hat{\mathbf{w}}(k), \mathbf{u}(k)) = \frac{1}{2} (h_i(k) - \phi_i(k))' (h_i(k) - \phi_i(k)).$$

5.4 Aplicação do Paradigma e Resultados de Simulações em Computador

Nesta seção, o paradigma proposto é aplicado na identificação do modelo não linear multivariável de um manipulador robótico com dois graus de liberdade (dois **GL**) acionado por motores elétricos de corrente contínua. Esta identificação é feita por simulações em computador, e usando o modelo de manipulador apresentado na seção 4.4. A tabela 5.1 mostra os parâmetros usados no modelo do manipulador. Estes parâmetros são os mesmos usados por Yu e Lloyd (1995).

Por construção de modelo, é considerado que os componentes do vetor de estado do sistema que podem ser diretamente medidos são apenas os correspondentes aos deslocamentos angulares de cada junta [i.e., posição angular de cada junta]. Este sistema é genericamente observável, pois é um difeomorfismo e apresenta pontos de equilíbrio isolados no sentido de Lyapunov [i.e., é diferenciável e inversível e também é um mapeamento de Morse, veja a definição 18, e as seções 4.4 e 5.3].

As características das **MLPs** usadas são apresentadas na tabela 5.2, e as simulações foram executadas considerando critérios de demanda *on-line*, onde:

- (i) O programa de simulação foi escrito em linguagem C++ e usando programação orientada a objetos. Esta escolha foi feita principalmente para permitir que as **MLPs** pudessem ser criadas com valores pré-estabelecidos para $w_{2_{\max}}^0 \in \mathbb{R}_+^*$, $u_{m_{\max}}^0 \in \mathbb{R}_+^*$, $\kappa_{\phi_1} \in \mathbb{R}_+^*$, and $\kappa \in (1; \infty) \subset \mathbb{R}$.
- (ii) O parâmetro $\alpha \in [0; 1) \subset \mathbb{R}$ foi pré-estabelecido empiricamente usando as idéias apresentadas na seção 3.2, e o passo ou taxa de aprendizagem, η , foi automática e adaptativamente determinado por meio do teorema 8 e do uso do algoritmo 1.
- (iii) Os vetores de parâmetros iniciais de cada **MLP**, $\hat{\mathbf{w}}^0$, foram iniciados aleatória e uniformemente com valores distribuídos dentro do intervalo $(-0.5; 0.5) \subset \mathbb{R}$, conforme o método *easiest-path* discutido na seção 3.4.1.

Tabela 5.1: Parâmetros do Manipulador de Yu e Lloyd (1995).

<i>Parâmetros</i>	<i>Valores</i>
R_1, R_2	1.6 Ω
N_1, N_2	62.55
L_1, L_2	0.0048 H
K_{t_1}, K_{t_2}	0.35 N m/A
K_{em_1}, K_{em_2}	0.35 Volt s/rad
J_{m_1}, J_{m_2}	0.0377 kg m ²
B_{m_1}, B_{m_2}	0.00175 N m s/rad
k_m	$\frac{1}{\pi}$ Volt/rad
k_v	15
m_1	15.91 kg
m_2	11.36 kg
l_1	0.432 m
l_2	0.432 m
l_{c1}	0.216 m
l_{c2}	0.216 m
I_1	0.247 kg m ²
I_2	0.177 kg m ²

Tabela 5.2: Parâmetros para as **MLPs**.

<i>Parâmetro</i>	<i>Valor</i>	<i>Characteristic</i>
n_e	52	Fixo e estabelecido pelo usuário
n_s	1	Fixo e estabelecido pelo usuário
u_0	1	Fixo e pré-definido internamente
$\phi_{1,0}$	1	Fixo e pré-definido internamente
$w_{2\max}^0$	0.5	Adaptativo e pré-definido internamente
$u_{m\max}^0$	0.1	Adaptivo e pré-definido internamente
η		Adaptativo e determinado internamente
J_0	0.0001 rad^2	Fixo e estabelecido pelo usuário
κ_{ϕ_1}	1	Fixo e pré-definido internamente
κ	1.01	Fixo e pré-definido internamente

A escolha do número de neurônios da primeira camada de neurônios processadores das **MLPs** foi feita por experimentação. Algumas experiências foram feitas usando identificadores com grupos de três, dez, e quarenta neurônios (veja as figuras 5.4-5.9), e com a observação do comportamento das funções objetivos e as saídas do sistema e do identificador.

Do ponto de vista de rapidez [i.e., ordem de convergência], o identificador com grupos de três neurônios foi superior aos outros. Em princípio, poder-se-ia pensar que usando-se um identificador que tenha um grande número de neurônios na primeira camada processadora da rede **MLP**, sua eficiência na identificação do modelo do sistema sempre seria melhor. Porém, isto requereria uma demanda computacional mais elevada, que poderia tornar inviável o ajuste de parâmetros *on-line*, além do fato de que, para estes casos, o passo de convergência do método do gradiente seria menor do que em uma rede com poucos neurônios (veja o teorema 8).

Para estas simulações, os parâmetros α foram fixados em 0.50 para cada rede

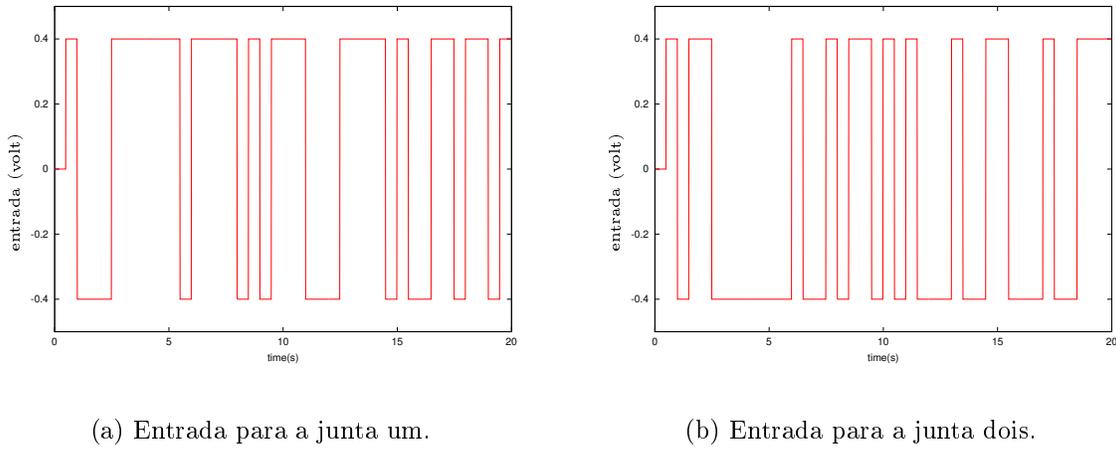


Figura 5.4: Sinais de Entrada.

MLP. Neste caso em particular, como as articulações têm características semelhantes, e em geral terão, os mesmos valores foram adotados para os α relativos a cada rede neural dos identificadores. Porém, dependendo do desempenho do identificador haveria a necessidade de valores mais adequados para cada um deles.

Nas figuras 5.4(a) e 5.4(b) são mostrados os respectivos sinais de entrada aplicados às juntas robóticas, com duração de 20 segundos e com 10 milésimos de segundo de período de amostragem. Estes sinais são binários e não correlatos um ao outro, sendo que foram gerados aleatoriamente, como é sugerido por Ljung (1987) com o objetivo de aumentar o número de componentes de frequências que compõem os sinais de entrada.

As saídas do sistema e dos identificadores são traçadas sobrepostas, para cada junta do manipulador na figura 5.5. Foram usados diferentes quantidades de neurônios na primeira camada processadora da rede **MLP**, com a seguinte seqüência de apresentação: para três neurônios as subfiguras 5.5(a) e 5.5(b); para dez neurônios as subfiguras 5.5(c) e 5.5(d); e para quarenta neurônios as subfiguras 5.5(e) e 5.5(f).

A figura 5.6 mostra as evoluções das funções objetivos para cada saída de identificador. A seqüência de figuras para as diferentes quantidades de neurônios na primeira

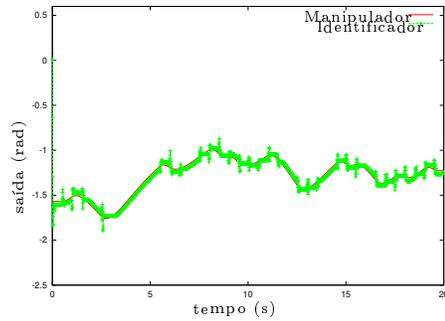
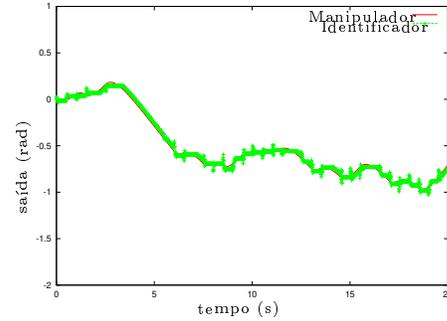
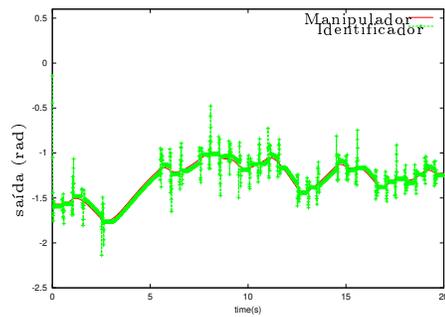
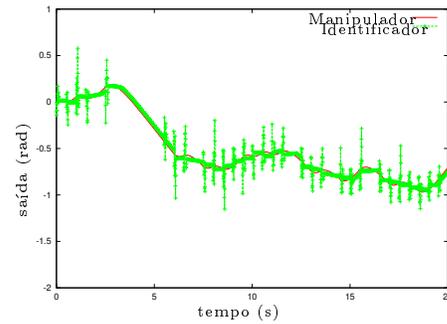
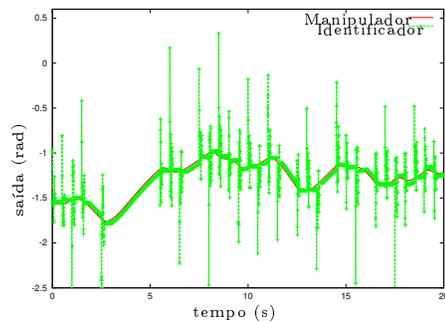
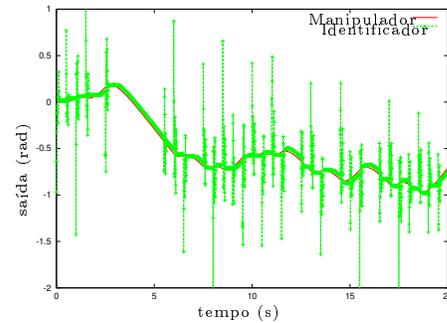
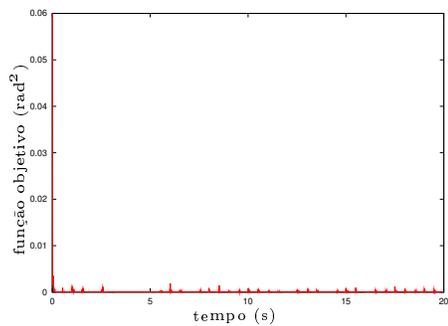
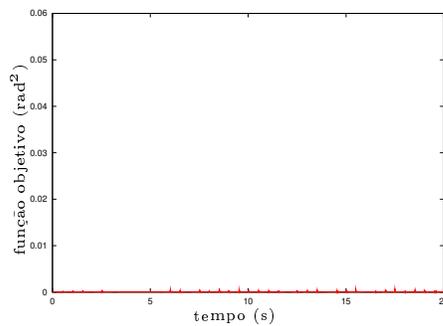
(a) Saídas para junta um ($N = 3$).(b) Saídas para junta dois ($N = 3$).(c) Saídas para junta um ($N = 10$).(d) Saídas para junta dois ($N = 10$).(e) Saídas para junta um ($N = 40$).(f) Saídas para junta dois ($N = 40$).

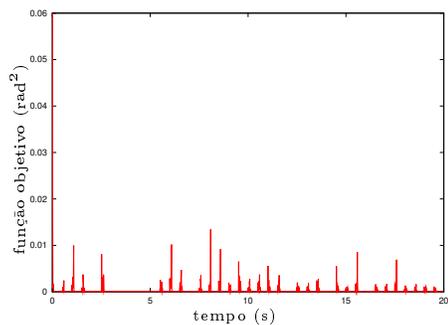
Figura 5.5: Saídas do sistema e dos identificadores.



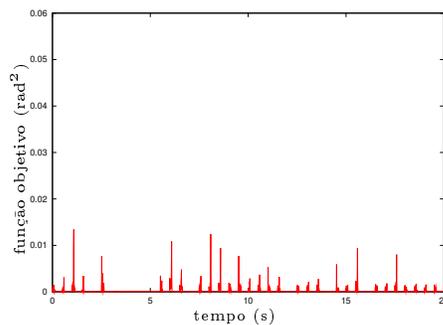
(a) MLP para junta um ($N = 3$).



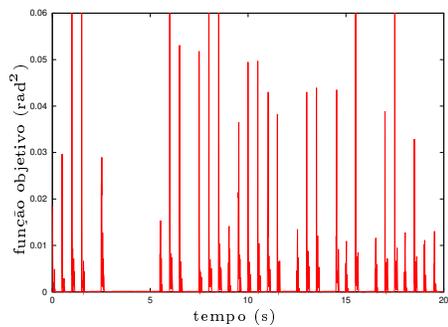
(b) MLP para junta dois ($N = 3$).



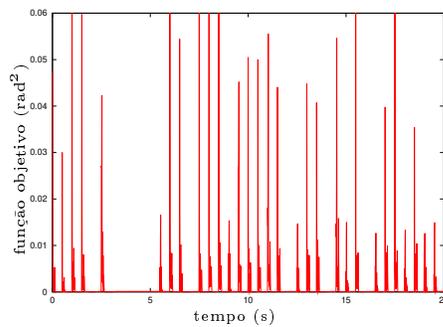
(c) MLP para junta um ($N = 10$).



(d) MLP para junta dois ($N = 10$).

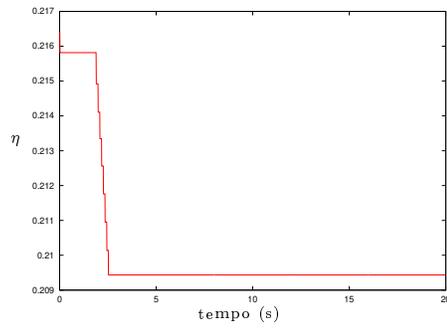
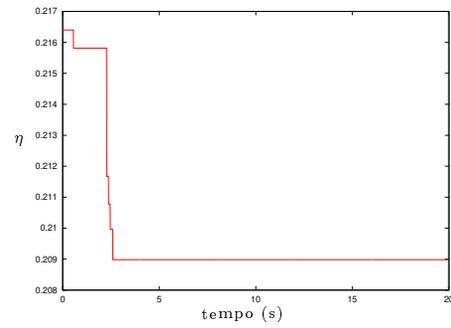
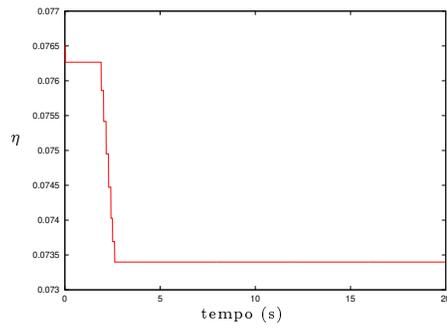
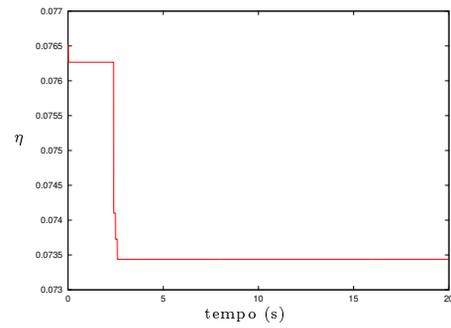
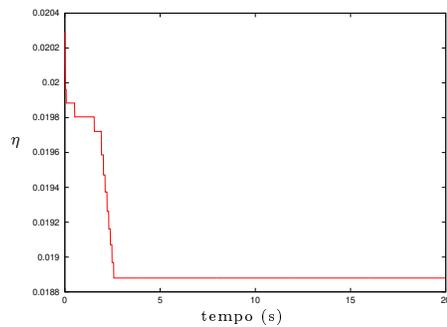
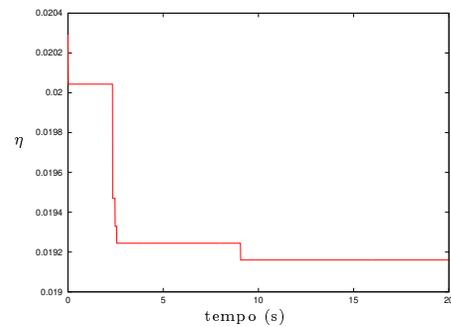


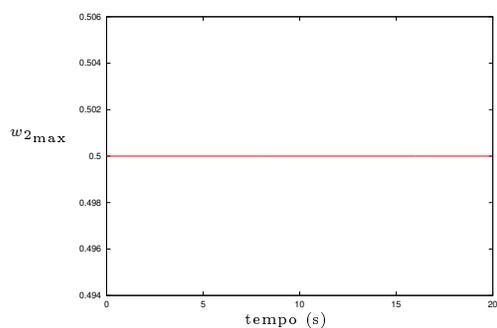
(e) MLP para junta um ($N = 40$).



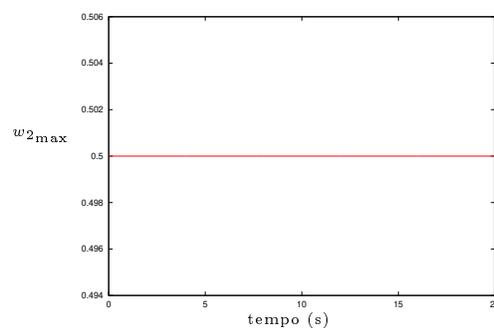
(f) MLP para junta dois ($N = 40$).

Figura 5.6: Evoluções das funções objetivo.

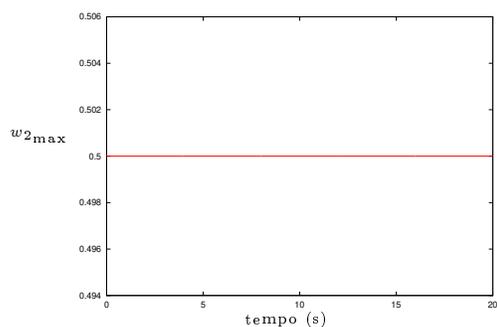
(a) MLP para junta um ($N = 3$).(b) MLP para junta dois ($N = 3$).(c) MLP para junta um ($N = 10$).(d) MLP para junta dois ($N = 10$).(e) MLP para junta um ($N = 40$).(f) MLP para junta dois ($N = 40$).Figura 5.7: Evoluções dos η .



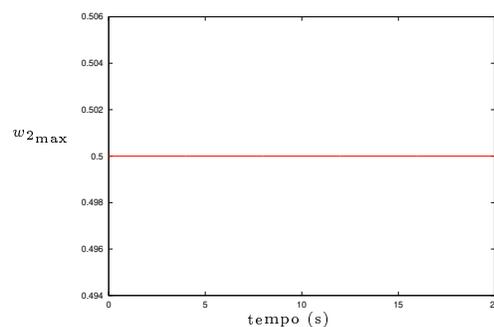
(a) MLP para junta um ($N = 3$).



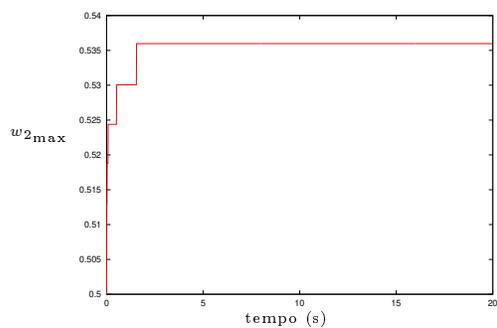
(b) MLP para junta dois ($N = 3$).



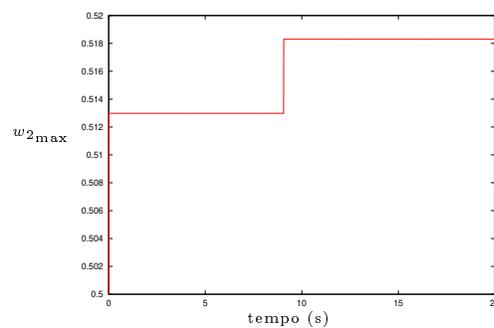
(c) MLP para junta um ($N = 10$).



(d) MLP para junta dois ($N = 10$).



(e) MLP para junta um ($N = 40$).



(f) MLP para junta dois ($N = 40$).

Figura 5.8: Evoluções dos $w_{2_{\max}}$.

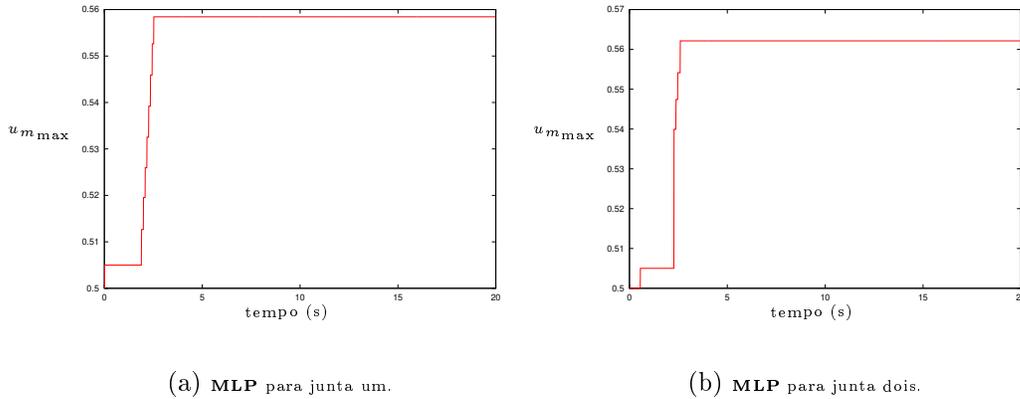


Figura 5.9: Evoluções dos $u_{m\max}$.

camada processadora é: para três neurônios as subfiguras 5.6(a) e 5.6(b); para dez neurônios as subfiguras 5.6(c) e 5.6(d); e para quarenta neurônios as subfiguras 5.6(e) e 5.6(f).

O uso de diferente números de neurônio para a primeira camada processadora das MLPs de cada identificador corresponderam ao uso de diferentes regiões de densidade para cada identificador. Assim, cada identificador possuiu a capacidade de aproximar regiões não lineares de diferentes tamanhos. Uma região de densidade grande para um identificador corresponde a um número grande de parâmetros a serem ajustados e também, por meio do teorema 8, corresponde a uma redução do intervalo de aceitação do passo (veja na figura 5.7 como os passos diminuem em função dos aumentos dos número de neurônio na primeira camada). Isto reduz velocidade de ajuste de parâmetros do identificador, como mostrado nas figuras 5.5 e 5.6. O identificador com três neurônios na primeira camada processadora teve um desempenho melhor que os outros. Suas funções objetivo tiveram flutuações menores que os outros.

As evoluções dos passos ou taxas de aprendizagem são mostradas na figura 5.7. A seqüência de figuras para as diferentes quantidades de neurônios na primeira camada processadora é: para três neurônios as subfiguras 5.7(a) e 5.7(b)); para dez neurônios

as subfiguras 5.7(c) e 5.7(d); e para quarenta neurônios as subfiguras 5.7(e) e 5.7(f).

As evoluções dos $w_{2_{\max}}$ são mostradas na 5.8. A seqüência de figuras para as diferentes quantidades de neurônios na primeira camada é: para três neurônios as subfiguras 5.8(a) e 5.8(b); para dez neurônios as subfiguras 5.8(c) e 5.8(d); e para quarenta neurônios as subfiguras 5.8(e) e 5.8(f).

As evoluções dos $u_{m_{\max}}$ são mostradas na figura 5.9. Os sinais de entrada e os parâmetros $u_{m_{\max}}^0$ foram os mesmos para todas as simulações. Consequentemente, esta figura mostra apenas duas subfiguras [figuras 5.9(a) e 5.9(b)] com as evoluções de $u_{m_{\max}}$.

A evolução dos passos em função dos $w_{2_{\max}}^0$ e $u_{m_{\max}}^0$ pode ser compreendido por meio das figuras 5.7-5.9. Os identificadores com respectivamente três e dez neurônios na primeira camada processadora das **MLP**s tiveram os seus passos mudados apenas por causa das variações dos $u_{m_{\max}}^0$. O identificador com quarenta neurônios na primeira camada das **MLP**s teve os passos mudados por causa de variações de ambos, $w_{2_{\max}}^0$ e $u_{m_{\max}}^0$. Estas figuras mostram como a rotina de gerenciamento do algoritmo adaptativo 1 decide sobre a aplicação de teorema 8.

Tomando-se como base a figura 5.6, a figura 5.10 apresenta uma expansão das funções objetivos dentro de uma janela de tempo entre 7 e 9 segundos de simulação. A seqüência de figuras para as diferentes quantidades de neurônios na primeira camada processadora é: para três neurônios as subfiguras 5.10(a) e 5.10(b); para dez neurônios as subfiguras 5.10(c) e 5.10(d); e para quarenta neurônios as subfiguras 5.10(e) e 5.10(f). Esta figura mostra com mais detalhes a convergência *on-line* do ajuste de parâmetros e o acompanhamento que o identificador faz da dinâmica do sistema.

Em princípio, alguém observando a figura 5.6 pode concluir que as funções objetivo comportam-se como seqüências de impulsos, mas com a observação da figura 5.10 nota-se que as funções objetivo não comportam-se como seqüências de impulsos. Todas as vezes que o sistema segue para uma região ainda não aproximada o identificador tenta acompanhá-lo, sendo que para isso o processo de ajuste de parâmetros tem que

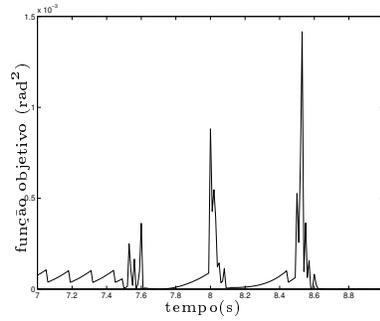
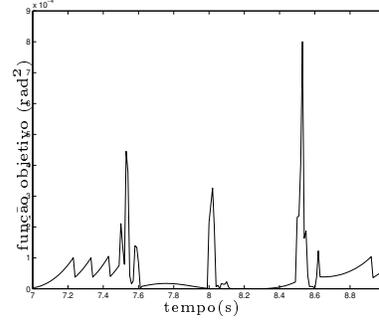
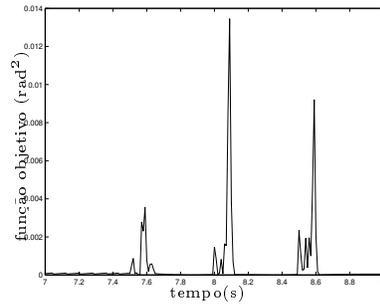
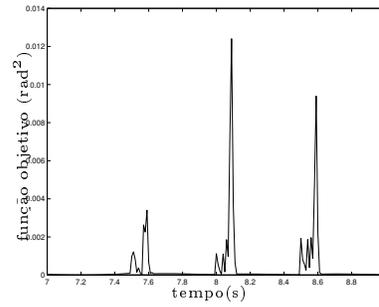
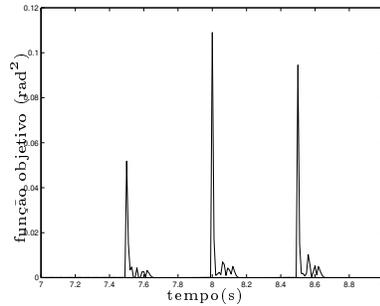
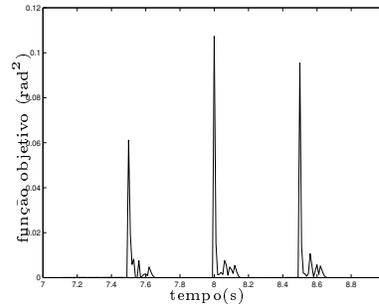
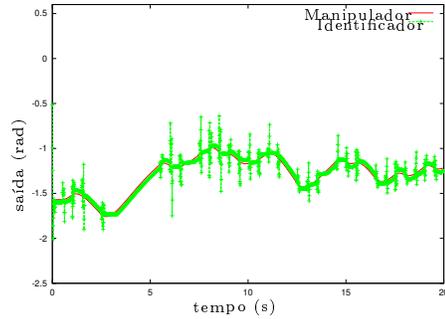
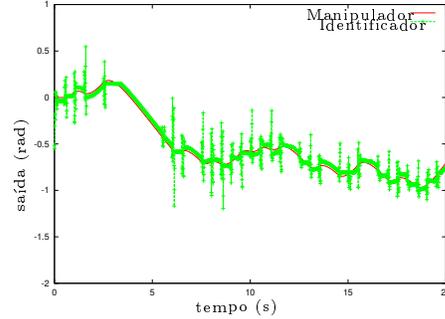
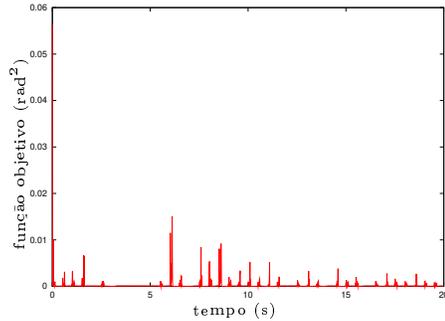
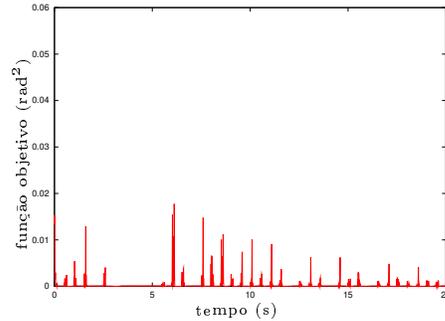
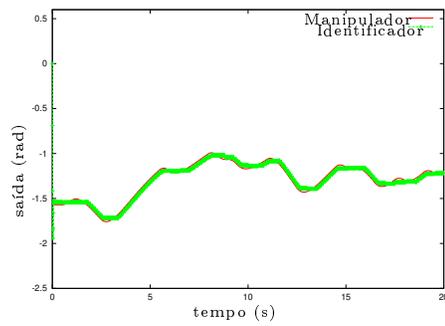
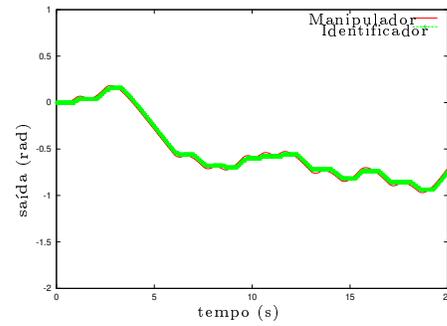
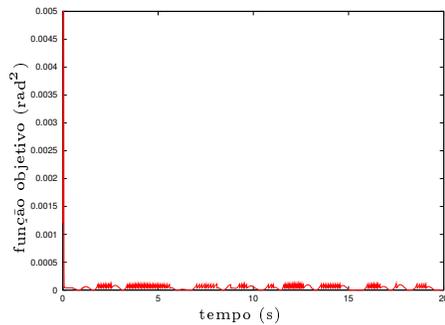
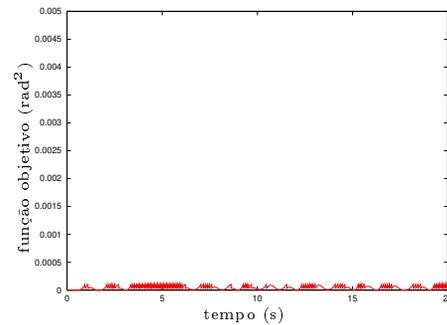
(a) MLP junta um ($N = 3$).(b) MLP junta dois ($N = 3$).(c) MLP junta um ($N = 10$).(d) MLP junta dois ($N = 10$).(e) MLP junta um ($N = 40$).(f) MLP junta dois ($N = 40$).

Figura 5.10: Evoluções das funções objetivo na janela de tempo entre 7 e 9 segundos.

(a) Saídas para junta um ($N = 3$).(b) Saídas para junta dois ($N = 3$).(c) **MLP** para junta um ($N = 3$).(d) **MLP** para junta dois ($N = 3$).Figura 5.11: Simulação usando iniciação de parâmetros no intervalo $(-1; 1)$.

ser reiniciado. As diferenças entre as saídas do sistema e as saídas do identificador são observadas nas funções objetivos nas formas apresentadas nas figuras. O identificador com três neurônios na primeira camada processadora foi o mais eficiente de todos do ponto de vista de convergência, e por isso apresenta funções objetivos com menores desvios absolutos.

As figuras 5.11 e 5.12 mostram simulações feitas com os parâmetros ajustáveis de cada **MLP** iniciados aleatoriamente e uniformemente com valores distribuídos respectivamente dentro dos intervalos $(-1; 1)$ e $(-0.01; 0.01)$, usando $N = 3$ e respectivamente $w_{2\max}^0 = 1$ e $w_{2\max}^0 = 0.01$. As figuras 5.11(a), 5.11(b), 5.12(a) e 5.12(b) mostram os

(a) Saídas para junta um ($N = 3$).(b) Saídas para junta um ($N = 3$).(c) MLP para junta um ($N = 3$).(d) MLP para junta dois ($N = 3$).Figura 5.12: Simulação usando iniciação de parâmetros no intervalo $(-0.01; 0.01)$.

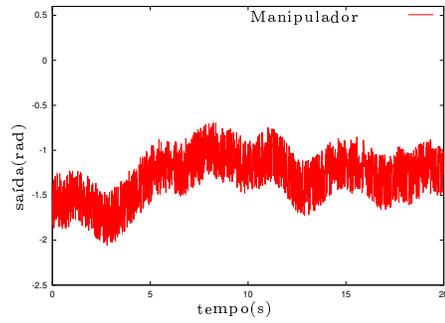
sinais de saída do sistema e dos identificadores traçados juntos, e as figuras 5.11(c), 5.11(d), 5.12(c) e 5.12(d) mostram as evoluções das funções objetivo para as saídas de cada identificador. Todas as outras simulações foram feitas usando o intervalo de iniciação de $(-0.5; 0.5)$ e $w_{2_{\max}}^0 = 0.5$. Estes diferentes intervalos de iniciação correspondem a subespaços \mathcal{W}^0 possuindo tamanhos diferentes, e também a diferentes condições iniciais para o vetor $\widehat{\mathbf{w}}^0$. Com isto, é possível observar como diferentes condições iniciais influenciam no desempenho do identificador.

O identificador mostrado na figura 5.12 teve um desempenho melhor que o identificador mostrado nas figuras 5.5(a), 5.5(b), 5.6(a), 5.6(b) e 5.11. Isto ocorreu porque o identificador mostrado na figura 5.12 foi iniciado com os subespaços \mathcal{W}^0 e com os valores para $w_{2_{\max}}^0$ das redes **MLP**s menores que os dos outros. Com isto, seus passos, ou taxas de aprendizagem, foram maiores que os dos outros, e, por conseguinte, suas velocidades de ajuste de parâmetros foram mais rápidas que as velocidades de ajuste de parâmetros dos outros identificadores (veja a seção 2.6).

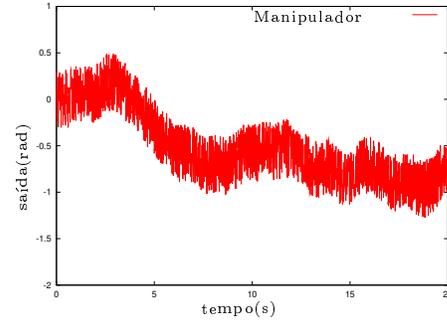
O uso de condições iniciais diferentes mostra que o paradigma proposto converge a uma solução independentemente da condição inicial, tanto da rede **MLP** como do sistema robótico. Além disso, mostra que um método mais elaborado para determinar uma condição inicial boa poderia ser útil (veja, por exemplo, o método proposto por Castro e Von Zuben (1998) e as análises feitas por Redondo e Espinosa (2000)).

Adicionalmente, note que existem duas maneiras de aumentar o intervalo de aceitação para η , e por conseqüência a velocidade de ajuste de parâmetros para o identificador:

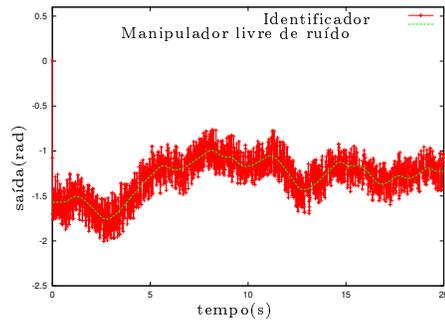
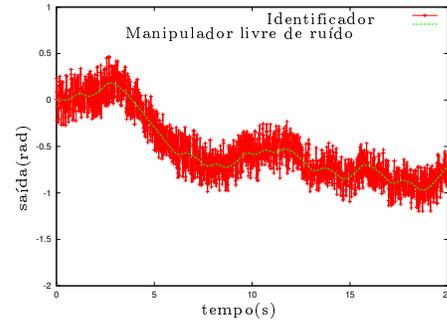
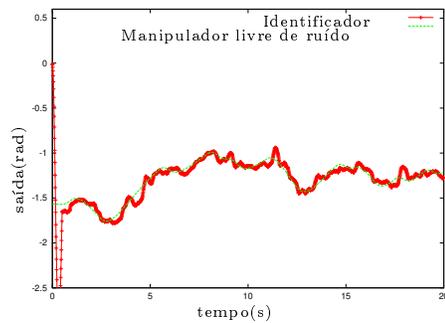
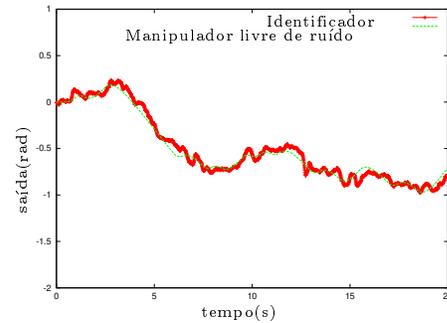
- (i) Fazendo os espaços $\mathcal{U} \subset \mathbb{R}^m$ e $\mathcal{Y} \subset \mathbb{R}^m$ pequenos [i.e., por fazer uma contração do espaços por meio do uso de sensores para as variáveis de entrada e saída do sistema que possuam níveis de sinais pequenos]. Com isto, $u_{m_{\max}}^0$ poderia ser pequeno e, pelo teorema 8, um incremento para o intervalo de aceitação de η seria adquirido;
- (ii) Pode ser feito $\kappa_{\phi_1} < 1$ (veja o teorema 8). Usualmente é feito $\kappa_{\phi_1} = 1$, mas nada



(a) Saídas para junta um.



(b) Saídas para junta dois.

(c) Saídas para junta um ($\alpha = 0.1$).(d) Saídas para junta dois ($\alpha = 0.1$).(e) Saídas para junta um ($\alpha = 0.99$).(f) Saídas para junta dois ($\alpha = 0.99$).Figura 5.13: Simulação usando $N = 3$ e ruído nos sinais de saída.

impede o uso de $\kappa_{\phi_1} < 1$.

A figura 5.13 mostra simulações feitas com $N = 3$ e ruído aditivo nos sinais de saída do sistema robótico. O ruído é branco e com nível equivalente a 10% do nível máximo do sinal de saída. As figuras 5.13(a) e 5.13(b) mostram os sinais de saída do sistema corrompidos pelo ruído. As figuras 5.13(c) e 5.13(d) mostram os sinais de saída do sistema livre de ruído sobreposto com saída do identificador para um $\alpha = 0.1$. As subfiguras 5.13(e) e 5.13(f) mostram os sinais de saída do sistema livre de ruído sobreposto com saída do identificador para um $\alpha = 0.99$.

As figuras 5.13(c) e 5.13(d) comparadas com as figuras 5.13(e) e 5.13(f) mostram o efeito da escolha de diferentes valores de α para o ajuste de parâmetros em redes **MLP**s quando os sinais do sistema são ruidosos (veja a seção 3.2). O identificador com $\alpha = 0.1$ apresenta uma capacidade de rejeição à distúrbios fraca em comparação com o identificador com $\alpha = 0.99$. O identificador com $\alpha = 0.1$ acompanhou a dinâmica do sistema, mas com muito ruído na saída [i.e., seu ajuste de parâmetros foi muito sensível ao ruído]. O identificador com $\alpha = 0.99$ acompanhou o sistema e filtrou a maior parte dos distúrbios externos [i.e, seu ajuste de parâmetros foi pouco sensível ao ruído]. A escolha de um valor apropriado para α sempre irá depender do conhecimento a respeito dos ruídos e perturbações observados no sistema a ser identificado. Em princípio, a escolha de um valor grande para α garantiria o efeito da filtragem com estabilidade sob perturbação, contudo a velocidade de ajuste de parâmetros seria significativamente reduzida. Logo, existe também um compromisso entre a filtragem e a velocidade de ajuste de parâmetros. Em adição, as simulações apresentada na figura 5.13 comprovam o colorário 2 e o lema 3 para redes do tipo **MLP**.

5.5 Sumário

Neste capítulo foi discutido a identificação de sistemas dinâmicos por meio do uso de **RNAs** do tipo **MLP**, que é uma ferramenta de modelagem matemática não

linear, e feita a aplicação do paradigma para um sistema manipulador robótico de dois **GL** cujas dificuldades de modelagem analítica são elevadas, como pode ser visto no capítulo 4.

Inicialmente, foi feita uma breve apresentação sobre a identificação de sistemas não lineares, com a indicação de alguns resultados importantes divulgados na literatura técnica. Na seção 5.2, foi discutida a questão da relação entre as regiões de densidade do sistema e do identificador. Esta seção mostrou que a situação ideal, que é quando a região de densidade do identificador é maior ou igual à do sistema, é muito difícil de ser obtida em situações reais, tendo em vista a existência de pouca informação sobre o sistema. A situação alternativa, e mais realística, é quando a região de densidade do identificador é menor que a do sistema. Foi mostrado que esta alternativa tornar-se-ia viável se o identificador fosse capaz de rastrear as trajetórias do sistema, todas as vezes que este seguisse para uma região ainda não aproximada, por meio de um método de ajuste de parâmetros adaptativo, convergente, e com velocidade de ajuste de parâmetros rápida o suficiente para acompanhar a dinâmica do sistema.

A seção 5.3 apresentou o modelo do identificador não linear multivariável baseado na propriedade da observabilidade genérica. Foram introduzidas as suposições necessárias para tornar o sistema tratável pela ótica da observabilidade genérica. O modelo do identificador possui uma rede **MLP** associada com cada saída do identificador, e com ajustes de parâmetros independentes e baseada no método do gradiente apresentado no capítulo 3.

Na seção 5.4 foi apresentada a aplicação do paradigma, por meio de simulação, ao sistema manipulador robótico articulado apresentado na seção 4.4. Estabelecidas condições iniciais para o sistema e para o identificador, foram inicialmente feitas simulações com o identificador possuindo diferentes números de neurônios artificiais na primeira camada da rede **MLP**, correspondendo assim a identificadores com diferentes regiões de densidade. Este experimento mostrou que o identificador com menor região de densidade foi o mais eficiente, conforme o paradigma proposto. Foi feito um outro

experimento em que foram variados os intervalos de iniciação dos parâmetros das redes **MLP**, correspondendo à variação dos tamanhos dos subespaços \mathcal{W}^0 . Verificou-se que uma contração neste subespaço [i.e., uma redução no seu tamanho] possibilita a obtenção de um maior intervalo de aceitação para η , e com o uso do algoritmo 1, uma maior velocidade no processo de ajuste de parâmetros das redes. Finalizando, estes experimento juntamente com um outro em que era variado o parâmetro α , *momentum*, do método de ajuste de parâmetro durante a identificação do sistema na presença de ruídos aditivos na saída possibilitou a verificação da convergência do método em função das variações de α , η e de distúrbios externos.

Capítulo 6

Considerações Finais

O ajuste de parâmetros baseado no método do vetor gradiente para redes **MLP**'s apresentados na literatura técnica carecem de análises teóricas sobre a convergência (veja Rumelhart *et al.* (1986) e Jin *et al.* (1995) por exemplo). A análise de convergência do ajuste de parâmetros para modelos matemáticos não lineares genéricos, apresentada na seção 3.2, baseado no método de gradiente, é útil e foi aplicada a rede **MLP** (Cerqueira *et al.*, 2000a; Cerqueira *et al.*, 2000c). Se as suposições 1 a 3 forem satisfeitas, a análise da convergência do ajuste de parâmetros pode ser aplicada a qualquer modelo matemático não linear. Esta análise aplicada ao modelo de rede **MLP** resultou no algoritmo adaptativo 1, que é capaz de assegurar bom desempenho ao processo de ajuste de parâmetros.

Um identificador não linear e multivariável para sistemas genericamente observáveis, usando redes **MLP**s como ferramenta de modelagem matemática não linear e a sistemática de ajuste de parâmetros apresentada no capítulo 3, foi desenvolvido e aplicado, por meio de simulações, a um sistema manipulador robótico com dois graus de liberdade. Os resultados mostraram a eficiência do paradigma. O modelo de identificador proposto possibilita o desenvolvimento de sistemas de controle adaptativos e/ou robustos para casos em que haja dificuldades de projeto pelo uso da abordagem linear (Cerqueira e Badan Palhares, 1999; Cerqueira *et al.*, 2000b; Cerqueira *et al.*, 2000c).

O paradigma proposto mostrou eficiência para aplicações *on-line*. A estimação de parâmetros baseado no método do vetor gradiente, junto com um modelo de identificador com região de densidade de pequeno tamanho, aumenta a velocidade do processo de ajuste de parâmetros. As regiões de densidade de pequeno tamanho reduzem o número de parâmetros a serem ajustados, e por conseguinte a demanda computacional. Adicionalmente, no caso de **RNA's** do tipo **MLP**, pelo teorema 8 o intervalo de aceitação do passo de ajuste ou taxa de aprendizagem, η , é incrementado. O algoritmo adaptativo proposto usa o limite superior do intervalo de aceitação, e uma rotina interna de gerenciamento protege o método proposto contra a instabilidade sob perturbação.

Durante a identificação, pode ser usado um sub-sistema de monitoramento e detecção de falhas para o sistema com o propósito de detectar falhas ou perturbações indesejáveis nas condições de funcionamento (veja Vemuri *et al.* (1998) por exemplo). Se qualquer condição indesejável de funcionamento acontecer, as suposições feitas sobre o sistema podem não ser adequadamente preenchidas, e o paradigma proposto pode não trabalhar com um bom desempenho. O sub-sistema de monitoramento e detecção de falhas pode interromper o sistema em três passos: detecção, diagnóstico, e acomodação. O sistema pode retornar para as condições de funcionamento normais após a atuação do sub-sistema de monitoramento e detecção de falhas. Com isto, o paradigma proposto pode trabalhar sempre com o sistema livre do efeito das falhas e das perturbações.

Os desenvolvimentos e análises feitos no capítulo 5 podem ser usados para tratar qualquer identificador não recorrente [i.e, para qualquer outra ferramenta matemática que realize mapeamentos estáticos e que tenha sido provada como sendo um aproximador universal, como alguns modelos *fuzzy* por exemplo].

6.1 Sugestões de Futuros Trabalhos

- (i) Aplicar o teorema 7 a outros algoritmos de ajuste de parâmetros para **RNAs**, como por exemplo o algoritmo proposto por Jin *et al.* (1995).

- (ii) Aplicar o teorema 7 a **RNAs** recorrentes, como por exemplo os modelos abordados por Ku e Lee (1995).
- (iii) Aplicação dos desenvolvimentos feitos na seção 3.2 para outros aproximadores universais, como por exemplo modelos fuzzy.
- (iv) Extensão do algoritmo adaptativo apresentado na seção 3.4 para outros modelos de aproximadores universais, como modelos *fuzzy* por exemplo.
- (v) Estudo de outras estruturas de identificadores usando a propriedade da observabilidade genérica.
- (vi) Desenvolvimento de técnicas ou métodos que usem identificadores baseados na propriedade da observabilidade genérica para cancelamento da dinâmica do sistema para aplicações em controle de sistemas robóticos. Algumas das estruturas de controle apresentadas por Lewis *et al.* (1999) podem ser adaptadas para o uso destes identificadores.

Referências Bibliográficas

- Abdallah, C., D. Dawson, P. Dorato e M. Jamshidi (1991). Survey of robust control of rigid robots. *IEEE Control Syst. Mag.* **11**(1), 24–30.
- Aeyels, D. (1981). Generic observability of differentiable systems. *SIAM J. Control and Optimization* **19**(5), 595–603.
- Álvarez, A. I. M. (1994). Estabilidade e Caos em Sistemas Dinâmicos não Lineares: Aplicação no Sistema **PLL**-Dual. Tese de Doutorado. Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, SP, Brasil.
- Bacciotti, A. (1992). *Local Stabilization of Nonlinear Control Systems*. Series on Advances in Mathematics for Applied Sciences. World Scientific.
- Baldi, P. (1995). Gradient descent learning algorithm overview: A general dynamical systems perspective. *IEEE Transactions on Neural Networks* **6**(1), 182–195.
- Ballini, R. e F. J. Von Zuben (2000). Book review: “Application of Neural Networks to Adaptive Control of Nonlinear Systems”. *Automatica* **36**(12), 1929–1933.

- Barron, A. R. (1993). Universal approximation bounds for superposition of a sigmoidal function. *IEEE Transactions on Information Theory* **39**(3), 930–945.
- Brent, R. P. (1991). Fast training algorithms for multilayer neural nets. *IEEE Transactions on Neural Networks* **2**(3), 346–354.
- Cabrera, J. B. D. e K. S. Narendra (1999). Issues in the application of neural networks for tracking based on inverse control. *IEEE Transactions on Automatic Control* **44**(11), 2007–2027.
- Cassandras, C. G. (1993). *Discrete Event Systems: Modeling and Performance Analysis*. Associates Series in Electrical and Computer Engineering. Aksen Associates Incorporated publishers.
- Castro, J. L. (1995). Fuzzy logic controllers are universal approximator. *IEEE Transactions on Systems, Man, and Cybernetics* **25**(4), 629–635.
- Castro, J. L. e M. Delgado (1996). Fuzzy systems with defuzzification are universal approximators. *IEEE Transactions on Systems, Man, and Cybernetics* **26**(1), 149–152.
- Castro, L. N. e F. J. Von Zuben (1998). A hybrid paradigm for weight initialization in supervised feedforward neural network learning. Em: *Proceeding of International Computer Symposium (Workshop on Artificial Intelligence)*. Tainan, Taiwan. pp. 30–37.
- Cerqueira, J. J. F., A. G. Badan Palhares e M. K. Madrid (2000a). A complement to the back-propagation algorithm: An upper bound for the learning rate. Em: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Vol. IV. Como, Itália. pp. 517–522.
- Cerqueira, J. J. F., A. G. Badan Palhares e M. K. Madrid (2000b). Identification of the dynamical model in robotic systems using only information about the position.

- Em: *Preprints of the 1st IFAC Conference on Mechatronic Systems*. Darmstadt, Alemanha. pp. 755–760.
- Cerqueira, J. J. F., A. G. Badan Palhares e M. K. Madrid (2000c). A nonlinear identifier based on artificial neurons: Application for a robotic system. Artigo submetido em 16 de outubro de 1998 a *Automatica - A Journal of IFAC the International Federation of Automatic Control*. Revisado em 13 de agosto de 1999, em 5 de abril de 2000 e em 21 de fevereiro de 2001.
- Cerqueira, J. J. F. e A. G. Badan Palhares (1999). Identificação dinâmica de sistemas robóticos via redes neurais artificiais e usando apenas informações sobre a posição. Em: *Anais do 4^o Simpósio Brasileiro de Automação Inteligente*. São Paulo, Brasil. pp. 130–135.
- Chen, C.-T. (1970). *Linear System Theory and Design*. HRW Series in Electrical and Computer Engineering. Holt, Rinehart and Wilson, Inc.
- Chen, F.-C. e H. Khalil (1995). Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control* **40**(5), 791–801.
- Chen, G., Y. Chen e H. Ogmen (1997). Identifying chaotic systems via a wiener-type cascade model. *IEEE Control Systems Magazine* **17**(5), 29–36.
- Close, C. M. e D. K. Frederick (1995). *Modeling and Analysis of Dynamic Systems*. John Willey & Sons, Inc.
- Cybenko, G. (1989). Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals, and Systems* **2**, 303–314.
- Dandy, G. C. e R. F. Warner (1989). *Planning and Design of Engineering Systems*. Unwin Hyman Ltd.. London, UK.

- Denavit, J. e R. S. Hartenberg (1955). A kinematics notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics* **77**, 215–221.
- Glendinning, P. (1994). *Stability, Instability and Chaos: An Introduction to the Theory of Nonlinear differential Equations*. Cambridge Texts in Applied Mathematics. Cambridge University Press.
- Golovko, V. e Y. Savisky (1999). New approach of the recurrent neural networks. Em: *Proceedings of the International Conference on Neural Networks and Artificial Intelligence*. Brest, Bielorrussia. pp. 32–35.
- Golovko, V., Y. Savisky, T. Laopoulos, A. Sachenko e L. Grandinetti (2000). Technique of learning rate estimation for efficient training of **MLP**. Em: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Vol. I. Como, Itália. pp. 323–329.
- Golubitsky, M. e V. Guillemin (1973). *Stable Mappings and Their Singularities*. Springer-Verlag Berlin, Inc.
- Gori, M. e M. Maggini (1996). Optimal convergence of on-line backpropagation. *IEEE Transactions on Neural Networks* **7**(1), 251–254.
- Guckenheimer, J. e P. Holmes (1983). *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Vol. 42 de *Applied Mathematical Sciences*. Springer-Verlag New York, Inc.
- Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks*. The MIT Press.
- Hermann, R. e A. J. Krener (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control* **AC-22**(5), 728–740.
- Hertz, J., A. Krogh e R. G. Palmer (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company.

- Hornik, K., M. Stinchcombe e H. White (1989). Multi-layer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366.
- Hsu, L. (1990). Variable structure model reference adaptive control using only I/O measurement: General case. *IEEE Transactions on Automatic Control* **35**, 1238–1243.
- Hsu, L. e F. Lizarralde (1993). Variable structure adaptive tracking control of manipulators without joint velocity measurement. Em: *Proceeding of the 12th IFAC World Congress*. Sydney, Australia. pp. 145–148.
- Hsu, L. e F. Lizarralde (1995). Experimental results on variable structure adaptive robot control without joint velocity measurement. Em: *Proceeding of the American Control Conference*. Seattle, USA. pp. 2317–2321.
- Hunt, K. J., D. Sbarbaro, R. Żbikowski e P. J. Gawthrop (1992). Neural networks for control systems - a survey. *Automatica* **28**(6), 1083–1112.
- Jaffard, S. e P. Laurentçot (1992). *Wavelets: A tutorial*. Cap. Wavelets and PDE. Academic Press.
- Jagannathan, S. e F. L. Lewis (1996). Identification of nonlinear dynamical systems using multilayered neural networks. *Automatica* **32**(12), 1707–1712.
- Jin, L., P. N. Nikiforuk e M. Gupta (1995). Fast neural learning and control of discrete-time nonlinear systems. *IEEE Transactions on Systems, Man, and Cybernetics* **25**(3), 478–488.
- Juditsky, A., H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg e Q. Zhang (1995). Nonlinear black-box model in system identification: Mathematical foundations. *Automatica* **31**(12), 1725–1750.
- Kalman, R. E. (1960). On the general theory of control systems. Em: *Proceeding of the First International Congress of Automatic Control*. Moscow, Russia. pp. 481–493.

- Kalman, R. E. e J. E. Bertram (1960a). Control system analysis and design via second method of lyapunov - part I, continuous-times. *Transactions of ASME, Journal of Basic Engineering, Series D* **3**(2), 371–393.
- Kalman, R. E. e J. E. Bertram (1960b). Control system analysis and design via second method of lyapunov - part II, discrete-times. *Transactions of ASME, Journal of Basic Engineering, Series D* **3**(2), 393–400.
- Kamen, E. W. e B. S. Heck (1999). *Fundamentals of Signal and Systems: Using the web and Matlab*. second ed.. Prentice Hall.
- Klafter, R. D., T. A. Chmielewski e M. Negin (1989). *Robotic Engineering: An Integrated Approach*. Prentice-Hall, Inc. Englewood Cliffs, New Jersey (USA).
- Koenig, H. E., Y. Tokad e H. K. Kesavan (1967). *Analysis of Discrete Physical Systems*. Electrical and Electronic Engineering Series. McGraw-Hill Book Company.
- Kreisselmeier, G. e D. O. Anderson (1986). Robust model reference adaptive control. *IEEE Transactions on Automatic Control* **AC-31**(2), 127–133.
- Krick, E. V. (1969). *Engineering and Engineering Design*. John Willey & Sons, Inc.
- Ku, C. C. e K. Y. Lee (1995). Diagonal recurrent neural networks for dynamic systems control. *IEEE Transactions on Neural Networks* **6**(1), 144–156.
- Kuan, C. M. e K. Hornik (1991). Convergence of learning algorithms with constant learning rates. *IEEE Transactions on Neural Networks* **2**(5), 484–489.
- Labinaz, G., M. M. Bayoumi e K. Rudie (1996). Modeling and control of hybrid systems: A survey. Em: *Proceeding of the 13th IFAC World Congress*. Vol. J. San Francisco, USA. pp. 293–303.
- Lane, D. M. (1998). Introduction to the Special Issue of the *International Journal of Systems Science* on the AMADEUS dextrous underwater grasping system. *International Journal of Systems Science* **29**(5), 445–454.

- Leng, D. Y. e M. Chen (1997). Robot trajectory planning using simulation. *Robotics & Computer-Integrated Manufacturing* **13**(2), 121–129.
- Leone, R., R. Capparuccia e E. Merelli (1998). A successive overrelaxation backpropagation algorithm for neural-network training. *IEEE Transactions on Neural Networks* **9**(3), 381–388.
- Leontaritis, I. J. e S. A. Billings (1985a). Input-output parametric model for non-linear systems part i: Deterministic non-linear systems. *International Journal of Control* **41**(2), 303–328.
- Leontaritis, I. J. e S. A. Billings (1985b). Input-output parametric model for non-linear systems part ii: Stochastic non-linear systems. *International Journal of Control* **41**(2), 329–344.
- Levin, A. U. e K. S. Narendra (1993). Control of nonlinear dynamics systems using neural networks: Controllability and stabilization. *IEEE Transactions on Neural Networks* **4**(2), 192–206.
- Levin, A. U. e K. S. Narendra (1995). Recursive identification using feedforward neural networks. *International Journal of Control* **61**(3), 533–547.
- Levin, A. U. e K. S. Narendra (1996). Control of nonlinear dynamics systems using neural networks - part II: Observability, identification, and control. *IEEE Transactions on Neural Networks* **7**(1), 30–42.
- Lewis, F. L., S. Jagannathan e A. Yesildirek (1999). *Neural Network Control RObot Manipulators and Nonlinear Systems*. Taylor & Francis Inc.
- Lizarralde, F. e J. T. Wen (1995). Attitude control without angular velocity measurement: A passivity approach. Em: *IEEE International Conference on Robotics and Automation*. Osaka, Japan. pp. 2701–2706.

- Lizarralde, F. e J. T. Wen (1996). Attitude control without angular velocity measurement: A passivity approach. *IEEE Transactions on Automatic Control* **41**(3), 468–472.
- Lizarralde, F., J. T. Wen e L. Hsu (1995). Quaternion-based coordinated control of a subsea mobile manipulator with only position measurements. Em: *Proceeding of the 34th IEEE Conference on Decision and Control*. New Orleans, USA. pp. 3996–4001.
- Ljung, L. (1987). *System Identification - Theory for the User*. Prentice Hall. Englewood Cliffs, NJ.
- Luenberger, D. G. (1984). *Linear and Nonlinear Programming*. second ed.. Addison-Wesley Publishing Company.
- Lyapunov, A. M. (1892). The General Problem of the Stability Motion. Tese de Doutorado. University of Kharkov, Russia. Texto escrito originalmente em russo.
- Magoulas, G. D., V. P. Plagianakos e M. N. Vrahatis (2000). Development and convergence analysis of training algorithms with local learning rate adaptation. Em: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Vol. I. Como, Itália. pp. 21–26.
- Mangasarian, O. L. (1994). *Nonlinear Programming*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics. Philadelphia, USA.
- Marino, R. e P. Tomei (1995). *Nonlinear Control Design: Geometric, Adaptive, & Robust*. Prentice Hall Information and System Sciences Series. Prentice Hall.
- Mayer, Y. (1990). *Ondelettes et Opérateurs*. Hermann. Paris, FR.
- McCulloch, J. L. e W. Pitts (1943). A logical calculus of ideas imminent in nervous activity. *Bulletin of Mathematical Biophysics* **5**, 115–133.

- Mehrotra, K., C. K. Mohan e S. Ranka (1997). *Elements of Artificial Neural Networks*. Complex Adaptive Systems. The MIT Press.
- Mhaskar, H. N. (1992). Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied Mathematics* **13**, 350–373.
- Minsky, M. L. e S. A. Papert (1969). *Perceptrons*. MIT Press. Cambridge, Massachusetts (USA).
- Narendra, K. S. e K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks* **1**(1), 4–27.
- Narendra, K. S. e S. Mukhopadhyay (1996). *Intelligent Control Systems: Theory and Applications*. Cap. Intelligent Control Using Neural Networks, pp. 151–186. IEEE Press.
- Neuman, C. P. e J. J. Murray (1987). The complete dynamic model and customized algorithms of the puma robot. *IEEE Transactions on Systems, Man and Cybernetics* **SMC-17**(4), 635–644.
- Ng, G. W. (1997). *Application of Neural Networks to Adaptive Control of Nonlinear Systems*. UMIST Control Systems Centre Series. Research Studies Press Ltd.
- Ortega, R. e M. W. Spong (1989). Adaptive motion control of rigid robots: a tutorial. *Automatica* **25**(6), 877–888.
- Parker, T. S. e L. O. Chua (1989). *Practical Numerical Algorithms for Chaotic Systems*. Springer-Verlag New York, Inc.
- Petrushev, P. e V. Popov (1985). *Rotational Approximation of Real Functions*. Cambridge University Press.

- Redondo, M. F. e C. H. Espinosa (2000). A comparison among weight initialization methods for multilayer feedforward networks. Em: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*. Vol. IV. Como, Itália. pp. 543–548.
- Robitaille, B., B. Marcos, M. Veillette e G. Payre (1996). Modified quase-newton methods for training neural networks. *Computers Chem. Engng* **20**(9), 1133–1140.
- Rosenberg, R. C. e D. C. Karnopp (1983). *Introduction to Physical System Dynamics*. McGraw-Hill Series in Mechanical Engineering. McGraw-Hill Book Company.
- Rosenblatt, F. (1962). *Principles of neurodynamics*. New York: Spartan.
- Ruelle, D. (1989). *Elements of Differentiable Dynamics and Bifurcation Theory*. Academic Press, Inc.. London, UK.
- Rumelhart, D. E., G. E. Hinton e R. J. Williams (1986). *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*. Cap. Learning internal representations by error propagating. MIT Press, Cambridge, MA.
- Sage, H. G., M. F. De Mathelin e E. Ostertag (1999). Robust control of robot manipulators: A survey. *International Journal of Control* **72**(16), 1498–1522.
- Schraudolph, N. N. (1998). Centering neural network gradient factors. Em: *Neural Networks: Tricks of the Trade* (G. B. Orr e K. R. Müller, Eds.). Vol. 1524 de *Lecture Notes in Computer Science*. Springer Verlag, Berlin.
- Seibert, P. (1963). Stability under perturbations in generalized dynamical systems. Em: *Nonlinear differential Equations and Nonlinear Mechanics* (J. P. LaSalle e S. Lefchetz, Eds.). pp. 463–473. Academic Press, New York.
- Shukla, K. K. e L. P. Raghunath (1999). An efficient global algorithm for supervised training of neural networks. *Computer and Electrical Engineering* **25**, 195–218.

- Sjöberg, J. e L. Ljung (1995). Overtraining regularization and searching for a minimum, with application to neural networks. *International Journal of Control* **62**(6), 1391–1407.
- Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarsson e A. Juditsky (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica* **31**(12), 1691–1724.
- Spong, M. W. e M. Vidyasagar (1989). *Robot Dynamics and Control*. John Willy & Sons.
- Suykens, J. A. K., B. L. R. De Moor e J. Vandewalle (1995). Nonlinear system identification using neural state space models, applicable to robust control design. *International Journal of Control* **62**(1), 129–152.
- Tarn, T. J., A. K. Bejczy, X. Yun e Z. Li (1991). Effect of motor dynamics on nonlinear feedback robot arm control. *IEEE Transactions on Robotics and Automation* **7**(1), 114–121.
- Unberhauen, H. e G. P. Roa (1998). A review of identification in continuous-time systems. Em: *Annual Reviews in Control*. Vol. 22. pp. 145–171.
- Van Milligen, B. P., V. Tribaldos, J. A. Jiménez e C. Santa-Cruz (1998). Comments on “An accelerated learning algorithm for multilayer perceptrons: Optimization layer by layer”. *IEEE Transactions on Neural Networks* **9**(2), 339–341.
- Vemuri, A. T., M. M. polycarpou e S. A. Diakourtis (1998). Neural network based fault detection in robotic manipulator. *IEEE Transactions on Robotic and Automation* **14**(2), 342–348.
- Vidyasagar, M. (1993). *Nonlinear Systems Analysis*. second edition ed.. Prentice-Hall, Inc.. Englewood Cliffs, New Jersey, USA.

- Wang, G. J. e C. C. Chen (1996). A fast multilayer neural-networks training algorithm based on the layer-by-layer optimizing procedures. *IEEE Transactions on Neural Networks* **7**(3), 768–775.
- Wang, L. X. (1992). Fuzzy systems are universal approximators. Em: *Proceedings of the IEEE International Conference on Fuzzy Systems*. San Diego, USA. pp. 1163–1170.
- Yu, H. e S. Lloyd (1995). Adaptive control of robot manipulators including motor dynamics. *Proc. Instn. Mech. Engrs* **209**, 207–217.

Índice Remissivo

- acelerômetros, 84
- álgebra de dioides, 11
- amplificadores de torque, 80
- aprendizagem, 43
 - em *batch*, 50
 - incremental, 50
- aproximador universal, 36, 42, 101
- atuadores, 72
- automato formais, 11
- back-propagation*, 43
- base de atração, 22
- bias, 57, 59
- camada
 - de entrada, 58
 - de saída, 58
- escondida, 58
- intermediária, 58
- círculo unitário, 38
- conjunto
 - aberto, 17
 - compacto, 17
 - convexo, 31
 - denso, 18
- constante de Lipschitz, 19
- continuidade Lipschitz local, 19
- coordenadas generalizadas, 74
- coordenadas-base, 75
- Degree-Of-Freedom*, 72
- Denavit e Hartenberg, 73
- difeomorfismo, 18
- domínio de atração, 22

- easiest-path*, 68
- elos, 71
- encoders*, 84
- equações diferenciais ordinárias, 13
- espaço de estado, 14
- estabilidade sob perturbação, 24
- estados distinguíveis, 26
- eventos discretos, 11

- filtro, 51
- fator de correção, 52
- fluxo, 17
- forças
 - centrípetas, 77
 - de coriolis, 77
- função
 - contínua, 17
 - convexa, 32
 - de ativação, 42
 - de base radial, 58
 - de classe C^r , 17
 - de Lyapunov, 23
 - de transferência, 37
 - definida positiva, 23
 - decrecente, 23
 - diferenciável, 17
 - radial base*, 58
 - sigmoidal, 58, 64
 - suave, 17
- fuzzy*, 12, 36

- identificação, 41, 46

- juntas, 71
 - prismáticas, 71
 - rotacionais, 71

- learning*, 43
- links*, 71

- manifold, 18
- manipulador robótico, 71
 - articulado, 73, 85
 - cartesiano, 73
 - cilíndrico, 73
 - esférico, 73
- mapeamento
 - contínuo, 17
 - de classe C^r , 17
 - de Morse, 20
 - diferenciável, 17
 - injetor, 18
 - inversível, 18
 - sobrejetor, 18
 - suave, 17
- matriz, 13
 - de inércia, 77
 - definida positiva, 28
 - hessiana, 29, 42
 - indefinida, 28
 - jacobiana, 19
- máximos, 13

- metasistema, 8
- método
 - de Lyapunov, 21, 54
 - de Newton, 33
 - do gradiente, 33, 44
 - recursivo, 33, 49
- mínimo
 - global, 30
 - local, 29
- modelagem matemática, 41, 46
- modelos, 10
 - abstratos, 10
 - descritivos, 10
 - diagramáticos, 10
 - gráficos, 10
 - matemáticos, 10
 - concreto, 10
 - físicos
 - icônicos, 10
 - protótipos, 10
 - matemáticos, 10
 - determinísticos, 12
 - estocástico, 12
 - fenomenológicos, 11
 - formais, 11
 - híbridos, 11
 - identificados, 11
 - lineares, 11, 12
 - não formais, 11
 - não lineares, 11, 12
 - probabilístico, 12
- momentum*, 33, 49
- motor elétrico, 78
- multi-layer feedforward perceptron*, 42
- Nomenclatura, 13
- norma, 13
 - Euclidiana, 13
 - Frobenius, 13
- números, 13
 - complexos, 13
 - naturais, 13
 - reais, 13
 - não negativos, 13
- observabilidade, 25
 - genérica, 101
- on-line*, 43
- órbita, 17
- ordem de convergência, 34
 - super-linear, 35
- passo, 33, 49
- perceptron, 57
- pólos, 38
- ponto
 - de equilíbrio, 20
 - assintoticamente estável, 22
 - atrativo, 20
 - estável, 21

- instável, 21
- isolado, 20
- interior, 29
- problema convexo, 34
- processo, 9
- propriedade genérica, 25
- recorrência, 16
- redes
 - de Petri, 11
 - neurais artificiais, 12, 36, 41
- redutores de movimento, 80
- região de densidade, 18
- regra de ajuste
 - em grupo, 50
 - incremental, 50
- representação de estado
 - discreta, 16
- revolutas, 71
- ruídos, 51
- sensores, 83
- shortest-path*, 68
- sistema, 7, 12
 - adaptativo, 8
 - ajustável, 8
 - autônomo, 17
 - computacional, 9
 - controle, 9
 - dinâmico, 8
 - contínuos, 11
 - discreto no tempo, 15
 - estáticos, 8
 - eventos discretos, 11
 - físico, 8, 13
 - forçado, 16
 - hierárquico, 8
 - inteligente, 9
 - modelo, 10
 - monitoramento, 9
 - não autônomo, 17
 - não forçado, 16
 - observável, 26
 - realizável, 9
 - representação, 10
 - de estado, 14
 - robótico, 71
 - análise cinemática, 72
 - análise dinâmica, 72
 - cinemática direta, 75
 - cinematicamente redundante, 72
 - trajetória, 76
 - robusto, 8
 - supervisão, 9
- spline*, 36
- subsistema, 8
- tacômetros, 84
- tempo discreto, 13, 16
- torques gravitacionais, 77

- trajetória, 17
- transformação pontual, 16
- transformada \mathcal{Z} , 37, 53

- valor absoluto, 13
- variável, 8
 - de entrada, 13
 - de estado, 14
 - de saída, 13
- variável de junta, 71
- variedade, 18
- vetor, 13
 - de diferenças, 33, 49
 - de estado, 14
 - gradiente, 28, 42
- vizinhança, 18

- wavelet*, 12, 36

- zona-morta, 51