Universidade Estadual Paulista

“ Júlio de Mesquita Filho “

FACULDADE DE ENGENHARIA DE ILHA SOLTEIRA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

IDENTIFICAÇÃO E CONTROLE DE SISTEMAS DINÂMICOS UTILIZANDO REDES WAVELETS

Luiz Henrique Maricato Grassi

Dissertação apresentada à Faculdade de Engenharia de Ilha Solteira da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Mecânica.

Orientador: Prof. Dr.João Antonio Pereira

Ilha Solteira - SP, Maio de 2004

Agradecimentos

A meus pais, à minha irmã, e a toda família pelo apoio e carinho oferecido. Agradeço em especial ao amigo/orientador João Antonio Pereira pela força, garra e entusiasmo desprendido e por todo aprendizado que a convivência nos proporcionou. Ao grande amigo Edison Righeto, que enobreceu o trabalho e com afinco dividiu seus conhecimentos. Agradeço os professores, técnicos e funcionários do Departamento de Engenharia Mecânica, pela ajuda no desenvolvimento do trabalho e ao Programa de Pós-Graduação e a Capes pelo apoio técnico e financeiro.

Sinto-me honrado com as amizades e tantas oportunidades de aprender. O mestrado possibilitou não apenas um aprimoramento profissional, mas também a consolidação de grandes amizades.

Muito obrigado UNESP – Ilha Solteira pelos sete anos maravilhosos que me ofereceu. Encerro este trabalho com a certeza de que os melhores resultados dele foram as amizades que ele consolidou.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

A necessidade de controle no tratamento de sistemas dinâmicos, com complexidade crescente e diante de fatores de incerteza, tem levado à reavaliação dos métodos convencionais e à proposição de métodos conceitualmente mais elaborados de controle. Estas novas propostas incluem, por exemplo, níveis hierárquicos de decisão, planejamento e aprendizagem, que são necessários quando um alto grau de autonomia do sistema é desejável. Assim as metodologias baseadas em redes neurais, que utilizam modelos matemáticos e técnicas numéricas inspiradas no cérebro humano e/ou sistema nervoso, representam um passo natural na evolução da teoria de controle, principalmente junto àqueles que envolvem não-linearidades. Este trabalho apresenta um estudo da técnica denominada *wavenet*, que combina redes neurais e transformada *wavelet*, como um direcionamento alternativo para a solução de problemas de identificação e controle de plantas não lineares. A transformada *wavelet* utiliza janelas com escala variável que possibilitam analisar faixas de altas e baixas freqüências em um mesmo sinal, e é exatamente essa capacidade de manipulação da janela de observação que a torna uma boa alternativa como função de ativação, realizando um mapeamento local do sinal. Isso proporciona uma identificação mais eficiente, principalmente em sinais não lineares e variantes no tempo. Vários testes simulados envolvendo não linearidade foram analisados visando estudar o comportamento do algoritmo *wavenet* e definir quais os tipos de funções de ativação, Morlet, Rasp ou Polywog, poderiam fornecer melhores resultados. Utilizou-se o método de otimização de Levenberg-Marquadt, o qual apresentou um desempenho melhor quando comparado com o método do gradiente descendente utilizado por outros autores, no processo de minimização do erro entre a saída da rede e a saída da planta. Os testes buscaram definir melhorias no algoritmo *wavenet*, com relação ao processo de identificação, pois trata-se de uma etapa primordial no projeto de neurocontroladores. Com os parâmetros da planta identificados, é possível construir um neurocontrolador. Neste caso utiliza-se a técnica de controle inverso, buscando definir um neurocontrolador capaz de manipular/controlar uma planta desconhecida para seguir um sinal de referência.

Palavras Chave: Identificação, Controle de sistemas dinâmicos, Redes neurais e Wavelets.

Abstract

The necessity of dynamic systems treatment control, with upper complexity and uncertain factors, has lead to reevaluation of conventional methods and the proposition of conceptly methods more elaborate of control. These new proposals include, for instance, hierarchic levels of decision, planning and learning, which are needed when a high degree of system autonomy is desirable. Thus the methodologies based in neural nets, which use mathematical models and numerical techniques inspired in human brain and/or nervous system, represent a natural step in evolution of control theory, mainly join to those which involve no-linearity. This work shows a technique study called *wavenet*, it combine neural nets and *wavelet* transformed, as an alternative leading for the solution of identification problems and non linear plants control. The transformed *wavelet* uses windows with variable scale and it makes possible analyze strips high and low frequencies in a same signal, and it is exactly this capacity of manipulation of observation window and it becomes a good alternative as activation function, achieving a local map of the signal. A identification more efficient is provided, mainly in non-linear signals and time variants. Several simulate tests involving non linear was analyzed, seeking to study the behavior of the algorithm *wavenet* and to define which the types of activation functions, Morlet, Rasp or Polywog, could give better results. The optimization method of Levenberg-Marquadt was used, and that one show a better performance when compared with the descendent gradient method used by other authors, in minimization of error process between the net and plant exit. The tests looked for to define improvements in algorithm *wavenet*, in relation to identification process, because it is primordial stage in the project of neurocontrolers. The identified plant parameters, it is possible to define a neurocontroler. In this case the technique of inverse control is used, looking for to define neurocontroler capable to handle/control an unknown plant to follow a reference signal.

Keywords: Identification, Dynamics control systems, Neural networks and Wavelets.

Capítulo 1

Introdução

1.1 Motivação do Trabalho

Um grande desejo do homem, a partir do momento em que as máquinas começaram a se desenvolver tem sido a criação de máquinas que possam operar independentemente do controle humano, máquinas, cuja independência seja desenvolvida de acordo com seu próprio aprendizado e necessidades, que tenham a capacidade de interagir com ambientes incertos (desconhecidos por ela), e de se adaptar a novas necessidades, uma máquina, que possa ser chamada de autônoma, inteligente ou cognitiva. A reunião destas características em uma máquina pode ser denominada de Inteligência Artificial (*IA*).

O sucesso de uma máquina autônoma depende claramente de sua capacidade de lidar com uma variedade de eventos inesperados que ocorrem no ambiente em que opera. Estas máquinas dotadas de inteligência artificial teriam maior capacidade de aprender tarefas de alto nível cognitivo que não são facilmente manipuladas por máquinas convencionais, e continuariam a se adaptar e realizar essas tarefas gradativamente com maior eficiência, mesmo que em condições de ambiente imprevisíveis.

Uma das fontes de inspiração para o desenvolvimento das máquinas dotadas de *IA* é o organismo humano, que proporciona diversas pistas para o desenvolvimento de algoritmos de aprendizado e adaptação que possibilitam algum tipo de tomada de decisão. A diferença básica das máquinas inspiradas na biologia em relação às máquinas atuais se encontra no fato de que essas baseiam seu processamento explicitamente em modelos matemáticos, enquanto que as máquinas dotadas de *IA* possuem um dado grau de autonomia para a tomada de decisão e operam

com um maior número de incertezas e variáveis, principalmente nos casos em que é muito complicado obter um modelo matemático.

Neste contexto surgem as Redes Neurais Artificiais que se baseiam na visão da *IA*. Acredita-se que construindo um modelo matemático que simule a estrutura cerebral humana, este apresentará "inteligência" e será capaz de aprender, assimilar, errar, e com esta experiência adquirir novos conhecimentos (IANET, Inteligência Artificial, 2004).

Desta forma, as Redes Neurais tendem a representar uma estrutura interconectada similar à estrutura do cérebro humano e com isto resolver situações não facilmente resolvidas na computação convencional. Assim as Redes Neurais tornam-se de fundamental importância no desenvolvimento dos sistemas de *IA*, em que há necessidade de resolução de problemas não lineares, bem como no reconhecimento de padrões, na otimização e na previsão de sistemas complexos.

Redes neurais artificiais representam, portanto, um direcionamento alternativo para a solução de problemas em identificação e controle, principalmente junto àqueles que envolvem não-linearidades. O uso de redes neurais em sistemas de controle pode ser visto como um passo natural na evolução da metodologia de controle no sentido de buscar a superação de suas limitações. Analisando as técnicas existentes, a evolução na área de identificação e controle tem sido motivada por três grandes necessidades:

1. Tratar sistemas de grande complexidade, a qual vem crescendo continuamente;
2. Acompanhar o aumento de demanda por eficiência e qualidade, presente nos requisitos de projeto;
3. Tratar estes requisitos sob condições de incerteza e imprecisão.

Hoje, a necessidade de controle no tratamento de sistemas dinâmicos com complexidade crescente e diante de fatores de incerteza, tem levado à reavaliação dos métodos convencionais de controle e à proposição de métodos conceitualmente mais elaborados. Estas novas propostas incluem, por exemplo, níveis hierárquicos de decisão, planejamento e aprendizagem, que são necessários quando um alto grau de autonomia do sistema é desejável.

A teoria convencional de controle: Diagrama de Nyquist, Bode, Espaço de Estados, Controle Ótimo e Lugar das Raízes são técnicas mais adequadas quando direcionadas ao projeto de algoritmos para o tratamento de processos lineares. Além disto, os métodos convencionais

baseiam-se na linearidade e comportamento invariante no tempo dos processos a serem controlados.

Na teoria convencional de controle, os objetivos do projeto do controlador são fixos e definidos pelo projetista (Coelho, 2000). Esta abordagem é limitada pelas situações do mundo real, devido ao fato que sistemas complexos requerem maior flexibilidade.

Os projetos de controle de sistemas baseados em linearização são amplamente difundidos em diversos domínios, mas a crescente demanda por qualidade (maior capacidade de manipulação de imprecisão e incerteza), e o contínuo aumento na complexidade (maior intensidade dos fenômenos não-lineares envolvidos), dos problemas de identificação de sistemas e controle de processos têm evidenciado as limitações desta abordagem. Particularmente na presença de incertezas e não-linearidades significativas às técnicas mais avançadas, desenvolvidas para o tratamento de sistemas lineares, não são capazes de responder ao conjunto de necessidades requeridas para atender aos critérios de desempenho, geralmente adotados em identificação e controle de processos atualmente em demanda (Narendra e Annaswamy, 1989). Sendo assim, embora a suposição da linearidade de sistemas dinâmicos tenha sido feita para desenvolver a teoria de controle, ela acaba reduzindo o alcance dos resultados, a ponto de dificultar a abordagem dos problemas de engenharia mais avançados atualmente.

Os sistemas não-lineares, por sua vez, podem apresentar comportamentos dinâmicos mais complexos, por exemplo, com uma mudança qualitativa de comportamento para diferentes pontos de operação e com muito mais flexibilidade em sua evolução no tempo. Isto, no entanto, dificulta a formulação de uma teoria sistemática e geral, aplicável a projetos de identificação e controle não-linear. Esta impossibilidade atual de recorrer a ferramentas lineares efetivas, ou não-lineares genéricas, cria a necessidade de aplicação de ferramentas dedicadas, inerentemente não-lineares e com grande poder de adaptação, dentre as quais se destacam as redes neurais artificiais (Hunt et al., 1992).

Neste cenário, o projeto de controladores auto-ajustáveis são ferramentas de grande interesse, e o desenvolvimento de métodos eficientes de auto-ajuste são aspectos importantes a serem considerados na construção dos controladores adaptativos. A utilização e adaptação de novas ferramentas matemáticas, principalmente, ferramentas capazes de trabalhar com não linearidades são de grande interesse.

As funções *wavelets*, utilizadas na teoria de análise de sinais, cuja principal característica é tratar sinais variantes no tempo e com comportamento não linear, surgem como uma alternativa para aproximação de sinais, buscando encontrar um conjunto de *wavelets*-filhas (construídas por

dilatação/compressão e translações da *wavelet* mãe) que melhor represente o sinal analisado. Uma *wavelet* comprimida caracteriza-se por captar mudanças rápidas (altas freqüências), já uma *wavelet* alongada caracteriza-se por captar mudanças lentas (baixas freqüências), no sinal, o que poderia ser explorado na identificação de sistemas.

Pesquisas objetivando agregar essas características das *wavelet*s na teoria de redes neurais são encontradas na literatura e a combinação delas tem sido uma proposta bem interessante. Uma dessas propostas, discutida inicialmente por Zhang e Benveniste (1992), é a denominada *wavenet*. Essa proposta introduz uma super *wavelet*, a qual é definida a partir de uma combinação linear de *wavelets* filhas que são tratadas como uma *wavelet* convencional. A *wavelet* filha é uma versão dilatada e/ou deslocada da *wavelet* mãe. A grande vantagem nesta aproximação é que as características da super *wavelet* permitem moldá-la para que ela se torne adequada a um problema particular, processo este, que vai além da adaptação dos parâmetros de uma *wavelet* de perfil fixo.

Pretende-se neste trabalho explorar a utilização de redes neurais e transformada *wavelet* como um direcionamento alternativo para a solução de problemas em identificação e controle de sistemas não lineares, considerando uma extensão do trabalho de Righeto (2003). O algoritmo *wavenet* apresentado nos trabalhos de Lekutai (1997) e Righeto (2003), é discutido, e uma nova abordagem do processo de otimização dos parâmetros da rede neural e das funções *wavelet*s, utilizadas na construção da rede neural adaptativa é proposta. São avaliados aspectos referentes à arquitetura da rede (número de neurônios e funções de ativação), eficiência e rapidez do algoritmo. A busca por melhorias no algoritmo *wavenet* recaem sobre o processo de identificação, que se trata de uma etapa primordial na obtenção de um bom projeto do neurocontrolador que é baseado na inversa da planta, tratando-se de um controle inverso/indireto.

1.2 Organização da Dissertação

A organização da dissertação apresenta inicialmente uma discussão geral a respeito de inteligência artificial, envolvendo redes neurais e controle adaptativo, justificando a sua aplicação à identificação e controle de sistemas não lineares.

No capítulo 2 é apresentada uma revisão bibliográfica evidenciando aspectos que envolvem a teoria de redes neurais e transformada *wavelet*, relacionadas à identificação e controle.

O capítulo 3 aborda de forma sucinta a teoria de redes neurais e apresenta tópicos relativos à classificação das redes neurais, o porquê de se utilizar as redes neurais, os tipos de treinamento, quais as funções de ativação mais comuns, um tópico abordando redes neurais em identificação de processos, e por fim traz uma tabela com as principais aplicações das redes neurais.

No capítulo 4 apresenta-se, também de forma sucinta, a teoria da transformada *wavelet* relatando todo um histórico acompanhado dos aspectos e limitações que levaram ao seu desenvolvimento em análise de sinais.

O capítulo 5 descreve o algoritmo *wavenet*, que interage as técnicas de redes neurais e *wavelet*s, discutindo a sua utilização na identificação e controle de processos, e também aborda algumas técnicas para o projeto de um controlador.

No capítulo 6 são apresentados os resultados simulados relativos à identificação e os resultados referentes ao controle utilizado para seguir uma dada referência. Por fim, no capítulo 7, são apresentadas as discussões e conclusões tiradas ao longo da dissertação e dos resultados apresentados.

Capítulo 2

Revisão Bibliográfica

O atual crescimento do interesse pelo estudo e implementação de modelos dotados de Inteligência Artificial é justificado com base no aumento do poder de processamento computacional e armazenamento de informação a baixo custo, principalmente em arquiteturas convencionais de computadores, geralmente empregadas para operar em "laboratórios de simulação e validação". No entanto, para propósito de identificação e controle de processos, os modelos convencionais de redes neurais artificiais apresentam limitações, devidas principalmente a três fatores (Ronco e Gawthop, 1995):

1. Não existe um modo sistemático para determinar a estrutura de modelagem requerida para um dado sistema. Isto torna a capacidade de aproximação universal, associada às redes neurais artificiais, de pouco efeito prático;
2. O algoritmo iterativo de aprendizado (*retropropagação*), essencialmente baseado no gradiente descendente, nem sempre leva à convergência em direção à solução global, ou a uma solução local de boa qualidade, e geralmente requer um número excessivo de iterações, mesmo quando encontrada uma arquitetura adequada para a rede neural;
3. A característica distributiva e não-linear do processamento dificulta e até pode impedir, em alguns casos, a análise eficiente dos modelos conexionistas resultantes, durante e após o treinamento da rede.

Na tentativa de propor soluções para tais problemas, diferentes arquiteturas de redes neurais foram adotadas durante estes últimos anos. O objetivo era modelar o sistema em questão

de maneira mais transparente e permitir um maior domínio sobre a flexibilidade das estruturas resultantes. Deste esforço surgiram, por exemplo, as redes neurais modulares, com computação local de informação (cada componente do vetor de entrada é computado por somente uma parte da arquitetura). A decomposição do espaço de entradas é, para o caso da rede modular, realizada de acordo com um conhecimento prévio acerca do sistema (Ronco e Gawthop, 1995; Ronco *et al*., 1996).

Estes conceitos podem ser ainda mais estendidos, de modo a atribuir cada vez maior significado e importância ao papel individual que cada unidade de processamento, ou neurônio artificial, pode exercer dentro da estrutura conexionista. Seguindo esta tendência, é possível descrever uma rede neural como um sistema multi-agentes, capaz de construir soluções computacionais dedicadas para problemas de engenharia (Lima., 2000).

Camargos e Khater (2001) discutem a aplicação de redes neurais na identificação de sistemas não-lineares. Os autores usaram o algoritmo de retropropagação de erro, fazendo o estudo da influência dos parâmetros da rede neural através de sucessivos treinamentos. Posteriormente tiveram a necessidade de encontrar métodos numéricos de otimização que pudessem fornecer maior acurácia nos resultados, reduzindo, porém o tempo e o número de iterações.

A aplicação conjunta de transformada *wavelet* e redes neurais são apresentadas no trabalho de Likutai, (1997), o autor propõe um processo de mapeamento entrada/saída, denominado *wavenet*, tornando-se uma alternativa para aproximar funções arbitrárias. Em outras palavras, *wavenet* é uma rede neural ativada por *wavelets* especiais que possibilita a utilização de diferentes tipos de funções de ativação. Essa é uma técnica ainda em desenvolvimento que oferece possibilidades em várias linhas de pesquisa, dentre elas cita-se o uso da técnica para o projeto de neurocontroladores. No artigo clássico de Narendra e Parthasarathy, (1990) são apresentadas diversas estruturas de projeto de rede neural em problemas de identificação de processos não-lineares.

Para obtermos redes classificatórias mais confiáveis, que respondam corretamente a dados não apresentados no conjunto de treinamento, há necessidade de se tratar o sinal de entrada, retirando-se a informação não relevante. O trabalho de Moutinho e Biondi (2002) tem como intuito comparar os mais diversos métodos usados no processamento dos sinais antes da sua aplicação às redes neurais, e escolher aquele ou a combinação deles que possibilite a melhor taxa de reconhecimentos corretos.

Hunt *et al*. (1992) apresentam um levantamento das técnicas e as aplicações de topologias de redes neurais em controle de processos. Hagan e Demuth (1999) apresentam um tutorial de abordagens das aplicações de redes neurais em controle de processos.

Hornik *et al*. (1989) discute a habilidade apresentada pelas redes neurais *feedforward* simples/multi-camadas para aprender um mapa de informações a partir de dados discretos, incluindo um grande número de demonstrações e provas matemáticas para explicar a habilidade incomum destas redes para aproximações de mapas. Entretanto, o treinamento (*backpropagation*) é computacionalmente complexo e usualmente requer computação off-line.

Ho *et al*., (2001) inspirados na teoria de análise multi-resolução, transformada *wavelet* e o conceito fuzzy, propõem, a união destas técnicas para aproximação de funções não-lineares arbitrárias. O conceito fuzzy é aplicado às redes *wavelets* criando sub-redes, que proporcionam diferentes resoluções, possibilitando uma melhor aproximação de funções não-lineares.

A literatura mostra que na área de Inteligência Artificial a combinação de controle inteligente/adaptativo juntamente com as técnicas redes neurais, *wavelet*s, fuzzy e algoritmos evolutivos constituem um assunto que desperta grande interesse entre os pesquisadores e o conceito de neurocontroladores tem sido aplicado com sucesso em várias situações, pois conseguem se adequar à não-linearidades significativas Narenda e Parthasarathy, (1990).

Alguns trabalhos preliminares sobre a utilização de algoritmos de controle avançado e teoria de controle não-linear são apresentados em Coelho e Coelho (1997b, 1998b). Neste caso são tratadas aplicações em processos não lineares mono-variáveis e aspectos de educação em controle.

A grande parte dos problemas de controle no meio industrial podem ser resolvidos com controladores clássicos simples do tipo *PI* (proporcional-integral) ou *PID* (proporcional-integral-derivativo). Entretanto, algumas malhas de controle, principalmente da indústria química, podem beneficiar-se de técnicas de controle avançado, pois os métodos clássicos apresentam limitações quando aplicados a processos apresentando características complexas, tais como: não linearidades, instabilidade, atraso de transporte não-unitário e comportamento não-estacionário (Coelho *et al*., 1998a, 1999).

Apesar do controlador clássico *PID* ter um baixo nível de inteligência, devido a sua estrutura fixa, este tipo de *PID* não tem auto-reconfiguração e a habilidade de tomada de decisões é baseada no cálculo da resposta de controle em termos da realimentação do erro. Por exemplo, um controlador *PID* adaptativo apresenta um nível médio de inteligência, porque têm um mecanismo de reconfiguração dos ganhos com uma estrutura fixa e alguma capacidade de

aprendizado, em termos de identificação de sistemas, usualmente, via algoritmos recursivos do tipo mínimos quadrados.

Vários trabalhos têm mostrado a superioridade do controle adaptativo. Zhang Kun e Wen-Kai, (2002) mostram esse fato, onde um controlador fuzzy adaptativo baseado em redes *wavelet* é descrito. Para construir este controlador, um sistema *fuzzy* foi usado para treinar a rede *wavelet*. As simulações dos resultados demonstraram claramente o avanço na estabilidade e robustez do sistema em relação ao método direto de *redes wavelet*.

Segundo Åström e Wittenmark (1995), um controlador adaptativo é um controlador com parâmetros ajustáveis e apresenta um mecanismo para ajuste dos parâmetros. Dizem ainda que, um engenheiro de controle, deve conhecer os sistemas adaptativos porque eles têm propriedades úteis, que podem ser usadas como vantagens para projetar sistemas de controle com aprimoramento de desempenho e funcionalidade.

O artigo de Maitelli e Rezende (2001) apresenta uma metodologia de controle neural em um sistema de temperatura. O controlador usa uma série de métodos de treinamentos otimizados para melhorar a velocidade de convergência e o treinamento é feito totalmente em tempo real. Os resultados apresentados mostraram a capacidade do controlador neural em problemas que envolvem dificuldades como não-linearidade e mudança na sua estrutura.

Capítulo 3

Redes neurais (RN)

3.1 Introdução as Redes neurais

Redes neurais artificiais são sistemas computacionais, de implementação em hardware e/ou software, que imitam algumas funções do sistema nervoso biológico, usando um grande número de elementos básicos interconectados, chamados neurônios artificiais. Da interação destes elementos relativamente simples emerge um comportamento global coerente, e que pode ser adaptado a diversos contextos de aplicação (Lima, 2000).

Na tentativa de justificar como tal comportamento pode emergir a partir de implementações artificiais, em computadores digitais, surgiram na Inteligência Artificial (IA) dois paradigmas. O primeiro, chamado de simbolista, foi fortemente influenciado pelos estudos realizados em psicologia. Estes estudos se iniciaram com os trabalhos pioneiros de Mccarthy (1963), Minsky (1961) e Newell e Simon (1972). A abordagem simbolista defende que a solução de problemas pode ser obtida através de um processo essencialmente algorítmico. Estas idéias geraram, por exemplo, linguagens de programação simbólicas voltadas para aplicações em IA, como *Lisp* e *Prolog* (Russell e Norvig, 1995).

Em contrapartida, o segundo paradigma, chamado conexionista, defende que é impossível transformar em algoritmos, isto é, reduzir a uma seqüência finita de passos lógicos e aritméticos, as diversas tarefas que a mente humana executa com facilidade e rapidez, envolvendo reconhecimento de imagens, visão, fala, compreensão e uso da linguagem, evocação de memória por associação, etc.

O conexionismo defende que a resolução de tarefas complexas está fundamentada no modo de operação de um sistema nervoso, de natureza completamente distinta daquele concebido para as arquiteturas computacionais baseadas na computação paralela e distribuída de elementos básicos e funcionalmente simples, denominados neurônios. As propriedades provêm do comportamento coletivo destes elementos. Suas características básicas são: capacidade de aprendizado, generalização e adaptação. O princípio básico da aprendizagem está no ajuste dos pesos das conexões sinápticas em resposta a estímulos recebidos.

As motivações da pesquisa em redes neurais artificiais variam de acordo com a área de interesse. No campo biológico, podem ser mencionadas: a compreensão da estrutura do cérebro para explicar funções e comportamentos; a memória; os estudos de disfunções cerebrais, como a epilepsia e os efeitos causados por drogas. Na engenharia: a construção de sistemas de processamento de informação mais eficientes e a aplicação em controle e identificação de processos. Na ciência da computação: compreensão do processo de tratamento da informação empregado pelo cérebro, visando construir computadores e sistemas de representação eficientes e o uso de novas tecnologias para a resolução de problemas de alto grau de complexidade, como reconhecimento de imagens, recuperação de informação a partir de dados contendo ruídos e aprendizado de máquina.

3.2 Breve histórico

A idéia da utilização das *RNs* como uma ferramenta potencial à resolução de problemas não é nova, e tem origem, em 3000 a.C., com os trabalhos de Hipócrates. O pesquisador William James, *Psychology (Briefer Course*), em 1890, tratou os tópicos relativos à atividade do cérebro, mas foi Alan Turing o primeiro pesquisador a inspirar-se no cérebro para configurar um paradigma computacional, em 1936 (Nelson & Illingworth, 1991).

Os fisiologistas têm desenvolvido modelos de aprendizado humano devido aos progressos da neuroanatomia e neurofisiologia. Um modelo, que possui aplicações no aprendizado das *RN*s, foi proposto por Donald Hebb, no trabalho *The Organization of Behavior*, em 1949. Neste modelo propõe-se uma regra de aprendizado, sendo este o marco inicial de algoritmos de treinamento das *RN*s.

Um dos trabalhos pioneiros na área foi o de McCullogh e Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity*, em 1943, quando propõem um modelo para o neurônio, demonstrando que as associações de neurônios artificiais implementam qualquer função lógica finita. Este foi o primeiro sucesso teórico do conexionismo. O neurônio tinha um número finito de entradas e uma saída. As entradas eram caracterizadas pelos estados excitatório (+1) ou inibitório (-1).

A primeira onda de entusiasmo com as *RNs* surgiu com o *perceptron* de Frank Rosenblatt, publicado no estudo *The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain*, em 1958. O *perceptron* consegue aprender a classificação de padrões a partir de exemplos. Assim, pela primeira vez, tinha-se a configuração de um modelo de aprendizado e percepção com resultados concretos.

Nas décadas de 50 e 60 foram publicados trabalhos relevantes quanto a utilização de *RNs* em computadores. Nathaniel Rochester e outros pesquisadores dos laboratórios da *IBM* implementam um *software* de simulação de *RN*s, baseados no trabalho de Donald Hebb. John Von Neumann, com *The Computer and the Brain*, em 1958, sugere a imitação de funções de neurônios simples para utilização em relés de telégrafos e tubos a vácuo.

Em 1959, Bernard Widrow e Marcian Hoff desenvolvem os modelos de *RNs* denominados *ADALINE (ADAptive LINear Element*s) e *MADALINE (Multiple ADALIN*E). Esta foi a primeira aplicação de *RN*s, em problemas do mundo real, constituindo-se de filtros adaptativos para eliminação de ruídos em linhas telefônicas.

Ao entusiasmo seguiu-se uma grande crise com a descoberta das limitações do *perceptron*. No livro *Perceptron*s, Minsky e Papert (1969) mostram as deficiências do *perceptron* e provam que as *RNs* de uma camada, então em utilização, são incapazes de resolver muitos problemas simples, incluindo a execução de uma função *ou exclusivo*. Em síntese, o *perceptron* de uma única camada é incapaz de resolver problemas linearmente não-separáveis. Neste trabalho onde todas estas dificuldades são apontadas, acompanhadas da declaração de uma crença pessoal (e errônea) dos autores, de que a extensão do modelo neural seria inútil, marca o ocaso das pesquisas em *RNs* na década de 70.

Entretanto, deve ser mencionado, que Minsky e Papert (1969) apontam a solução do *ou exclusivo*, através de alguma transformação pela adição de uma camada ao *perceptron* conectando todas as entradas.

As conseqüências desta publicação foram o desencorajamento de pesquisadores e o redirecionamento dos fundos de agências governamentais. Alguns poucos cientistas, entre os

quais, Teuvo Kohonen, Stephen Grossberg e James Anderson continuaram suas pesquisas. Na década de 70 e início dos 80, alguns trabalhos isolados foram publicados em jornais.

O ressurgimento do interesse em *RNs* ocorre com a utilização do algoritmo de treinamento por retropropagação de erro, por um grupo de pesquisadores denominado *PDP (Parallel Distributed Processing)*, em 1986. Suas pesquisas resultam na extensão do *perceptron* para várias camadas de neurônios, e assim, superam as dificuldades daquele modelo (Rumelhart *et al*., 1986). A partir deste marco, observa-se uma explosão de aplicações de *RN*s, nos mais variados campos do conhecimento. Os detalhes sobre as origens das *RNs* e sua evolução cronológica são encontrados em Wasserman (1993), Haykin (1994) e Gupta e Rao (1994).

3.3 Modelo de um neurônio

Os modelos de neurônios artificiais foram desenvolvidos baseados no funcionamento dos neurônios biológicos. Vários modelos foram propostos na literatura. O modelo de McCulloch-Pitts (1943), é o mais empregado, principalmente em problemas de reconhecimento de padrão.

Uma *RN* constitui-se de modelos de processamento paralelo e distribuído. A unidade básica de uma *RN* é o neurônio. Os neurônios são capazes de modificar o seu comportamento após a realização de um treinamento dinâmico. Os neurônios são interconectados por conexões (sinapses) com valores variáveis denominados pesos.

Cada neurônio pode ter várias entradas, porém somente uma saída. Cada saída pode ser utilizada como entrada a vários neurônios (através de ramificações). Assim, como cada neurônio pode receber várias entradas procedentes de outros neurônios.

Em essência, um conjunto de entradas (vetor x) é aplicado a um neurônio artificial, cada entrada é multiplicada por um peso correspondente (vetor w), que são somados para determinar o nível de ativação do neurônio. Em outras palavras, um neurônio corresponde a uma soma ponderada de entradas, soma esta aplicada a uma função de ativação não-linear (ou mesmo linear), contínua e diferenciável, conforme apresentado na figura 3.1.

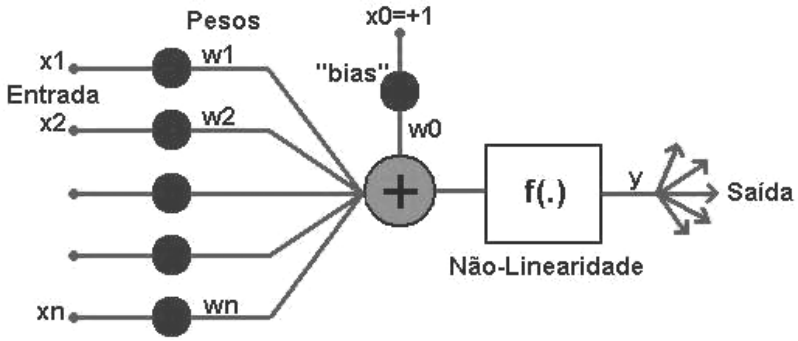


Figura 3.1: Neurônio de McCulloch-Pitts

A estrutura do neurônio apresentada na fig. 3.1 pode ser expressa na seguinte forma:

$$y = f\left(\sum_{i=1}^{n} w_i x_i\right) + w_0 x_0 \tag{3.1}$$

o qual *f* é a função de ativação, x_i o conjunto de entradas e w_i são os pesos do neurônio.

As funções de ativação mais empregadas são: relé, lógica *threshold* e sigmóide, conforme é apresentado na seção seguinte. O neurônio de McCulloch-Pitts pode conter também um peso bias w_0 alimentado por uma constante $x_0 = +1$ que desempenha o controle do nível de saída do neurônio.

3.4 Funções de Ativação

A arquitetura das redes neurais descritas até aqui empregam a função de ativação $f(\cdot)$ condicionando a saída de um neurônio ao nível de atividade em sua entrada. As funções de ativação convertem um domínio infinito para um outro domínio finito, delimitado por uma faixa de valores predeterminados.

3.4.1 Função Linear

A função de ativação linear produz uma saída linear a partir de uma entrada *x*, descrita pela seguinte equação.

$$f(x) = kx \tag{3.2}$$

sendo *k* um escalar positivo.

3.4.2 Função Degrau

A função de ativação tipo degrau, produz somente dois valores β e δ, escalares positivos. Se o valor do sinal de entrada *x* é igual ou maior que um dado valor de referência x_k, então a função de ativação produz o valor β, caso contrário produz o valor $-\delta$. Matematicamente esta função pode ser descrita como:

$$f(x) = \begin{cases} \beta & \text{se } x \geq x_k \\ -\delta & \text{se } x < x_k \end{cases} \tag{3.3}$$

É comum nas redes neurais do tipo Hopfield, usar função degrau bipolar, ou seja,

$$f(x) = \begin{cases} 1 & \text{se } x \geq x_k \\ -1 & \text{demais casos} \end{cases} \tag{3.4}$$

3.4.3 Função Rampa

A função de ativação tipo rampa é uma combinação das funções degrau e linear. A função rampa produz valores que variam linearmente entre dois pontos de saturação simetricamente dispostos em torno da origem; fora dessa faixa, produz valores constantes.

$$f(x) = \begin{cases} \rho & \text{se } x \geq \rho \\ x & \text{se } |x| < \rho \\ -\rho & \text{se } x \leq -\rho \end{cases} \tag{3.5}$$

sendo ρ o valor de saturação da função. Os pontos $x = \rho$ e $x = -\rho$, são descontinuidades na função f.

3.4.4 Função Sigmoidal

A função de ativação sigmoidal é uma versão contínua da função rampa. A função sigmóide é uma função monotônica, não decrescente, tem um domínio finito e produz respostas não-lineares dentro de uma faixa predefinida. A função sigmoidal mais comum é matematicamente definida como:

$$f(x) = \frac{1}{1+e^{-\alpha x}} \tag{3.6}$$

sendo que $\alpha > 0$ (geralmente $\alpha = 1$) promove um valor da saída na faixa ente 0 e 1. Ainda, dois outros tipos de funções alternativas a função sigmoidal são encontrados, a função tangente hiperbólica, que produz valores na faixa de +1 a -1,

$$f(x) = \tanh(\gamma x) = \frac{1-e^{-2\gamma x}}{1+e^{-2\gamma x}}, \ \gamma > 0 \tag{3.7}$$

e a função das razões quadráticas aumentada (Racional), que produz valores na faixa entre 0 e 1.

$$f(x) = \begin{cases} \dfrac{x^2}{1+x^2} & se \ x > 0 \\ 0 & \text{se } x \leq 0 \end{cases} \tag{3.8}$$

3.4.5 Função Gaussiana

A função de ativação gaussiana é simétrica em relação a origem e requer um valor predefinido de variância $\sigma > 0$.

A maioria das aplicações que utilizam esta função de ativação o faz em conjunto com as ligações, ou sinapses do tipo dual, ou média-variância.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{3.9}$$

sendo μ o valor médio da entrada, e σ é a variância previamente definida.

A tabela 3.1 resume os tipos básicos de funções de ativação.

Tabela 3.1: Funções de ativação.

NOMES	**DEFINIÇÃO**
Linear	$f(x)=kx$
Degrau (usualmente: $\beta=1, \delta=0, x_k=0$)	$f(x)=\begin{cases}\beta \ se \ x \geq x_k \\ \delta \ se \ x < x_k\end{cases}$
Rampa	$f(x)=\begin{cases}\rho \ se \ x \geq \rho, & \rho>0 \\ x \ se \ \lvert x\rvert < \rho \\ -\rho \ se \ x \leq -\rho\end{cases}$
Sigmóide	$f(x)=\frac{1}{1+e^{-\alpha x}}, \ \alpha>0$
Tangente hiperbólica	$f(x)=\tanh(\gamma x)=\frac{1-e^{-2\gamma x}}{1+e^{-2\gamma x}}, \ \gamma>0$
Racional	$f(x)=\begin{cases}\frac{x^2}{1+x^2} & se \ x>0 \\ 0 & \text{se } x \leq 0\end{cases}$
Gaussiana	$f(x)=\frac{1}{\sqrt{2\pi\sigma}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$

3.5 Classificação das Redes Neurais

A classificação das *RNs* é importante no contexto de comparar estruturas e melhor diferenciá-las. Um esquema de classificação simples (Redgers e Aleksander, 1995) pode ser realizado com algumas informações que descrevem uma *RN*, tais como:

i. *Topologia*: como as funções são interconectadas;
ii. *Arquitetura*: o tipo e a utilização da *RN*;
iii. *Modelo do neurônio*: o que as funções são;
iv. *Algoritmo de treinamento*: como os parâmetros são configurados;
v. *Escalonamento da operação*: a sincronização das interações entre as funções.

A topologia de uma *RN* é um conjunto das conexões entre as entradas, as saídas e os nodos (nós). Todas topologias são decorrentes de subconjuntos da topologia totalmente conectada. As *RNs* podem ser classificadas, quanto à conectividade interna, em *feedforward* e *feedback* (recorrentes), conforme apresentado na figura 3.2. Na literatura, as diversas estruturas de *RNs* são mencionadas em aplicação na área de identificação e controle de processos. As *RNs perceptron* multicamadas, Hopfield, função de base (radial, *wavelet* e polinomial) e híbridas com técnicas de controle adaptativo têm polarizado a maioria das aplicações.

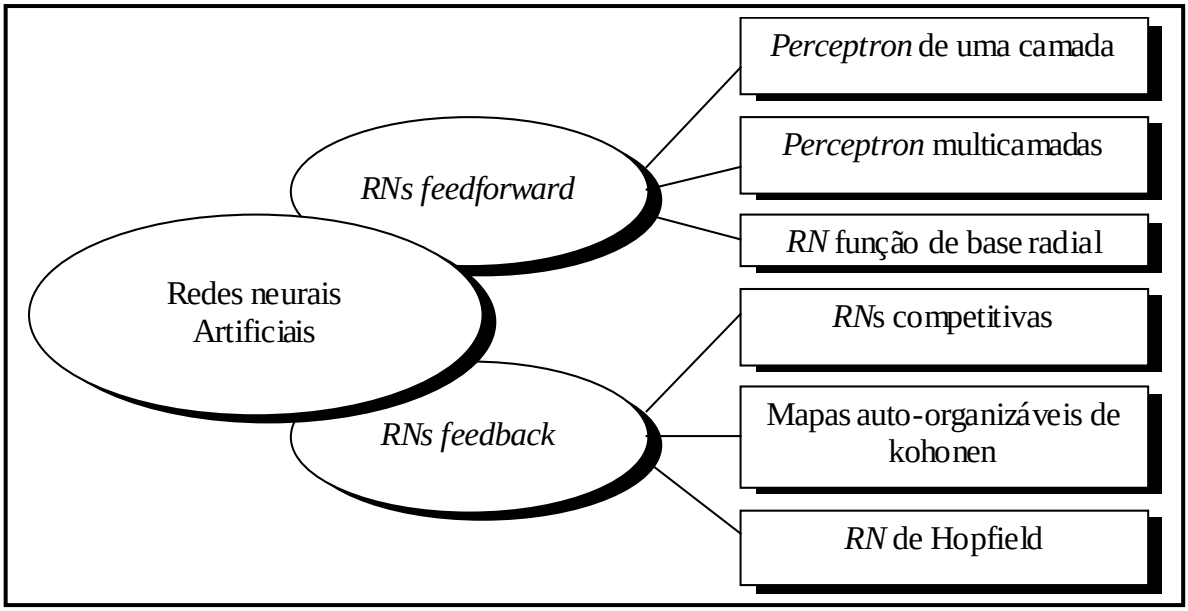


Figura 3.2: Taxionomia das *RNs feedforward e feedback.*

Quanto aos diferentes tipos de conexões, as *RNs* utilizam conexões entre as camadas e entre as conexões na mesma camada. As *RNs* podem ser classificadas sob outro ponto de vista, pela diferenciação entre as que generalizam globalmente e localmente. A generalização da *RN* é global se um (ou mais) dos seus parâmetros adaptativos (os pesos) podem afetar a saída da *RN*, para um (ou todo) ponto no espaço de entrada. O *perceptron* multicamadas é um exemplo de *RN* que generaliza globalmente. A generalização local ocorre em *RNs* quando somente poucos pesos afetam a resposta da saída da *RN*, para os "pontos locais" no espaço de entrada. As *RNs* função de base radial e *B-Splines* (HARRIS *et al.*, 1996) são exemplos de *RN*s, com generalização local, em que a interferência no aprendizado é minimizada e o aprendizado é relativamente rápido, devido ao número mínimo de pesos ajustados a cada par de treinamento apresentado à *RN* (HARRIS *et al*., 1993).

3.6 Por que utilizar as *RN*s?

As características que tornam as *RNs* atraentes são (HAYKIN, 1994):

(i) *Habilidade de tratar sistemas não-lineare*s: relevante nas aplicações em identificação de sistemas dinâmicos e classificação de padrões;

(ii) *Tolerância à falhas*: o conhecimento é distribuído pela *RN*, mais que em uma simples localização de memória. Uma parte das conexões pode estar inoperante, sem mudanças significativas no desempenho de toda a *RN*;

(iii) *Adaptabilidade*: capacidade da *RN* em se auto-ajustar. Os aspectos de aprendizado, auto-organização, generalização e treinamento estão intrinsecamente ligados a esta característica;

(iv) *Aprendizado*: uma *RN* pode modificar seu comportamento em resposta ao ambiente. Quando é apresentado um conjunto de entradas, as *RNs* se ajustam para gerar as respostas apropriadas;

(v) *Generalização*: consiste na *RN* mapear entradas similares em saídas similares;

(vi) *Treinamento*: é a forma pela qual a *RN* aprende;

(vii) *Processamento paralelo*: as *RNs* são estruturalmente paralelas. A seqüência de processamento das *RNs* é realizada em paralelo e simultaneamente;

(viii) *Abstração*: muitas *RNs* são capazes de abstrair a essência de um conjunto de entradas;

3.7 Tipos de Treinamento de Redes Neurais

A propriedade mais importante das redes neurais é a habilidade de aprender e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicados a seus pesos que correspondem ao treinamento. Denomina-se algoritmo de treinamento a um conjunto de regras bem definidas para a solução de um problema de treinamento. Existem muitos tipos de algoritmos de treinamento específicos para determinados modelos de redes neurais. Estes algoritmos diferem entre si, principalmente, pelo modo como os pesos são modificados.

Outro fator importante é a maneira pela qual uma rede neural se relaciona com o ambiente. Nesse contexto existem, basicamente, os seguintes paradigmas de treinamento:

3.7.1 Treinamento Supervisionado

A disponibilidade de arquiteturas de rede neurais de importância prática está associada à existência de algoritmos de otimização eficientes para realizar os ajuste dos parâmetros no processo de aproximação resultante, denominado treinamento supervisionado.

O algoritmo de aprendizagem possui dois momentos distintos: em primeiro lugar, quando um padrão (ou uma seqüência de padrões de entrada) é apresentado à rede, o fluxo é alimentado para frente, isto é, propagado adiante até a camada de saída *(forward)*. Após este processo, a saída obtida é comparada com a saída desejada, em caso de existir erro, isto é, se a saída desejada não corresponder à obtida, então é feita uma correção nos pesos das conexões sinápticas (Widrow e Lehr, 1990), ajustando-os na direção oposta à do gradiente do erro instantâneo (este é o instante da aprendizagem).

O ajuste é proporcional ao gradiente, segundo um fator de proporcionalidade denominado *taxa de aprendizagem*, que pode ser fixa ou variável.

O ajuste dos pesos é feito de trás para frente, isto é, da última camada em direção à camada de entrada *(backward)*. Este procedimento caracteriza o algoritmo de retropropagação, backpropagation (BP), proposto por Werbos em 1974. Estes dois procedimentos são repetidos até que haja convergência do erro para um valor satisfatório. Neste caso, diz-se que a rede aprendeu o mapeamento, ou que está treinada.

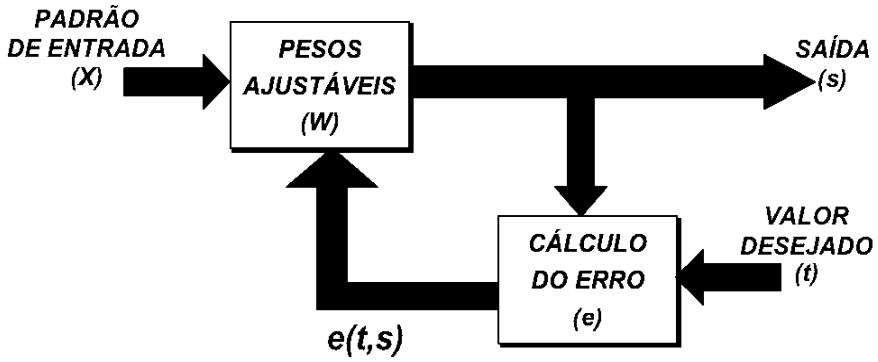


Figura 3.3: Treinamento supervisionado.

O treinamento de uma rede neural usa, em essência, a regra da cadeia, do cálculo diferencial multivariável, para calcular o modo segundo o qual os pesos serão ajustados. Seu objetivo, normalmente, é reduzir o erro quadrático da aproximação feita pela rede, e para tanto emprega o método do passo na direção oposta à do gradiente da superfície de erros, que é função de todos os pesos da rede.

3.7.2 Treinamento Não-supervisionado

É baseado apenas nos estímulos recebidos pela rede neural. Basicamente, a rede deve aprender a categorizar os estímulos. Neste caso, não há um professor ou crítico inspecionando o processo de aprendizado (Kohonen, 1997). Em outras palavras, não existe um exemplo específico da função a ser aprendida pela rede. Ao contrário, seus parâmetros livres são ajustados no sentido de extrair alguma tendência estatística presente nos dados de entrada.

Uma vez que a rede concluiu um processo de auto-organização baseado nos dados de entrada, ela explora a representação interna gerada para discriminar características presentes na entrada.

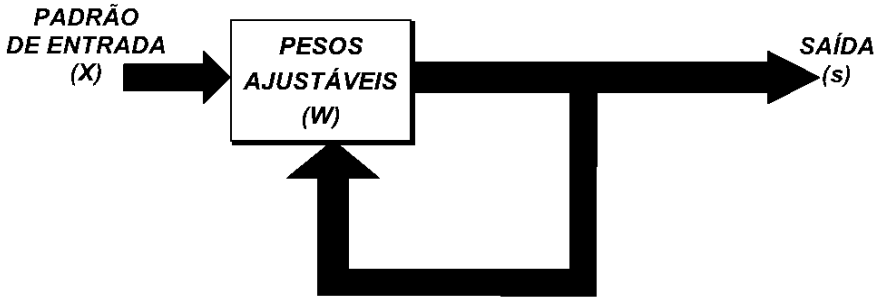


Figura 3.4: Treinamento não-supervisionado.

3.8 Modelos de Redes Neurais

Como abordado anteriormente, uma rede neural é constituída de um arranjo de neurônios funcionando em paralelo e dispostos em camadas. Deste modo, nesta seção, serão abordados os modelos da rede neural e os mecanismos de treinamento.

Em 1959, Bernard Widrow e Marcian Hoff desenvolvem os modelos de *RN*s denominados *ADALINE (ADAptive LINear Elements) e MADALINE (Multiple ADALIN*E). Esta foi a primeira aplicação de *RN*s, em problemas do mundo real, constituindo-se de filtros adaptativos para eliminação de ruídos em linhas telefônicas.

3.8.1 O Modelo ADALINE

O modelo de neurônio ADALINE *(ADAptive LInear Element*) de Widrow e Lehr (1990) é mostrado na figura 3.5.

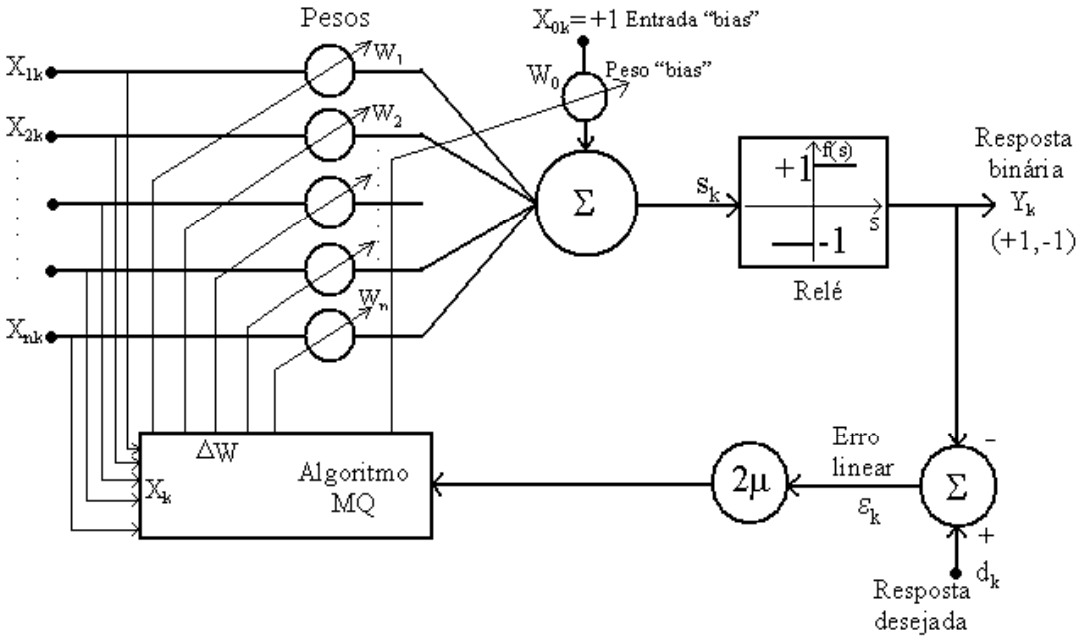


Figura 3.5: Rede neural ADALINE.

A saída é uma combinação linear das entradas. Na implementação discreta, estes elementos recebem, no instante k, um vetor padrão de entrada:

$$x_k = [x_{1k}\ x_{2k}\ \ldots\ x_{nk}]^T \tag{3.10}$$

e uma resposta desejada d_k. Os componentes do vetor padrão de entrada são ponderados por um conjunto de coeficientes, ou seja, pelo vetor de pesos:

$$w = [w_1\ w_2\ \ldots\ w_n]^T \tag{3.11}$$

A soma das entradas ponderadas é, então, avaliada (calculada), produzindo uma combinação linear correspondente ao produto interno:

$$s_k = < x_k\ ,\ w > \tag{3.12}$$

Os componentes de x_k podem ser valores reais ou binários. Porém, os pesos são valores essencialmente reais.

Durante o processo de treinamento, os padrões de entrada e de respostas desejadas correspondentes são apresentados à rede neural. Um algoritmo de adaptação ajusta, automaticamente, os pesos de tal forma que as saídas fiquem próximas dos valores desejados.

A rede neural ADALINE consiste no mecanismo de adaptação linear em série com relé (função não-linear), que é empregada para produzir uma saída binária +1:

$$y_k = \text{sgn}(s_k) \tag{3.13}$$

sendo, sgn(.) = função sinal.

3.8.2 O Modelo da Rede Neural MADALINE

A rede neural MADALINE *(Multi-ADALIN*E) é constituída por vários elementos ADALINE. Contudo, o processo de treinamento é bem mais complexo, se comparado ao ADALINE. Deve-se ressaltar que o treinamento de redes multineurônios/multicamadas é bastante complexo. O treinamento (processo adaptativo) é um procedimento que emprega algum método de otimização para o ajuste de pesos. Sabe-se que os métodos de otimização determinísticos, via de regra, empregam derivadas de funções. Deste modo, o relé, que é uma função não-diferenciável, é inadequado para uso em redes MADALINE. Deve-se, portanto, buscar outras alternativas de funções para uso neste tipo de redes neurais, *e.g.*, as funções sigmoidais que são a base do desenvolvimento de redes neurais com treinamento via algoritmo backpropagation. Este assunto será abordado na seqüência.

3.9 Algoritmo de treinamento backpropagation (BP)

O algoritmo *BP* é baseado na regra de aprendizado pela correção do erro, que pode ser vista como uma generalização do algoritmo da filtragem adaptativa ou mesmo um caso especial do algoritmo dos mínimos quadrados. O algoritmo *BP* é um método iterativo de gradiente

projetado para minimizar a soma do erro quadrático, entre a saída atual e a saída desejada. Para minimizar esta função objetivo, o algoritmo *BP* utiliza uma técnica de busca baseado em gradiente, a regra delta generalizada.

É relevante mencionar que o algoritmo *BP* não tem, aparentemente, nenhuma relação com sistemas de aprendizado biológico. Não parece haver nenhum indício de retropropagação do erro no sistema nervoso animal. Entretanto, se um algoritmo de aprendizado revela resultados promissores, a inexistência de um paralelo natural (plausibilidade biológica) conhecido, não costuma ser levada em consideração em aplicações de controle e em outras aplicações industriais de engenharia.

O *BP* consiste basicamente de duas fases através das diferentes camadas da *RN*, que são: a fase *forward* e a fase *backward*. Na fase *forward*, um padrão de atividade (vetor entrada) é aplicado a *RN* e o seu efeito é propagado, camada por camada. Finalizando, um conjunto de saídas produz a resposta atual da *RN*. Durante a fase *forward*, os pesos da *RN* são fixos (Haykin, 1994). Na fase *backward*, os pesos são ajustados de acordo com uma regra de correção do erro. Especificamente, a saída atual da *RN* é subtraída da saída desejada para o cálculo do erro. O erro é propagado para trás *(backward*) em direção às entradas, através da *RN*, atualizando os pesos. Os pesos são ajustados de forma que a saída atual da *RN* aproxime-se da saída desejada. As etapas que regem o *BP* são sintetizadas por:

(i) inicializar aleatoriamente os pesos das conexões da *RN*;

(ii) aplicar o conjunto de treinamento constituído das entradas e das saídas desejadas (dados do processo) à *RN*;

(iii) calcular a(s) saída(s) da *R*N, propagando as saídas dos neurônios de cada camada da *RN* para os neurônios da próxima camada, passando pela função de ativação dos neurônios (passo *forwar*d);

(iv) especificar a saída desejada e calcular os erros das camadas;

(v) ajustar os pesos pelos "gradientes locais" da *RN* (passo *backward*) utilizando o algoritmo recursivo, começando das unidades de saída e propagando-se em direção à primeira camada oculta, utilizando as equações:

$$w_i^{novo} = w_i^{velho} + \Delta w_i$$
$$\Delta w_i = \eta\left(y_j^{desejado} - y_j^{obtido}\right)x_{i.j}$$

no qual w_i é o peso, η é o ganho que representa o coeficiente de aprendizado, Δw_i é o termo de erro;

(vi) repetir os passos (iv) e (v), enquanto a função do erro da saída não apresentar um valor de tolerância aceitável (critério de parada do treinamento);

(vii) utilizar conjunto de teste para analisar o desempenho do treinamento. Após a conclusão do treinamento, a *RN* atua como uma *RN feedforwar*d, pois os pesos das conexões permanecem constantes, exceto no caso da necessidade de um novo treinamento com um novo conjunto de dados. A figura 3.6 ilustra um exemplo de uma *RN multi-camadas* com treinamento *BP*. Os detalhes da formulação matemática da regra delta generalizada e do algoritmo *BP* podem ser encontrados em Oliveira (1998).

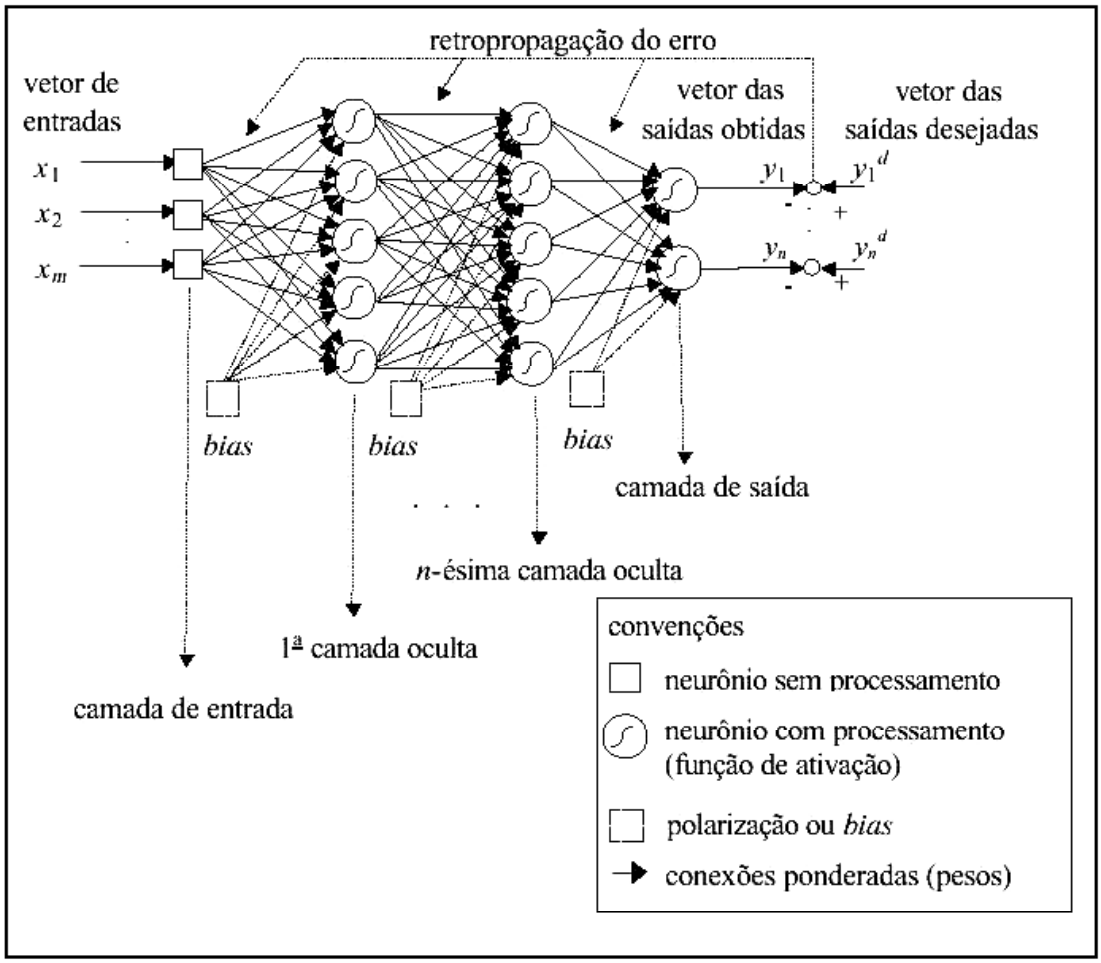


Figura 3.6: *RN multi-camadas* com treinamento *BP*.

O *BP* é uma das mais populares técnicas de aprendizado supervisionado de *RN*s. É uma metodologia simples de implementar, e sua capacidade de generalização é atraente para um grande espectro de aplicações em problemas de reconhecimento e classificação de padrões. Apesar do sucesso aparente do *BP*, existem alguns aspectos que fazem com que este algoritmo apresente algumas deficiências, tais como:

(i) Justificar suas respostas é praticamente impossível;
(ii) Custo computacional significativo;
(iii) e baixa velocidade de aprendizado.

O aumento da velocidade de aprendizado, através de procedimentos heurísticos, pela modificação adaptativa dos parâmetros e prescrição de uma seleção específica dos pesos iniciais de uma *RN* multi-camadas são alternativas viáveis.

3.10 *RNs* em identificação de processos

As metodologias de controle neural mais simples são centradas em procedimentos de inversão de processos, onde a inversa da dinâmica do processo é utilizada para o controle em malha aberta. A *RN* é muito utilizada para os propósitos de identificação e controle de processos, mas não existe garantia que a *RN* seja bem sucedida em controlar um determinado processo. O modelo matemático do processo, obtido através da *RN*, tem representação estrutural, ou seja, modelo caixa-preta. Esta característica é proibitiva quanto à análise das propriedades do modelo aprendidas.

A capacidade de identificação não-linear das *RNs* pode ser explorada para aprimorar as metodologias de controle preditivo, baseadas em modelo, pois um modelo preciso do processo é parte essencial na aplicação eficiente desta metodologia (Ronco e Gawthrop, 1997).

A literatura é extensa quanto a estudos relativos a aspectos de modelagem e identificação de processos através de *RN*s. Os estudos incluem a teoria e as aplicações em identificação caixa-preta não-paramétrica, identificação paramétrica, identificação da inversa do processo, tanto para

processos *SISO* quanto para *MIMO*, principalmente não-lineares (Sjöberg, 1995, Hunt *et al*., 1992).

3.10.1 Seleção do sinal de treinamento

Para os sistemas lineares, existe uma base teórica para a seleção de sinais de treinamento. Entretanto, os sinais de ruído branco são freqüentemente utilizados (Ljung, 1987). Em identificação de sistemas não-lineares, o sinal ruído branco também é utilizado, porém não existe base teórica para esta escolha.

Segundo Van Can *et al.* (1995), a escolha do sinal de treinamento influencia, severamente, o desempenho da *RN*. Assim, o conjunto de dados de treinamento deve conter o máximo de informação relevante sobre a dinâmica do processo. Devido à falta de base teórica para a seleção de um sinal ótimo de treinamento, nenhuma especificação geral pode ser dada. Além disso, espera-se que o sinal de treinamento contenha todas as freqüências e os espectros de amplitude relevantes. Dependendo das propriedades estatísticas do estimador, existe a necessidade de obtenção dos dados de treinamento agrupados em um intervalo específico de operação.

3.10.2 Validação e seleção da configuração das *RNs*

O vetor de entrada de uma *RN* pode conter um ou mais atrasos de tempo das entradas e saídas do processo, dependendo da ordem do mesmo. Os processos práticos e correlações não-lineares apresentam ordem desconhecida.

No caso das *RNs feedforward* e *WNN* necessita-se definir o número de neurônios, na camada oculta, para a definição da configuração do modelo neural. Em geral, entretanto, encontrar a configuração de *RN* adequada significa obter um número considerável de configurações a serem treinadas e testadas na análise do potencial de previsão. A adoção de algoritmos de treinamento eficientes diminui, razoavelmente, o tempo de treinamento e de teste dos modelos neurais.

Uma outra observação relevante é a necessidade de especificação de um critério para a análise de desempenho das configurações neurais. Isto é, usualmente, realizado pelo cálculo de uma função dos erros, usualmente, quadrática.

3.11 Aplicações

As aplicações das redes neurais cada vez mais vem crescendo na abordagem de sistemas complexos em que não há modelos disponíveis ou são muito difíceis de serem obtidos, e também nas análises em tempo real, onde o tempo de processamento é, basicamente, gasto para a execução do treinamento (o diagnóstico é instantâneo). Dessa forma a tabela 3.2 traz as principais áreas de aplicações de redes neurais.

Tabela 3.2: Aplicações para Redes Neurais

Área	Principais Aplicações
Biologia	◆ Genótipo e fenótipo ◆ Obtenção de modelos para identificação de odores
Medicina	◆ Diagnósticos ◆ Análise de imagens ◆ Controle hospitalar ◆ Controle de doenças ◆ Determinação de modelos de doenças
Odontologia	◆ Análise de imagens ◆ Diagnósticos
Sistemas Agrícolas e Pecuária	◆ Comportamento de plantas em função do solo, clima e nutrientes ◆ Otimização de recursos ◆ Previsões ◆ Auxílio no tratamento e criação de animais
Astronomia	◆ Pesquisa e reconhecimento de corpos celestes
Engenharias e áreas afins	◆ Reconhecimento de padrões ◆ Análise em tempo real ◆ Robótica, Controle, Otimização. ◆ Processamento de sinais ◆ Reconhecimento de voz ◆ Visão artificial
Física	◆ Auxílio na pesquisa de novos materiais (polímeros, supercondutores).

	◆ Análise de sistemas complexos
Música	◆ Composição musical ◆ Auxílio no estudo sobre compositores
Segurança	◆ Detecção de fraudes ◆ Sistemas militares (defesa, estratégia, míssil inteligente, sonar e radar (análise de sinais)).

Capítulo 4

Transformada Wavelet

4.1 Introdução a Transformada *Wavelet* (TW)

O termo *wavelet* significa “ondinha”. Essa ondinha para ser admitida como *wavelet* deve apresentar alguma oscilação e cair rapidamente a zero. Esta propriedade, de admissibilidade, é a mesma que se coloca sobre uma função para que ela seja transformável via *wavelet*.

Conjuntos de *wavelets* são empregados para aproximar um sinal e o objetivo é encontrar uma coleção de *wavelets*-filhas (construídas por dilatação/compressão e translações da *wavelet* original) que melhor se ajuste ao sinal analisado. Uma *wavelet* comprimida é adequada para captar mudanças rápidas (altas freqüências no sinal sob análise). Uma *wavelet* alongada é adequada para captar mudanças lentas (baixas freqüências no sinal sob análise).

A transformada *wavelet* é uma operação que integra, de menos a mais infinito em princípio, o produto de uma função por algum núcleo. O núcleo é chamado *wavelet-mãe* e suas modificações são as *filhas*.

4.2 Histórico

Embora desde o começo do século 20 a literatura registre tratamentos matemáticos semelhantes à TW (Daubechies, 1992 e Meyer, 1993), foi no final da década de 70 que a mesma passou a ter uma identidade própria.

Nessa ocasião, o francês Jean Morlet, trabalhando para a companhia de petróleo Elf Acquitaine, propôs uma modificação na Transformada de Fourier (TF), para melhor tratar sinais geofísicos. Por falta de embasamento matemático, Morlet inicialmente encontrou muitos opositores. Esses, como lembra Daubechies (1996), teciam críticas do seguinte tipo: "*Se isso estivesse correto, já estaria nos livros de matemática. Como não está, provavelmente é inútil*". No entanto, alguns anos depois, as "*wavelets*" de Morlet atraíram a atenção de Yves Meyer, um matemático que ajudou a enriquecer e amadurecer a nova teoria, encontrando paralelos surpreendentes com diversos outros campos da matemática, antes estudados separadamente.

Logo em seguida, Stéphane Mallat, um estudante de processamento de imagens, desenvolveu um algoritmo para calcular a TW de forma computacionalmente eficiente, abrindo então as portas à comunidade de processamento de sinais. Como se vê, desde seu início, a teoria de *wavelet*s se mostrou interdisciplinar, o que em parte explica a grande popularidade adquirida nos últimos anos e a opinião de alguns autores (Abbate e Koay, 1997) que a reputam como o "*evento matemático de maior relevância na década de 80*".

4.3 Aspectos teóricos e limitações que levaram ao desenvolvimento da TW

Os dados sísmicos estudados por Morlet exibiam conteúdos de freqüência que mudavam rapidamente ao longo do tempo, para os quais TF não era adequada como ferramenta de análise.

De fato, a TF não permite uma análise local do conteúdo de freqüência do sinal. Eventos que venham a ocorrer em intervalos de tempos distintos e mesmo bastante remoto contribuem de maneira global para a transformada, afetando a representação como um todo. A equação abaixo ilustra a TF de um sinal contínuo no tempo *x(t)*. É importante notar que a transformada está baseada na integração de toda a função para o cálculo de cada freqüência.

$$X(f) = \int_{-\infty}^{\infty} x(t) * e^{-2j\pi ft} dt \qquad (4.1)$$

Em sua versão discreta a TF de um sinal *x(t)* com *N* pontos é calculada como:

$$X_k(f) = \sum_{t=0}^{N-1} x_k(t) * e^{-2j\pi ft} \tag{4.2}$$

Uma das limitações da TF encontra-se no fato de que ela não permite analisar em separado, diferentes trechos do sinal.

Assim, caso um trecho seja extremamente ruidoso ou contenha pontos anômalos, tais como em sinais de impactos, choques e inicio e fim de eventos, o processamento de todo o sinal é comprometido.

A fim de resolver esse problema, foi introduzida a Transformada de Fourier Janelada (TFJ) ou a Transformada de Fourier de Curta Duração (STFT), que consiste em dividir o sinal em regiões e aplicar a TF a cada uma delas. Matematicamente, a TFJ de um sinal $x_k(t)$ é dada por (Qian e Chen, 1996):

$$X_k(f,b) = \sum_{t=0}^{N-1} x_k(t) * w(t-b) * e^{-2j\pi f} \tag{4.3}$$

sendo, *w(t)* uma função de janelamento, responsável pela delimitação do trecho que está sendo considerado no sinal. A posição da janela dentro do sinal é dada pelo parâmetro *b*. A TFJ transforma um sinal no domínio do tempo em uma função bidimensional no domínio tempo-freqüência (*f*, *b*), a qual pode ser representada através de um espectrograma fig. 4.1.

Figura 4.1: Espectrograma que representa TFJ ou STFT.

A título de ilustração, a figura 4.2 mostra um espectro de um sinal dividido em 8 janelas retangulares.

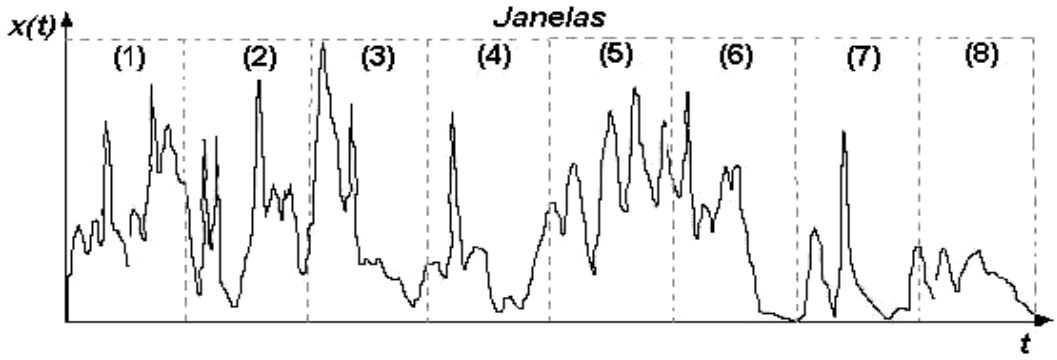


Figura 4.2: Espectro dividido em 8 regiões por janelas retangulares.

No exemplo da figura 4.2, a função *w(t)* realiza cortes abruptos, dividindo o sinal em janelas bem delimitadas. Contudo, esse procedimento causa um problema conhecido como "efeito de borda" ou "fenômeno de Gibbs" (Gottlieb e Shu, 1997), que pode ser minimizado usando-se janelas com decaimento suave, tais como a gaussiana:

$$w(t) = e^{\frac{-t^2}{2}} \tag{4.4}$$

Além do efeito de borda, outro problema relevante é a escolha da largura para a função de janelamento. Suponha, por exemplo, que sejam usadas janelas retangulares com L variáveis t. Então, cada janela dá origem a L coeficientes *Fourier* e, portanto, a preservação do número total de N variáveis requer o uso de N/L janelas. Note-se que, aumentando L, melhora-se a análise de cada janela (maior número de coeficientes *Fourier*), mas perde-se resolução espacial (menor número de janelas). Esse problema é conhecido na área de processamento de sinais como Princípio da Incerteza de Heinsenberg (Cerqueira *et al.*, 2000).

Para ilustrar a dificuldade em se escolher uma janela de largura conveniente, pode-se notar na figura 4.2 que a região "2" precisaria ser dividida em janelas mais estreitas que a região "8", ou seja, uma janela larga permite uma boa resolução no domínio da freqüência, mas resolução pobre no domínio do tempo, e vice versa. Então, com a TFJ não é possível obter uma boa resolução no domínio do tempo e da freqüência simultaneamente.

Para resolver ou contornar os problemas da TFJ, ou STFT, foi introduzida a TW, que utiliza janelas sem cortes abruptos, à semelhança da gaussiana, mas que, ao contrário desta, permite uma fácil reconstrução do sinal a partir dos coeficientes da transformada. Além disso, a TW utiliza janelas de largura variável, que podem assim ser mais bem ajustada às características de cada trecho do sinal. Isso permite que eventos de alta freqüência possam ser localizados com uma maior resolução temporal, ao passo que componentes de baixa freqüência possam ser analisados com maior resolução no domínio de Fourier. Na figura 4.3 a seguir é possível analisar no plano tempo-freqüência como cada transformada trata o sinal.

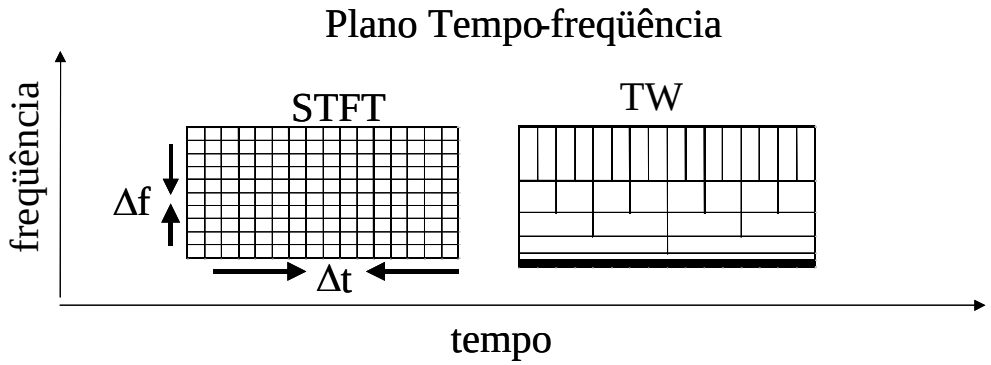


Figura 4.3: Análise no Plano Tempo-Freqüência das Transformadas.

4.4 Formulação Matemática

A TW de um sinal *x(t)* é expressa como:

$$Wx(a,b) = \int_{-\infty}^{+\infty} x(t)\psi_{a,b}(t)dt \tag{4.5}$$

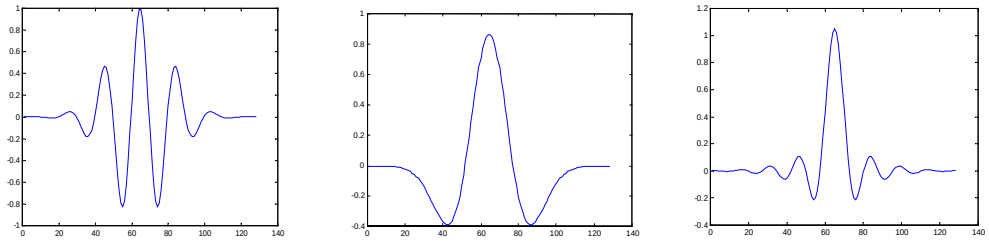
que, aproximando a integral por um somatório, torna-se

$$Wx(a,b) = \sum_{t=0}^{N-1} x(t)\psi_{a,b}(t) \tag{4.6}$$

para um sinal discreto com *N* pontos. A função $\psi_{a,b}(t)$, chamada "*wavelet*", é derivada de uma função $\psi(t)$ através da seguinte transformação:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{4.7}$$

Há uma ampla gama de possíveis escolhas para a função $\psi(t)$, denominada "*wavelet-mãe*". Algumas alternativas estão apresentadas na figura 4.4.



Morlet Mexican Hat Meyer

Figura 4.4: Exemplos de *Wavelet*s.

Comparando-se as equações 4.3 e 4.6, vê-se que ambas realizam o produto do sinal por funções que apresentam oscilações dentro de uma janela. A diferença encontra-se na natureza dos parâmetros das transformadas. Na TFJ, *b* e *w* representam, respectivamente, posição e freqüência, estando fixada a largura da janela. Na TW, *b* também representa posição (ou "translação" da *wavelet*), mas *a* (chamado parâmetro de "escala") está associado à largura da janela. Como se pode ver na figura 4.5, em tracejado, encontra-se a *wavelet-mãe* e, em linha cheia, a *wavelet* com escala a = 2.

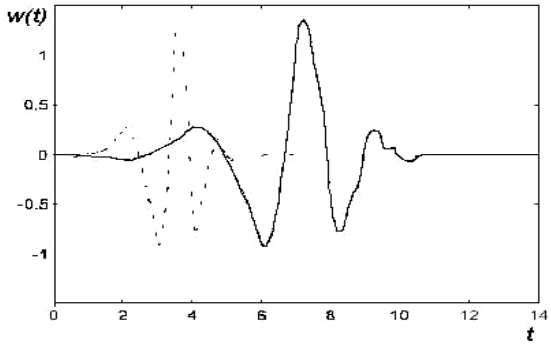


Figura 4.5: Efeito de modificações na escala para a *wavelet* Daubechies.

4.5 Tipos de Filtros *Wavelets*

4.5.1 *Wavelet* Morlet

A função *wavelet*-mãe de Morlet é o produto da exponencial complexa pela janela gaussiana:

$$h(t) = \exp(j\omega_0 t)\exp(-t^2/2) \tag{4.8}$$

A *wavelet* cosseno-gaussiano, parte real de *h*, é uma função real par, cuja transformada de Fourier deslocada, respectivamente para ω_0 e $-\omega_0$ é uma função real par e positiva:

$$H(\omega) = \sqrt{2\pi}\left(\exp\left[-(\omega-\omega_0)^2/2\right] + \exp\left[-(\omega+\omega_0)^2/2\right]\right) \tag{4.9}$$

A figura 4.6 mostra os gráficos da *wavelet* Morlet e seu espectro de Fourier, com $\omega_0 = 4$.

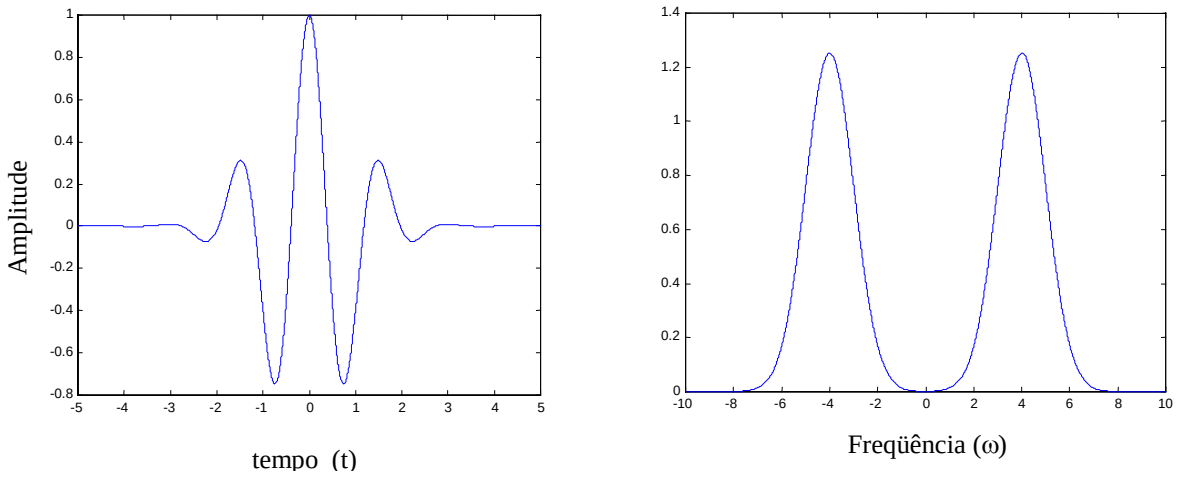


Figura 4.6: *Wavelet* cos-gaussiano de Morlet h(t) e sua transformada de Fourier H(ω).

As *wavelets* Morlet não satisfazem, em geral, a condição de admissibilidade porque:

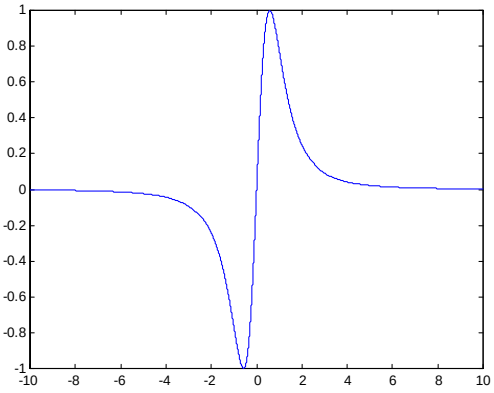
$$H(0)=\sqrt{2\pi}\exp(-\omega_0^2/2)\neq 0 \qquad (4.10)$$

e daí $c_h=+\infty$. Entretanto, se ω_0 é bastante grande, digamos $\omega_0=4$, $H(0)$ torna-se muito próximo de zero e pode ser tomado como zero, computacionalmente.

4.5.2 *Wavelet*s RASP

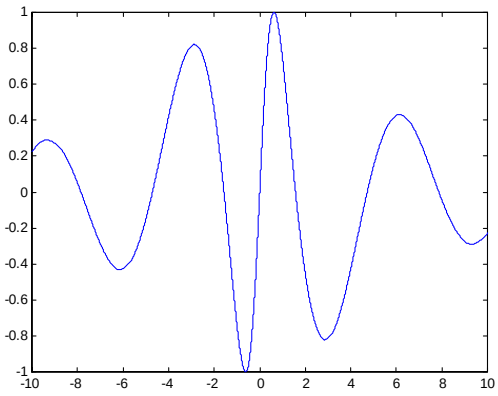
O teorema do resíduo para funções de variáveis complexas pode ser usado para mostrar que algumas funções reais tem média zero, proporcionando a possibilidade de candidaturas a *wavelets*-mães.

É o caso, por exemplo, das funções racionais com pólos de segunda ordem (RASP), três delas são apresentadas na figura 4.7.



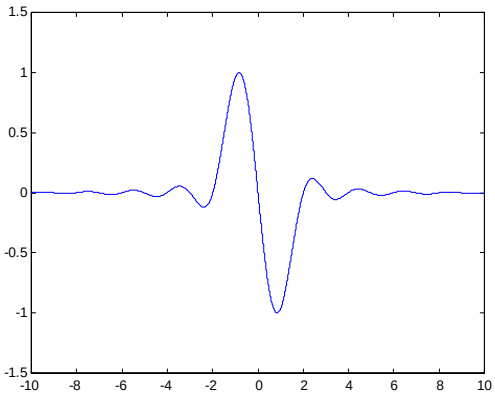
$$\frac{k_1 t}{\left(t^2+1\right)^2} \quad , \quad k_1=3{,}0788$$

RASP1



$$\frac{k_2\, t\cos(t)}{(t^2+1)} \quad , \quad k_2 = 2{,}7435$$

RASP2



$$\frac{k_3\,\mathrm{sen}(\pi t)}{(t^2-1)} \quad , \quad k_3 = 0{,}6111$$

RASP3

Figura 4.7: Funções Racionais com pólos de segunda ordem (exemplos de RASP).

Teorema de Resíduo

Seja C um caminho fechado, $C \subset C$ (corpo dos números complexos), tal que f é analítica sobre C e no interior de C exceto num número finito de pontos singulares $z_1, z_2, \ldots, z_n$. Então,

$$\oint_C f(z)\,dz = 2\pi j \sum_{i=1}^{n} \mathrm{Res}[f; z_i] \qquad (4.11)$$

em que a integral é calculada no sentido anti-horário ao longo de *C* e se f tem um pólo de ordem k em z_i. O termo Res[*f ; z*] é dado por:

$$\text{Res}[f;z_i]=\lim_{z\to z_i}\frac{1}{(k-1)!}\frac{d^{k-1}}{dz^{k-1}}\left\{(z-z_i)^k\ f(z)\right\} \tag{4.12}$$

A wavelet RASP2 $h(t)=\dfrac{t\cos(t)}{t^2+1}$ é aceita como *wavelet*-mãe, pois tem média zero, como se verifica aplicando o teorema do resíduo, observando-se que outros exemplos são tratados similarmente. Considere-se então a integral no plano complexo $\oint_c \dfrac{z\,e^{jz}}{z^2+1}dz$, sendo *C* a curva fechada simples mostrada na figura 4.8.

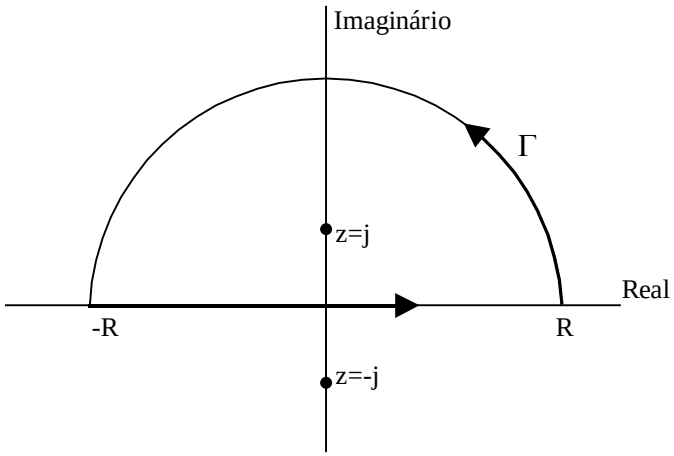


Figura 4.8: *C*, uma curva fechada simples.

O integrando $\dfrac{z\,e^{jz}}{z^2+1}$ tem pólos simples em $z=\pm j$ e destes apenas $z=j$ ocorre dentro de *C*.

O resíduo de $h(z)$ em $z=j$ é dado por:

$$\text{Res}(h;j)=\lim_{z\to j}(z-j)\frac{z\,e^{jz}}{(z+j)(z-j)}=\frac{1}{2e} \tag{4.13}$$

e em virtude de (4.11), tem-se

$$\oint_c \frac{z\,e^{jz}}{z^2+1}dz = 2\pi\, j\,\mathrm{Re}\,s[h;\,j] = \frac{\pi}{e}\,j \tag{4.14}$$

ou

$$\int_{-R}^{R} \frac{x\,e^{jk}}{x^2+1}\,dx + \int_{\Gamma} \frac{z\,e^{jz}}{z^2+1}dz = \frac{\pi}{e}\,j \tag{4.15}$$

isto é,

$$\int_{-R}^{R} \frac{x\cos(x)}{x^2+1}dx + j\int_{-R}^{R} \frac{x\,\mathrm{sen}\,(x)}{x^2+1}dx + \int_{\Gamma} \frac{z\,e^{jz}}{z^2+1}dz = \frac{\pi}{e}\,j \tag{4.16}$$

e assim a parte real é

$$\int_{-R}^{R} \frac{x\cos(x)}{x^2+1}dx + \int_{\Gamma} \frac{z\,e^{jz}}{z^2+1}dz = 0\,. \tag{4.17}$$

Fazendo-se $R \to \infty$, segue-se que

$$\int_{\Gamma} \frac{z\,e^{jz}}{z^2+1}dz = \lim_{R\to\infty} \int_0^{\pi} \frac{R\,e^{j\theta}\; e^{jR\,e^{j\theta}}\; R\,e^{j\theta}\; j d\theta}{R^2\,e^{j2\theta}+1} \le \int_0^{\pi} \lim_{R\to\infty} \frac{R^2\left|e^{-R\,\mathrm{sen}\,\theta}\right| d\theta}{R^2-1}\,. \tag{4.18}$$

Note-se que a desigualdade $|A+B| \ge |A|-|B|$ foi usada no denominador e $\left|\int AB d\theta\right| \le \int |A|\,|B|\,d\theta$ no numerador da equação (4.17).

Note-se também que $\lim_{R\to\infty} e^{-R\,\mathrm{sen}\,\theta} = 0$ e este fato levado à equação (4.17) assegura que $\int_{\Gamma} \frac{z\,e^{jz}}{z^2+1}dz = 0$ e da equação (4.17) segue-se que $\lim_{R\to\infty} \int_{-R}^{R} \frac{x\cos(x)}{x^2+1}dx = 0$ ou que $h(t) = \frac{t\cos(t)}{t^2+1}$ tem média zero podendo assim ser empregada como *wavelet*-mãe.

4.5.3 *Wavelet* SLOG

Uma função estritamente crescente, diferenciável (suave) cujo gráfico lembra um S alongado é nomeada de logística, usualmente empregada como ativação no neurônio de saída de uma rede neural (Haykin, 1994). A superposição de um número finito de logísticas (SLOG) devidamente ponderadas e deslocadas constituem uma família de *wavelets*-mães bem adaptadas para problemas de representação e classificação oriundas de rede neural. Uma função logística sigmoidal é por exemplo:

$$f_{\log}(t)=\frac{1}{1+e^{-t}} \tag{4.19}$$

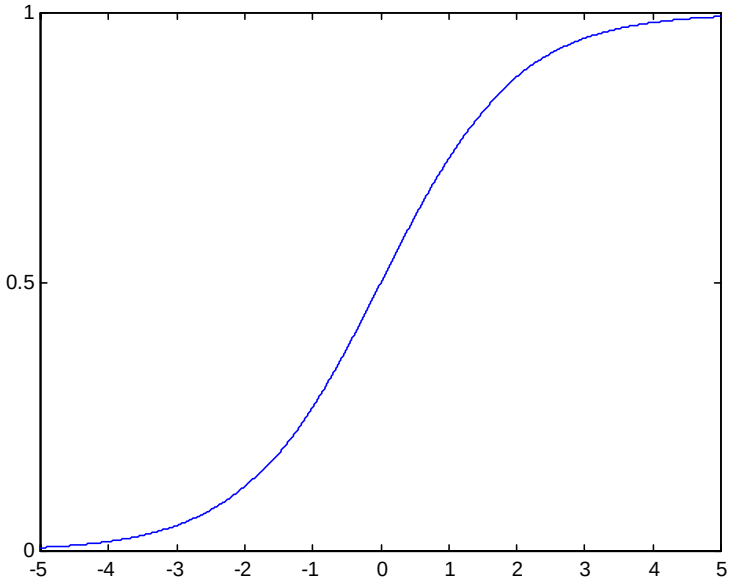


Figura 4.9: Função Logística Sigmoidal

Essa família de *wavelets*-mães aparecem naturalmente quando redes neurais são utilizadas em questões de representação e classificação. A habilidade com que uma função entrada-saída de rede pode prover uma *wavelet*, sugere que ela tem propriedades de *aproximação* semelhante, isto é, representação de mapas arbitrários de $L^2(\Re)$.

Pati e Krishnaprasad (1993), provaram construtivamente o teorema de aproximação universal em $L^2(\Re)$ mostrando que a implementação de uma rede neural feedforward garante a

construção de um frame afim para $L^2(\Re)$. O teorema afirma que *redes neurais feedforward com funções de ativação sigmoidais e uma única camada escondida pode representar qualquer função* $f \in L^2(\Re)$.

A decomposição do sinal em multiresolução sustenta a prova do teorema. Para se constituir um frame, a transformada *wavelet* discreta gerada amostrando os parâmetros (tempo/escala) em um reticulado, não igualmente espaçado, devem satisfazer a condição de admissibilidade e os pontos do reticulado devem estar suficientemente próximos para reter as informações essenciais. Assim, a questão da reconstrução do sinal a partir dos seus valores transformados depende do espaçamento no dito reticulado.

Um reticulado mais fino permite reconstrução mais fácil, embora com possível redundância. Um reticulado mais grosseiro resulta em perda de informação. O cuidado deve estar presente para que se obtenha um conjunto de funções *wavelets* que seja minimal, isto é, não redundante.

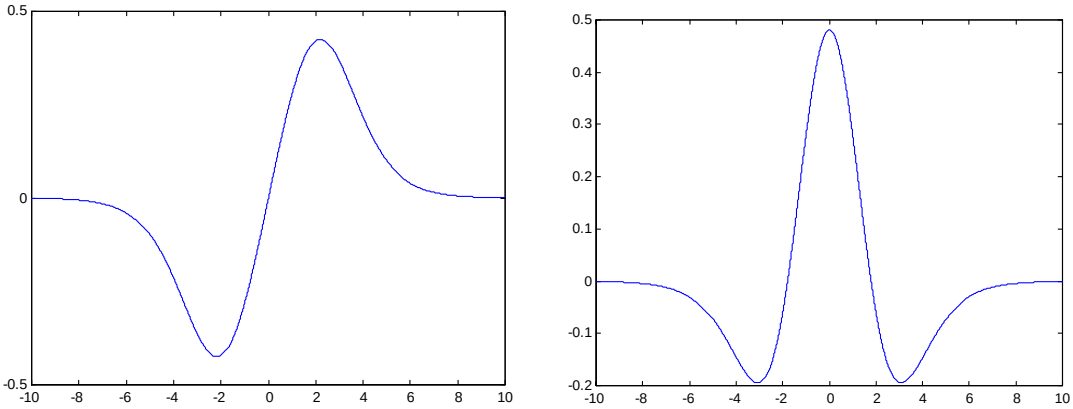
A figura 4.10a representa a primeira *wavelet*-mãe SLOG exibindo a seguinte superposição de sigmóides logísticas:

$$h_{s\log 1}(t) = \frac{1}{1+e^{-t+1}} - \frac{1}{1+e^{-t+3}} - \frac{1}{1+e^{-t-3}} + \frac{1}{1+e^{-t-1}} \tag{4.20}$$

Esta *wavelet*-mãe apresenta uma oscilação mínima. A figura (4.10b) apresenta outra *wavelet* SLOG tendo uma meia oscilação extra. Neste segundo exemplo ainda, uma soma finita de funções logísticas ponderadas e deslocadas é empregada:

$$h_{s\log 2}(t) = \frac{3}{1+e^{-t-1}} - \frac{3}{1+e^{-t+1}} - \frac{1}{1+e^{-t-3}} + \frac{1}{1+e^{-t+3}} \tag{4.21}$$

Da mesma forma, somando e ou subtraindo sigmóides logísticas deslocadas, novas famílias de *wavelets* SLOG podem ser geradas com mais ou menos meia-oscilação.



(a) SLOG1 (b) SLOG2

Figura 4.10: Exemplos de SLOG

4.5.4 *Wavelet* POLYWOG

Inúmeras *wavelets*-mães aparecem de funções do tipo gaussiano janeladas por polinômios (POLYWOG).

Por exemplo, todas as derivadas de uma função gaussiana são POLYWOGs sendo pois admissíveis como *wavelets*-mães. A seguir são apresentados quatro exemplos de tais funções:

POLYWOG1:

$$h_{POLYWOG1} = R_1\, t \exp\left(-\frac{t^2}{2}\right),\ R_1 = \sqrt{e} \tag{4.22}$$

POLYWOG2:

$$h_{POLYWOG2} = R_2 \left(t^3 - 3t\right) \exp\left(-\frac{t^2}{2}\right),\ R_2 = 0{,}7246 \tag{4.23}$$

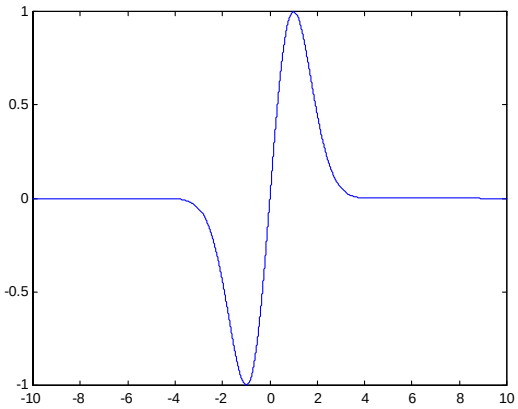
POLYWOG3:

$$h_{POLYWOG3} = R_3\left(t^4 - 6t^2 + 2\right)\exp\left(-\frac{t^2}{2}\right),\ R_3 = \frac{1}{3} \tag{4.24}$$

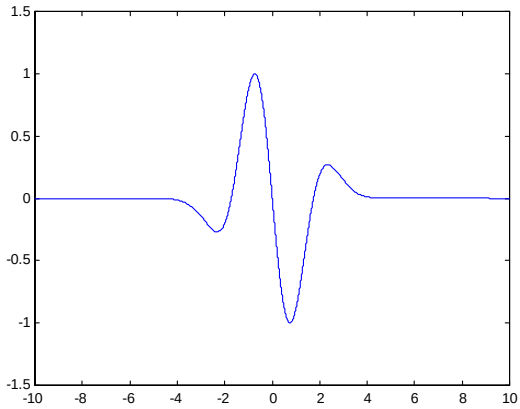
POLYWOG4:

$$h_{POLYWOG4} = \left(1 - t^2\right)\exp\left(-\frac{t^2}{2}\right) \tag{4.25}$$

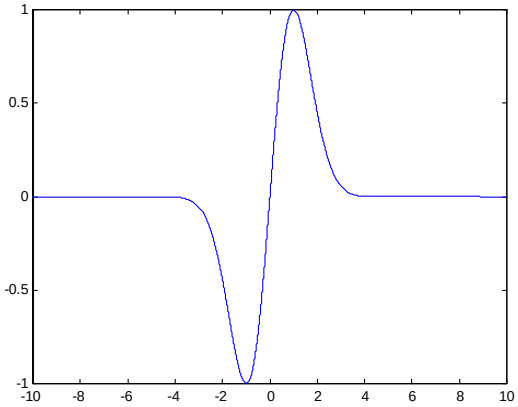
A figura 4.11 a seguir mostra exemplos de POLYWOGs.



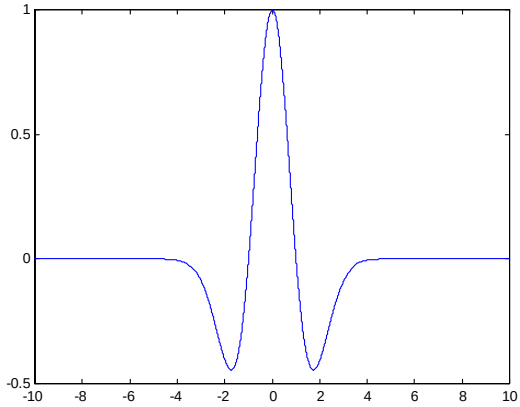
POLYWOG1



POLYWOG2



POLYWOG3



POLYWOG4

Figura 4.11: Exemplos de *wavelets* POLYWOG

A *hermiticidade* de um operador pode ser outro caminho para construção de *wavelets* POLYWOG.

É oportuna uma revisão sobre operadores *hermitianos* ou *auto-adjuntos* e como eles marcam presença na representação de quantidades físicas no mesmo sentido que a "*frequência*" surge com a transformada de Fourier. Expanda-se um sinal na forma

$$f(t)=\int_{-\infty}^{\infty} F(a)u(a,t)\,da \tag{4.26}$$

em que $u(a,t)$ são as funções base e $F(a)$ são os *coeficientes da expansão* ou a "*transformada integral do sinal"*. Será mostrado que $F(a)$ é dado por:

$$F(a)=\int_{-\infty}^{+\infty} f(t)u^{*}(a,t)\,dt \tag{4.27}$$

em que * denota conjugação complexa. As funções base (base no sentido de Hilbert, conjunto de funções apropriadas para fins de representação), $u(a,t)$ são funções da variável t (tempo) e uma outra quantidade física *a*. Sabemos que $F(a)$ é uma avaliação da importância do particular a para o sinal sob análise. Se $F(a)$ for significativamente grande apenas em uma região particular, então o sinal é dito ser concentrado naqueles valores de *a*.

Um operador linear $\mathbf{A}:E\to E$, num espaço vetorial munido de produto interno, chama-se *auto-adjunto (hermitiano)* quando $(\mathbf{A}u,v)=(u,\mathbf{A}v)$ para quaisquer $u,v\in E$, em que $(\cdot,\cdot)$ representa um produto interno em E. Presentemente, de interesse tem-se no papel de E, o espaço vetorial das funções de quadrado integrável, $E=L^{2}(\Re)$, munido do produto interno usual.

Assim, repetindo, um operador $\mathbf{A}$ é *hermitiano* se para todo par de funções $f(t)$ e $g(t)$ tem-se

$$\int_{-\infty}^{+\infty} g^{*}(t)\,\mathbf{A}(f(t))dt=\int_{-\infty}^{+\infty} f(t)(\mathbf{A}(g(t)))^{*}\,dt \tag{4.28}$$

A hermiticidade do operador garante que o conjunto das autofunções é *completo* (no sentido de Riesz-Hilbert) e ortogonal.

Isto significa que as autofunções satisfazem

$$\int_{-\infty}^{+\infty} u^*(a',t)u(a,t)dt = \delta(a-a') \tag{4.29a}$$

$$\int_{-\infty}^{+\infty} u^*(a,t')u(a,t)da = \delta(t-t') \tag{4.29b}$$

Sendo δ a representação clássica da distribuição de Dirac.

São estas propriedades que respaldam as transformadas entre $f(t)$ e $F(a)$ como dadas nas equações (4.26) e (4.27). Em particular, para se obter $F(a)$, multiplica-se (4.26) por $u^*(a',t)$ e integra-se com respeito a *t*:

$$\int_{-\infty}^{+\infty} f(t)u^*(a',t)dt = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(a)u(a,t)u^*(a',t)\,da\,dt \tag{4.30a}$$

$$= \int_{-\infty}^{+\infty} F(a)\delta(a-a')da \tag{4.30b}$$

$$= F(a') \tag{4.30c}$$

provando-se assim, a relação inversa (4.27).

Outro fato notável é que os operadores hermitianos possuem autovalores reais e isto é significativo porque na natureza tudo que é mensurável é real. Entretanto, operadores não-hermitianos, alguns deles, são importantes, embora, não seja possível agregar aos seus autovalores conteúdo físico mensurável. Provaremos a seguir que a diferenciação é um operador hermitiano. Consideremos o operador $\mathbf{A} = \frac{1}{j}\frac{d}{dt}$ definido em $L^2(\Re)$ e sejam $f(t)$ e $g(t)$ quaisquer duas de suas funções. Integrando-se por partes,

$$\int_{-\infty}^{+\infty} g^*(t)\frac{1}{j}\frac{d}{dt}f(t)\,dt = \frac{1}{j}fg^*\Big|_{-\infty}^{+\infty} - \frac{1}{j}\int_{-\infty}^{+\infty} f(t)\frac{d}{dt}g^*(t)\,dt \tag{4.31}$$

Desde que $f(t),\, g(t) \in L^2(\Re)$, segue que $fg^*\Big|_{-\infty}^{+\infty}$ se anula e a equação (4.31) pode ser escrita alternativamente como

$$\int_{-\infty}^{+\infty} g^*(t)\frac{1}{j}\frac{d}{dt}f(t)\,dt = \int_{-\infty}^{+\infty} f(t)\left(\frac{1}{j}\frac{d}{dt}g(t)\right)^* dt$$

ou

$$\left(\mathbf{A}(f), g\right) = \left(f, \mathbf{A}(g)\right) \tag{4.32}$$

provando assim que $\mathbf{A}$ é hermitiano.

Sejam $f(t)$ e $g(t)$ elementos de $L^2(\Re)$ onde $f(t)$ é uma função polinomial janelada e $g(t)$ é uma gaussiana. A partir da equação (4.32) tem-se

$$\int_{-\infty}^{+\infty} \exp\left(-\frac{t^2}{2}\right)\frac{d}{dt}f(t)\,dt = -\int_{-\infty}^{+\infty} f(t)\,t\,\exp\left(-\frac{t^2}{2}\right)dt \tag{4.33}$$

Como exemplo de *wavelets*-mães produzidas por derivação pode-se citar a POLYWOG4 (fig. (4.11)), a qual é a derivada segunda da função gaussiana, também conhecida por chapéu mexicano (Gabor):

$$h_{POLYWOG4} = \left(1-t^2\right)\exp\left(-\frac{t^2}{2}\right)$$

A transformada de Fourier do chapéu mexicano é

$$H(\omega) = -\omega^2 \exp\left(-\frac{\omega^2}{2}\right) \tag{4.34}$$

uma função real e par, satisfazendo a condição de admissibilidade $H(0)=0$, isto é,

$$\int_{-\infty}^{+\infty}\left(1-t^2\right)\exp\left(-\frac{t^2}{2}\right)dt=0 \tag{4.35}$$

Usando a hermiticidade da diferenciação, combinando as equações (4.33) e (4.35) obtém-se:

$$\int_{-\infty}^{+\infty}\left(1-t^2\right)\exp\left(-\frac{t^2}{2}\right)dt=\int_{-\infty}^{+\infty}\left(t-\frac{t^3}{3}\right)t\exp\left(-\frac{t^2}{2}\right)dt=0 \tag{4.36}$$

e daí uma nova *wavelet*-mãe POLYWOG com média zero e boas propriedades de localização.

A figura (4.12) exibe uma nova *wavelet*-mãe, POLYWOG5:

$$h_{POLYWOG5}=R_3\left(3t^2-t^4\right)\exp\left(-\frac{t^2}{2}\right) \tag{4.37}$$

em que $R_3=0{,}8244$ surge para garantir normalização.

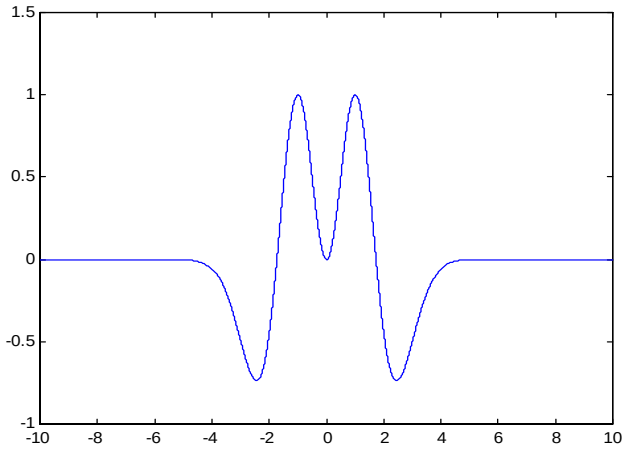


Figura 4.12: *Wavelet* POLYWOG5, Hermiticidade da derivada.

4.5.5 *Wavelet* de Shannon

Um exemplo de *wavelet*-mãe que gera uma *base ortonormal* para $L^2(\Re)$ é a de Shannon, descontínua em frequência e portanto espalhada no tempo. Para que alguma função $\phi(t) \in L_2(\Re)$ forme uma base ortogonal, $\{\phi(t-n), n \in \mathrm{N}\}$ deve cumprir a condição de ortonormalidade:

$$(\phi(t-\ell),\ \phi(t-k)) = \delta(k-\ell) \tag{4.38}$$

ou no domínio de Fourier, basta que

$$\sum_{k=-\infty}^{k=+\infty} \left|\Phi\left(\omega + 2k\ -\ k\pi\right)\right|^2 = 1 \tag{4.39}$$

Por exemplo, o teorema da amostragem de Shannon garante que as funções:

$$\phi(t-k) = \frac{\operatorname{sen}\pi(t-k)}{\pi(t-k)} \tag{4.40}$$

constituem uma base ortonormal desde que a transformada de Fourier Φ tem suporte compacto $L^2[-\pi, \pi]$ e daí toda função $f(t)$ pode ser expressa como:

$$f(t) = \sum_{k=-\infty}^{\infty} f(k)\,\phi(t-k) \tag{4.41}$$

A verificação da ortonormalidade é simples, no domínio da freqüência.

$$\phi(t) = \frac{\operatorname{sen}(\pi t)}{\pi t} \leftrightarrow \Phi(\omega) = \begin{cases} 1 & se\ |\omega| < \pi \\ 0 & se\ |\omega| \geq \pi \end{cases} \tag{4.42}$$

e o produto interno definido em (4.5) garante que

$$\int_{-\infty}^{+\infty} \phi(t-k)\phi^*(t-\ell)dt = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-j\omega k}\,\Phi(\omega)e^{j\omega\ell}\,\phi^*(\omega)d\omega \tag{4.43a}$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-j\omega(k-\ell)}d\omega = \delta(k-\ell) \tag{4.43b}$$

Analogamente, se uma função *wavelet* básica $h(t) \in L^2(\Re)$ satisfaz a condição de admissibilidade e sua transformada de Fourier $H(\omega)$ satisfaz a condição de ortonormalidade (4.39):

$$\sum_{k=-\infty}^{\infty} |H(\omega + 2k\pi)|^2 = 1 \tag{4.44}$$

as translações inteiras da *wavelet*, $h(t-k)$, também formam uma base ortonormal. A equação (4.44) é satisfeita por, exemplificando,

$$H(\omega) = \begin{cases} 1 & se \quad \pi < |\omega| < 2\pi \\ 0 & se \quad |\omega| > 2\pi \quad ou \quad |\omega| < \pi \end{cases}$$

produzindo a *wavelet* Shannon

$$h_{Shannon}(t) = \frac{\text{sen}(2\pi t) - \text{sen}(\pi t)}{\pi t} \tag{4.45}$$

cujo gráfico está mostrado na fig. (4.13);

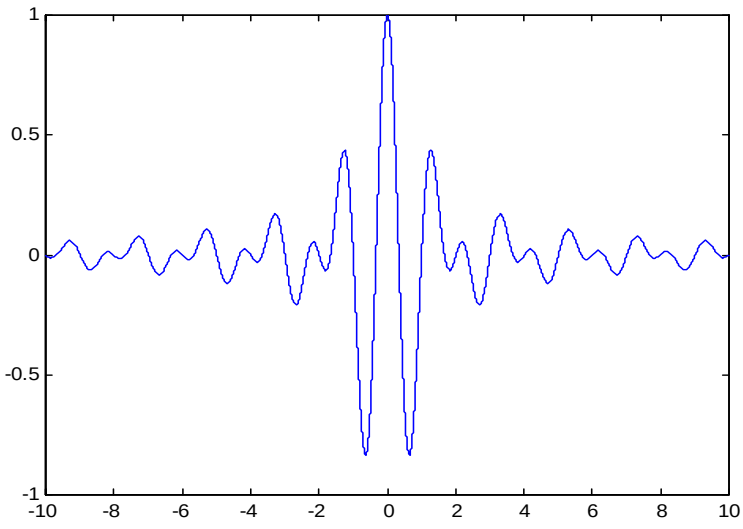


Figura 4.13: *Wavelet*-mãe de Shannon

Até o momento apresentou-se a transformada *wavelet* como ferramenta para análise de sinais, dando ênfase as suas vantagens em relação à transformada de Fourier. A familiaridade com a teoria da transformada *wavelet* oferece uma alternativa promissora para aproximar funções arbitrárias quando combinada com redes neurais. Uma indicação de como isso ocorre será feita em seguida, no próximo capítulo.

Neste trabalho, iremos utilizar a decomposição em *wavelets* de uma forma diferente, nossa representação será baseada em uma técnica proposta por Lekutai denominada *wavenet*. Nesta técnica, uma função é representada por um número fixo de *wavelets*, que correspondem aos neurônios de uma rede neural. Os coeficientes das *wavelets* são representados pelos pesos sinápticos desta rede. Mediante um procedimento de otimização, os pesos e parâmetros das *wavelets* (dilatação *a* e translação *b*) são determinados de forma a melhor aproximar a função considerada. É importante salientar que os parâmetros das *wavelets* são obtidos a partir do espaço contínuo de fase, permitindo uma melhor aproximação para um considerado número de *wavelets*.

Capítulo 5

Wavenet

5.1 Introdução

Combinando a teoria da transformada *wavelet* com o conceito básico de redes neurais, faz surgir um processo de mapeamento entrada/saída denominado *wavenet* (Lekutai, 1997), que surge como uma alternativa para redes neurais *feedforward* aproximar funções não-lineares arbitrárias. Em outras palavras, *wavenet* é uma rede neural ativada por *wavelets* especiais.

O algoritmo *wavenet* consiste em um processo que objetiva minimizar a função erro, entre a saída da planta e a saída da rede, adaptando os parâmetros da rede neural e das funções *wavelets*. A utilização das *wavelet*s como funções de ativação da rede neural, proporciona um mapeamento local das funções, produzindo melhores resultados com menor esforço computacional (Grassi *et al*., 2003).

5.2 Algoritmo Wavenet

A arquitetura da rede fig. 5.1 aproxima qualquer sinal esperado $y(t)$ pela combinação linear de um conjunto de *wavelets-filha*s $h_{a,b}(t)$, em que $h_{a,b}(t)$ são geradas por dilatação *a*, e translação *b*, a partir da *wavelet-mãe* $h(t)$:

$$h_{a,b}(t) = h\left(\frac{t-b}{a}\right) \tag{5.1}$$

com fator de dilatação $a > 0$.

O sinal aproximado fornecido pela rede, $\hat{y}(t)$, pode ser representado por:

$$\hat{y}(t) = u(t) \sum_{k=1}^{K} w_k \; h_{a_k, b_k}(t) \tag{5.2}$$

em que *K* é o número de janelas *wavelets*, e w_k são os pesos.

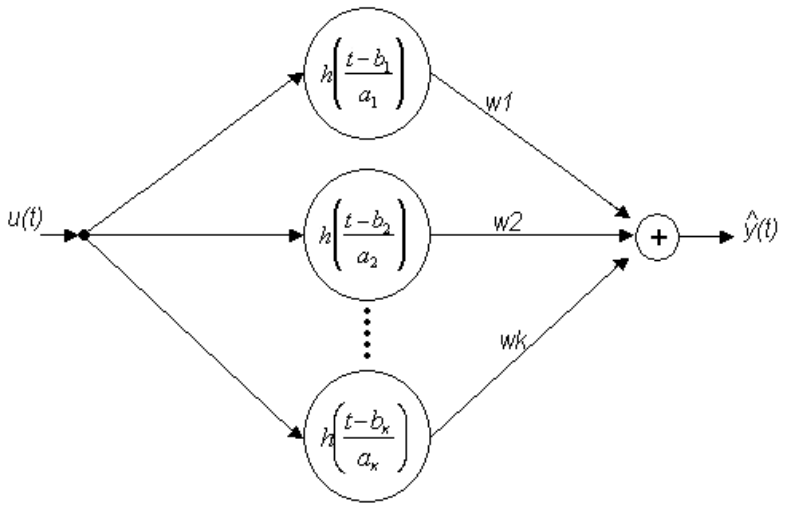


Figura 5.1: Estrutura de Rede *Wavelet*.

A rede *wavelet* faz parte de um mecanismo de identificação, que ajusta os parâmetros da rede w_k, a_k e b_k, objetivando minimizar o erro entre *y(t)* e $\hat{y}(t)$ conforme apresentado na estrutura da figura 5.2.

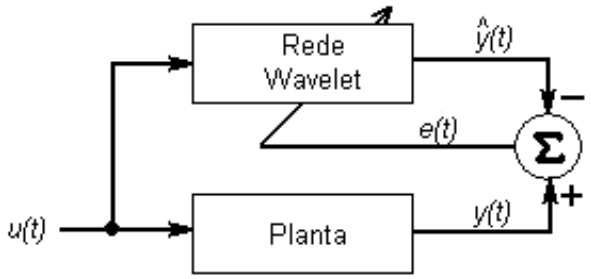


Figura 5.2: Estrutura de Identificação.

Os parâmetros da rede w_k, a_k e b_k devem ser otimizados no sentido dos mínimos quadrados, minimizando a função custo, E, por todo tempo t. Denotando-se o erro, variável com o tempo, por:

$$e(t) = y(t) - \hat{y}(t) \tag{5.3}$$

em que $y(t)$ é a resposta desejada (meta). A função custo é definida por:

$$E = \frac{1}{2}\sum_{t=1}^{T} e^2(t) \tag{5.4}$$

Para minimizar *E* nós podemos utilizar vários métodos dentre os quais apresentaremos aqui o mais empregado na literatura, o método do gradiente descendente, o qual requer os gradientes $\frac{\partial E}{\partial w_k}$, $\frac{\partial E}{\partial a_k}$ e $\frac{\partial E}{\partial b_k}$, que são necessários no processo de atualização dos parâmetros w_k, a_k, b_k, respectivamente. Os gradientes de *E* são:

$$\frac{\partial E}{\partial w_k} = -\sum_{t=1}^{T} e(t)h(\tau)u(t) \tag{5.5}$$

$$\frac{\partial E}{\partial b_k} = -\sum_{t=1}^{T} e(t)u(t)\, w_k \frac{\partial h(\tau)}{\partial b_k} \tag{5.6}$$

$$\frac{\partial E}{\partial a_k} = -\sum_{t=1}^{T} e(t)u(t)\, w_k \tau \frac{\partial h(\tau)}{\partial b_k} \tag{5.7}$$

em que $\tau = \frac{t - b_k}{a_k}$

As mudanças incrementais de cada coeficiente são:

$$\Delta \mathrm{w} = -\frac{\partial \mathrm{E}}{\partial \mathrm{w}},\ \Delta \mathrm{b} = -\frac{\partial \mathrm{E}}{\partial \mathrm{b}},\ \Delta \mathrm{a} = -\frac{\partial \mathrm{E}}{\partial \mathrm{a}} \tag{5.8}$$

e assim, serão atualizados de acordo com a regra

$$w(n+1)=w(n)+\mu_w\,\Delta w \tag{5.9a}$$

$$b(n+1)=b(n)+\mu_b\,\Delta b \tag{5.9b}$$

$$a(n+1)=a(n)+\mu_a\,\Delta a \tag{5.9c}$$

em que μ é a taxa de aprendizado previamente fixada.

O objetivo do processo de aprendizado é ajustar os parâmetros livres da rede para minimizar a função custo, resultando em um problema de otimização irrestrito.

De forma a melhorar a eficiência do método de gradiente, pode efetuar-se uma modificação na taxa de aprendizado μ, introduzindo para o efeito uma matriz positiva definida *P*, em função de cada parâmetro.

O sucesso da identificação da planta, neste caso, está diretamente associada à escolha da família *wavelet* e ao número de janelas usadas, bem como a escolha dos valores iniciais das escalas, parâmetro *a*.

A tabela 5.1 mostra algumas famílias *wavelet* e suas respectivas derivadas.

Tabela 5.1 - *Wavelet*s e suas derivadas

NOME	$h(\tau)$	$\dfrac{\partial h(\tau)}{\partial b}$
MORLET	$\cos(\omega_0\,\tau)\exp(-0{,}5\tau^2)$	$\dfrac{1}{a}\left[\omega_0\,\mathrm{sen}(\omega_0\,\tau)\exp(-0{,}5\,\tau^2)+\tau h(\tau)\right]$
RASP1	$\dfrac{\tau}{(\tau^2+1)^2}$	$\dfrac{1(3\tau^2-1)}{a(\tau^2+1)^3}$
RASP2	$\dfrac{\tau\cos(\tau)}{\tau^2+1}$	$\dfrac{\tau\left(\dfrac{\tau^2+1}{a}\right)\mathrm{sen}(\tau)+\left(\dfrac{\tau^2-1}{a}\right)\cos(\tau)}{(\tau^2+1)^2}$
RASP3	$\dfrac{\mathrm{sen}(\pi\tau)}{\tau^2-1}$	$\dfrac{\left(\dfrac{2\tau}{a}\right)\mathrm{sen}(\pi\tau)-\pi\left(\dfrac{\tau^2-1}{a}\right)\cos(\pi\tau)}{(\tau^2-1)^2}$

SLOG1	$\frac{1}{1+e^{-\tau+1}}-\frac{1}{1+e^{-\tau+3}}+$ $-\frac{1}{1+e^{-\tau-1}}+\frac{1}{1+e^{-\tau-3}}$	$\frac{1}{a}\left[-\frac{e^{\tau+1}}{\left(1+e^{-\tau+1}\right)^2}+\frac{e^{-\tau+3}}{\left(1+e^{-\tau+3}\right)^2}+\right.$ $\left.+\frac{e^{-\tau-3}}{\left(1+e^{-\tau-3}\right)^2}-\frac{e^{-\tau-1}}{\left(1+e^{-\tau-1}\right)^2}\right]$
SLOG2	$\frac{3}{1+e^{-\tau-1}}-\frac{3}{1+e^{-\tau+1}}+$ $+\frac{1}{1+e^{-\tau-3}}+\frac{1}{1+e^{-\tau+3}}$	$\frac{1}{a}\left[-\frac{3e^{-\tau-1}}{\left(1+e^{-\tau-1}\right)^2}+\frac{3e^{-\tau+1}}{\left(1+e^{-\tau+1}\right)^2}+\right.$ $\left.+\frac{e^{-\tau-3}}{\left(1+e^{-\tau-3}\right)^2}-\frac{e^{-\tau+3}}{\left(1+e^{-\tau+3}\right)^2}\right]$
POLYWOG 1	$\tau\exp\left(-\frac{\tau^2}{2}\right)$	$\frac{1}{a}\left(\tau^2-1\right)\exp\left(-\frac{\tau^2}{2}\right)$
POLYWOG 2	$\left(\tau^3-3\tau\right)\exp\left(-\frac{\tau^2}{2}\right)$	$\frac{1}{a}\left(\tau^4-6\tau^2+3\right)\exp\left(-\frac{\tau^2}{2}\right)$
POLYWOG 3	$\left(\tau^4-6\tau^2+3\right)\exp\left(-\frac{\tau^2}{2}\right)$	$\frac{1}{a}\left(\tau^5-10\tau^3+15\tau\right)\exp\left(-\frac{\tau^2}{2}\right)$
POLYWOG 4	$\left(1-\tau^2\right)\exp\left(-\frac{\tau^2}{2}\right)$	$\frac{1}{a}\left(3\tau-\tau^3\right)\exp\left(-\frac{\tau^2}{2}\right)$
POLYWOG 5	$\left(3\tau^2-\tau^4\right)\exp\left(-\frac{\tau^2}{2}\right)$	$\frac{1}{a}\left(-\tau^5+7\tau^3-6\tau\right)\exp\left(-\frac{\tau^2}{2}\right)$
SHANNON	$\frac{\operatorname{sen}2\pi\tau-\operatorname{sen}\pi\tau}{\pi\tau}$	$\frac{\pi}{a}\frac{(-\pi\tau\cos\pi\tau-2\pi\cos2\pi\tau+\operatorname{sen}\pi\tau+\operatorname{sen}2\pi\tau)}{(\pi\tau)^2}$

Embora seja o mais empregado, o método do gradiente descendente nem sempre trabalha corretamente, já que, somente a informação da primeira derivada do erro é utilizada ao longo do processo iterativo de otimização, tornando-o muitas vezes lento e ineficiente.

A superfície de erro de uma rede complexa é composta por ‘colinas’ e ‘vales’. Devido ao gradiente descendente a rede pode ficar bloqueada (‘*trapped*’) num mínimo local embora exista um mínimo ‘mais profundo’ na sua proximidade.

Uma possibilidade de sair desta armadilha é a utilização, por exemplo, de métodos probabilísticos. Uma outra possibilidade é aumentar o número de unidades escondidas, o que leva a aumentar a dimensão do espaço dos erros e a uma menor chance da rede cair num mínimo

local (Ribeiro B., 2002/2003). De qualquer modo, neste caso, o aumento de unidades (neurônios) pode levar a uma dimensão exagerada da rede caindo-se facilmente no problema da falta de generalização.

Momentaneamente e com o intuito de simplificar a notação, vamos tratar apenas do parâmetro *w(peso da rede)* para apresentar o problema de mínimos locais, que são dados por:

1. Em geral estes métodos encontram um mínimo local w^* tal que $E(w^*) < E(w)$ para todos os vetores peso w numa vizinhança de w^*.
2. Apenas condições especiais como, por exemplo, a convexidade de $E(w^*)$ garantirão que o mínimo local é também um mínimo global.
3. Na prática, pode ser usada uma extensa gama de vetores pesos iniciais w_0, cada qual levando a uma solução w^*.
4. O vetor de pesos w^* associado ao $E(w^*)$ será selecionado como a melhor escolha possível.

Para acelerar o processo de treinamento, alguns algoritmos de otimização de segunda ordem, baseados no gradiente e na informação da segunda derivada (ou Hessiana) têm sido propostos (Hwang e Lewis, 1990), como por exemplo, o método de Newton, Quasi-Newton, Gradiente Conjugado, Gauss-Newton, bem como o método de Levenberg-Marquardt.

5.3 Métodos de Otimização

Como mencionado anteriormente, existe uma grande variedade de algoritmos que implementam métodos de minimização do erro, sendo classificados pela informação que usam (Ribeiro B. 2002/2003):

1. Métodos de 1º ordem ou de gradiente (usam a 1º derivada de $E(w)$):
   – Algoritmo de Retropropagação (BP)
2. Métodos de pesquisa (usam a direção de pesquisa)
   – Método do Gradiente Conjugado (GC)
3. Métodos de 2º ordem (usam a 2º derivada de $E(w)$ - Hessiana)

– Método de Newton (N)

– Método de Gauss Newton (GN)

– Método de Levenberg- Marquardt (LM)

Estes métodos de otimização numérica podem ser aplicados às redes neurais, dado que temos basicamente um problema de utilização de uma função de custo em ordem a parâmetros que são, neste caso, os pesos *w*.

5.3.1 Método de Newton

Se no método de gradiente descrito anteriormente para o algoritmo *wavenet* $M = \mu$, no método de Newton a matriz *P* é definida como sendo a inversa da Hessiana,

$$\mathrm{P}(\mathrm{w}) = \mathrm{H}^{-1} \tag{5.10}$$

que por sua vez é definida como a matriz das segundas derivadas:

$$H = \frac{\partial^2 E}{\partial w \partial w^T} \tag{5.11}$$

5.3.2 Método de Gauss-Newton

Uma desvantagem do método de Newton é a necessidade de usar informação de segunda ordem e a possibilidade de existência de singularidades, que dificulta, ou mesmo impossibilita o cálculo da inversa. Os métodos de Quasi-Newton tentam, em certa medida, ultrapassar o problema estimando o valor da Hessiana com base apenas em informação de primeira ordem. No método Gauss-Newton é calculada uma aproximação da Hessiana, H_a, tendo em conta apenas o cálculo do gradiente, desprezando-se o termo de segunda ordem.

$$H_a \approx \frac{\partial E^T \partial E}{\partial w \partial w} \tag{5.12}$$

5.3.3 Método de Levenberg-Marquardt

Contudo, mantém-se a necessidade de calcular a inversa de uma matriz. Uma forma possível de ultrapassar o problema consiste em adicionar uma restrição adicional, proposta por Levenberg-Marquardt (1963).

$$H_a \approx \frac{\partial E^T \partial E}{\partial w \partial w} + \gamma I \tag{5.13}$$

A variável γ é uma constante real positiva e I é uma matriz identidade de dimensões apropriadas. Para $\gamma = 0$ o método reduz-se ao método de Newton, para γ elevado obtém-se o método de gradiente. Este método é particularmente adequado a problemas mal condicionados, de que é exemplo o problema em causa, a estimação de parâmetros em redes neurais.

5.4 Redes Neurais e Controle de Processos

A aplicação de redes neurais em controle tem sido uma área de pesquisa ativa desde os anos 80, em virtude da dificuldade de controlar sistemas dinâmicos com alto grau de não-linearidade, incerteza e imprecisão, empregando técnicas convencionais de controle. Redes neurais apresentam algumas propriedades desejáveis para modelar sistemas não-lineares e, com isto, podem superar algumas dificuldades encontradas por outras técnicas.

A habilidade das redes neurais em representar mapeamentos não-lineares, e assim modelar sistemas não-lineares, é a característica mais explorada na síntese de controladores. Recentemente, foram propostos diversos métodos de projeto de controle não-linear, incluindo principalmente técnicas de otimização, que têm um importante papel no ajuste dos parâmetros dos modelos envolvidos.

Uma área estabelecida em teoria de controle moderno, com grande relevância aqui, é a teoria de sistemas adaptativos (Åstrom e Wittenmark, 1994). Esta área tem produzido muitos resultados teóricos interessantes nos últimos anos. Embora a teoria de sistemas adaptativos seja

fundamentada na suposição de sistemas lineares e invariantes no tempo, os conceitos e resultados encontrados na literatura estão sendo estendidos para uso no campo de redes neurais. No entanto, muitos resultados acabam sendo verificados apenas através de simulações, necessitando de uma maior fundamentação teórica.

Existem duas abordagens distintas filosoficamente para o controle de um sistema sob incerteza (Coelho, 2000), controle direto e controle indireto.

No controle direto, os parâmetros do controlador são ajustados para reduzir alguma norma do erro de saída. O sucesso desta abordagem depende do conhecimento a priori do jacobiano da planta, ou pelo menos do sinal de seus elementos. Logo, o estimador é utilizado para obter os parâmetros do controlador diretamente a partir das medidas de entrada e saída, fig. 5.3.

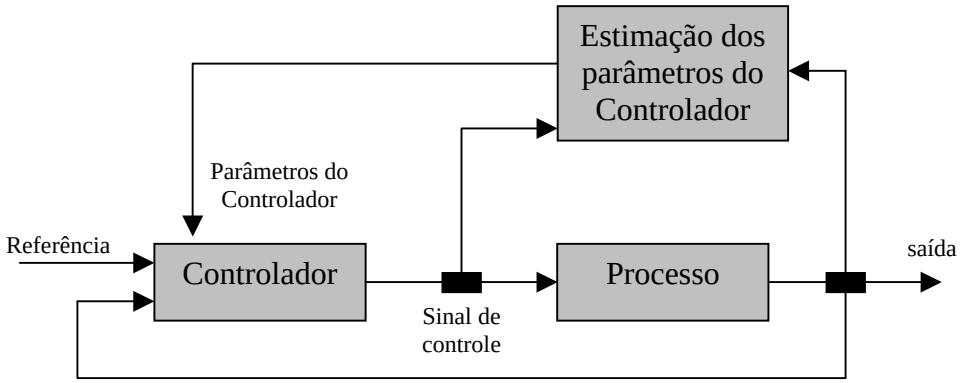


Figura 5.3: Esquema de controle adaptativo direto.

Por outro lado, no controle indireto, os parâmetros da planta são estimados como elementos de um vetor j(k), para algum instante k, e os parâmetros q(k) do controlador são escolhidos supondo que j(k) representa o valor verdadeiro do vetor de parâmetros "*p*" da planta. Em outras palavras, no procedimento de projeto indireto, baseado na técnica *self-tuning*, o controle é calculado supondo que os parâmetros do processo sejam conhecidos. Um estimador é utilizado para obter os parâmetros do processo a partir das medidas de entrada e saída e, a seguir, substituem-se os parâmetros pelos valores estimados, de forma recursiva. A partir dos parâmetros estimados calculam-se os parâmetros do controlador (ver fig. 5.4). Mesmo quando a planta é suposta ser linear e invariante no tempo, os controles adaptativos direto e indireto conduzem a problemas de otimização não-lineares.

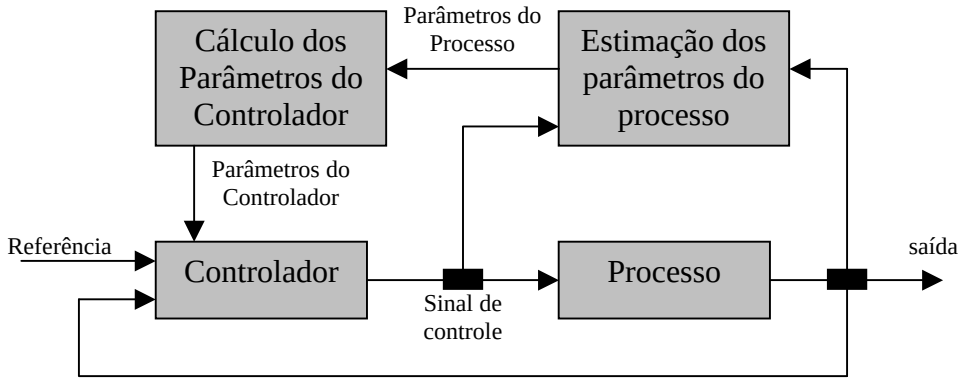


Figura 5.4: Esquema de controle adaptativo indireto.

Conseqüentemente, uma metodologia de controle direto pode ser considerada como "adaptativo por desempenho", enquanto controle indireto é "adaptativo por parâmetro".
Conforme já mencionado, a aplicação de redes neurais em controle de processos representa uma alternativa às técnicas tradicionais.

5.5 Abordagens tradicionais para controle de sistemas dinâmicos não-lineares

Os sistemas dinâmicos são tipicamente modelados por equações diferenciais ordinárias. Geralmente, recorre-se a um sistema acoplado de equações de primeira ordem, $\dot{x} = F(x,u,t)$, onde os vetores $\boldsymbol{x}$ e $\boldsymbol{u}$ representam respectivamente o estado e a entrada do sistema e F é um campo vetorial não-linear (Isidori, 1989). O objetivo de controle é encontrar a entrada u, a qual conduz o sistema a apresentar características desejáveis, por exemplo, estabilidade e alto desempenho ao seguir uma referência. No caso do controle por realimentação, a entrada será uma função de algumas variáveis do próprio sistema, por exemplo, $u = g(x)$.

Embora exista uma grande variedade de ferramentas para projeto de sistemas de controle lineares, este não é o caso para sistemas de controle não-lineares, e mesmo quando existe alguma técnica para solução de problemas de controle não-linear, ela não é aplicável genericamente na prática devido à sua complexidade e especificidade. A maior dificuldade é que nenhuma formalização genérica está disponível para o tratamento de todos os problemas não-lineares.

Muitos métodos empregam aproximações, visto que as soluções exatas para as equações diferenciais não-lineares são raramente conseguidas. Entre as técnicas mais freqüentemente usadas temos: *simulação*, *linearização local* e *ganho de escalonamento*, *linearização por realimentação global*, *funções descritivas*, *ganhos equivalentes* e *estratégias baseadas no segundo método de Lyapunov* (Phillips e Müller-Dott, 1992*).*

O comportamento dinâmico acoplado de equações de 1º ordem, utilizadas na geração do modelo, pode ser revelado através da solução numérica obtida por simulação em computadores digitais. Os resultados de simulação não conduzem a conclusões fortes sobre a natureza qualitativa, como, por exemplo, a garantia ou não de estabilidade do sistema, mas permitem avaliar o desempenho sob certas condições.

Linearização local é uma das técnicas mais comuns para análise e projeto de sistemas não-lineares, desde que ela aproxime localmente o sistema não-linear por um sistema linear. A validade da aproximação linear se restringe a uma região limitada em torno de um ponto de operação específico (Lima, 2000). Uma vez que existe um grande número de ferramentas para projeto e análise de sistemas lineares, a abordagem por linearização parte do conjunto de equações diferenciais não-lineares e produz um conjunto de equações diferenciais lineares para um ponto de operação específico. Esta técnica de linearização local torna-se ineficiente para sistemas altamente não-lineares, uma vez que um número não razoável de pontos de operação discretos pode ser requerido para modelar o sistema não-linear com um grau de precisão aceitável. Em se tratando de controle, esta tendência é altamente prejudicial, uma vez que tal projeto deve ser repetido para cada modelo linear.

Linearização por realimentação é uma abordagem de natureza mais abrangente, a qual requer um controlador não-linear com a função específica de remover um comportamento não-linear indesejado da planta. Esta técnica exige um amplo conhecimento da planta e essencialmente requer a construção de um controlador não-linear por realimentação para cancelar as não-linearidades da planta. O projeto de tal controlador não é trivial e os efeitos da modelagem incompleta e de distúrbios da planta ainda não foram plenamente investigados (Isidori, 1989).

O segundo método de *Lyapunov* (Vidyasagar, 1993; Wang e Yeh, 1990) pode ser empregado na análise de estabilidade de sistemas não-lineares sob bases teóricas. O método foi desenvolvido para examinar a estabilidade de equações diferenciais não-lineares.

Um importante aspecto do segundo método de *Lyapunov* é que ele não requer a solução do sistema de equações diferenciais. Embora o método nos poupe a tarefa de solucionar o

sistema de equações diferenciais, ele requer a escolha de uma função de *Lyapunov*, para a qual não existem procedimentos sistemáticos de obtenção. A idéia básica do segundo método de *Lyapunov* é que um sistema físico pode somente armazenar uma quantidade finita de energia. Se puder ser mostrado que a energia está sempre sendo dissipada, exceto para pontos de equilíbrio, então o sistema deve se aproximar do ponto de equilíbrio, que representa um estado de energia mínima. A função de energia para o sistema, que pode não ser única, é chamada de *função de Lyapunov*. Cada uma destas técnicas convencionais de projeto de controle não-linear tem todos os inconvenientes associados às abordagens lineares. É por causa destas dificuldades que têm sido propostas alternativas para controle não-linear, como por exemplo, o uso de redes neurais artificiais.

5.6 Técnicas de neurocontrole baseadas em dinâmica inversa

Muitas das aplicações de controle utilizando redes neurais, também denominadas neurocontrole foram projetadas para conduzir à determinação da dinâmica inversa do sistema (Ortega e Camacho, 1996; Steck *et al*., 1996). Esta abordagem é especialmente útil para plantas que apresentam dinâmica rápida.

5.6.1 Controle realimentado e controle neural baseado no sistema inverso

Métodos de controle neural baseado no sistema inverso, embora já tenham apresentado resultados práticos interessantes, principalmente em robótica (Miller *et al*., 1990), continuam a manifestar sérios inconvenientes. O mais comum diz respeito à falta de realimentação. Se o controlador não é um modelo inverso preciso do sistema, este pode levar a ações de controle incoerentes, que conduzem a erros que não podem ser detectados e corrigidos sem realimentação. No entanto, não é trivial aprender completamente a dinâmica inversa de um

sistema de ordem elevada e sob efeito de ruído, como ocorre em boa parte dos sistemas do mundo real (Gorinevsky, 1993). Mesmo se obtivermos o verdadeiro modelo inverso do sistema, o controlador ainda assim pode ser muito sensível a distúrbios e atrasos no sistema.

Um outro problema do controle inverso é que nem todo sistema é inversível. Estes problemas restringem fortemente a amplitude de aplicação do método de controle neural baseado no sistema inverso.

Fica então devidamente justificado o emprego de realimentação junto à malha de controle, quando isto for possível. No entanto, quando a realimentação é introduzida como entrada do controlador o risco de instabilidade cresce. Isto implica que resultados mais abrangentes ainda são necessários para desenvolver esquemas de aprendizado de controle realimentado.

Em breve, os sistemas de malha fechada deverão ser empregados para realizar ações de controle mais precisas. Sistemas de malha fechada são mais interessantes quando o processo é lento e a regulação é importante (Schmit, 1982). Entretanto, sistemas de malha aberta, sendo essencialmente servos-mecanismos, deverão ser aplicados pelo menos para controle de sistema de movimento rápido, tal como movimento de braços em robótica (Kawato, 1989).

5.7 Controladores Neuro Wavenet Auto-Ajustáveis

Consideremos agora um sistema dinâmico de uma única entrada e única saída (SISO) representado pelas equações de estado:

$$x(k+1) = f\left(x(k), u(k), k\right) \tag{5.14}$$

$$y(k) = g\left(x(k), k\right) \tag{5.15}$$

em que $x(k), u(k)$ *e* $y(k)$ são reais. Admitamos que as funções desconhecidas f e g sejam de classe $C^1(\Re^3, \Re)$. Os únicos dados acessíveis são u (entrada tanto da rede quanto da planta, fig. 5.2) e y (saída da planta).

Levin e Narendra mostraram em 1995 que se o sistema linearizado em torno de um estado de equilíbrio for *observável*, então sua representação poderá ter a forma:

$$y(k+1) = F\left(y(k),\ y(k-1),\ldots,\ y(k-n+1),\ u(k),\ u(k-1),\ldots,\ u(k-n+1)\ \right) \tag{5.16}$$

isto é, existirá uma função $F(\cdot)$ mapeando $y(k)$ e $u(k)$, e seus $n-1$ valores passados, em $y(k+1)$.

Assim, um modelo de rede *wavenet* $\hat{F}$ poderá ser treinada objetivando aproximar a F numa certa região de interesse. Não importa se o modelo da planta é acessível. Importante é a adaptação do modelo pela atualização dos parâmetros do controle *on-line*.

Um enfoque diferente (e adotado neste trabalho) foi proposto por Narendra, Balakrishnan e Ciliz em 1995 para uma planta descrita pela equação:

$$\begin{aligned} y(k+1) = {} & F\left(y(k),\ y(k-1),\ldots,\ y(k-n+1),\ u(k),\ u(k-1),\ldots,\ u(k-n+1)\ \right) \ + \\ & G\left(y(k), y(k-1),\ldots, y(k-n+1), u(k), u(k-1),\ldots, u(k-n+1)\right)\cdot u(k) \end{aligned} \tag{5.17}$$

Em busca de simplicidade e concisão, ela será reescrita na forma:

$$y(k+1) = F\left(y(k)\right) + G\left(y(k)\right)\cdot u(k) \tag{5.18}$$

em que $y(k)$ e $u(k)$ denotam a saída e entrada (da planta) no k-ésimo instante de tempo.

Se as não linearidades $F(\cdot)$ e $G(\cdot)$ fossem conhecidas exatamente, então, o controle $u(k)$ necessário para encontrar a saída desejada, $r(k+1)$, seria calculado em cada instante de tempo por:

$$u(k) = \frac{r(k+1) - F\left(y(k)\right)}{G\left(y(k)\ \right)} \tag{5.19}$$

Entretanto, na prática, $F(\cdot)$ e $G(\cdot)$ são desconhecidos e assim, o enfoque é a busca por um modelo de rede neural adaptativa que se aproxime do sistema dinâmico, baseado nas figuras 5.5 e 5.6, isto é,

$$\hat{y}(k+1) = \hat{F}\left(y(k), \Theta_{\Phi}\right) + \hat{G}\left(y(k)\ , \Theta_{\Gamma}\right) u(k) \tag{5.20}$$

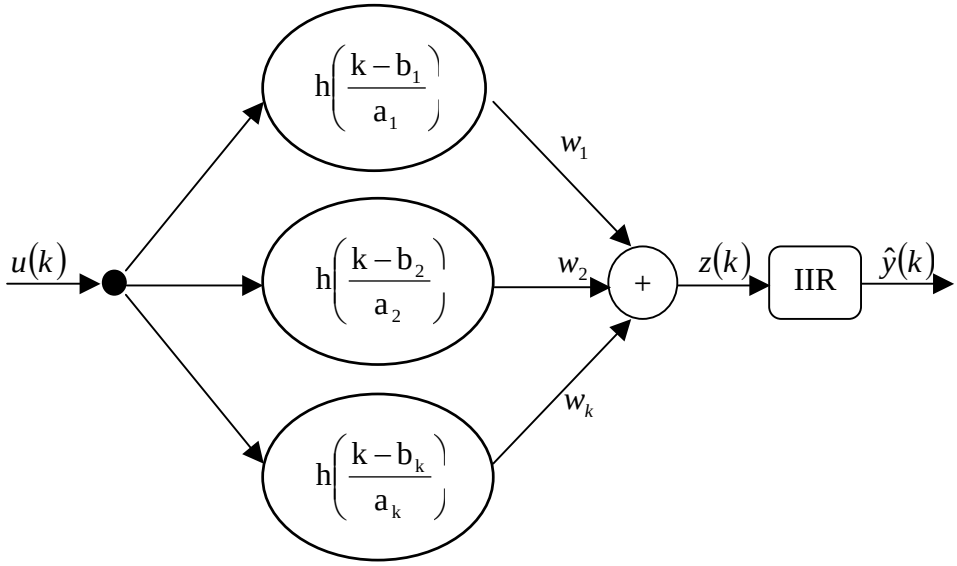


Figura 5.5: Estrutura da *wavenet* adaptativa.

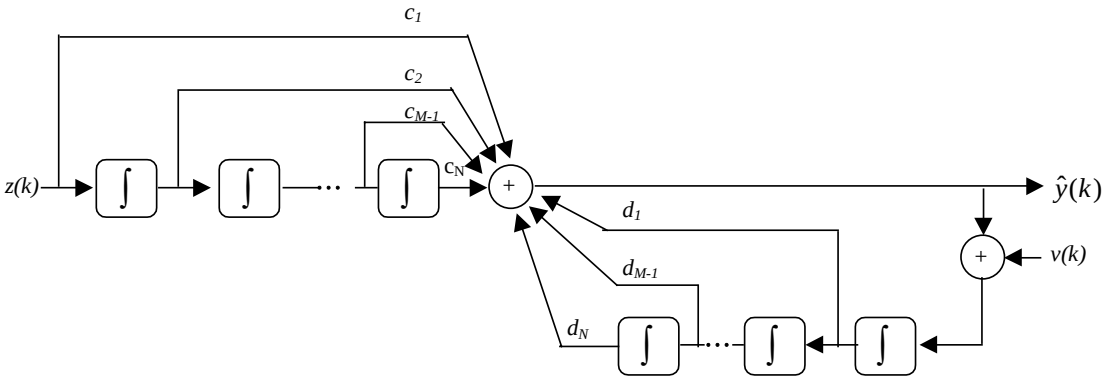


Figura 5.6: Detalhes do filtro resposta ao impulso infinito (IIR).

Observando que a entrada $u(k)$ excita a rede neural produzindo uma saída $z(k)$, entrada do filtro IIR (Resposta ao Impulso Infinito, fig. 5.6) emergindo $\hat{y}(k)$ modelado por:

$$\hat{y}\,(k+1) \;=\; \sum_{i=0}^{M} c_i \;\; z\,(k-i) \;\; u\,(k) \;+\; \sum_{j=1}^{N} d_j \;\; \hat{y}\,(k-j) \;\; v\,(k) \tag{5.21}$$

então, comparando termo a termo, eq. 5.20 e eq. 5.21, conclui-se que

$$\hat{F}\left(y\,(k),\, \Theta_{\Phi}\right) = \sum_{j=1}^{N} d_j \;\hat{y}\,(k-j)\; v\,(k) \tag{5.22}$$

$$\hat{G}\left(y\,(k),\,\Theta_{\Gamma}\right) = \sum_{i=0}^{M} c_i \;\; z\,(k-i)\,u(k) \tag{5.23}$$

em que

$$z\,(k) = \sum_{\ell=1}^{K} w_\ell \; h\left(\frac{k-b_\ell}{a_\ell}\right) \tag{5.24}$$

K é o número de janelas *wavelets*, *M* e c_i são o número de atrasos feedforward e coeficientes do filtro IIR, respectivamente, *N* e d_j são o número de atrasos feedback e coeficientes do filtro IIR, respectivamente, e o sinal $v(k)$ é a co-entrada (fig. 5.6), que influencia para garantir estabilidade do processo de aproximação. Após a substituição das não-linearidades $F\,(\cdot)$ e $G\,(\cdot)$ pelas suas aproximações $\hat{F}\,(\cdot)$ e $\hat{G}\,(\cdot)$ com parâmetros ajustáveis (pesos w_k, escalas a_k, translações b_k, coeficientes *feedforward* IIR c_k e coeficientes feedback IIR d_k) o controle $u(k)$ necessário para localizar a saída desejada $r\,(k+1)$ será calculado em todo instante por

$$u\,(k) = \frac{r(k+1) - \hat{F}\left(y\,(k),\,\Theta_{\Phi}\right)}{\hat{G}\left(y\,(k),\,\Theta_{\Gamma}\right)} \tag{5.25}$$

em que Θ_{Φ} e Θ_{Γ} sintetizam os parâmetros envolvidos no processo de ajuste e adaptação.

Um esquema de neurocontrolador para sistemas de controle auto-ajustável é proposto na figura 5.7, onde o objetivo é que a rede neural identifique a planta fornecendo os parâmetros para o controlador, que por sua vez, encontra a entrada u(k), a qual conduz o sistema a apresentar estabilidade e alto desempenho ao seguir uma referencia r(k).

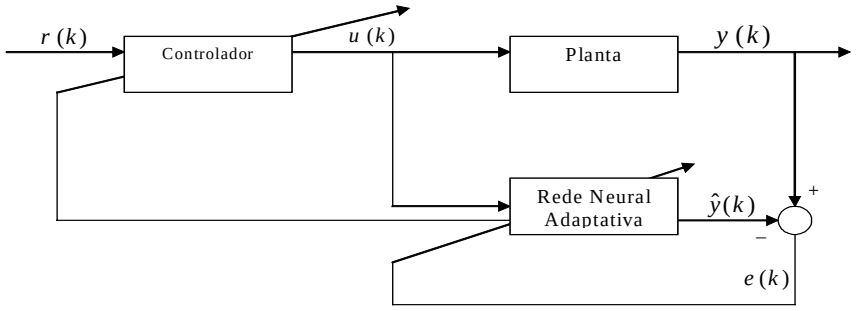


Figura 5.7: Sistema de neurocontrolador auto-ajustável.

Capítulo 6

Resultados

Os capítulos anteriores discutiram aspectos envolvidos no processo de identificação e controle, agregando as técnicas de redes neurais e transformada *wavelet*, explorando o algoritmo denominado *wavenet*. Neste capítulo serão apresentadas algumas peculiaridades do comportamento do algoritmo *wavenet*, discutindo, uma nova abordagem do processo de otimização dos parâmetros, e aspectos referentes à arquitetura da rede (número de neurônios e funções de ativação), a partir de dados simulados, tentando encontrar qual configuração do algoritmo produz resultados mais eficientes no processo de identificação e controle de sistemas não lineares.

O primeiro passo dado foi verificar a potencialidade do algoritmo *wavenet*, alvo principal do estudo, avaliando em que aspectos e condições o funcionamento do algoritmo era capaz de realizar uma boa identificação da planta, já que o controle pretendido se baseia na inversa da planta, e exige uma boa identificação para obter uma resposta eficaz do controlador.

Buscou-se reproduzir alguns testes utilizando os exemplos apresentados em Lekutai, 1997 e Righeto, 2003, aproveitando a experiência prévia para definir a arquitetura da rede, verificando a influência do número de neurônios na camada intermediária e quais funções de ativação *wavelet* proporcionavam melhores resultados no processo de identificação de plantas não lineares.

Outro aspecto analisado e que foi objeto de estudo, foram os métodos de otimização para o desempenho do algoritmo *wavenet*. Relatos apresentados na literatura apontam alguns problemas que podem vir a ocorrer quando se utiliza o método do gradiente descendente, e sugerem métodos de otimização de segunda ordem para tratar problemas não lineares. Dentre os pesquisados, optou-se pelo método de Levenberg-Marquadt, o qual foi testado e confrontado com o gradiente descendente apresentado no trabalho de Lekutai.

6.1 Identificação de um Sinal

Antes de entrar em detalhes nos resultados das simulações realizadas, precisamos definir os tipos de arquiteturas de redes neurais (*wavenet*) e algoritmos de treinamento supervisionado implementados e utilizados para realizar um estudo comparativo. Dentre os modelos, temos:

1. Rede neural com uma camada intermediária de *n* neurônios com funções de ativação (*wavelets*) idênticas e parâmetros ajustados via método do gradiente descendente;
2. Rede neural com uma camada intermediária de *n* neurônios com funções de ativação (*wavelets*) idênticas e parâmetros ajustados via método de Levenberg-Marquadt.

O desempenho do algoritmo, na identificação de um sinal referente à voz feminina do fonema "*a*", pronunciando a palavra "had" da língua inglesa, é avaliado neste item. Em função da não disponibilidade do sinal, o mesmo foi obtido dos dados de Lekutai. Neste caso, o sinal analisado foi simulado utilizando a definição *wavelet* e os parâmetros *a*, *b* e *w* retirados do trabalho de Lekutai, 1997. Com o sinal de saída da planta gerado (referente à voz feminina do fonema "*a*"), realizou-se a identificação da planta utilizando a mesma configuração de rede proposta por Lekutai, e esquematizado na figura 5.1, conforme discutido anteriormente no capítulo 5. A configuração da rede é composta por 10 neurônios na camada intermediária e utiliza como função de ativação uma *wavelet* mãe do tipo Rasp1. As figuras e tabelas a seguir mostram os resultados da identificação obtidos com o método de Levenberg-Marquadt e os resultados obtidos por Lekutai utilizando o método do gradiente descendente.

A figura 6.1 mostra a superposição do sinal original e do sinal identificado, e como pode ser observado, a identificação é quase perfeita, o que pode ser constatado também na figura 6.2 que aponta o erro calculado durante o processo.

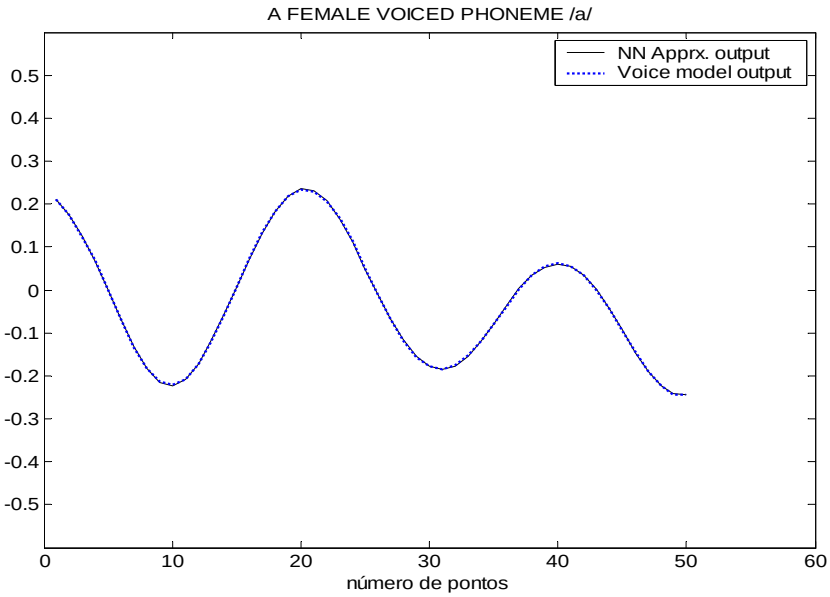


Figura 6.1: Identificação utilizando Levenberg-Marquadt.

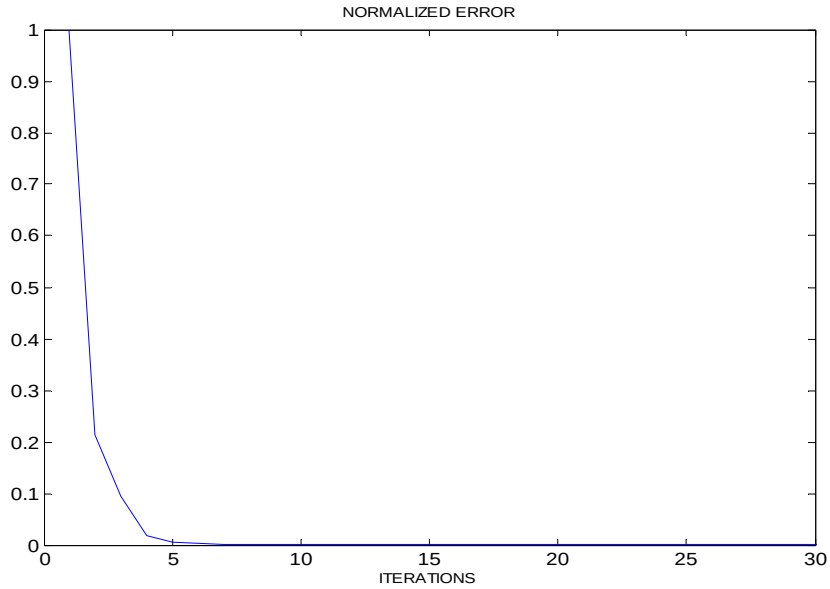
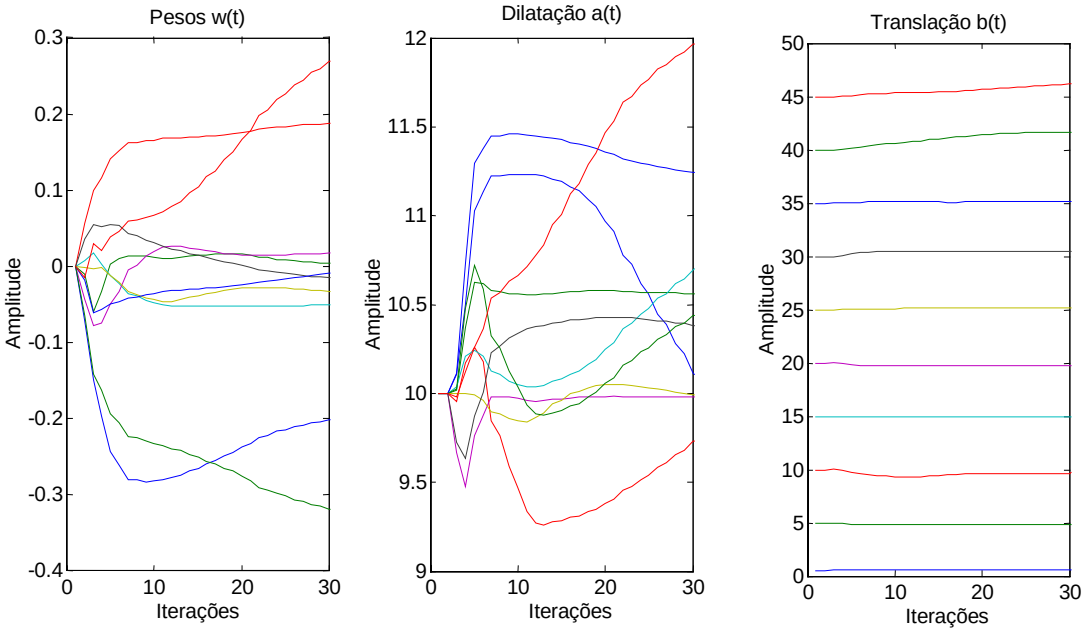


Figura 6.2: Curva do erro na identificação utilizando Levenberg-Marquadt.

As figuras 6.3(a), 6.3(b) e 6.3(c), mostram a evolução dos parâmetros da rede *wavenet* durante o processo de identificação, sendo eles, o peso "*w*" e os coeficientes de dilatação "*a*" e translação "*b*" da *wavelet*, utilizando o método de Levenberg-Marquadt. A tabela 6.1 mostra os valores utilizados na inicialização do algoritmo e os valores obtidos no final do processo de identificação.



(a) (b) (c)

Figura 6.3: Identificação utilizando Levenberg-Marquadt

Tabela 6.1. Parâmetros Iniciais e Finais.

Inicial			**Final**		
w	a	b	w	a	b
0.0000	10.0000	0.5000	-0.1468	9.0000	0.5500
0.0000	10.0000	5.0000	0.0455	10.6083	4.8140
0.0000	10.0000	10.0000	0.2415	10.2666	11.0000
0.0000	10.0000	15.0000	-0.0931	12.1294	14.9273
0.0000	10.0000	20.0000	0.0123	10.0789	19.6370
0.0000	10.0000	25.0000	-0.0354	9.4181	24.4868
0.0000	10.0000	30.0000	-0.0217	10.3038	30.6236
0.0000	10.0000	35.0000	-0.0151	11.0885	35.1281
0.0000	10.0000	40.0000	-0.3762	11.1457	41.8198
0.0000	10.0000	45.0000	0.4404	13.000	46.5489

Com o intuito de comparar os dois processos de identificação, utilizando diferentes métodos de minimização, entretanto utilizando a mesma configuração de rede e parâmetros, são reproduzidos a seguir os resultados obtidos por Lekutai, utilizando o método do gradiente descendente. As figuras. 6.4 e 6.5, apresentam os resultados da identificação e a evolução do erro.

A evolução dos parâmetros, peso "*w*" e os coeficientes de dilatação "*a*" e translação "*b*" da *wavelet* são mostrados nas figuras 6.6(a), 6.6(b) e 6.6(c) respectivamente. A tabela 6.2 mostra as condições iniciais utilizadas no processo de identificação e os resultados finais obtidos.

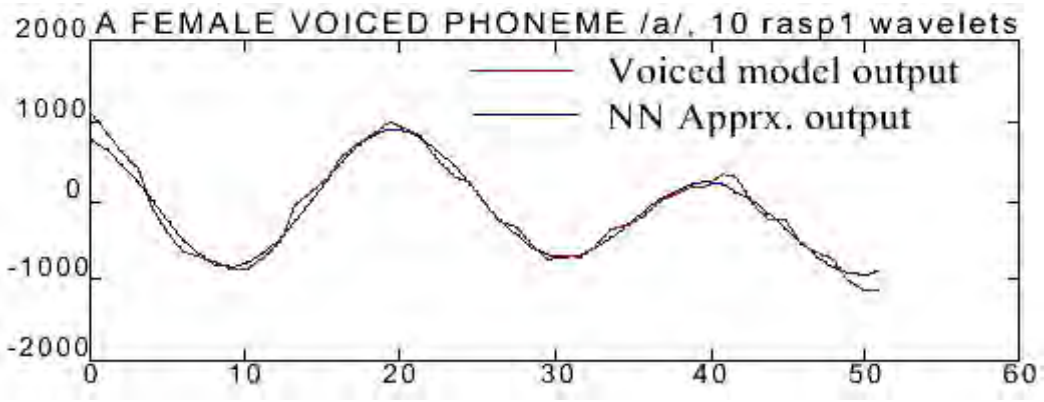


Figura 6.4: Identificação utilizando Gradiente Descendente (Lekutai, 1997)

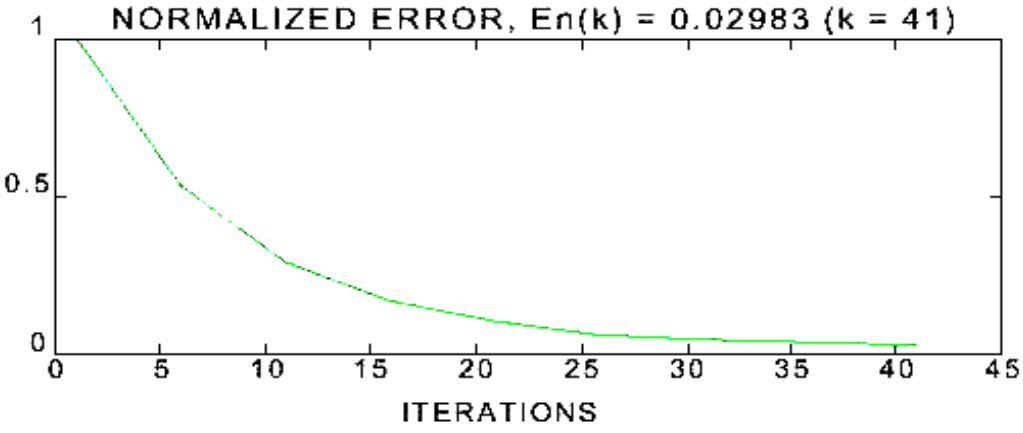
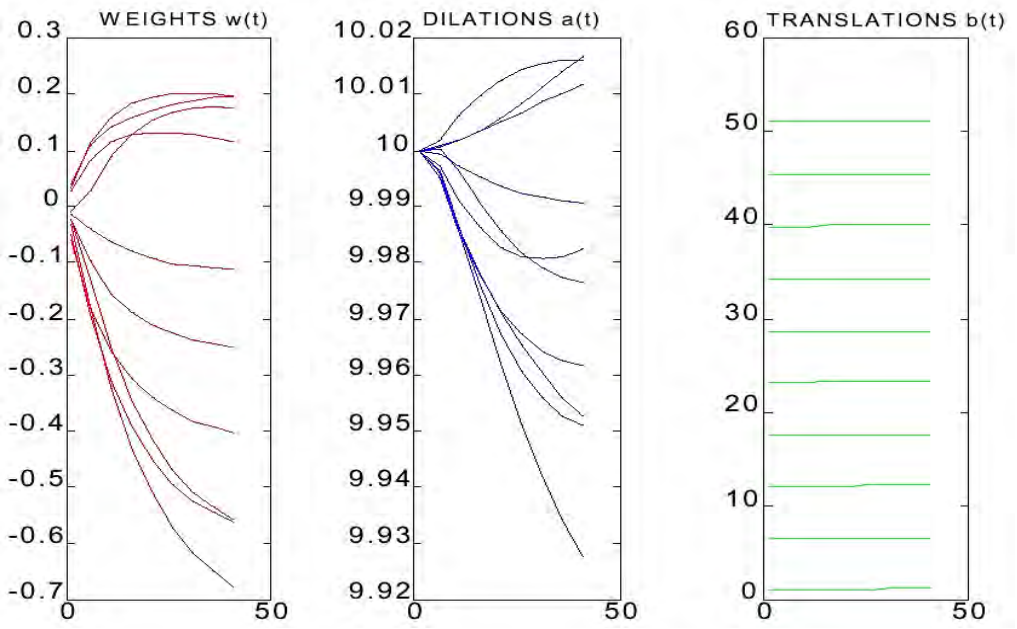


Figura 6.5: Curva do erro utilizando Gradiente Descendente (Lekutai, 1997).



(a) (b) (c)

Figura 6.6: Ajuste dos parâmetros da Rede.

Tabela 6.2. Parâmetros Iniciais e Finais.

Initial			**Final**		
w	a	b	w	a	b
0.0000	10.0000	1.0000	-0.2541	10.0160	1.0000
0.0000	10.0000	6.5556	-0.6818	9.9264	6.4449
0.0000	10.0000	12.1111	0.2002	9.9519	12.1928
0.0000	10.0000	17.6667	0.1946	9.9614	17.5843
0.0000	10.0000	23.2222	-0.5632	9.9825	23.2888
0.0000	10.0000	28.7778	-0.4034	10.0173	28.6582
0.0000	10.0000	34.3333	0.1747	9.9763	34.3482
0.0000	10.0000	39.8889	-0.1116	10.0119	39.9264
0.0000	10.0000	45.4444	-0.5615	9.9507	45.4442
0.0000	10.0000	51.0000	0.1138	9.9907	51.0732

Um dos objetivos deste item, conforme discutido anteriormente, foi comparar o efeito do método de segunda ordem, que utiliza informação da 2º derivada (Levenberg-Marquadt), no processo de minimização. Fazendo uma análise e comparação dos gráficos de erro, obtidos nos

dois casos, fica visível à superioridade do método de Levenberg-Marquadt. O método proporciona um erro significativamente menor do que o erro do gradiente descendente já nas primeiras iterações. A partir da 5º iteração, o erro obtido é menor do que o erro final obtido por Lekutai, evidenciando uma maior rapidez e eficiência do algoritmo. Outros exemplos, não discutidos aqui, mostraram um comportamento similar. Portanto, os próximos itens e condições avaliadas serão baseados no método de otimização de Levenberg-Marquadt.

Entretanto, deve ser notado que a definição dos parâmetros de inicialização da rede, merece alguns cuidados específicos, revelando que uma boa inicialização dos parâmetros de entrada, proporciona maior possibilidade de convergência do algoritmo.

6.2 Influência do número de neurônios na convergência da rede neural.

O número de neurônios utilizados na definição da arquitetura da rede tem uma forte influência nos resultados obtidos. Vários testes foram realizados objetivando avaliar a sua influência no processo de identificação. Entretanto, para não tornar os resultados muito repetitivos serão apresentados apenas os resultados mais relevantes.

A identificação de um sistema de segunda ordem, com uma não linearidade estática é discutida em termos do número de neurônios da rede. O sistema é descrito pela equação diferença, dada pela expressão (6.1).

$$y(n) = 0.3y(n-1) - 0.6y(n-2) + f(u(n)) \tag{6.1}$$

A expressão representa um sistema dinâmico linear, com uma entrada (excitação) não linear, dada pela função não linear *f(.)*, eq. (6.2).

$$f(u) = u^3 + 0.3u^2 - 0.4u \tag{6.2}$$

em que $u(n) = \sin(2\pi n/250)$, $n = 0, 1, 2, ..., 200$.

O sistema acima representa uma planta desconhecida a ser identificada utilizando uma rede *wavenet*. Três diferentes configurações de rede, utilizando uma *wavelet* mãe do tipo Morlet como função de ativação, foram analisadas. Variando o número de neurônios da camada intermediária, buscando avaliar a sua influência na identificação de plantas, a arquitetura da rede é definida de forma que, cada neurônio seja ativado por uma *wavelet* filha, ou seja, uma decomposição da *wavelet* mãe Morlet, dada por, ($h(\tau) = cos(w_0\tau)*exp(-0.5\tau^2)$).

Neste caso foram simuladas redes com configurações de 5, 10 e 20 neurônios na camada intermediária. Dizer que uma rede neural é composta por 20 neurônios na camada intermediária, neste trabalho, significa dizer que o sinal identificado é subdividido em 20 janelas *wavelet*s, ou seja, é subdividido em 20 decomposições da *wavelet* mãe.

As figuras 6.7, 6.8 e 6.9 apresentam a evolução do processo de identificação para 5, 10 e 20 neurônios respectivamente na camada intermediária.

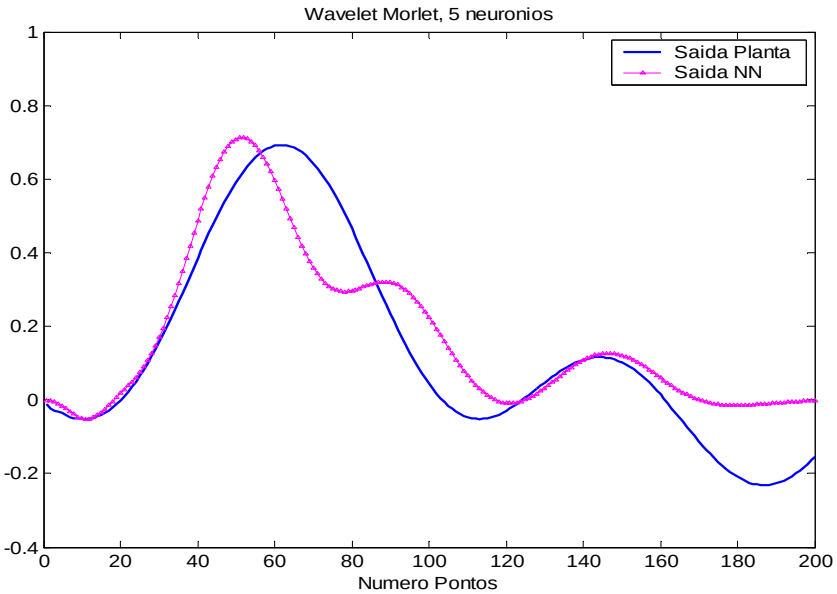


Figura 6.7: Identificação com 5 neurônios na camada intermediária.

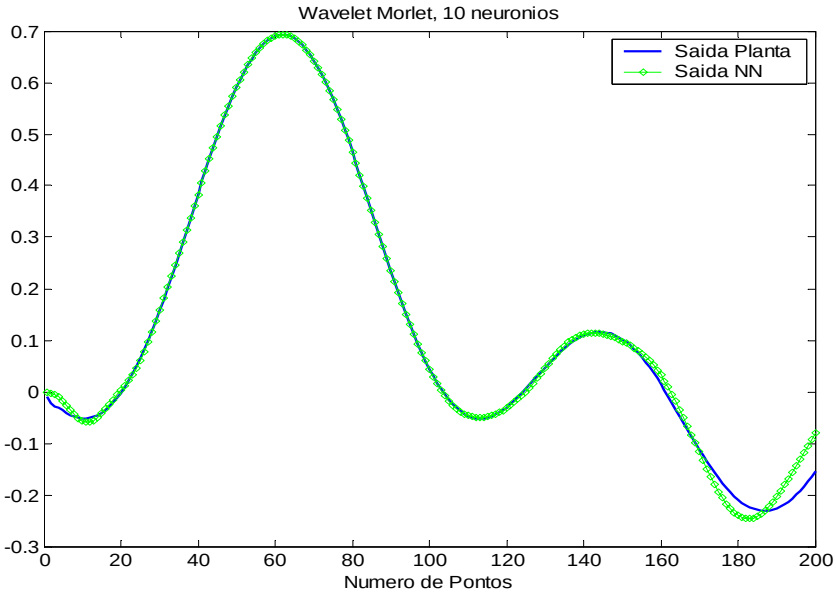


Figura 6.8: Identificação com 10 neurônios na camada intermediária.

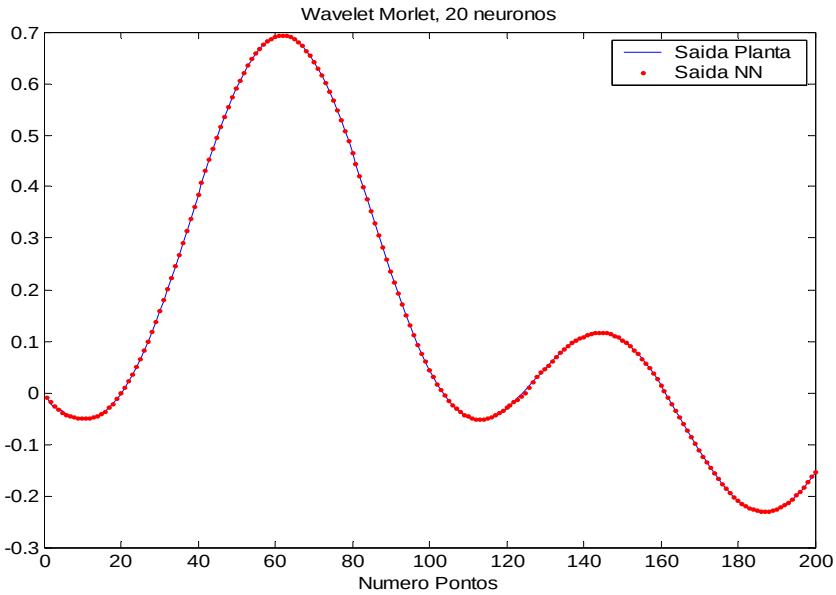


Figura 6.9: Identificação com 20 neurônios na camada intermediária.

Os gráficos mostraram que a eficiência da identificação cresce significativamente quando se aumenta o número de neurônios na camada intermediária. No entanto isso não é sempre verdade, pois existe um número ótimo para cada caso ou sinal analisado, de maneira que, o processo de identificação se estagna e para de convergir. Um número demasiadamente grande de

neurônios torna o processo inviável, já que o número de parâmetros adaptados cresce bastante, fazendo com que o algoritmo fique muito lento.

A fig. 6.10 mostra o comportamento do erro normalizado a cada iteração para as três configurações de rede.

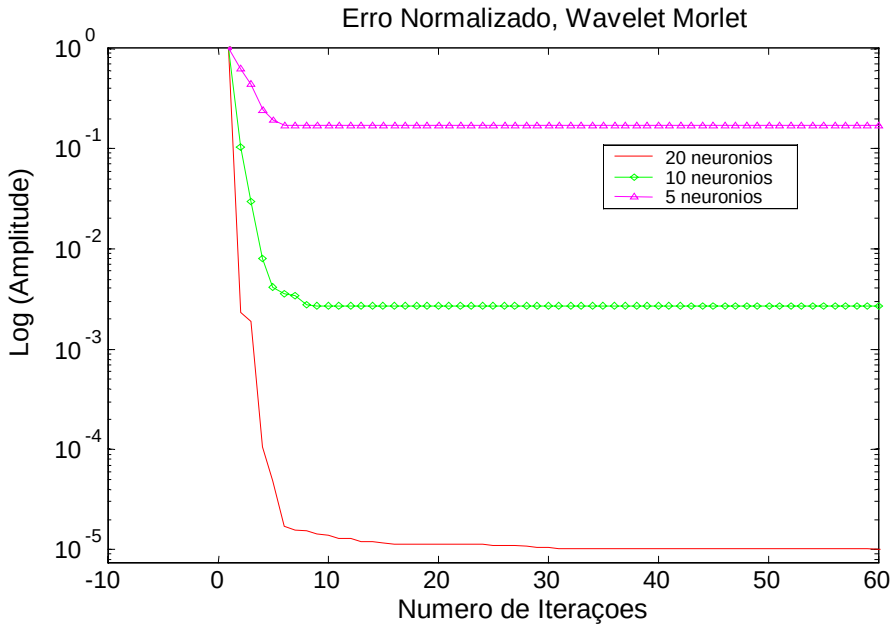


Figura 6.10: Comportamento do Erro variando a Arquitetura da Rede Neural.

A análise do gráfico de erros deixa claro a importância do número de neurônios, que se apresenta como um parâmetro crucial para uma boa convergência da rede *wavenet*. É possível observar que a eficiência do algoritmo melhora quando é utilizado um número adequado de neurônios. Com isso, pode-se concluir que, o mais importante nesta análise, e que ainda é feito heuristicamente, é definir o menor número de neurônios necessários para representar o sinal de saída da planta.

6.3 Influência das funções de ativação considerando diferentes tipos de *wavelet*s.

Como apresentado no capítulo 4, existem vários tipos de *wavelet*s das mais diversas formas. A escolha de uma família *wavelet* adequada, como função de ativação no processo de identificação, está diretamente ligada à forma/estrutura do sinal. Dessa maneira fica justificado realizar um estudo verificando qual família *wavelet* é mais eficiente para identificar ou representar um dado sinal.

O sistema que será utilizado para avaliar o comportamento do algoritmo *wavenet* no processo de identificação, quando estudamos o efeito das funções de ativação, trata-se de um sistema não linear que possui uma não linearidade dinâmica, excitado por uma entrada linear. O sistema simula o comportamento de uma planta, que é descrita pela equação diferença não linear, dada pela equação (6.3).

$$y(n) = \frac{y(n-1)}{1+y^2(n-1)} + u(n) \tag{6.3}$$

em que a excitação é dada por, $u(n) = 0.8 \sin(2\pi n/50)$, $n = 0, 1, 2, ..., 200$.

Dentre as várias funções *wavelet*s disponíveis escolhemos as *wavelet*s Morlet, Rasp1 e Polywog5, porque apresentaram um comportamento mais coerente para avaliar o desempenho da *wavenet* no processo de identificação. Neste caso os testes foram realizados para uma única configuração da rede, que possui 10 neurônios na camada intermediária, para os três tipos de *wavelet*s mãe.

As figs. 6.11, 6.12 e 6.13 mostram a influência das funções de ativação no processo de identificação para a *wavelet* mãe Morlet, Rasp1 e Polywog5, respectivamente.

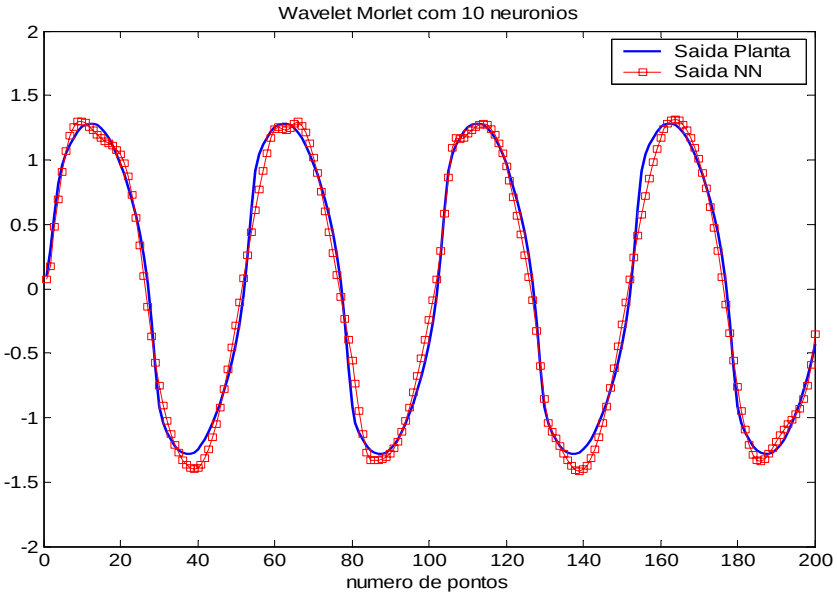


Figura 6.11: Identificação com *Wavelet* Morlet.

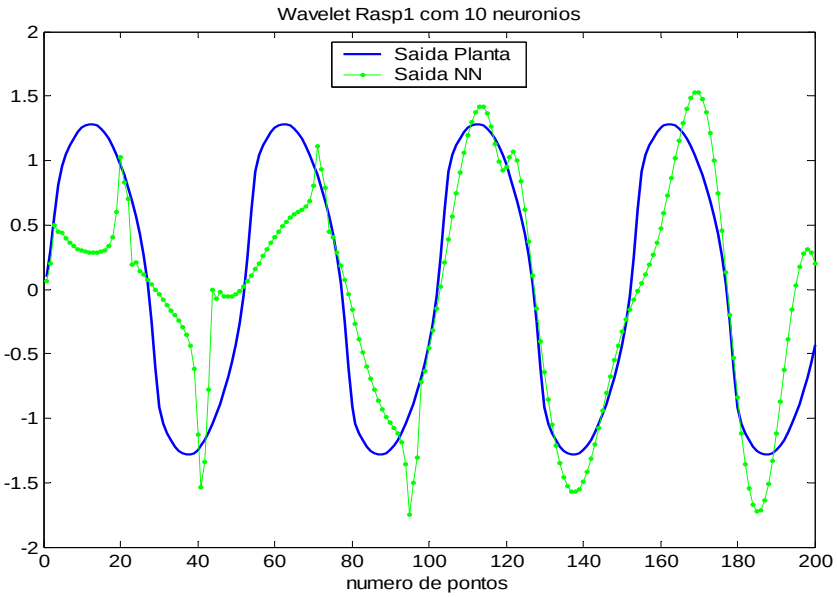


Figura 6.12: Identificação com *Wavelet* Rasp1.

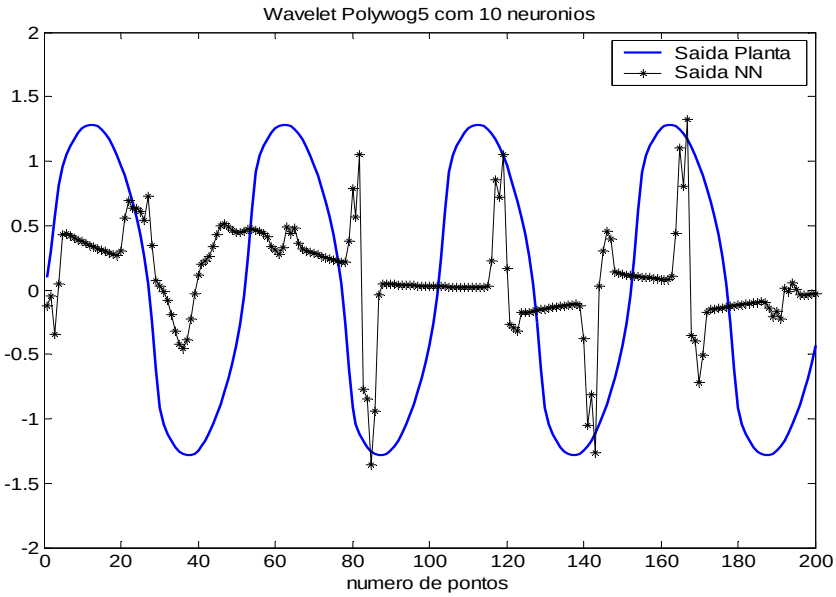


Figura 6.13: Identificação com *Wavelet* Polywog5.

A fig. 6.14 mostra o comportamento do erro normalizado a cada iteração para os três tipos de funções de ativação utilizados na rede *wavenet*.

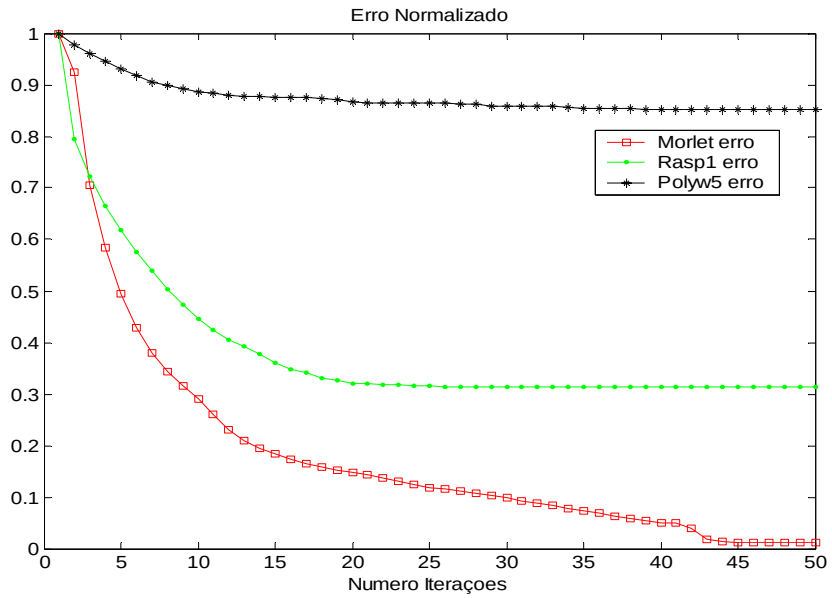


Figura 6.14: Comportamento do Erro considerando 3 funções *Wavelet*.

Os gráficos apresentados anteriormente mostraram a importância da escolha das funções de ativação. A escolha da função está diretamente associada à estrutura ou forma do sinal a ser analisado. No entanto, apesar das *wavelet*s serem bastante versáteis, possibilitando deslocar a *wavelet* ao longo do sinal, e também de perceber regiões com alta e baixa freqüência, adaptando o seu parâmetro de escala (contraindo e dilatando a *wavelet*), em alguns casos a incompatibilidade da janela do sinal e a *wavelet* fica muito acentuada, tornando a identificação bastante onerosa e lenta. A diferença entre utilizar uma *wavelet* ou outra, fica evidente e é decisiva para melhorar a eficiência do algoritmo *wavenet* e levar a uma boa convergência da rede.

6.4 Identificação e Controle de Plantas Não Lineares.

O tipo de controle discutido no trabalho caracteriza-se por um controle inverso/indireto, que utiliza as informações dos parâmetros da planta na tomada de decisões. O que difere este trabalho da maioria dos trabalhos apresentados na área de redes neurais e controle, é que, estes geralmente fazem um pré-treinamento da rede neural com diferentes padrões de entrada, ou seja, são apresentados diversos padrões para a rede e a mesma é treinada para posterior reconhecimento do padrão de referência. Neste trabalho isso não acontece, pois a rede utilizada para identificar a planta é treinada com base em um único padrão de referência (saída da planta). O processo de identificação ocorre através da minimização do erro entre o sinal de saída da planta e o sinal de saída da rede, e os parâmetros da rede treinada são utilizados para projetar o neurocontrolador.

Desse modo, tem-se duas fases distintas, a primeira refere-se à identificação, em que são obtidos os parâmetros do modelo, e a segunda refere-se ao controle propriamente dito, em que são utilizados os parâmetros obtidos na identificação para projetar o controlador, conforme o esquema da figura 6.15, inicialmente apresentada no capítulo 5.

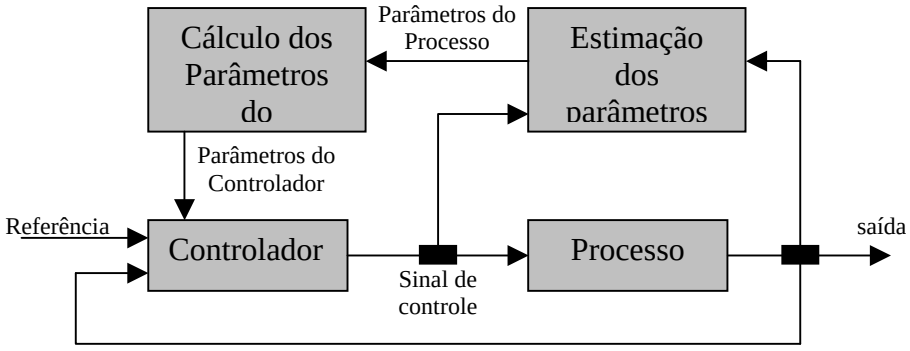


Figura 6.15: Esquema de controle adaptativo indireto.

Seja um sistema dinâmico genérico de uma única entrada e uma única saída (SISO), representado pelas equações de estado:

$$x(k+1) = f\left(x(k), u(k), k\right)$$
$$y(k) = g\left(x(k), k\right)$$

Admitindo que as funções f e g são desconhecidas, os únicos dados acessíveis são *u* e *y*, que correspondem a entrada e a saída da planta respectivamente, sendo que *u* também é entrada da rede.

O modelo da planta, apresentado no capítulo 5, pode ser expresso em termos de *f* e *g*, pela expressão abaixo.

$$y(k+1) = F\left(y(k),\ y(k-1),\ldots, y(k-n+1),\ u(k),\ u(k-1),\ldots,\ u(k-n+1)\right) \quad +$$

$$G\left(y(k), y(k-1),\ldots, y(k-n+1), u(k), u(k-1),\ldots, u(k-n+1)\right) \cdot u(k)$$

Em uma forma mais compacta tem-se:

$$y(k+1) = F\left(y(k)\right) + G\left(y(k)\right) \cdot u(k)$$

Se as funções $F(\cdot)$ e $G(\cdot)$ são conhecidas exatamente, então, o controle $u(k)$ necessário para encontrar a saída desejada, $r(k+1)$, é calculado em cada instante de tempo, pela equação a seguir.

$$u(k) = \frac{r(k+1) - F\left(y(k)\right)}{G\left(y(k)\right)}$$

Entretanto, $F(\cdot)$ e $G(\cdot)$ são desconhecidos e assim, o enfoque é a busca por uma rede neural adaptativa capaz de identificar as funções $\hat{F}(\cdot)$ e $\hat{G}(\cdot)$ de forma que o modelo se aproxime do sistema dinâmico, isto é,

$$\hat{y}(k+1) = \hat{F}\left(y(k), \Theta_{\Phi}\right) + \hat{G}\left(y(k), \Theta_{\Gamma}\right) u(k)$$

Comparando a expressão acima com a equação (5.20), tem-se:

$$\hat{F}\left(y(k), \Theta_{\Phi}\right) = \sum_{j=1}^{N} d_j \, \hat{y}(k-j) \, v(k) \text{ e}$$

$$\hat{G}\left(y(k), \Theta_{\Gamma}\right) = \sum_{i=0}^{M} c_i \, z(k-i)$$

Substituindo as funções $F(\cdot)$ e $G(\cdot)$ pelas suas respectivas aproximações $\hat{F}(\cdot)$ e $\hat{G}(\cdot)$, o controle $u(k)$ necessário para seguir a saída desejada $r(k+1)$ será calculado em cada instante pela expressão a seguir.

$$u(k) = \frac{r(k+1) - \hat{F}\left(y(k), \Theta_{\Phi}\right)}{\hat{G}\left(y(k), \Theta_{\Gamma}\right)}$$

em que Θ_{Φ} e Θ_{Γ} sintetizam os parâmetros envolvidos no processo de ajuste e adaptação.

6.4.1 Aplicação a problemas de identificação e controle de sistemas dinâmicos.

O modelo de planta utilizado para avaliar o controlador é um sistema não linear que possui uma não linearidade dinâmica. O sistema descrito pela equação (6.3), exibida anteriormente e reproduzida abaixo, é excitado por uma entrada linear, *u(n) = 0.8 sin(2πn/50),*com *n* = 0, 1, 2, ..., 200.

$$y(n) = \frac{y(n-1)}{1 + y^2(n-1)} + u(n)$$

A configuração da rede para identificação, utilizando o algoritmo *wavenet* com filtro IIR acoplado, é formada por 20 neurônios na camada intermediária, ativados por *wavelets* do tipo Morlet. A escolha dessa configuração baseou-se nos resultados anteriores em que foram considerados outros tipos de *wavelet*s como função de ativação, e a *wavelet* Morlet foi a que apresentou os melhores resultados. A *wavenet* com filtro IIR acoplado é utilizada porque o modelo de controle adotado depende dos coeficientes *c* e *d* do filtro IIR, que são ponderações dos atrasos do sinal de saída da rede.

Os resultados da fase de identificação e posteriormente os resultados obtidos com o controlador serão mostrados a seguir.

A figura 6.16 mostra a superposição da saída da planta e o sinal identificado, e a figura 6.17 apresenta o comportamento do erro ao longo do processo de minimização que utiliza o método de Levenberg-Marquadt como método de otimização.

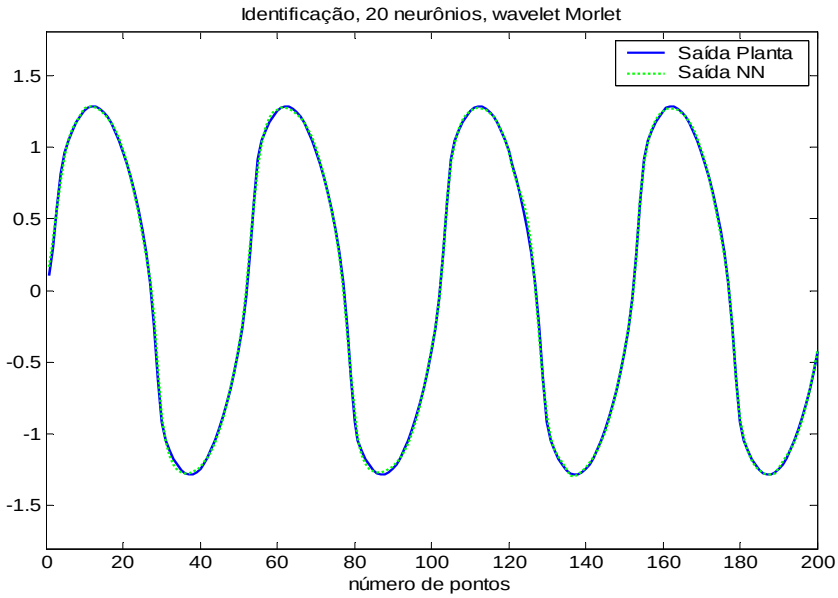


Figura 6.16: Resultado de Identificação

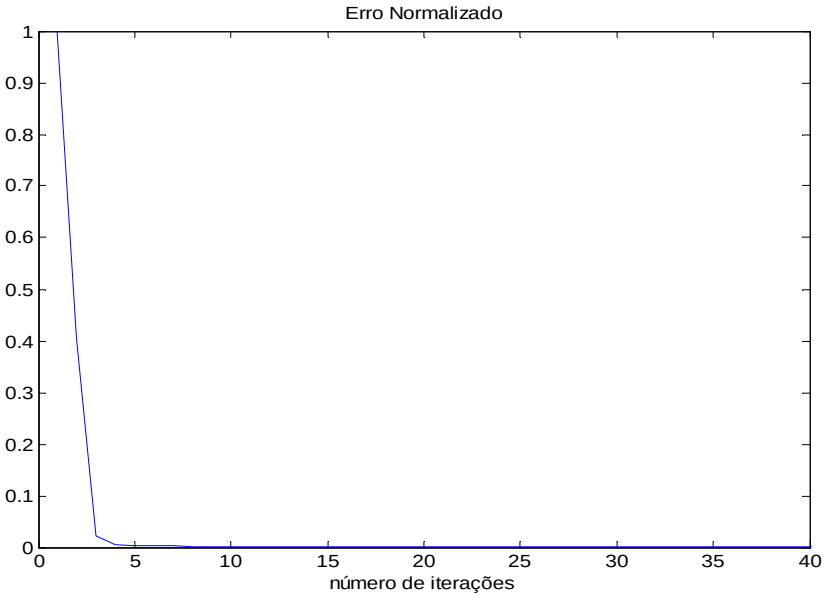


Figura 6.17: Comportamento do Erro a cada Iteração.

Considerando que o processo de identificação dos parâmetros foi satisfeito, e que o erro esperado foi alcançado, o neurocontrolador é projetado baseado no modelo identificado.

As figuras a seguir mostram os resultados do controle aplicado em uma planta não linear, cuja função é controlar a entrada da planta u(k) para que a sua saída siga um sinal de referência r(k). Serão apresentadas também algumas variações na forma do sinal de referência, com o intuito de analisar o comportamento do controlador a essas mudanças.

O primeiro sinal de referência que será aplicado no controlador é representado pela equação (6.4), e a função do controlador é gerar uma entrada para planta que resulte em uma saída que acompanhe o sinal de referência.

$$r(k) = \partial(k-200) - 1.5\ \partial(k-300) + 1.5\ \partial\ (k-400) \qquad (6.4)$$

$k = 200,\ 201\ldots,\ 500$, em que

$$\partial(k) = \begin{cases} 1 & \text{para} \quad k \geq 0 \\ 0 & \text{para} \quad k < 0 \end{cases}$$

A Figura 6.18 ilustra a ação do controlador, onde inicialmente foi realizada a identificação do processo referente aos primeiros 200 pontos do sinal. Uma vez identificado o modelo, passou-se a ação do controle a partir do ponto 200.

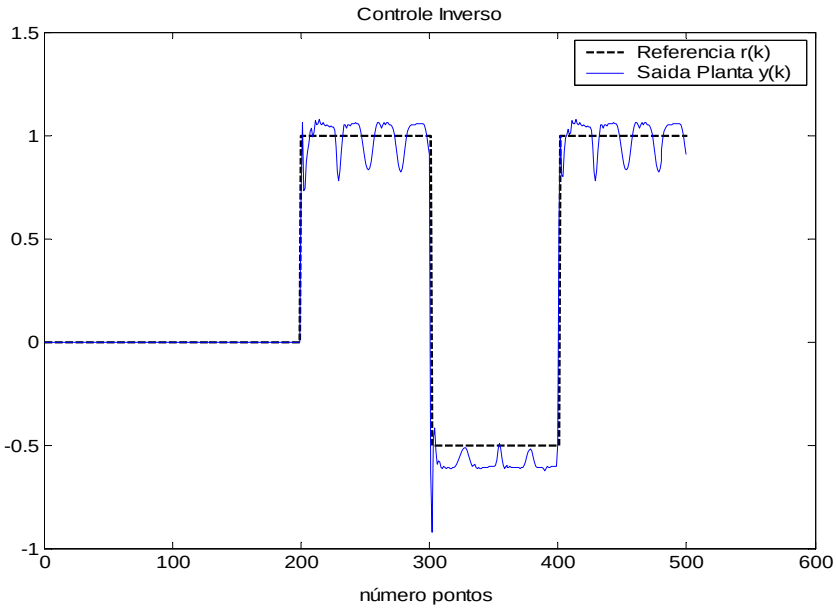


Figura 6.18: Ação de Controle u(k)

O gráfico mostra que o controlador não apresenta um bom comportamento, oscilando bastante em torno da referência. Uma possível solução seria realizar um pré-treinamento da rede, o que poderia amenizar esse tipo de comportamento ou então, acoplar um outro controlador do tipo PID (Proporcional, Integral e Derivativo), convencional, para suprir essas oscilações e os erros de regime.

Diferentes formas do sinal de referência r(k) foram utilizadas para verificar como o controlador se comporta. Foram realizadas algumas alterações no sinal r(k), da equação (6.4) em relação à amplitude do sinal e ao intervalo de variação do sinal de referência (degrau), como pode ser acompanhado nas figuras 6.19 e 6.20 a seguir.

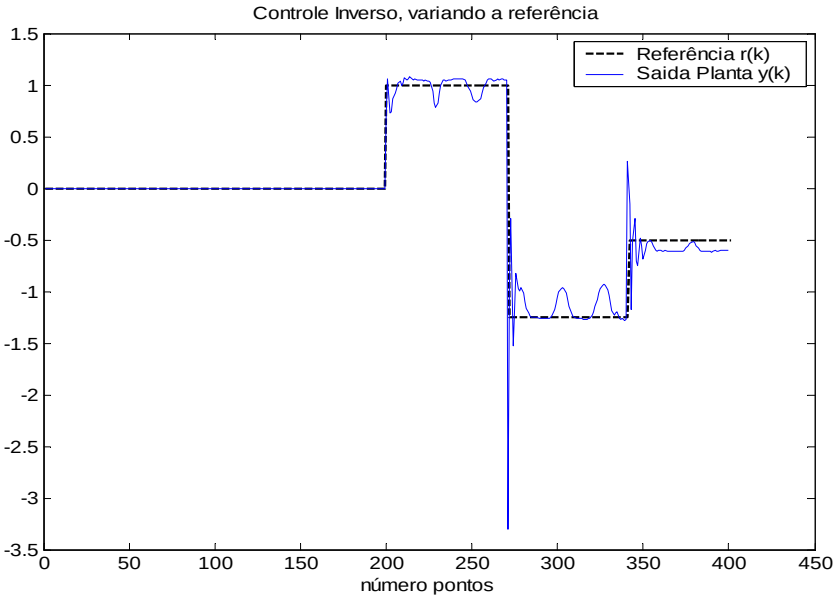


Figura 6.19: Ação de Controle u(k), variando a Referência.

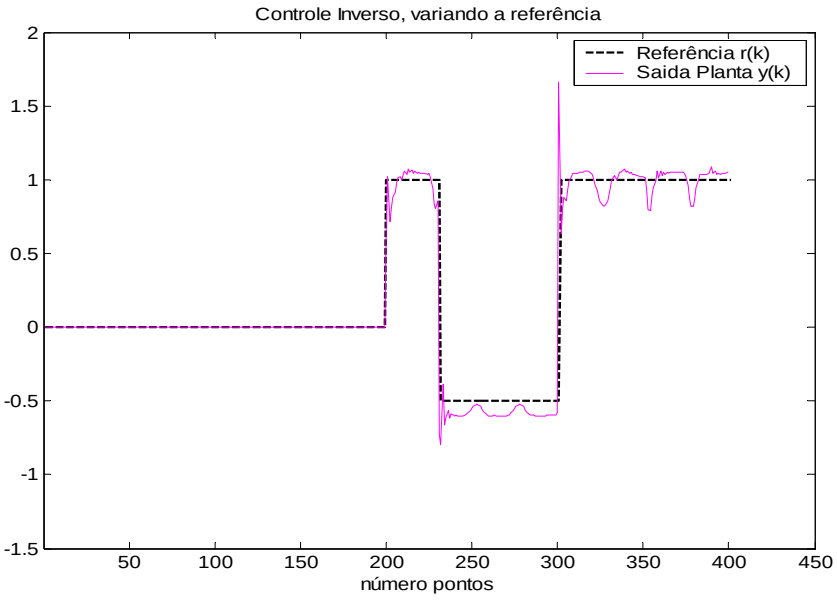


Figura 6.20: Ação de Controle u(k), variando a Referência.

Nota-se que o controlador responde de maneira diferente para cada sinal de referência, ficando mais instável em alguns casos do que em outros. Uma razão para essa variação da ação de controle, provavelmente está associada a ausência de treinamento prévio do controlador.

Um outro aspecto que chama atenção e que teoricamente era “esperado”, é a instabilidade do controlador, visto que o ajuste dos parâmetros da rede *wavenet* nem sempre produzem um

sistema inversível. A técnica de identificação e controle por redes neurais não estabelece nenhum cuidado a respeito da estabilidade.

A figura 6.21 mostra este tipo de problema que ocorreu utilizando os parâmetros ajustados, apresentando um comportamento totalmente incoerente e tendendo ao infinito.

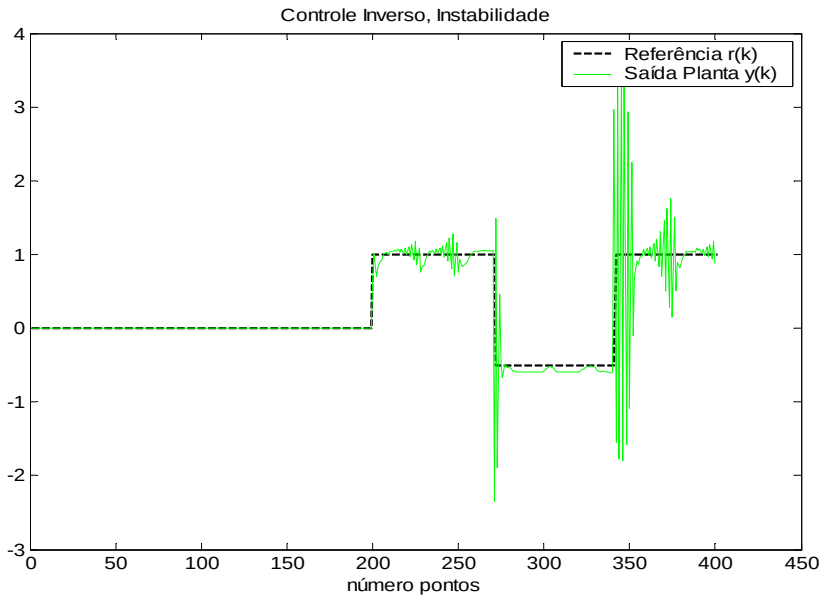


Figura 6.21: Instabilidade do Controle u(k).

Foram realizados ainda testes para sinais de referência com um comportamento mais suave, tanto com excitação linear como não linear. A referência utilizada neste caso representa a expressão de um sistema dinâmico linear, com uma excitação não linear, dada pela expressão (6.5).

$$r(k) = 0.3r(k-1) - 0.6r(k-2) + f(u(k)) \tag{6.5}$$

em que, $f(u) = u^3 + 0.3u^2 - 0.4u$ e $u(k) = \sin(2\pi k/250)$, $k = 200, 201, 202, ..., 400$.

A figura 6.22 mostra a superposição da saída da planta a ação do controle *u(k)*, tentando seguir o sinal de referência *r(k)*, sendo que os primeiros 200 pontos são referentes a identificação.

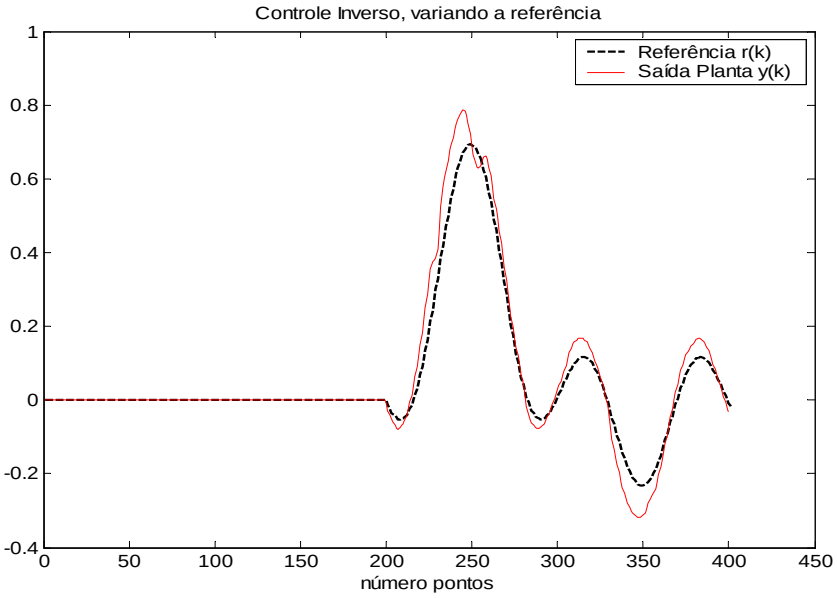


Figura 6.22: Ação de Controle u(k), variando a Referência.

Uma outra referência analisada é referente a um sistema não linear que possui uma não linearidade dinâmica excitada por uma entrada linear. O sistema é descrito pela equação diferença não linear, eq.(6.6).

$$r(k) = \frac{r(k)*r(k-1)*(r(k)+2.5)}{1+r^2(k)+r^2(k-1)} + u(k) \qquad (6.6)$$

sendo $u(k) = sin(2\pi k/50)$, k = 200, 201, 202, ..., 400.

A figura 6.23 mostra o comportamento da saída da planta à ação do controlador. É observado que a saída da planta aproxima-se da referência, apresentando oscilações apenas na região inferior do sinal de referência, região em que o sinal apresenta uma maior variação no tempo.

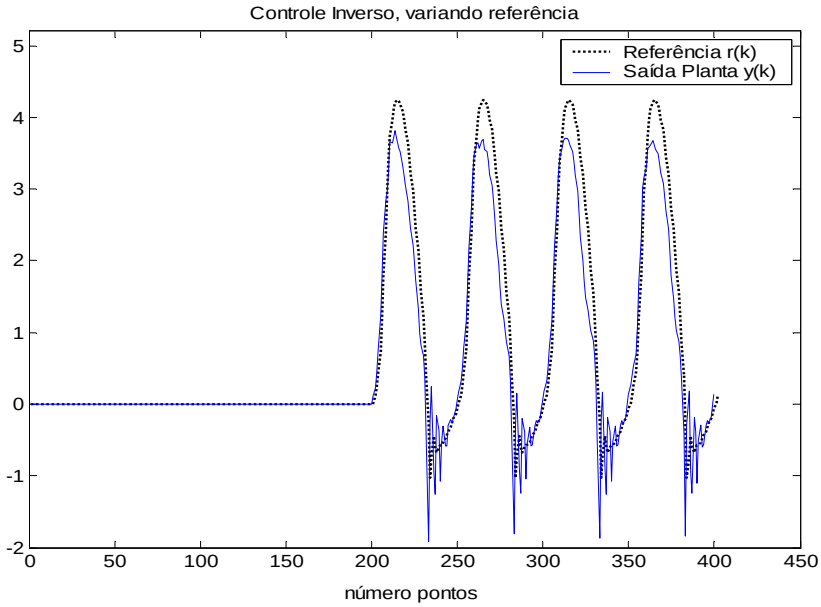


Figura 6.23: Ação de Controle u(k), variando a Referência.

O comportamento do controlador em relação aos vários tipos de referências é bem distinto, e mostram-se mais estáveis a ação do controle as referências com formas mais suaves do que as referências com comportamento em degrau. Um outro ponto que ficou evidente observando os gráficos, é que o controlador em algumas situações não produz uma entrada eficiente o bastante para a planta seguir exatamente a referência. Esse problema seria amenizado se o controlador fosse treinado previamente para vários tipos de sinais de referência, considerando que eles pudessem excitar todos os modos da planta. No entanto fica claro a necessidade de ajuste dos parâmetros do controlador, e a necessidade de se trabalhar em malha fechada, o que não é trivial se optarmos por um tratamento on-line.

A seguir é apresentado na figura (6.24) o resultado obtido fazendo-se o ajuste dos parâmetros do controlador (pesos w_k, escalas a_k, translações b_k, coeficientes feedforward IIR c_k e coeficientes feedback IIR d_k), considerando o ajuste em uma janela de 200 pontos, para o sinal de referência dado pela equação (6.4). O ajuste é realizado minimizando-se o erro entre o sinal de referência que entra no controlador e o sinal de saída da planta, utilizando-se o método de otimização de Levenberg-Marquadt.

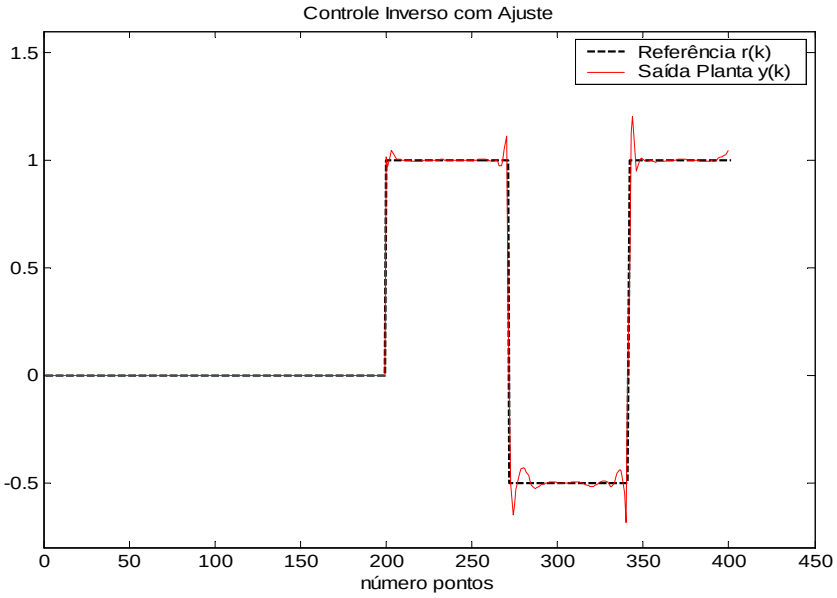


Figura 6.24: Ação de Controle u(k) com Ajuste dos Parâmetros.

Com o ajuste dos parâmetros do controlador, foi possível realizar uma ação de controle mais eficiente, atenuando as oscilações excessivas apresentadas sem o ajuste.

O resultado mostrou uma melhora do controle com ajuste, mas os testes não são suficientes para uma análise conclusiva. Avaliar o controlador com um ajuste ponto a ponto e confrontá-lo com um ajuste por janela, entre outros testes, seria uma boa opção para avaliar e entender melhor as oscilações nas extremidades do degrau.

Capítulo 7

7.1 Conclusões

A utilização de redes neurais para a identificação e controle de sistemas tem apresentado um grande avanço em vários setores da engenharia, entretanto, o tratamento de sistemas mais complexos, principalmente envolvendo sistemas com não linearidades ainda necessitam de uma maior atenção. Neste caso, o desenvolvimento de um modelo de rede neural apresenta vários pontos de estudo que podem melhorar a eficiência da rede, tais como, tipo de treinamento, arquitetura da rede (número de neurônios e número de camadas escondidas), escolha das funções de ativação entre outras peculiaridades que dificultam a definição de um modelo de rede neural que engloba ou que possa ser utilizado por mais de um problema.

As *wavelet*s que são funções diferenciáveis e comumente utilizadas em análise de sinais, trouxeram grande avanço no tratamento de sinais não estacionários, cuja transformada do sinal obtém boa resolução tanto no tempo quanto em freqüência, solucionando algumas deficiências da transformada de Fourier. A transformada *wavelet* usa uma janela com escala variável que possibilita analisar faixas de alta e baixa freqüência em um mesmo sinal, e é exatamente essa capacidade de manipulação de sua janela que a torna uma boa alternativa como função de ativação, pois em conjunto com a rede neural ela realiza uma análise ou mapeamento local do sinal, proporcionando uma identificação mais eficiente principalmente para sinais não lineares e variantes no tempo.

Ao longo do trabalho tentou-se expor de maneira sucinta as técnicas de redes neurais e transformada *wavelet* e como essas técnicas podem ser combinadas para formar uma rede neural adaptativa, denominada *wavenet*, trazendo dessa forma, benefícios na área de identificação e controle de sistemas não lineares. Vários testes foram realizados utilizando o algoritmo *wavenet* e diferentes famílias de *wavelet*s e estruturas de rede foram avaliadas para identificação de plantas não lineares. Uma das observações notadas no algoritmo é que utilizar o método de

otimização gradiente descendente para minimizar a função custo, não garante uma aproximação adequada e deixa o treinamento da rede muito lento e ineficiente, já que somente a informação da primeira derivada (ou gradiente) do erro de treinamento é utilizada ao longo do processo iterativo de otimização.

Em se tratando de um controle baseado na inversa da planta, a etapa de identificação é de fundamental importância na obtenção de um bom controle. Dessa forma é decisivo que a etapa de identificação busque obter melhores resultados. Uma saída foi encontrar um novo método de otimização que minimize a função custo (erro) com mais eficiência. Para acelerar o processo de treinamento, alguns algoritmos de otimização de segunda ordem, baseados no gradiente e na informação da segunda derivada (ou Hessiana) foram pesquisados.

Um estudo comparativo entre os métodos de otimização para treinamento da rede neural mostrou um desempenho superior do método de segunda ordem Levenberg-Marquadt em relação ao método do gradiente descendente, apresentando-se mais rápido e preciso. Geralmente os métodos de otimização de segunda ordem proporcionam desempenho superior àqueles de primeira ordem, no entanto o cálculo da Hessiana, necessário em alguns métodos, apresenta um custo computacional adicional, além da possibilidade de existência de singularidades, que dificulta, ou mesmo impossibilita o cálculo da inversa.

As redes neurais ativadas por *wavelets* apresentaram resultados promissores. Entretanto, o sucesso da identificação, depende fortemente da escolha da família *wavelet* no papel de função ativadora e do número de neurônios, necessários para cobrir eficazmente toda a extensão do sinal (planta). Essas escolhas ainda são feitas de forma heurística e a definição de uma arquitetura apropriada da rede neural não é um trabalho trivial e, na maioria dos casos, depende da experiência do usuário.

Mesmo obtendo bons resultados na identificação o controle inverso apresentou algumas limitações. A mais evidente constatada diz respeito à falta de realimentação do processo. Se o controlador não é um modelo inverso preciso do sistema, este pode levar a ações de controle incoerentes, que conduzem a erros que não podem ser detectados e corrigidos sem realimentação. Aprender completamente a dinâmica inversa de um sistema de ordem elevada não é trivial, e mesmo obtendo-se o verdadeiro modelo do sistema, nem todo sistema é inversível, o que ficou claro na apresentação dos resultados, evidenciando a instabilidade em alguns casos, justificando então o emprego de realimentação junto à malha de controle.

Um aspecto importante observado no controle inverso foi o seu comportamento a diferentes formas de referências, mostrando-se mais eficiente em alguns casos do que em outros.

Esse problema poderia ser melhor absorvido, realizando um pré treinamento da rede a partir de uma redefinição do sinal de entrada que propiciasse uma excitação mais adequada das dinâmicas do processo não-linear em toda a região de interesse.

De forma geral, o trabalho conseguiu abordar os principais aspectos referentes à utilização de redes neurais e funções *wavelets* na identificação e controle de plantas não lineares. Apresenta resultados que deverão nortear o desenvolvimento de novos estudos nesta área de pesquisa, e tem a função de despertar interesse na busca por tecnologias e técnicas que desenvolvam a inteligência artificial.

7.2 Trabalhos Futuros

Um das questões importantes fundamentalmente verificadas no trabalho é a estabilidade do processo. A utilização de redes neurais artificiais na identificação e controle do processo não garante a estabilidade, pois a análise de estabilidade do processo não é avaliada pela rede neural. Uma proposta a ser estudada para contornar essa limitação seria a utilização do segundo método de Lyapunov juntamente com a rede.

Um outro aspecto que pode ser tratado para melhoria no processo de identificação e controle seria realizar um treinamento considerando um projeto mais elaborado do sinal de entrada, para uma excitação mais adequada das dinâmicas do processo não-linear em toda a região de interesse.

E finalmente, a comparação da técnica utilizada no trabalho com técnicas alternativas para solução de problemas em identificação e controle de sistemas não lineares e um confronto dos resultados simulados com um problema real.

Referências

ABBATE, A.; KOAY, J. Ultrasonics, Ferroeletrics, And Frequency Control . **IEEE Trans**., v. 44, p. 14, 1997.

ÅSTROM, K.J.; WITTENMARK B. **Adaptive Control. Reading**, Ma: Addison-Wesley, 1994.

ÅSTRÖM, K. J. E WITTENMARK, B. **Adaptive Control**, 1995.

ÅSTRÖM, K. J Adaptive Control Around 1960. **IEEE Control Systems**, v. 16, n. 3, p. 44-49, 1996.

CAMARGOS, H. S. AND KHATER, E. **Identificação de Sistemas Não Lineares através de Redes Neurais usando o Método de Levenberg-Marquardt**, 2001.

CERQUEIRA, E. O; POPPI, R. J.; KUBOTA, L.; MELLO. C.; QUIM. **Nova**, v. 23, p. 690, 2000.

COELHO, L. S.; COELHO, A. A. R. Computação Evolucionária em Identificação e Controle de Processos: Fundamentos, Análise e Aplicações, In: **III Congresso Argentino de Ciências de La Computación, IV Workshop Sobre Aspectos Teóricos de La Inteligencia Artificial-Teoría,** La Plata, Argentina, p. 1063-1079, 1997a.

COELHO, L. S.; COELHO, A. A. R. Metodologia Para Ensino de Automação: Uma Visão dos Aspectos Teóricos, Práticos E Pedagógicos. In: **XXV Congresso Brasileiro de Ensino de Engenharia,** Salvador, BA, v. 1, p. 227-242, 1997B.

COELHO, L. S.; SILVA, A. C.; COELHO, A. A. R.; Genetic Algorithms And Evolution Strategies Applied In Identification And Control: Case Study. In: Chawdhry, P. K.; Roy, R.; Pant, R. K. (Eds.). **Soft Computing In Engineering Design And Manufacturing**. Chapter 8: Dynamic System, Identification And Control, Springer-Verlag: London, p. 430-438, 1998a.

COELHO, L. S.; SILVA, A. C.; COELHO, A. A. R.; Siscon ¾ Ferramenta de Ensino e Aplicação em Tempo Real de Controle PID Convencional e Adaptativo. In: **XII Congresso Brasileiro de Ensino de Engenharia,** São Paulo, SP, p. 324, 1998b.

COELHO, L. S.; SILVA, A. C.; COELHO, A. A. R.; **Automatic Tuning of PID and Gain Scheduling PID Controllers By A Derandomized** (Coelho, L. S.; Silva, A. C.; Coelho, A. A. R.) Evolution Strategy. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, v. 13, n. 5, p. 365-373, 1999.

COELHO, L. S. **Identificação e Controle de Processos Multivariáveis Via Metodologias Avançadas E Inteligência Computacional**, Tese de Doutorado.

DAUBECHIES I.; Ten Lectures on Wavelets. **CBMS-NSF Regional Conference Series in Applied Mathematics,** Philadelphia, Penssynlvania, 1992.

DAUBECHIES, I.; "Where Do Wavelets Come From?-A Personal Point of View", **IEEE Special Issue on Wavelets,** v.84, n.4, p. 510-513, April 1996.

GORINEVKY, D. M.; Modeling of Direct Motor Program Learning. **Fast Human Arm Motions. Biological Cybernetics**, v. 69, p. 219-228, 1993.

GOTTLIEB, D.; SHU, C. W.; **Siam Review**, v.39, p. 644, 1997.

GRASSI L. H.; RIGHETO E.; PEREIRA J. A.; Nonlinear Plant Identification by Wavelets **COBEM2003**, São Paulo, 2003.

GUPTA, M. M.; RAO, D. H.; Neuro-Control Systems: A Tutorial. In: Neuro-Control Systems: Theory and Applications. **IEEE** Press: Piscataway, NJ, USA. p. 1-43, 1994.

HAGAN, M. T.; DEMUTH, H. B.; Neural Networks for Control. **American Control Conference**, San Diego, Ca, Usa p. 1642-1656, 1999.

HARRIS, C. J.; BROWN, M.; BOSSLEY, K. M.; MILLS, D. J.; MING, F.; Advances in Neurofuzzy Algorithms for Real-Time Modeling and Control. **Engineering Applications of Artificial Intelligence**, v. 9, n. 1, p. 1-16, 1996.

HARRIS, C. J.; MOORE, C. G.; BROWN, M.; Intelligent Control: Aspects of Fuzzy Logic and Neural Nets. **World Scientific Series** In Robotics and Automation Systems, v. 6, World Scientific: London, 1993.

HAYKIN, S.; **Neural Networks: A Comprehensive Foundation**, Prentice-Hall, Upper Saddle River, New Jersey, USA, 1994.

HO, D. W. C., ZHANG PING-AN, AND XU JINHUA; Fuzzy Wavelet Networks for Function Learning. **IEEE** Transactions On Fuzzy Systems, v. 9, n. 1, February, 2001.

HORNIK, K. M. STINCHCOMBE, AND WHITE, H.; "Multilayer Feedforward Networks Are Universal Approximators." **Neural Networks**, v. 2, p. 359-366, 1989.

HUNT, K.J., SBARBARO, D., ZBIKOWSKI, R.,GAWTHROP, P.J. Neural Networks For Control Systems – **A Survey. Automática**, v. 28, n. 6, p. 1083-1112, 1992.

HWANG, J.N., LEWIS, P.S. From Nonlinear Optimization to Neural Network Learning. **Conference on Signals Systems Computation**. Pacific Grove, CA, p. 985-989, November 1990.

IANET, Inteligência Artificial Disponível Em : **<http:// www.lokall.com/ianet/rdneuro.htm>**. Acesso Em 28 De Janeiro 2004.

ISIDORI A.; Nonlinear Control Systems, Springer-Verlag, 1989.

KAWATO M. **Adaptation and Learning in Control of Voluntary Movement by the Central Nervous System**. Advanced Robotics, v. 3, n. 3, p. 229-249, 1989.

KOHONEN T. **Self-Organizing Maps,** 2º ed. Springer Series in Information Sciences, 30. Springer Verlag, June, 1997.

LEKUTAI, G.; **Adaptive Self-Tuning Neuro Wavelet Network Controllers**, Dissertation, 1997.

LEVIN A. U. AND NARENDRA K. S.; "Control of Nonlinear Dynamical Systems Using Nerual Networks, Part II: Observability, Identification and Control," **IEEE** Transactions on Neural Networks, 1995.

LIMA C. A. M.; **Emprego de Teoria de Agentes no Desenvolvimento de Dispositivos Neurocomputacionais Híbridos e Aplicação ao Controle e Identificação de Sistemas Dinâmicos**, Dissertação, 2000.

LJUNG L.; System Identification: Theory for the user. **Prentice-Hall** : New York, 1987.

MAITELLI, A. L. E REZENDE J. A. D.; "Um Neurocontrolador Com Treinamento Em Tempo Real Aplicado A Uma Planta de Temperatura". **V Simpósio Brasileiro de Automação Inteligente**, Canela, RS, 2001.

MCCARTHY, J.; A Basis for Mathematical Theory of Computation, p. 33-70. In P. Braffort And D Herschberg, Editors, **Computer Programming And Formal Systems**, p. 33-70, 1963.

MCCULLOCH W. S., AND PITTS W. "A Logical Calculus of the Ideas Immanent In Nervous Activity", Bulletin of Mathematical Biophysics, n. 9, p. 127-147, 1943.

MEYER, Y.; Wavelets-Algorithms and Applications; **SIAM**; Philadelphia, 1993.

MILLER, W.T., SUTTON, R.S., WERBOS, P.J; Neural Networks for Control. **MIT Press**. Cambridge, Massachusettes, 1990.

MINSKY, M.L; Steps Towards Artificial Intelligence. **Proceedings of the Institute of Radio Engineers**, v. 49, p. 8-30, 1961.

MINSKY, M.; PAPERT, S.; Perceptrons. **MIT Press**: Cambridge, MA 1969.

MORLET, J. AND GROSSMANN, A.; Decomposition of Hardy Functions into Squared Integrable Wavelets of Constant Shape. **SIAM J. Math. Analysis**, 15:723-736, 1984.

MOUTINHO E BIONDI; "Método de Pré-Processamento de Sinais Aplicados ao Treinamento de Redes Neurais Artificiais". **II Congresso Brasileiro de Computação** – Cbcomp., 2002.

NARENDRA, K.S., ANNASWAMY, A.M.; **Stable Adaptive Systems**. Englewood Cliffs, NJ, Prentice-Hall, 1989.

NARENDRA, K. S. AND PARTHASARATHY, K.; "Identification and Control of Dynamical Systems Using Neural Network," **IEEE** Transactions on Neural Networks, v.1, n.1, p 4-27, March , 1990

NARENDRA K. S., BALAKRISHNAN J. AND CILIZ M. K.; "Adaptation and Learning Using Multiple Models, Switching, and Tuning," **IEEE** Control Systems Magazine, V15, N3, Pp 37-51, June 1995.

NELSON, M. M.; ILLINGWORTH, W. T.; **A Practical Guide to Neural Nets**. Addison-Wesley: Reading, MA, USA, 1991.

NEUMANN J. V.; **The Computer and the Brain**, 1958.

NEWELL A., SIMON H.A.; Human Problem Solving. **Prentice-Hall**, 1972.

OLIVEIRA R. C. L.; **Rede Neural Com Dinâmica Interna Aplicada A Problemas de Identificação e Controle Não-Linear**. Tese de Doutorado, UFSC, 1998.

ORTEGA J. G., CAMACHO E. F.; **Mobile Robot Navigation. A Partially Structured Static Environment, Using Neural Predictive Control**. Control Engineering Practice, v. 4, n. 12, p. 1669-1679, 1996.

PATI Y. AND KRISHNAPRASAD P.; "Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transformation," **IEEE** Trans. Neural Networks, v.4, n.1, p. 73-85, Jan. 1993.

PHILLIPS S. M., MÜLLER-DOTT C.; Identification and Control. **International Journal of Analog Integrated Circuits and Signal Processing**, v. 2, n. 4, p. 353-363, 1992.

QIAN S., CHEN D.; Joint Time-Frequency Analysis-Methods and Applications. **Prentice Hall** PTR; Upper Saddle River, 1996.

REDGERS A., ALEKSANDER I.; Digital Neural Networks, In: IRWIN, G. W.; Warwick, K.; Hunt, K. J. (Eds.). Neural Network Applications in Control, **IEEE** Control Engineering Series 53, The Institution Of Electrical Engineers. Chapter 2, p. 17-32, 1995.

RIBEIRO B.; **Computação Adaptativa Redes Neuronais**: Aula 8, 3º Ano de Engenharia Informática, disciplina 1º semestre 2002/ 2003, Departamento de Engenharia Informática, DEI-FCTUC, 2002/2003

RIGHETO E.; **Identificação de Plantas Não Lineares Via Wavenets**. Unesp-Ilha Solteira, Dissertação de Mestrado, 2003.

RONCO E., GAWTHOP P. J.; **Modular Neural Neural Networks**: A State of The Art. Technical Report CSC-95026. Centre for System and Control. Faculty of Mechanical Engineering, University of Glasgow, UK, 1995. Available at http://www.mech.gla.ac.uk/~ericr/pub/surveymnm.ps.

RONCO E., GOLLEE H., GAWTHROP P. J.; **Modular Neural Network and Self Decomposition**. Technical Report CSC-96012. Centre for System and Control, University. of Glasgow, UK, 1996. Available At Www.Mech.Gla.Ac.Uk/~Ericr/Pub/Mnsd.Ps.Z.

RONCO E.; GAWTHROP P. J.; **Neural Networks for Modelling and Control.** Glasgow, UK. Technical Report CSC97008, Centre for System and Control, Department of Mechanical Engineering, University Of Glasgow, 1997.

RUMELHART D. E., HINTON G. E., WILLIAMS R. J.; Parallel Distributed Processing: Explorations In The Microstructure Of Cognition. v. 1, **The MIT Press**: Cambridge, MA, USA, 1986.

RUSSELL S. J., NORVIG P.; Artificial Intelligence: A Modern Approach, Englewood Cliffs, NJ, **Prentice Hall**, 1995.

SCHMIDT R. A.; Motor Control and Learning. **Human Kinetics Publishers**. Champaign, Illinois, 1982.

SHOURESHI R. A.; Intelligent Control Systems: Are They For Real? Transactions of the ASME, J**ournal of Dynamic Systems**, Measurement, and Control, v. 115, p. 392-401, 1993.

SJÖBERG J.; **Non-Linear System Identification with Neural Networks**. Linköping, Sweden. Phd. Thesis, Department of Electrical Engineering, Linköping University, 1995.

STECK J. E., ROKHSAZ K., SHUE S. P.; Linear and Neural Networks Feedback for Flight Control Decoupling. **IEEE Control Systems Magazine**, v. 16, n. 4, p. 22-30, 1996.

TOVAR S. N.; El Control Avanzado En Los Procesos Industriales. **Boletin Informativo de La Associacion Colombiana de Automatica**, n. 3, p. 2-3, Agosto, 1996.

VAN CAN H. J. L., BRAAKE H. A. B., HELLINGA C., KRIJGSMAN A. J., VERBRUGGEN H. B., LUYBEN K. C. A. M., HEIJNEN J. J.; Design and Real Time Testing of a Neural Model Predictive Controller for a Nonlinear System. **Chemical Engineering Science,** v. 50, n. 15, p. 2419-2430, 1995.

VIDYASAGAR M.; Nonlinear Systems Analysis. New Jersey: **Prentice-Hall**, Englewood Cliffs, NJ, 2nd Edition, 1993.

WASSERMAN P. D.; **Advanced Methods in Neural Computing**. Van Nostrand Reinhold: New York, NY, USA, 1993.

WANG S., YEH H.; Self-Adaptive Neural Architecture for Control Applications. **IEEE** International Conference on Neural Networks, v. 3, n. 1, p. 309-314, 1990.

WIDROW B., LEHR M. A.; 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. Proceedings of the **IEEE**, v. 78, p. 1415-1442, 1990.

WILLIS M. J., THAM M. T.; **Advanced Process Control**. Newcastle, UK. Technical Report, Department of Chemical and Process Engineering, University of Newcastle Upon Tyne, UK, 1994.

ZHANG Q. AND BENVENISTE A.; "Wavelet Networks," **IEEE** Trans. Neural Networks, v.3, n.6, p 889-98, Nov. 1992.

ZHANG KUN AND CHEN WEN-KAI; "On-Line Adaptive Controller with Wavelet Neural Network" **IEEE** Trans. Neural Networks, 2002.