UNIVERSIDADE ESTADUAL PAULISTA
FAC. DE ENGENHARIA DE ILHA SOLTEIRA
DEPARTAMENTO DE ENGENHARIA MECÂNICA
DEM/FEIS-UNESP

DETECÇÃO E LOCALIZAÇÃO DE FALHAS ATRAVÉS DE REDES NEURAIS ARTIFICIAIS

Tese apresentada à Faculdade de Engenharia de Ilha Solteira – FEIS/UNESP para obtenção do título de Livre Docente junto ao Departamento de Engenharia Mecânica.

1210001072

Vicente Lopes Junior

ILHA SOLTEIRA
Agosto/ 1998

Proc. 040/98 - NRD 96
UNESP - "CAMPUS" DE ILHA SOLTEIRA
SERVIÇO DE BIBLIOTECA E DOCUMENTAÇÃO

DATA DE CHEGADA	DATA DE TOMBO
29.09.98	30.10.98
REGISTRADO POR	TOMBO
Célia	Te. 1072
AQUISIÇÃO	CLASSIFICAÇÃO
Doação autor R$ 10,00	LD 864 d

3050407 - Controle de sistemas mecânicos

BCpIS - FEIS - UNESP

AGRADECIMENTOS

O trabalho apresentado faz parte de um programa geral de manutenção preditiva que vem sendo desenvolvido, em conjunto, por um grupo de professores e alunos do Departamento de Engenharia Mecânica da UNESP/Ilha Solteira, onde busca-se a detecção, localização e quantificação de falhas em estruturas mecânicas utilizando como ferramenta básica redes neurais artificiais.

Este trabalho apresenta, no capítulo 5, cinco trabalhos relacionados ao assunto. Os trabalhos são apresentados na íntegra, como foram publicados ou encaminhados para publicação, para que fique explicita a participação do grupo referido acima. Mesmo estando explicita a participação destas pessoas, são aqui nominalmente citadas devido a grande contribuição ao trabalho aqui apresentado.

Agradeço portanto os professores co-autores e amigos do Departamento de Engenharia Mecânica, UNESP/Ilha Solteira, Antônio Eduardo Turra, João Antônio Pereira e Max Suel Dutra. Agradecimento aos alunos co-autores e amigos Ádamo de Oliveira do Nascimento e Daniela Demarchi pelo trabalho conjunto e, um reconhecimento especial ao Prof. Hans Ingo Weber, co-autor de um dos trabalhos aqui apresentado e, de alguns outros, que por vários anos tem incentivado nosso grupo para atuar na área de manutenção preditiva e sido nosso consultor nas questões mais difíceis.

Ao Departamento de Engenharia Mecânica, pela amizade e alegria do dia a dia.

BCpIS - FEIS - UNESP

RESUMO

Redes Neurais Artificiais (ANN) tem sido considerada uma ótima ferramenta para aplicações em problemas de reconhecimento de padrões e, portanto pode ser aplicada em problemas de detecção e localização de falhas. Esta técnica oferece vantagens para a solução do problema variacional inverso, devido sua capacidade de interpolação e de reconhecimento de padrões. Para a identificação de falhas no modelo, a rede deve ser treinada usando os dados obtidos a partir do problema variacional direto. Então, as redes poderiam ser especificadas para classificar os padrões de entrada em classes pré definidas ou criar categorias de acordo com sua similaridade. A proposta principal deste trabalho é a implementação de um processo de monitoramento baseado em medidas de vibração. O trabalho apresenta uma proposta alternativa que utiliza a técnica de ajuste de modelos associada a redes neurais para a automação do processo de detecção de falhas. Também é apresentado a utilização de ANN para aplicação em problemas não lineares, para a solução do oscilador de Duffing e para determinar os ângulos de um robô em função da trajetória e dos passos para uma faixa específica de valores treinados. A técnica de ajuste de modelos procura melhorar os modelos obtidos por elementos finitos para que o modelo matemático seja representativo do sistema real. O modelo ajustado permite simular várias condições de falhas, que são necessárias para definir um padrão de entradas que represente cada tipo de falha que possa ocorrer nas condições normais de operação. O aprendizado da rede é realizado utilizando uma compressão no espectro do sinal, para cada tipo específico de falha. O método de compressão, baseado nos picos de amplitude máxima, tem como objetivo reduzir o número de pontos do sinal e construir um novo sinal equivalente e que tenha algumas vantagens em relação ao anterior. O método localiza os pontos de máxima freqüência e cria máscaras do sinal com uma largura de banda para cada pico de freqüência. Isto permite pequenas variações, pois em casos práticos é muito difícil que os dados calculados numericamente sejam exatamente iguais aos dados medidos. Como um exemplo desta metodologia, a detecção de falhas usando redes neurais e ajuste de modelos foi apresentada para o caso de uma estrutura. Outros exemplos mostraram a aplicação de redes em problemas não lineares e, também, a quantificação da falha.

unesp

ABSTRACT

Artificial Neural Networks (ANN) have recently emerged as a good tool for pattern recognition turning their application in promising tools for monitoring and fault classification of machines and equipments. The technique offers promise for solving inverse varational problem in the context of monitoring and fault detection because of their pattern recognition and interpolation capabilities. In order to identify the fault in the model, it must be trained using a set of solutions to its corresponding forward variational problem. In this case, the neural networks should be designed to classify the input patterns in pre-defined classes or to create categories of group patterns according to their similarity. In this work, the main proposal is an implementation of structural health monitoring process automation based on vibration measurements. The work presents an alternative approach which intent is to exploit the capability of model updating techniques associated to neural networks to be used in a process of automation of fault detection. It is, also, presented the application of ANN in nonlinear problems to determine the angles of the locomotor as function of the trajectory and step to a specific range of trained values. The updating approach performs an improvement of the FE-Model in order to define a representative mathematical model of the actual system. The improved model permits to create the various damage conditions required to define the reliable set of information data concerning each kind of failure liable to occur in the life cycle of the system. The learning of the network is performed using a compressed spectrum signal created for each specific type of fault. A compression method based on the maximum amplitudes of the spectrum signal has been used to reduce the number of points of the signal and to construct a new equivalent signature for the system. The method localizes the frequency points related with the maximum amplitudes and creates a masked signal with a band width of the frequency points of maximum amplitudes. This permits to comprise small deviations of the measured FRF(s) since, in practical cases, it is almost impossible to calculate FRF(s) that matches exactly the measured ones. Like an example of this methodology, a fault detection using neural networks and model updating for identifying the presence of structural damage has been presented and demonstrated with an application to a frame structure. Others examples show the application of ANN in nonlinear problems and fault quantification.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

ÍNDICE

I. INTRODUÇÃO 1

II. FUNDAMENTOS DE REDES NEURAIS 5

2.1 Modelo do Neurônio Biológico 5
2.2 Modelo do Neurônio Artificial 7
2.2.1 Redes Neurais de Camada Única 9
2.2.2 Redes Neurais Multicamadas 10
2.2.3. Redes Recorrentes e Não Recorrentes 11
2.3 Função de Transferência 11
2.4 Treinamento de Redes Neurais Artificiais 14
2.4.1 Treinamento Supervisionado 14
2.4.2 Treinamento Não Supervisionado 15
2.4.3 Aprendizagem Reforçada 16
2.3.4. Aprendizagem Competitiva 17
2.5 Algoritmos de Treinamento 17
2.6 Característica da Rede Neural Artificial 19

III. TREINAMENTO E CONFIGURAÇÃO DAS REDE 20

3.1 Perceptrons 20
3.1.1 Eficiência de Armazenagem 21
3.1.2 Aprendizado do Perceptrom 22
3.2 Retropropagação 24
3.3 Configuração de Redes Neurais Utilizando Lógica Fuzzy 29
3.3.1 Fluxograma do Algoritmo Fuzzy 39

IV DETECÇÃO E LOCALIZAÇÃO DE FALHAS 41

4.1 Isolamento, Detecção e Avaliação da Severidade das Falhas 44
4.2 Conceitos de Probabilidade para Reconhecimento de Padrões 45
4.3 Utilização da Transformada de Wavelet para Determinação de Padrões 47

unesp

4.3.1 Transformada de Wavelet 48
4.4 Análise Modal 52
4.4.1 Métodos de Análise Modal Experimental 59

V. RESULTADOS 62
5.1 Automation in Fault Detection Using Neural Network and Model Updating 63
5.2 Modeling of a Bipedal Locomotor Using Neural Network 78
5.3 Identificação de Não Linearidades em Sistemas Dinâmicos Utilizando Redes Neurais Artificiais 82
5.4 Identificação de Rigidez Bilinear em Sistemas Mecânicos Através de Redes Neurais 91
5.5 Diagnosis of Rotating Systems Using Artificial Neural Networks 99

VI CONCLUSÃO 113

BIBLIOGRAFIA 115

I. INTRODUÇÃO

Devido ao aumento na velocidade das máquinas e exigência de eficiência, a análise de sistemas dinâmicos torna-se cada vez mais importante para responder questões relativas as condições de operação das máquinas. Um dos métodos mais utilizados e tradicionais de análise do estado das máquinas é o diagnóstico através dos níveis de vibração. Estes métodos permitem distinguir o estado geral da máquina e também detectar ou localizar o tipo de falha. Esta teoria tem sido melhor desenvolvida nos sistemas definidos por operadores lineares usando técnicas de solução bem definidas para sistemas de equações diferenciais ordinárias lineares. A análise de sistemas não-lineares usualmente é feita para classes específicas de sistemas e envolve programas robustos.

Vários estudos tem sido realizados na área de identificação de sistemas mecânicos e há uma tendência de se introduzir inteligência artificial nos projetos de controle de sistemas e nos sistemas de monitoramento de sinais, os quais permitem uma automatização no processo de monitoramento e diagnose de sistemas complexos (Marsi, Chassiakos & Caughey, 1993). Então o interesse em redes neurais, após ter ficado por duas décadas quase desaparecidas, cresceu rapidamente nos últimos anos e diferentes profissionais do campo da engenharia, psicologia e fisiologia intrigados com o potencial oferecido por esta tecnologia procuram aplicações dentro de suas áreas.

Este ressurgimento de interesse em redes neurais ocorre tanto na teoria quanto na prática. Isto possibilitou o aparecimento de aplicação computacional em um domínio restrito da inteligência humana e tem sido aplicado na fabricação de máquinas, onde o aprendizado tem atingido uma semelhança com o processo mental humano e, também, a fim de dar um novo enfoque para o termo inteligência artificial.

O processamento da informação visual é um dos processos mais complexos do cérebro e, portanto, a organização do caminho ótico tem influenciado os modelos da cadeia neural. Para desenvolver uma rede neural artificial, um modelo matemático que melhor descreva o funcionamento do sistema biológico é desenvolvido. Um computador pode então simular o modelo rapidamente e algum grau de confiança pode então ser ganho com respeito à sua

unesp

operação e funcionamento. Mudanças podem ser feitas para o modelo realçar sua realização ou para simplificá-la. Para criar um modelo que simule um sistema biológico, de qualquer modo, deve-se primeiro estudar e entender este sistema detalhadamente.

Redes neurais são normalmente utilizadas em problemas de reconhecimento de padrões e, em alguns casos, tem mostrado sua superioridade sobre os métodos tradicionais. Alguns estudos têm sidos elaborados no campo de identificação de sistemas. Um dos possíveis usos do sistema de identificação não linear é a detecção e localização de falhas em sistemas mecânicos. As redes neurais multicamadas com retro-alimentação, são largamente usadas por causa de suas propriedades atrativas, tais como implementação arbitrária de decisão complexa, não fazendo nenhuma suposição com relação a forma das distribuições sobre camadas e assim por diante.

Redes neurais artificiais (RNA) são inspiradas biologicamente, isto é, elas são compostas de elementos que realizam, de maneira análoga, a maior parte das funções elementares do neurônio biológico. Estes elementos são organizados de modo que possam ser relacionados com o cérebro. A rede neural artificial exibe um número surpreendente de características do cérebro, como por exemplo, o aprendizado a partir de experiências prévias.

Em função desta semelhança funcional alguns otimistas defendem que as redes neurais artificiais logo duplicarão as funções do cérebro humano. No entanto a atual inteligência exibida pela mais sofisticada rede neural artificial é bastante limitada e o entusiasmo deve ser ajustado a realidade. Contudo não é correto ignorar o surpreendente desempenho de certas redes neurais artificiais. Estas habilidades, hoje limitadas, sugerem que um entendimento profundo da inteligência humana pode ser útil e, com isso, avançar rumo a aplicações revolucionárias.

Redes neurais artificiais podem mudar seus procedimentos em resposta ao seu meio. Este fato, mais que muitos outros, é responsável pelo interesse que tem recebido. Apresentando um grupo de entradas (que podem ter saídas desejadas), elas se auto ajustam para produzirem respostas consistentes. Uma enorme variedade de algoritmos de treinamento tem sido desenvolvido, cada um com suas vantagens e desvantagens.

Verificamos neste contexto a aplicação de redes neurais para detecção e classificação de defeitos em máquinas (Tadeusz, 1994). Esta técnica consiste na análise dos sinais de vibração vindos dos componentes móveis das máquinas para detectar caracteres que refletem o seu estado operacional. Geralmente os níveis de vibração são baixos e constantes quando a

unesp

máquina opera regularmente. Quando ocorrem falhas são alteradas algumas características dinâmicas da máquina e os níveis de vibração também são alterados. O problema de detecção de falhas ou localização com base na medida de informação é essencialmente um modelo de reconhecimento. Informações vindas do sistema sem defeito devem ser distinguíveis das informações do sistema com defeito. A tarefa de reconhecimento de uma falha é complicada por uma série de fatores que afetam o formato do espectro, dentre eles: ruído, geometria da máquina e principalmente mudanças da velocidade de operação.

A habilidade de Redes Neurais operar sobre ruídos a tornou uma excelente e poderosa ferramenta para modelos de reconhecimento aplicado à detecção de falhas em máquinas. O problema da escolha da estrutura de uma rede neural inclui o número de neurônios, o número de camadas, a escolha da função de transferência e o algoritmo de treinamento.

Muitos sistemas reais apresentam características de não-linearidade e não podem ser tratados satisfatoriamente usando a teoria de sistemas lineares. Redes Neurais, as quais apresentam grande habilidade para aprender sofisticadas relações de não-linearidade, fornecem características adequadas para o monitoramento de sistemas não-lineares complicados.

Simulações no computador mostram que redes treinadas podem identificar sistemas bilineares com diferentes razões de rigidez e consequentemente, diferentes razões de freqüência em relação aos sistemas bilineares usados no estágio de treinamento. Esses resultados são significativos para a detecção e localização de falhas e fornecem a possibilidade do uso de redes neurais para detectar e, em um estágio seguinte, quantificar o defeito através da identificação da rigidez bilinear (Lucifredi & Hou, 1994). Vários trabalhos tem sido desenvolvidos na área de identificação e controle de sistemas dinâmicos usando Redes Neurais Artificiais (Narenda and Parthasarathy – 1990 e Chen, Billings and Grant – 1990).

Uma proposta alternativa para localização de falhas estruturais, utilizando a potencialidade de redes neurais associada a técnica de ajuste de modelos ("model updating technique") é apresentada por Lopes Jr., Pereira e Weber (Lopes Jr., Pereira e Weber – 1997a e 1997b) e representa a base deste trabalho. Modificação Dinâmica Estrutural (SDM) é um termo usado com frequência em análise modal e refere-se à mudanças em modelos existentes, em função de novas informações obtidas de testes experimentais ou outras fontes, com o objetivo de melhorar a exatidão do modelo. SDM é utilizada para resolver problemas variacionais direto, isto é, para variações nas propriedades físicas do sistema determina-se as

unesp

variações correspondentes nas propriedades modais. Diferentemente do problema variacional direto, a localização ou detecção de falhas estruturais envolve a solução do problema inverso, isto é, para uma dada variação nas propriedades modais da estrutura deseja-se quantificar e/ou localizar as mudanças correspondentes nas propriedades físicas. Também neste trabalho é dada atenção especial para a identificação de sistemas dinâmicos não-lineares usando Redes Neurais Artificiais e para a quantificação de falhas estruturais.

No capítulo 2 são apresentados os fundamentos teóricos sobre redes neurais. No capítulo 3 são apresentados os fundamentos dos processos de treinamento e da escolha da configuração da rede, que representam a principal etapa deste processo. No capítulo 4 são apresentados conceitos sobre a avaliação da severidade das falhas e métodos (conceitos de probabilidade e de wavelets) que poderiam melhorar a qualidade do padrão de entradas. Também é apresentado neste capítulo uma revisão dos conceitos de análise modal e ajuste de modelos, os quais são aplicados na fase de pré processamento, ou seja, gerar os padrões numericamente para serem aplicados na fase de treinamento das redes. No capítulo 5, principal parte deste trabalho, são apresentados 05 (cinco) trabalhos, na forma original que foram publicados, ou encaminhados. Estes trabalhos foram desenvolvidos em cooperação com professores e alunos do curso de graduação do Departamento de Engenharia Mecânica da UNESP – Ilha Solteira e representam a proposta deste trabalho. Em função da opção de apresentar o capítulo de resultados com os trabalhos na íntegra, alguns conceitos foram repetidos. No capítulo 6 é apresentada a conclusão, a ligação entre os trabalhos e a proposta de continuidade deste estudo.

cm 1 2 3 4 5 6 7 unesp 1 12 13 14 15 16 17 18

II. FUNDAMENTOS DE REDES NEURAIS

2.1 Modelo do Neurônio Biológico

Os neurônios biológicos são constituídos basicamente por:

Corpo Celular: Corresponde a parte central do neurônio, responsável pela geração e recepção dos impulsos nervosos;

Dendrito: Tem a função de receber informações ou impulsos nervosos de outros neurônios e conduzí-los ao corpo celular;

Axônio: Um axônio possui, basicamente, a mesma função do dendrito, porém é longo. Um axônio pode medir de 0,1 milímetro a 1 metro. Próximo de sua extremidade o axônio divide-se em várias ramificações menores (dendritos) que se interligam com os demais neurônios através das sinapses;

Sinapse: É o ponto de conexão entre a terminação axônica de um neurônio e o dendrito de outro. Funcionam como válvulas, sendo capazes de controlar a transmissão de impulsos, ou seja, o fluxo de informação entre os neurônios que compõem a rede. Esta capacidade é conhecida como eficiência sináptica, (Storer, 1979).

A figura 2.1 mostra as partes componentes do neurônio biológico descrito acima:

Redes Neurais Artificiais são concepções em hardware/software baseadas no modelo de como o cérebro codifica e processa informações. Sua aplicação é destinada, principalmente, à soluções de problemas difíceis de serem obtidas por meio de programas convencionais. Estas redes são dotadas de aprendizado, reconhecimento de padrão e memória associativa, dentre outras funções especializadas. Uma Rede é composta por elementos de processamento (neurônios) dispostos em camadas e interligados entre si. Cada neurônio pode conter várias entradas mas somente uma saída e cada saída pode ser utilizada como entrada em vários neurônios seguintes através de suas interligações.

unesp

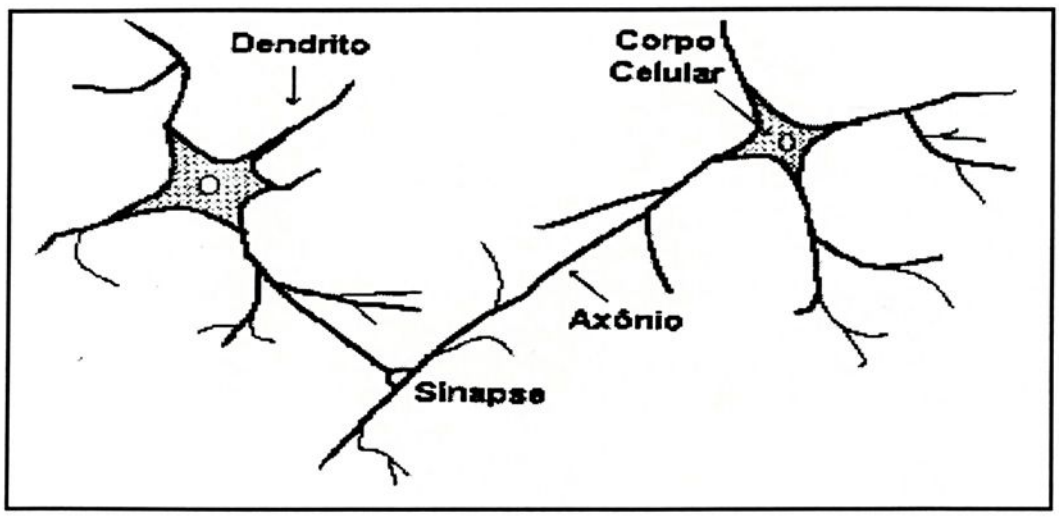


Figura 2.1 - Modelo de um neurônio biológico

Dentre vários trabalhos desenvolvidos na área de identificação e controle de sistemas dinâmicos usando redes neurais artificiais, duas classes têm sido utilizadas com maior frequência: Redes Neurais Multicamadas, as quais tem demonstrado grande sucesso nos problemas de reconhecimento de padrões e Redes Recorrentes que tem sido usadas para solução de problemas de otimização.

As Redes Neurais Multicamadas são largamente usadas por causa de suas propriedades atrativas, tais como implementação de decisões arbitrárias complexas e utilização tanto de dados contínuos como binários, não fazendo nenhuma suposição com relação a forma das distribuições sobre camadas (Worden and Tomlinson – 1992, Hush and Horne – 1993 e Lippmann – 1987).

O treinamento de uma Rede Neural é um processo de ajuste dos pesos e "bias". Existem dois tipos de treinamento: supervisionado e não supervisionado. O treinamento supervisionado consiste no ajuste de pesos de uma rede neural para fornecer saídas desejadas, considerando-se o conjunto de padrões de entrada. Este treinamento necessita de um par de vetores formado por um vetor de entrada e um vetor alvo representando a saída desejada. Juntos eles são chamados de par de treinamento.

unesp

2.2 Modelo do Neurônio Artificial

Os modelos de neurônios artificiais foram concebidos baseados no funcionamento dos neurônios biológicos. RNAs têm sido desenvolvidas com uma grande variedade de configurações. Apesar desta aparente diversidade, os problemas das redes têm sido comuns. As RNAs são inspiradas biologicamente e pesquisas são, usualmente, desenvolvidas sobre a organização do cérebro onde se considera as configurações e algoritmos da rede (Wasserman – 1989). Neste ponto a correspondência deve finalizar. O conhecimento sobre a operação global do cérebro é limitado e ainda há pouco conhecimento das funções correlacionadas. Consequentemente, o projetista de redes deve procurar, além do conhecimento biológico atual, estruturas que realizem funções usuais. Em muitos casos é necessário descartar a plausividade biológica e as redes são produzidas com suposições pouco prováveis sobre a anatomia e funcionamento do cérebro.

Uma rede neural multicamadas usualmente tem de três a quatro camadas de elementos, chamados neurônios, que são classificadas como camada de entrada, camadas intermediárias ou ocultas (uma ou mais) e camada de saída, como ilustrado na Figura 2.2 (Lucifredi and Hou – 1994).

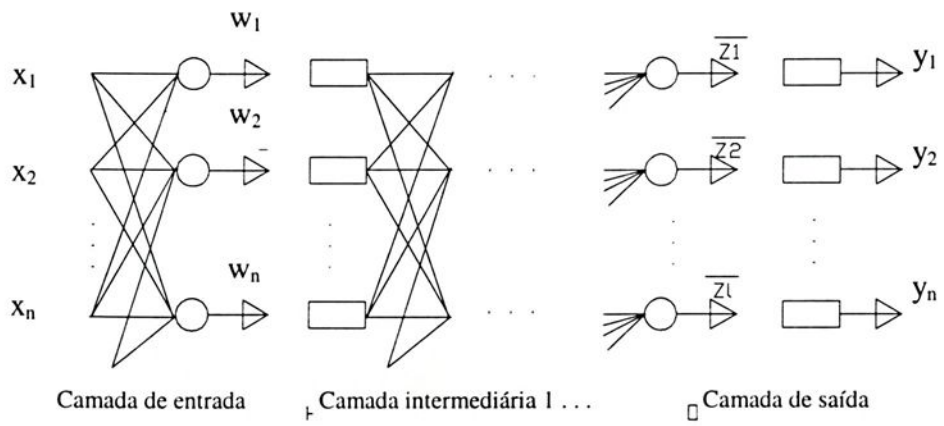


Figura 2.2- Esboço de uma rede neural multicamadas.

Cada neurônio em uma camada é ligado a todos os neurônios da camada seguinte. Essas ligações são ponderadas e os valores passam de um neurônio para o próximo. Os dados de entrada são ponderados e passados para e os neurônios da primeira camada intermediária, os quais são ativados e produzem saídas, que são baseadas na somatória dos

pesos dos valores e na função de ativação. Da mesma forma, a primeira camada intermediária passa os valores à segunda camada (se ela existir) e, assim, sucessivamente. No final, os valores são passados para a camada de saída que produz as saídas da rede.

O primeiro estágio e, também, o mais importante estágio de uma rede neural, é chamado "treinamento" ou "aprendizado". O treinamento é um processo de ajustes dos pesos em função das entradas da rede e das saídas desejadas.

Os valores de saída da rede são repetidamente comparados com os valores das saídas desejadas e, a cada iteração os pesos de conexão são ajustados na direção das saídas desejadas. Um dos algoritmos mais utilizados no treinamento de redes neurais multicamadas é o algoritmo da retro-propagação ou "backpropagation" (Demuth & Beale, 1994). Quando os erros de comparação são considerados pequenos o bastante, o treinamento termina. Se uma rede é bem treinada, isto é, se ela tem uma boa capacidade de generalização, ela pode fazer o trabalho de classificação correto para dados de entrada que não foram utilizados no treinamento.

Na parte deste trabalho relacionado com a identificação da rigidez bilinear, a rede é usada para classificar dois sistemas operando em duas faixas lineares, uma diferente da outra. Portanto, um neurônio na saída é suficiente.

Para o caso do oscilador de Duffing, as entradas da rede foram a força e a aceleração e as saídas o deslocamento e a velocidade. Portanto o número de neurônios da camada de saída é igual ao da camada de entrada. Para o caso do locomotor bípede e da estrutura as entradas e saídas são melhores definidas no capítulo 5. No entanto, de uma maneira geral, podemos expressar os dados de entrada pela seguinte fórmula,

$$\{a_i, a_{i-1}, ..., a_{i-n/2}, p_i, p_{i-1}, ..., p_{i-n/2}\} \; ; \quad i = 1, 2, ...N \qquad 2.1$$

onde a e p são dados de respostas da aceleração e de excitação para o caso do oscildador de Duffing ou frequências e amplitudes quando se considerou a função de resposta em frequência (FRF) ou os espectros de frequência medidos experimentalmente (frequência vs amplitude) para o exemplo da quantificação da falha. N é o número de dados de ajuste do treinamento e n é o número de dados de cada vetor de ajuste, correspondendo ao número de neurônios na camada de entrada. Nos casos apresentados, n é um número constante.

unesp

A determinação do tamanho da rede não é um trabalho fácil e geralmente é feito através de tentativas e erros. Após o treinamento da rede estar concluído, a sua capacidade de generalização é testada, pois a generalização é a medida da performance da rede. Este teste foi feito sobre a razão de rigidez (α) na operação da equação diferencial dos sistemas bilineares. Diferentes razões, α, foram inicialmente simuladas numericamente. Para os outros sistemas, de uma maneira geral, são fornecidas às redes entradas diferentes das utilizadas durante o processo de treinamento. Maiores informações a respeito da generalização de cada rede são fornecidas no capítulo 5, juntamente com cada exemplo analisado.

Em essência as entradas são aplicadas com as correspondentes saídas. Cada entrada é multiplicada por um peso correspondente, análogo ao comprimento sináptico e, todos os pesos de entrada são, então, somados a um determinado nível de ação do neurônio. Apesar da diversidade de paradigmas da rede, tudo é baseado nessa configuração. Aqui, um grupo de entrada (x_1, x_2,..., x_n) é aplicado ao neurônio artificial. Estas entradas coletivas referem-se ao vetor X, correspondendo aos sinais dentro da sinapse de um neurônio biológico. Cada sinal é multiplicado por um peso associado (w_1, w_2,..., w_n) e é aplicado em um bloco de soma, denominado somatório $\sum$. Cada peso corresponde ao comprimento de uma simples conecção sináptica biológica (o grupo de pesos é referido coletivamente como um vetor W). O bloco de soma, correspondente à célula biológica, adiciona todos os pesos de entrada algebricamente, produzindo uma saída que chamamos Y. Isto pode ser descrito em uma notação vetorial por:

$$Y = X W \qquad 2.2$$

2.2.1 Redes Neurais de Camada Única

Embora um simples neurônio possa realizar um modelo de detecção de padrão, o poder da computação neural vem da conexão dos neurônios em forma de redes. A rede mais simples é um grupo de neurônios arranjados em uma camada como mostra o lado direito da Figura 2.3. Nota-se que os círculos do lado esquerdo da Figura 2.3 servem somente para distribuir as entradas. Eles não realizam cálculo e, conseqüentemente, não são considerados para constituir uma camada. Por essa razão, são mostrados como círculos para distingui-los dos neurônios de computação, que estão mostrados como quadrados. O grupo de entradas X tem cada um dos seus elementos conectados a cada neurônio artificial. Inicialmente as redes neurais artificiais não eram mais complexas que esta. Cada neurônio de saída é uma soma

unesp

ponderada das entradas da rede. As redes artificiais e biológicas atuais podem ter muitas das conexões desativadas, mas toda conectividade é mostrada por razões de generalidade.

É conveniente considerar os pesos como elementos de uma matriz W. As dimensões da matriz são m linhas por n colunas, onde m é o número de entradas e n é o número neurônios. Por exemplo, o peso conectando a 3ª entrada com o 2° neurônio é $w_{3,2}$. Desta forma, calcula-se o sistema de saída Y.

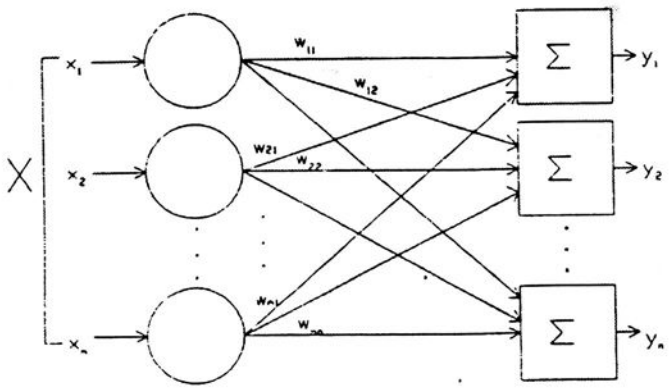


Figura 2.3 - Rede Neural com uma camada.

2.2.2 Redes Neurais Multicamadas

Em geral, as redes mais complexas oferecem maior capacidade computacional. Contudo, as redes têm sido construídas nas mais variadas configurações, imitando a estrutura de certas porções do cérebro. Essas redes de multicamadas apresentam capacidade computacional maiores do que aquelas de simples camadas e, nos últimos anos, algoritmos têm sido desenvolvidos para treiná-las. As redes multicamadas podem ser formadas em cascata com um grupo de camadas simples. A saída de uma camada fornece a entrada da camada subsequente. A Figura 2.4 mostra as conecções de uma rede multicamadas (Wasserman – 1989).

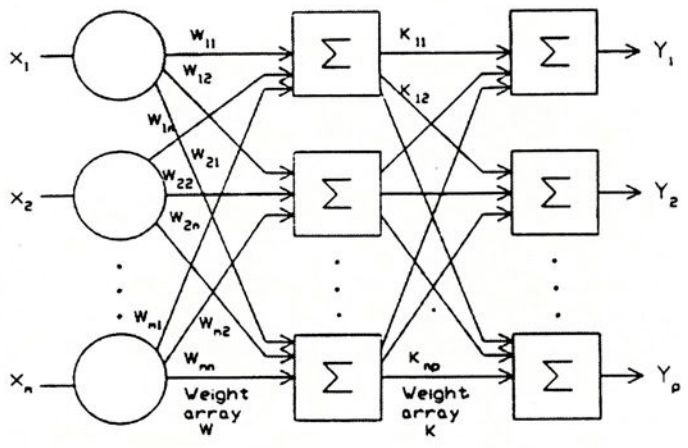


Figura 2.4 - Rede Neural multicamadas.

2.2.3 Redes Recorrentes e Não Recorrentes

As redes não-recorrentes ou redes "feedforward" são de considerável interesse e representam uma classe especial largamente aplicada. Estas redes não apresentam conexão de retorno, isto é, conecção da saída de uma camada para a entrada da mesma camada ou para camadas prévias. Redes não-recorrentes não possuem memória e sua saída é determinada pelas entradas reais e valores dos pesos.

As redes que contém conexões de retorno são chamadas de recorrentes. Em algumas configurações as redes recorrentes recirculam saídas prévias de volta às entradas, então a saída é determinada pela entrada real e pela saída prévia.

2.3 Função de Transferência

Função de transferência é a ligação virtual entre os neurônios de uma rede neural. Esta função possibilita que informações calculadas em um neurônio artificial sejam transmitidas para outros neurônios da rede. A função de transferência seria as trocas sinápticas se fizermos uma analogia ao neurônio biológico. O desenvolvimento das redes levou a adoção de vários tipos de funções de transferência. As principais são: rampa, degrau, logística, tangente hiperbólica e gaussiana (Kaminski – 1997).

unesp

• Função Degrau

$$y_i = f(a_i) = \begin{Bmatrix} \alpha & se \quad a_i \geq \mu \\ -\beta & se \quad a_i < \mu \end{Bmatrix} \tag{2.2}$$

caso especial: função degrau unitário $\quad y_i = f(a_i) = \begin{Bmatrix} 1 & se \quad a_i \geq 0 \\ 0 & se \quad a_i < 0 \end{Bmatrix}$

• Função Rampa

$$y_i = f(a_i) = \begin{Bmatrix} \alpha & se \quad a_i \geq \mu \\ a_i & se \quad |a_i| < \mu \\ -\alpha & se \quad a_i < -\mu \end{Bmatrix} \tag{2.3}$$

casos particulares:

$$y_i = f(a_i) = \begin{Bmatrix} 1 & se \quad a_i \geq 1 \\ a_i & se \quad |a_i| < 1 \\ -1 & se \quad a_i < -1 \end{Bmatrix}$$

$$y_i = f(a_i) = \begin{Bmatrix} 1 & se & a_i \geq 1 \\ a_i & se & 0 \leq a_i < 1 \\ 0 & se & a_i < 0 \end{Bmatrix}$$

• Função Logística

$$y_i = f(a_i) = \frac{1}{1+e^{-\alpha a_i}} \qquad \text{com } \alpha > 0 \tag{2.4}$$

• Função Tangente Hiperbólica

$$y_i = f(a_i) = \tanh(a_i) \tag{2.5}$$

• Função Gaussiana

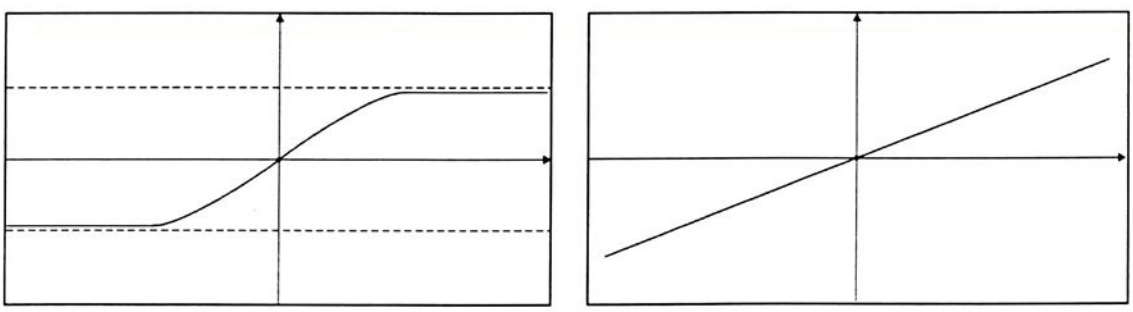
$$y_i = f(a_i) = e^{\frac{-a_i^2}{\upsilon}} \tag{2.6}$$

unesp

caso particular: $y_i = f(a_i) = e^{-a_i}$

Nos exemplos analisados neste trabalho foram utilizadas a função de transferência não linear sigmóide (logística) e a função linear (purelin) do programa Matlab e estão mostradas na figura 2.5.

A função sigmóide é uma aproximação de uma função real diferencial que é definida para todos os valores reais de entrada e possui sempre uma derivada positiva. Além disso, é assumida como sendo uma aproximação rápida de um limite superior fixo assintoticamente quando os argumentos crescem e, é assumida como uma aproximação rápida de um limite inferior fixo assintoticamente quando os argumentos decrescem. A parte central da função sigmóide (próxima de zero) é assumida como sendo uma aproximação linear. A inclinação média desta parte central é chamada de ganho da função. Um exemplo típico de uma função desta família é: $f(u) = 1/(1+e^{-u})$



Função Sigmóide Função *Purelin*

Figura 2.5 - Funções de Transferência.

As redes multicamadas não aumentam o poder computacional em relação a uma rede de simples camada a menos que haja uma função de transferência não linear entre as camadas. Calcular a saída de uma camada consiste em multiplicar o vetor de entrada pela 1ª matriz de peso e, então (senão houver uma função de ativação não linear) multiplicar o resultado do vetor pela 2ª matriz de peso. Isto deve ser expresso como:

$$(X W_1) W_2 \qquad (2.7)$$

Aplicando a propriedade associativa da multiplicação de matrizes obtém-se

$$X (W_1 W_2) \qquad (2.8)$$

unesp

Isto mostra que uma rede de duas camadas lineares é exatamente equivalente a uma camada simples tendo uma matriz de peso igual ao produto das duas matrizes de peso. Conseqüentemente, a rede linear de multicamadas pode ser transformada em uma rede de uma camada equivalente. As redes de camadas simples são limitadas em sua capacidade computacional e as funções de transferência não lineares são vitais para a expansão da capacidade de uma rede.

2.4 Treinamento de Redes Neurais Artificiais

De todas as características de interesse das redes neurais artificiais nenhuma desperta mais a imaginação que a sua capacidade em aprender. O treinamento mostra muitos paralelos com o desenvolvimento intelectual humano e o entendimento fundamental desse processo já foi alcançado. A euforia deve ser temperada com precaução pois o aprendizado em redes neurais artificiais é limitado e muitos problemas permanecem para serem solucionados. Apesar de demonstrações impressionantes terem sido formadas, muitas aplicações práticas são ainda emergentes.

Uma rede é treinada para aplicações de grupos de entradas produzirem um grupo de saída desejada. O treinamento é executado seqüencialmente aplicando os vetores de entrada, durante o ajuste dos pesos da rede de acordo com um procedimento pré determinado. Durante o treinamento, os pesos da rede convergem gradualmente para valores de modo que um vetor de entrada produza um vetor de saída desejável.

2.4.1 Treinamento Supervisionado

Os algoritmos de treinamento possuem duas categorias: supervisionado e não supervisionado. O treinamento supervisionado requer o pareamento de cada vetor de entrada com o vetor alvo representando a saída desejada. Juntos eles formam um par de treinamento. Usualmente, uma rede é treinada com um grande número de pares de treinamento.

Durante a sessão de treinamento de uma rede neural, um estímulo da quantidade que entra é aplicado e isso resulta em uma reação de saída. Esta reação é comparada com um sinal de saída prévio desejado. Se a saída da rede diferenciar da saída desejada, a cadeia neural gera

unesp

um sinal de erro, que é então usado para calcular o ajuste que deverá ser feito no peso da cadeia para que a saída real se aproxime da saída desejada.

Em outras palavras, o algoritmo procura minimizar o erro. O processo de minimização de erro requer um circuito especial de supervisão. Portanto, o nome aprendizagem supervisionada. A noção de um supervisor vem de observações biológicas, por exemplo, quando aprendemos uma linguagem, nós ouvimos o som de uma palavra (de um professor). O som é armazenado no banco de memórias do nosso cérebro e nós tentamos reproduzir o som. Quando ouvimos nosso próprio som, mentalmente comparamos isto (saída real) com o som armazenado (som alvo) e anotamos o erro. Se o erro é grande, tentamos outra vez e outra vez até isto tornar-se significantemente pequeno e, então, paramos o aprendizado. Com cadeia neural artificial a quantidade de cálculo requerida para minimizar o erro depende do algoritmo usado e isto é puramente uma ferramenta matemática derivada das técnicas de otimização.

Tais técnicas são extensivamente usadas em paradigmas da cadeia neural. Alguns parâmetros que deveriam ser observados são o tempo requerido por iteração e o número de iterações por padrão de entradas, para o erro alcançar um mínimo durante a sessão de treinamento.

2.4.2 Treinamento Não Supervisionado

Apesar de muitas aplicações com sucesso, treinamento supervisionado tem sido criticado por ser biologicamente implausível, por supor um mecanismo de treinamento do cérebro que compara saídas desejadas e as reais, alimentando um processo de retro correções na rede. Se este é o mecanismo do cérebro, de onde vem o modelo de saída desejada? Como o cérebro de uma criança pode executar uma organização própria já que tem um desenvolvimento precoce?

O treinamento não supervisionado é o mais plausível modelo de entendimento do sistema biológico. Desenvolvido por Kohonen (Kohonen – 1984) e muitos outros, ele não requer um vetor alvo para as saídas e, conseqüentemente, sem comparações pré determinadas. O algoritmo de treinamento modifica os pesos da rede para produzir vetores de treinamento ou aplicação de um vetor que, suficientemente similar, produzirá o mesmo modelo de saídas. O processo de treinamento, por essa razão, extrai as propriedades estatísticas do grupo de treinamento e grupos de vetores similares entre classes. Aplicando um vetor de uma dada classe para uma entrada, este produzirá um vetor de saída específico, mas não há um caminho

unesp

determinado a priori para o treinamento que produzirá um modelo específico de saída para uma dada classe de vetor de entrada. Consequentemente, as saídas da rede necessitam, geralmente, ser transformadas em forma compreensível para o processo de treinamento. Isto não representa um problema sério e é simples identificar a relação entrada-saída estabelecida pela rede.

Em contraste com a aprendizagem supervisionada a aprendizagem não supervisionada não requer um processo de revisão. Durante a sessão de treinamento a rede neural recebe, na sua quantidade que entra, muitas excitações diferentes, ou padrões de saída e, arbitrariamente, organiza o padrão em categorias.

Quando uma nova entrada é, então aplicada, a rede neural proporciona uma reação de saída indicando a classe para qual o estímulo pertence. Se uma classe não pode ser encontrada para o estímulo da quantidade que entra, uma nova classe é gerada. Por exemplo, mostre a uma pessoa um conjunto de diferentes objetos. Então peça a ela para separá-los em grupos, ou classificá-los, tal que objetos em um grupo tenham uma ou mais características comuns que os distinguem de outros grupos. Quando isto é feito, mostre à mesma pessoa um outro objeto e peça a ela para colocar o objeto em um dos grupos. Se ele não pertencer a qualquer dos grupos existentes, um novo grupo pode ser formado.

Ainda que aprendizagem não supervisionada não requeira um supervisor, ela requer princípios gerais para determinar como formar-se-ão os grupos. O agrupamento pode ser baseado em forma, cor, consistência material ou sob alguma outra propriedade do objeto. Portanto, se os princípios gerais não foram dados e, nem os tipos de características que devem ser usadas para o agrupamento dos objetos, ele pode ou não ser bem sucedido.

2.4.3 Aprendizagem Reforçada

Aprendizagem reforçada requer um ou mais neurônios na camada de saída e um supervisor que, ao contrário da aprendizagem supervisionada, não indica a que distância a saída real está da saída desejada, mas se a saída real é a mesma da saída desejada ou não. Durante a fase de aprendizagem, um estímulo da quantidade que entra é aplicada e uma reação de saída é obtida (considerar uma saída por simplicidade). O supervisor não apresenta o alvo de saída para a rede, mas apresenta somente uma indicação "passar/reprovar".

Deste modo, o sinal de erro gerado durante a sessão de treinamento é binário: passar ou reprovar. Se a indicação for reprovada, a rede reajusta seus parâmetros e tenta novamente

unesp

até conseguir a saída correta. Durante este processo, não há indicação se a reação de saída está se movendo na direção certa e qual a proximidade que está da reação correta. Portanto, o processo de correção de pesos segue uma estratégia diferente do processo de aprendizagem supervisionada.

Quando a aprendizagem reforçada é usada como uma técnica de treinamento, certas fronteiras devem ser estabelecidas. Assim, o aprendizado não deve continuar tentando conseguir a correta reação indefinidamente.

2.4.4 Aprendizagem Competitiva

Aprendizagem competitiva é uma outra forma de aprendizagem supervisionada, que é distinguida por suas características de operação e arquitetura. Neste esquema, vários neurônios estão na camada de saída. Quando um estímulo da quantidade que entra é aplicado, cada neurônio de saída compete com o outro para produzir o sinal de saída mais próximo do alvo. Esta saída se torna dominante e as outras saídas cessam a produção de sinais de saída para aquele estímulo. Para um outro estímulo, um outro neurônio de saída se torna o dominante e, assim por diante.

Desse modo, cada neurônio de saída é treinado a responder a diferentes estímulos da quantidade que entra. Aprendizagem competitiva pode também ser vista como um processo de especialização casual. Quando uma rede neural artificial com aprendizagem competitiva faz parte de um sistema de redes neurais artificiais maior, então, da questão de conectividade, esta especialização casual pode não ser sempre a desejada. Nesse caso, pode-se tentar a aprendizagem reforçada.

Aprendizagem competitiva é, freqüentemente, encontrada em grupos de pessoas onde cada membro do grupo foi selecionado e treinado para realizar tarefas específicas baseadas no princípio da pessoa certa, na hora certa, no lugar certo.

2.5 Algoritmos de Treinamento

Muitos dos algoritmos de treinamento atuais usam o conceito de Donald Hebb (Hebb – 1961). Ele propôs um modelo de treinamento não supervisionado baseado no fato de que quando um axônio da célula A está perto o bastante para estimular uma célula B e,

unesp

repetidamente, ou, persistentemente, permite a excitação dessa célula, algum processo metabólico de crescimento permite mudanças em uma ou ambas as células tal que a eficiência de A, com uma excitação da célula B, é aumentada.

Desse modo, a força sináptica (conhecida como peso W) entre a célula A e a célula B é modificada segundo o grau de atividade correlata entre as quantidades de entrada e saída. Este tipo de aprendizagem é chamado de aprendizagem Hebbian, um termo encontrado freqüentemente em RNA.

A ponderação (pesos) é incrementada se ambos, a origem e o neurônio de destino, são ativados. Deste modo, usando muitas vezes este caminho, a rede é fortalecida e o fenomenal hábito de aprender pela repetição direta é explicado.

Uma rede neural artificial usando o aprendizado Hebbian incrementa os pesos desta rede de acordo com o produto do nível de excitação da origem e dos neurônios de destinos, ou seja,

$$w_{ij}(n+1) = w_{ij}(n) + \eta.out_i out_j \qquad 2.5$$

onde:

$w_{ij}(n)$ = o valor do peso do neurônio i para o neurônio j antes do ajuste

$w_{ij}(n+1)$ = o valor do peso do neurônio i para o neurônio j depois do ajuste

η = o coeficiente de aprendizado

out_i = a saída do neurônio i e entrada para o neurônio j

out_j = a saída do neurônio j

Redes têm sido construídas usando o aprendizado Hebbian. Entretanto, algoritmos de treinamento mais eficientes têm sido desenvolvidos. Vários autores (Rosenblatt – 1962, Widrow -1961, 1962 e 1963 e outros) desenvolveram algoritmos de treinamento supervisionado, produzindo redes que aprendem com uma larga faixa de padrões de entrada e maiores razão de aprendizado, podendo então ser executado usando o simples aprendizado Hebbian. Atualmente existe uma grande variedade de algoritmos de treinamento em uso e um grande esforço é dedicado a este tópico

unesp

2.6 Característica da Rede Neural Artificial

Genericamente falando, a rede neural é uma rede adaptativa de comunicação que comunica uma *função objetivo* para uma saída desejada. Matematicamente falando, uma cadeia neural representa um sistema dinâmico que pode ser modelado como um conjunto de equações diferenciais acopladas.

Como se sabe da teoria de controle (retro-alimentação), um sistema assim depende dos valores dos parâmetros da equação. Pequenas mudanças nos parâmetros podem resultar em estabilidade, instabilidade (oscilações) ou mesmo, instabilidade catastrófica.

unesp

III. TREINAMENTO E CONFIGURAÇÃO DAS REDES

3.1 Perceptrons

Na década de 60, os perceptrons criaram um grande momento de interesse e otimismo. Rosenblatt (Rosenblatt – 1962) provou um notável teorema sobre o aprendizado do perceptron. Widrow (Widrow – 1959 e 1960) e outros autores fizeram demonstrações do sistema perceptron e, pesquisadores em todo o mundo estavam ansiosos com o potencial destes sistemas. A euforia inicial foi mudada pela desilusão. Nos perceptrons foi encontrada uma falha em uma simples tarefa de aprendizagem. Minsky (Minsky – 1969) analisou este problema com muito rigor e provou que havia severas restrições no que uma simples camada de perceptron poderia representar e, consequentemente, no que ela poderia aprender. Como não havia técnicas conhecidas para treinamento das redes de multicamadas, os pesquisadores se voltaram para áreas mais promissoras e as pesquisas com redes neurais artificiais foram quase totalmente esquecidas.

A descoberta recente dos métodos de treinamento para redes de multicamadas, mais que qualquer outro fator, tem sido responsável pelo ressurgimento do interesse pela pesquisa. O trabalho de Minsky diminuiu o ardor dos entusiastas do perceptron, no entanto proporcionou um período para a necessária consolidação e desenvolvimento da teoria básica. É importante notar que a análise de Minsky não tem sido refutada, ele deixou um importante trabalho e deve ser estudado para que os erros da década de 60 não sejam repetidos.

Apesar das limitações dos perceptrons, eles têm sido extensivamente estudados. Sua teoria é o fundamento para muitas outras formas de redes neurais artificiais e demonstra importantes princípios. Por estas razões, o perceptron é o ponto lógico inicial para o estudo das redes neurais artificiais.

O teorema de aprendizagem do perceptron demonstrou que ele poderia representar qualquer função que tivesse sido aprendida. É importante distinguir entre representação e aprendizagem. A representação refere-se a habilidade de um perceptron (ou outra rede) simular uma função específica. A aprendizagem requer a existência de um procedimento

unesp

sistemático para o ajuste dos pesos da rede para produzir aquela função. Para ilustrar a representação do problema, suponha que se tenha um grupo de cartões brilhantes com os números de 0 a 9. Suponha, também, que uma máquina hipotética é capaz de distinguir um cartão par de um cartão ímpar, acendendo um indicador no seu painel quando mostrar um cartão ímpar, Figura 3.1. Pode esta máquina ser representada por um perceptron? Isto é, pode um perceptron ser construído e os seus pesos ajustados tal que ele tenha a mesma capacidade discriminatória? Se isto for possível, dizemos que o perceptron pode representar a máquina desejada. Um perceptron de simples camada tem capacidade de representação limitada. Há muitas máquinas simples que o perceptron não pode representar.

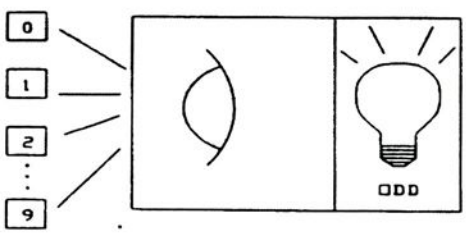


Figura 3.1- Sistema de reconhecimento de imagem.

3.1.1 Eficiência de Armazenagem

Há sérias questões sobre a eficiência de armazenagem dos perceptrons (e outras redes neurais artificiais) relativas à memória do computador convencional e de métodos de busca. Por exemplo, seria possível armazenar todos os padrões de entrada na memória de um computador juntamente com a classificação dos bits? O computador procuraria pelo padrão desejado e responderia com a sua classificação? Se uma informação exata não fosse encontrada, o critério do elemento mais próximo poderia ser usado para retornar o ajuste mais fino?

O número de bits requerido para armazenar a mesma informação nos pesos do perceptron pode ser substancialmente menor do que o método convencional de memória do computador, se a natureza dos padrões permitir uma representação compacta. Entretanto, Minsky mostrou exemplos em que o número de bits requeridos para representar os pesos cresce exponencialmente com o tamanho do problema. Nestes casos, a memória requerida expande rapidamente para níveis impraticáveis, quando o tamanho do problema aumenta. Se,

unesp

quanto as conjecturas, esta situação não é exceção, perceptrons freqüentemente poderiam ser limitados para problemas pequenos.

3.1.2 Aprendizado do Perceptron

A capacidade de aprendizagem das redes neurais é uma das mais intrigantes propriedades. Como sistemas biológicos, estas redes se auto modificam, como resultado da experiência para produzir o mais desejável modelo de comportamento.

Usando um critério de separabilidade linear é possível decidir se uma rede de simples camada pode ou não representar uma função desejada. Mesmo se a resposta for "Sim" poderá ser de pouca utilidade se não encontramos valores adequados para os pesos. Se a rede for de valor prático, necessita-se de um método sistemático (um algoritmo) para calcular estes valores. Rosenblatt forneceu isto num algoritmo de treinamento do perceptron, juntamente com a prova de que se um perceptron pode ser treinado para qualquer função ele poderá representá-la. O aprendizado do perceptron é do tipo supervisionado.

O algoritmo de treinamento do perceptron pode ser implementado com um computador digital ou outro hardware e a rede se auto ajusta. Por esta razão, o ato de ajuste dos pesos é comumente chamado de "treinamento" e a rede é chamada de "aprendiz". A prova de Rosenblatt foi um marco importante e favoreceu o grande interesse para a pesquisa nesse campo. Hoje, de uma forma ou de outra, os elementos do algoritmo de treinamento do perceptron são encontrados em muitas redes modernas.

Algoritmo de Treinamento do Perceptron

Um perceptron é treinado com a utilização de um grupo de padrões na entrada, um por vez, ajustando os pesos até que a saída desejada ocorra para cada um deles. Suponha que os padrões de entrada são cartões brilhantes. Cada cartão brilhante pode ser marcado dentro de quadrados e cada quadrado fornece uma entrada para o perceptron. Se o quadrado tem uma linha reta, sua saída é 1 se não, sua saída é 0. O grupo de quadrados representa o sistema dos 1 e 0 utilizados como entrada para o perceptron. O objetivo é treinar o perceptron para que, ao aplicar um grupo de entradas representando um número ímpar, sempre acenda uma luz, enquanto ele permanece desligado para os números pares.

unesp

A Figura 3.2 mostra a configuração deste sistema. Suponha que um vetor X represente o modelo do cartão brilhante a ser reconhecido. Cada componente (quadrado) de X, (x_1, x_2, ...,x_n), é multiplicado pelos componentes correspondentes do vetor peso W, (w_1, w_2,...,w_n). Estes produtos são então somados. Se a soma excede um limite θ, a saída do neurônio Y é 1 e a luz é acendida, caso contrário é 0. Esta operação pode ser representada compactamente na forma vetorial.

$$Y = XW \tag{3.1}$$

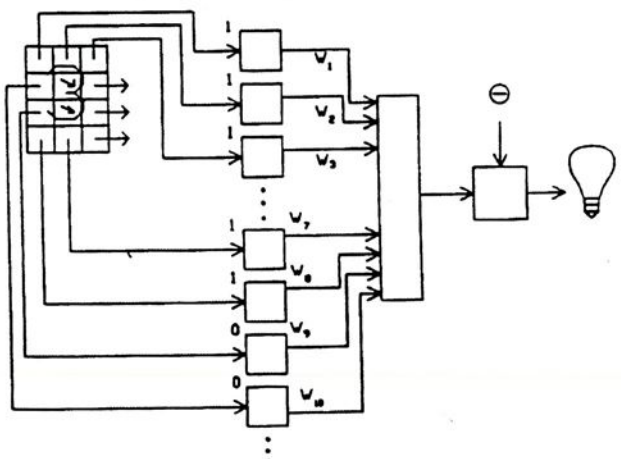


Figura 3.2- Sistema de reconhecimento de imagem com o Perceptron

Para treinar a rede, um modelo X é aplicado para a entrada e a saída Y é calculada. Se Y for correto nada é mudado. Contudo, se a saída for incorreta, os pesos de conexão tem seus valores modificados para diminuírem o erro. Por exemplo, assuma que um cartão brilhante carregando o número 3 é a entrada do sistema e a saída Y é 1 (indicando um número ímpar). Como a resposta esta correta, os pesos não são mudados. Se, contudo, um cartão brilhante com o número 4 for a entrada e a saída Y for 1 (ímpar), os pesos de conecção para as entradas devem ser diminuídos, pois tendem a produzir um resultado incorreto. Similarmente, se um cartão com número 3 produz uma saída 0, os pesos devem ser aumentados. Este método de treinamento pode ser esquematizado através dos seguintes passos:

Passo 1- Aplicar uma entrada padrão e calcular a saída Y.

Passo 2- Se a saída estiver correta, vá para o passo 1;

- Se a saída estiver incorreta e for 0, adicionar cada entrada aos pesos correspondentes; ou

- Se a saída estiver incorreta e for 1, subtrair cada entrada do peso correspondente;

Passo 3- Vá para o passo 1.

Em um número finito de passos, a rede aprenderá separar os cartões de entrada nas categorias par e ímpar, desde que os grupos das figuras sejam linearmente separáveis. Para todos os cartões ímpares a saída será maior que o limite e, para todos os cartões pares, será abaixo do limite. Note que este treinamento é global e a rede aprende para todo o grupo de cartões. Isto nos leva a questão sobre qual sistema deveria ser apresentado para minimizar o tempo de treinamento. Os grupos podem ser aplicados seqüencialmente, ou podem ser selecionados aleatoriamente? Há pouca teoria para guiar esta determinação. Uma importante generalização do algoritmo de treinamento do perceptron, chamado de Regra Delta é mostrado em Wasserman, (Wasserman – 1989).

3.2 Retropropagação

O algoritmo de retropagação, "backpropagation", foi criado como uma generalização da *regra do delta* de Widrow para redes de múltiplas camadas e funções de transferência não lineares. Os vetores de entrada e os correspondentes vetores de saída são usados para treinar a rede até ela possa aproximar uma função, associar vetores de entrada com os específicos vetores de saída, ou classificar vetores de entrada em um modo apropriado pré definido.

As regras do treinamento de retropropagação são usadas para ajustar os pesos (e os "bias") da rede com o objetivo de minimizar a soma dos erros quadráticos. Isto é feito pela mudança contínua dos valores de ponderação da rede na direção do passo descendente com respeito ao erro. Este procedimento é chamado de gradiente descendente.

Este algoritmo pode conduzir a um mínimo local ao invés do global. O mínimo local encontrado pode ser satisfatório. Caso contrário, uma rede com mais camadas e neurônios

unesp

pode fazer uma melhor tarefa. De qualquer modo, o número de neurônios ou camadas a aumentar pode não ser óbvio. Alternativamente, pode-se rodar o programa usando diferentes condições iniciais para verificar se as soluções permanecem constantes.

Os erros são calculados para a camada de saída da rede e, então retropropagados, através do vetor delta, para as camadas ocultas. Quando a soma dos erros quadrados de todos os vetores for menor que o erro meta, o treinamento termina. O processo de treinamento pode ser visto como tendo quatro fases:

- Fase de apresentação: apresenta um vetor de treinamento na entrada e calcula a saída sucessivamente, até que a última camada seja encontrada;
- Fase de parada: calcula o vetor erro da rede e a soma do erro quadrático para o vetor de entrada. Parar se a soma do erro quadrático de todos os vetores de treinamento for menor que o erro meta ou se o número máximo especificado de iterações (épocas) tenha sido alcançado, se não continuar o processo;
- Fase de retropopagação: calcula o vetor delta para a camada de saída e, então faz-se o processo de retropropagação do vetor delta para as camadas intermediárias;
- Fase de aprendizado: calcula a nova matriz de ponderação para cada camada. Então retorna a fase de apresentação.

A matriz inicial de ponderação para o treinamento de retropropagação é criada com elementos aleatórios entre -1 e 1. Estes possibilitam que os pesos e os "bias" sejam mudados facilmente e ainda permitem uma variação suficiente, que é vantajosa para a regra de aprendizado. Cada elemento pode ser obtido usando-se uma matriz simétrica da função aleatória. Nos exemplos apresentados no capítulo 5 usou se sempre a função *rands* do programa Matlab (Demuth and Beale – 1996).

O controle linear do algoritmo de retropropagação é usado para treinamento não linear de redes multicamadas, para aproximação de funções, associação de padrões e classificação de padrões.

unesp

O algoritmo de retropopagação descrito no quadro abaixo é uma generalização do algoritmo LMS. Ele usa a técnica do gradiente para minimizar uma função custo igual a diferença média dos quadrados entre a saída desejada e a saída real da rede. A rede é inicialmente treinada adotando pequenos pesos e "bias" aleatoriamente e então apresentando todos os dados do treinamento repetidamente. Os pesos são ajustados depois de cada iteração usando informações especificando a classe correta, até que os pesos convirjam e a função custo seja reduzida a um valor aceitável. Um componente essencial do algoritmo é o método iterativo descrito abaixo, que propaga o erro obtido nos nós da camada de saída para os nós das camadas inferiores para adequação dos pesos.

Algoritmo de treinamento (Lippmann – 1987)

É um algoritmo do gradiente iterativo para minimizar o erro quadrático médio entre a saída real de um perceptron de multicamadas e a saída desejada. Isto requer uma função diferenciável contínua não linear. Assumindo uma função de transferência não-linear sigmoidal f(α):

$$f(\alpha)=\frac{1}{1+e^{-(\alpha-\theta)}}$$

Passo 1 – Inicialização dos pesos

Adotar pequenos valores para os pesos e "bias" aleatoriamente;

Passo 2 – Apresentação das entradas e saídas desejadas

Apresentar um vetor de entrada x_0, x_1, ...x_{n-1} e especificar as saídas desejadas $d_0,d_1,....,d_{m-1}$. Se a rede é usada como um classificador então todas as saídas desejadas são tipicamente fixadas em 0, exceto aquela correspondente a classe de entrada. Aquela saída desejada é 1. A entrada poderia ser nova para cada tentativa ou, amostras de um conjunto de treinamento poderiam ser apresentadas ciclicamente até estabilizar os pesos.

Passo 3 – Cálculo das saídas reais

Usar a função não-linear sigmoidal, acima mencionada, e calcular as saídas y_0, y_1,

unesp

....y_{m-1}. De uma maneira geral as saídas podem ser representadas pela equação

$$Y = f\left(\sum(W.X - \theta)\right)$$

Passo 4 – Adaptação dos pesos

Usar um algoritmo recursivo, começando nos nós da camada de saída e trabalhando para trás, para as camadas intermediárias.

$$w_{ij}(t+1) = w_{ij}(t) + \eta\,\delta_j x'_i$$

Nesta equação w_{ij} é o peso de um nó i de uma camada oculta para uma entrada no nó j para o tempo t, x'_i a entrada do nó i, η o fator de aprendizagem, e δ_j é um termo de erro para um nó j. Se o nó j for um nó de saída então,

$$\delta_j = y_j(1-y_j)(d_j - y_j),$$

onde d_j é a saída desejada do nó j e y_j é a saída real.

Se o nó j for um nó oculto, então

$$\delta_j = x'_j(1-x'_j)\sum_k \delta_k w_{jk}$$

com k sendo todos os nós das camadas acima do nó j. Os valores dos "bias" são ajustados de maneira similar assumindo que eles são pesos de conecção nas ligações. A convergência, usualmente, é mais rápida se um termo momento é adicionado e os pesos são alterados por:

$$w_{ij}(t+1) = w_{ij}(t) + \eta\,\delta_j x'_i + \alpha\left(w_{ij}(t) - w_{ij}(t-1)\right)$$

onde $0 < \alpha < 1$.

Passo 5 – Retornar ao passo 2.

Este algoritmo tem sido testado com um grande número de problemas determinísticos, em problemas de reconhecimento de vozes e problemas relacionados a reconhecimento de padrões. Tem-se encontrado bom desempenho para os casos apresentados. Uma demonstração da força deste algoritmo foi apresentada por Sejnowski e Rosenberg (Sejnowski and

unesp

Rosenberg – 1986). Ele treinou um perceptron de duas camada com 120 nós e mais de 20.000 pesos para formar letras para regras de transcrição de fonemas. A entrada para esta rede foi um código binário indicando aquelas palavras do texto falado. A saída desejada era um código binário indicando a transcrição fonética das letras. Após 50 vezes através de um diálogo contendo 1024 palavras, o erro de transcrição foi de apenas 5%. Este erro aumentou para 22% para uma continuação daquele diálogo que não foi usado durante o treinamento.

A boa performance para este algoritmo é considerada surpreendente. Sugestões para melhorar a performance e reduzir a ocorrência de mínimos locais incluem a possibilidade de acrescentar uma unidade extra oculta, diminuir a razão de aprendizado usado para adaptar pesos e fazer muitos treinamentos, iniciando com diferentes pesos aleatoriamente. Quando usado com problemas de classificação, o número de nós pode ser especificado usando as considerações descritas acima. O problema de mínimo local, neste caso, corresponde a união de duas ou mais classes dentro de uma região. Isto pode ser minimizado usando múltiplas entradas com diferentes pesos aleatórios e um baixo ganho para adaptar os pesos.

Uma dificuldade que se nota neste algoritmo é que, em muitos casos, o número de apresentações de dados de treinamento, requeridos para a convergência tem sido grande. Contudo um número de adaptações mais complexas tem sido propostas para acelerar a convergência.

Um teorema interessante que elucida algo sobre a capacidade de um perceptron de multicamadas foi proposto por Kolmogorov (Lorentz – 1976). Este teorema estabelece que qualquer função contínua de N variáveis pode ser calculada usando somente a soma linear e não linear, mas continuamente aumentando funções de somente uma variável. Portanto um perceptron de 3 camadas com $N(2N + 1)$ nós, aumentando a não linearidade continuamente, pode calcular qualquer função contínua de N variáveis. Um perceptron de 3 camadas pode então ser usado para criar qualquer função contínua de probabilidade requerida em um classificador. Infelizmente o teorema não faz indicações de como os pesos ou as não linearidades da rede poderiam ser selecionados ou, qual a sensibilidade da saída em relação a variação dos pesos e das funções.

unesp

3.3. Configuração de Redes Neurais Utilizando Lógica Fuzzy

A lógica fuzzy (nebulosa) é baseada na teoria dos conjuntos fuzzy e difere dos sistemas lógicos tradicionais em suas características e seus detalhes. Nesta lógica, o raciocínio exato corresponde a um caso limite do raciocínio aproximado, sendo interpretado como um processo de composição de relações nebulosas.

Na lógica fuzzy, o valor verdade de uma proposição pode ser um subconjunto fuzzy de qualquer conjunto parcialmente ordenado, ao contrário dos sistemas lógicos binários, onde o valor verdade só pode assumir dois valores: verdadeiro (1) ou falso (0). Nos sistemas lógicos multi-valores, o valor verdade de uma proposição pode ser ou um elemento de um conjunto finito, num intervalo, ou uma álgebra booleana. Na lógica nebulosa, os valores verdade são expressos linguisticamente (exemplo: verdade, muito verdade, não verdade, falso, muito falso, ...), onde cada termo linguístico é interpretado como um subconjunto fuzzy do intervalo unitário.

Outras características da lógica fuzzy podem ser resumidas como se segue: nos sistemas lógicos binários, os predicados são exatos (exemplo: par, maior que, ...), ao passo que na lógica fuzzy os predicados são nebulosos (exemplo: alto, baixo, ...). Nos sistemas lógicos clássicos, o modificador mais utilizado é a negação, enquanto que na lógica fuzzy uma variedade de modificadores de predicados são possíveis (exemplo: muito, mais ou menos, ...). Estes modificadores são essenciais na geração de termos linguísticos (exemplo: muito alto, mais ou menos perto, ...)

A probabilidade no contexto da lógica clássica é um valor numérico ou um intervalo. Na lógica nebulosa existe a opção adicional de se empregar probabilidades linguísticas (exemplo: provável, altamente provável, improvável, ...), interpretadas como números fuzzy e manipuladas pela aritmética fuzzy (Gomide, Gudwin e Tanscheit – 1997). Também, em contraste com a lógica modal clássica, o conceito de probabilidade é interpretado utilizando-se subconjuntos fuzzy no universo dos reais (Zadeh – 1988).

Nas teorias clássicas e modernas de controle, como nos sistemas de identificação ou análise dinâmica de sistemas, o primeiro passo consiste na determinação do modelo matemático que descreve o processo. O procedimento requer que se conheça detalhadamente o processo a ser controlado, o que nem sempre é possível em função da complexidade do processo. Estas teorias existentes se aplicam a uma grande variedade de sistemas, onde o

unesp

processo é bem definido. Várias técnicas tem sido desenvolvidas e aplicadas com sucesso, neste contexto, em um grande número de problemas. Entretanto estas técnicas não são capazes de resolver problemas reais cuja modelagem matemática é impraticável. Por exemplo, em diversas situações, informações essenciais são apenas conhecidas a priori de forma qualitativa, ou critérios de desempenho só estão disponíveis em termos linguísticos. Estas restrições podem levar a imprecisões, ou falta de exatidão, no julgamento final do processo.

A módelagem fuzzy é uma técnica para se manusear informações qualitativas de uma maneira rigorosa. Esta técnica considera o modo como a falta de exatidão é descrita e com isto torna-se uma poderosa ferramenta para manipular de maneira conveniente o conhecimento. A teoria de modelagem fuzzy trata do relacionamento entre entradas/saídas e isto permite a consideração de processos complexos, de modo que os sistemas resultantes podem proporcionar resultados mais exatos, além de um desempenho estável e robusto. A simplicidade de implementação de sistemas fuzzy pode reduzir a complexidade de um projeto e portanto oferecer aplicações vantajosas, em certos casos, em relação às técnicas clássicas. A escolha de uma configuração ótima de redes neurais parece portanto se adequar a este tipo de procedimento.

Embora vários trabalhos têm sido publicados com relação a este assunto, para a escolha da configuração da rede neural ainda persiste o conceito de tentativa e erros. Kung e Hwang (Kung e Hwang – 1988) apresentaram um método de determinação do número ótimo de unidades ocultas e da razão de aprendizado da rede que usava o algoritmo de treinamento de retropropagação. Sua aproximação usava um método de projeção algébrica aplicado a um sistema linear formado por dados treinados. Baum e Haussler (Baum e Haussler –1989) mostraram um método para determinar o número de neurônios necessários para produzir generalizações para o mesmo problema. Este trabalho foi realizado para a rede feedforward com uma camada oculta e função de ativação linear de neurônios. Gutierrez et. al (Gutierrez et. al – 1989) sugeriram um método para estimar o número de unidades ocultas requeridas em um perceptron de duas camadas. O número de unidades ocultas é baseado no número de conflitos, um grupo de relações entrada/saída que requer soluções de peso incompatíveis quando as respostas na unidade de saída são treinadas num perceptron de simples camada. Karnim (Karnim –1990) oferece um método de partida com uma rede maior e diminui o número de neurônios das camadas intermediárias para um número mínimo aceitável. As conexões a serem cortadas são baseadas na sensibilidade do erro, para a inclusão/exclusão da conexão em consideração. Huang e Huang (Huang and Huang – 1991) apresentaram um

unesp

método baseado no número de neurônios ocultos necessários para separar (dividir) o espaço de entrada em diferentes regiões. Sartori e Antsaklis (Sartori and Antsaklis – 1991)afirmaram que para um treinamento de p padrões uma rede formada de uma camada oculta e p-1 neurônios pode implementar precisamente o treinamento. Fogel (Fogel – 1991) produziu um método de seleção da melhor configuração da rede baseado no modelo estatístico de identificação. Fogel desenvolveu uma medida da capacidade de generalização para uma dada configuração da rede e esta medida produziu um método de distinção entre as configurações. Sharpe e Chow (Sharpe and Chow – 1994) apresentam uma metodologia usando a técnica de lógica fuzzy para adequar uma configuração de rede atual para uma o mais próximo possível da configuração ótima. Existem portanto diferentes aproximações para se determinar o número de nós ocultos a serem utilizados, cada um com seu próprio mérito, mas não há ainda um trabalho definitivo para se obter o número de entradas e de nós ocultos necessários para uma configuração ótima.

O projeto da rede neural precisa levar em consideração vários fatores para obter uma performance num nível aceitável, usualmente, considera-se a exatidão, a sensibilidade, o tempo de treinamento e o número de neurônios. A exatidão da rede é baseada na diferença da saída desejada e da saída real da rede. A sensibilidade é uma medida da mudança na saída devida às flutuações das entradas (Guo and Uhrig – 1992), causadas possivelmente por ruídos. O tempo de treinamento é, usualmente, baseado no total de tempo requerido de CPU. O tempo de treinamento pode ser especialmente importante onde o treinamento deva ser executado em tempo real. Em geral, deseja-se ter uma rede exata, que seja rapidamente treinada, que tenha tolerância para entradas com ruídos e composta pelo menor número de neurônios possível. Para obter uma medida quantitativa da performance, uma função custo deve ser usada para se avaliar as prováveis variações da rede.

As quatro medidas mencionadas acima não tem valores definidos, cada projetista de rede tem suas próprias idéias e quais são aceitáveis em cada caso. Porém as arquiteturas conhecidas são as que têm pequeno erro, um pequeno tempo de treinamento, uma pequena sensibilidade e um pequeno número de neurônios. Estas medidas são nebulosa neste sentido; deste modo induzem o uso da lógica fuzzy, nestas aplicações (Zimmermann – 1991). Esta metodologia é apresentada para uma rede com uma camada de entrada, uma camada oculta, e uma camada de saída (tem sido mostrado que uma rede feedforward com três camadas é capaz de aprender funções contínuas complicadas). Outras estruturas de rede podem também ser configuradas usando a metodologia fuzzy, com algumas modificações. A rede será

unesp

designada por $\eta_{ni,nh,no}$, onde n_i é o número de entradas, n_h é o número de neurônios ocultos e n_o é o número de saídas. O número de entradas e saídas são fixados em função dos dados de treinamento e, entradas adicionais podem ser geradas por defasagem, figura 3.3, onde k é o índice do tempo amostrado e n_s é o número de medidas amostradas, $x = \{x_1 \ldots\ldots x_{ni}\}$. Estas entradas adicionais podem ser necessárias para melhorar a robustez da rede para situações com ruído.

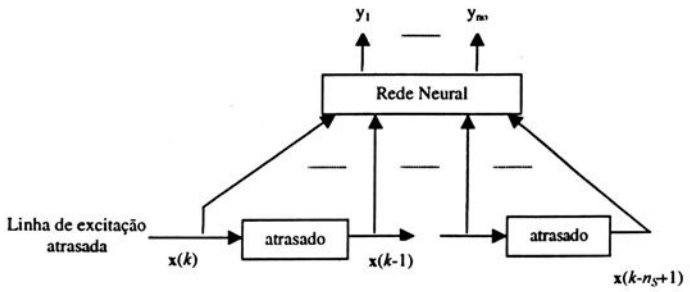


Figura 3.3 – Rede utilizando entradas com defasagem.

Zadeh (Zadeh –1965) introduziu a teoria de sistemas fuzzy como um método de tratamento da imprecisão do mundo físico real. Quatro critérios de avaliação tem sido mencionados em cálculos de configurações de rede: erro de treinamento, tempo de treinamento, sensibilidade e o número de neurônios usados. Antes do treinamento da rede estes parâmetros são desconhecidos. Estes parâmetros são também fuzzy no sentido em que nós desejamos um *pequeno* erro de treinamento, um *pequeno* tempo de treinamento, uma *baixa* sensibilidade e um *pequeno* número de neurônios. A lógica fuzzy trata da caracterização da situação da verdade de um evento. Usando funções de associação, variáveis lingüísticas, como *pequeno, baixa*, etc, podem ser transformadas em valores numéricos. Usando o controle conforme a concepção fuzzy, pode-se tratar estes valores numéricos para melhorar a configuração da rede.

Uma função custo deve ser introduzida para se avaliar as várias configurações de rede. Um procedimento de lógica fuzzy será usado para determinar o número de entradas e os nós na camada oculta de uma rede de duas camadas para minimizar a função custo através de alterações na configuração atual da rede.

unesp

Função Custo

Como mencionado, a exatidão, o tempo de treinamento, a sensibilidade e o número de neurônios são considerados quando se avalia uma dada configuração. Por estas razões, quatro medidas deverão aparecer na função custo, representando cada critério de avaliação. Em geral, a exatidão da rede neural é inversamente relacionada com o erro de medida. Deste modo, pode-se considerar a minimização do erro como uma maximização da exatidão. O erro, tempo de treinamento, sensibilidade e o número de neurônios serão denotados como E, T, S e N respectivamente. Existe um "custo" associado a cada critério, os quais serão denotados como J_E, J_T, J_S e J_N respectivamente. Uma função custo da forma

$$J(n_i, n_h) = q_1 J_E(n_i, n_h) + q_2 J_T(n_i, n_h) + q_3 J_S(n_i, n_h) + q_4 J_N(n_i, n_h) \qquad (3.2)$$

é usada para produzir um método de avaliação das várias configurações de rede. Os valores de q_i ($i = 1, 2, 3$ e 4) em (3.2) são os pesos que podem ser usados em função das preferências individuais do projetista da rede. O problema pode ser colocado como um problema de minimização, onde se deseja minimizar J através da seleção do número de nós de entrada, n_i, e do número de nós ocultos, n_h expressos como

$$\min_{n_i, n_h} J(n_i, n_h) \qquad (3.3)$$

Erro de Treinamento

Para medir a exatidão ou a performance da rede, a média dos erros médios quadráticos sobre o grupo de treinamento pode ser usada como o critério de avaliação. O erro e_m entre a saída da rede y_m e a saída desejada d_m para a entrada x_m, como definida pelo grupo de treinamento é dada por

$$e_m = y_m(x_m) - d_m(x_m) \qquad (3.4)$$

unesp

A média dos erros médios quadráticos pode ser expressa como

$$E = \frac{\sum_{m=1}^{n_p} \|e_m\|^2}{n_p} \tag{3.5}$$

onde $\|.\|$ denota a norma e n_p representa o número de padrões no grupo de treinamento.

Tempo de Treinamento

A condição de parada do treinamento, apresentada por Rumelhart et al (Rumelhart, Hinton and Williams – 1986) e Sharpe e Chow (Sharpe and Chow – 1994), para as várias configurações de redes foram "*nenhuma mudança no erro médio quadrático*", expressa como:

$$\overline{E} = \left| \frac{E_{ciclodetreinamento\ 50i} - E_{ciclodetreinamento\ 50(i-1)}}{E_{ciclodetreinamento50i}} \right| < \varepsilon \tag{3.6}$$

onde E é erro médio quadrático, i é usado para o índice de ciclos de treinamento em cinqüenta passos, $\overline{E}$ é a variação do erro médio quadrático em 50 ciclos e ε é a tolerância. Quando este critério de parada é alcançado, o treinamento termina. Neste ponto, continuar o treinamento não produzirá nenhum benefício, desconsiderando a possibilidade de ter encontrado um mínimo local.

O tempo de treinamento, T, pode ser calculado determinando o número de operações requeridas para completar o treinamento. Para um ciclo de treinamento de um perceptron de duas camadas, o número de operações aritméticas efetuadas estão listadas na tabela 3.1 (considerando a função de transferência sigmoidal). Estas expressões são obtidas considerando o número de operações numéricas requeridas em um algoritmo de treinamento de retropropagação para uma rede genérica consistindo de n_i nós de entrada, n_h nós nas camadas ocultas e uma saída. Com o número de ciclos de treinamento conhecido, o número

unesp

de operações aritméticas pode ser calculado usando a tabela 3.1. Baseado no tempo requerido para efetuar uma dada operação e no número de operações requeridas, uma medida do tempo de treinamento pode então ser calculado.

Tabela 3.1 - Números de operação para um ciclo de treinamento de um perceptron de duas camadas com n_i entradas, n_h nós ocultos e uma saída.

Operação	Operações por ciclo de treinamento
Multiplicação	$4\ n_h\ n_i + 5\ n_0\ n_h + 2(n_0+n_h)$
Adição	$3\ n_i\ n_h - 2\ n_h + 4\ n_0\ n_h - n_0$
Subtração	$3\ n_0 + 2\ n_h + n_0 + n_i\ n_h$
Cálculo da função de transferência	$n_h + n_0$

Sensibilidade

A Sensibilidade é baseada na expansão da série de Taylor e pode ser definida como a primeira derivada da saída em relação a entrada. Isto fornece a magnitude e a direção da variação da saída em função de uma perturbação do sinal de entrada. A matriz de sensibilidade, S_s, de uma rede neural com n_i entradas e n_o saídas é dada por (Chow and Yee – 1991 e Guo and Uhrig – 1992)

$$S_s = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_{n0}}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_{n0}}{\partial x_2} \\ \vdots & \vdots & & \vdots \\ \frac{\partial y_1}{\partial x_{ni}} & \frac{\partial y_2}{\partial x_{ni}} & \cdots & \frac{\partial y_{n0}}{\partial x_{ni}} \end{bmatrix} \tag{3.7}$$

unesp

A medida da sensibilidade, S, é definida como a média da norma quadrada de Frobenius da matriz de sensibilidade S_s, sobre o sistema de treinamento

$$S = \frac{\sum_{m=1}^{n_p} \|S_s\|^2}{n_p} = \frac{\sum_{m=1}^{n_p} \sum_{i=1}^{n_i} \sum_{j=1}^{n_0} S_{ij}^2}{n_p} \tag{3.7}$$

onde n_p representa o número de padrões de treinamento, m é usado para o índice dos padrões de treinamento e S_{ij} é um elemento de S_s.

Configuração da Rede

O objetivo do procedimento fuzzy é minimizar a função custo, equação 3.2, através de mudanças no número de entradas e no número de nós ocultos usados na implementação da rede. Baseado na experiência com o treinamento de vários paradigmas da rede neural, a base das regras fuzzy pode ser construída para traduzir noções heurísticas em regras fuzzy que reduzirão J. Estas noções podem ser convertidas em base do controle fuzzy:

1 - Se E é *GRANDE*, então um *GRANDE* aumento de n_h;

2 - Se T é *GRANDE*, então um *MÉDIO* decréscimo de n_i;

3 - Se T é *GRANDE*, então um *MÉDIO* decréscimo de n_h;

4 - Se S é *GRANDE*, então um *GRANDE* aumento de n_i;

5 - Se N é *GRANDE*, então um *MÉDIO* decréscimo de n_i;

6 - Se N é *GRANDE*, então um *MÉDIO* decréscimo de n_h.

Baseado nos valores atuais de E, T, S e N são feitas mudanças no número de entradas (Δn_i) e no número de nós na camada oculta (Δn_h) aplicando estas regras básicas e a estratégia de "defuzzyfication". As regras apresentadas aqui são genéricas e geralmente verdadeiras para a maioria dos treinamentos de redes neurais. Por exemplo, para aumentar a

unesp

exatidão da rede, a potência computacional da rede precisa ser aumentada pela adição do número de nós ocultos (Regra 1).

Desde que se usa o tempo efetivo de CPU como o tempo medido, a regra 3 é correta devido ao aumento do número de operações numéricas que são necessárias com redes mais complexas. Se maiores informações estão disponíveis em relação sobre os efeitos em J devido as mudanças em n_i e n_h, regras mais elaboradas e exatas podem ser desenvolvidas para conduzir a uma convergência mais rápida.

Estratégia de defuzzification

As regras básicas são conduzidas no sentido das alterações em n_i e n_h baseadas nos valores *de E, T, S* e *N*. Como a escolha do algoritmo ótimo de defuzzification é um problema dependente, duas estratégias populares são: o método da área central e a composição "*max-min*" (Klir and Folger – 1988 e Lee – 1990). Para determinar a regra básica de defuzzification, geralmente o método da área central produz resultados superiores para este tipo de problema. Cada regra será ponderada de acordo com os critérios sobre J_E, J_T, J_S e J_N na equação 3.2. A equação de ponderação da área central é

$$f_i = \frac{\sum_{r=1}^{n_r} q_r \mu_r \times U_r}{\sum_{r=1}^{n_r} q_r \mu_r}, \qquad i = 1, 2 \tag{3.8}$$

onde f_1 e f_2 representam alterações em n_i e n_h respectivamente. Na equação acima, n_r é igual ao número de regras para cada variável de controle. q_r corresponde ao peso colocado na regra r, como definido pela equação 3.2. Estes pesos fornecem a ligação entre a função custo e o procedimento da lógica fuzzy. μ_r representa a associação de *E*, *T*, *S* ou *N* no sistema fuzzy *GRANDE* para a regra r (por exemplo, $\mu_{grande}(E)$ definirá o erro de associação do sistema fuzzy *GRANDE*). U_r é a ação de controle representando *GRANDE* aumento e *MÉDIO* decréscimo como definido na tabela 3.2. Embora os valores numéricos na tabela 3.2 sejam subjetivo, o valor de *MÉDIO* decréscimo poderia ser próximo da metade de *GRANDE* aumento. Isto é devido ao fato de que o *MÉDIO* decréscimo aparece duas vezes em (3.9) e o

unesp

GRANDE aumento aparece somente uma vez. Isto impedirá (3.8) de influenciar sistematicamente o aumento ou diminuição do número de nós. Como um exemplo, a mudança em n_i é dada por:

$$f_1 = \frac{q_2\mu_{grande}(T)\times(MEDDEC)+q_3\mu_{grande}(S)\times(GRANAUM)+q_4\mu_{grande}(N)\times(MEDDEC)}{q_2\mu_{grande}(T)+q_3\mu_{grande}(S)+q_4\mu_{grande}(N)} \quad (3.9)$$

f_1 não é uma função de $\mu_{grande}(E)$, pois não existem regras definidas nas regras básicas relacionando mudanças em n_i em função da magnitude de E.

Tabela 3.2 – Mudanças da rede fuzzy.

Mudanças da rede fuzzy	Ação fuzzy
GRANDE aumento	1.0
MÉDIO decréscimo	-0.5

A equação 3.8 varia de negativo a positivo, em função dos valores listados na tabela 3.2. Para determinar as mudanças em n_i e n_h (Δn_i e Δn_h respectivamente) através de (3.8), os f_i's precisam ser números inteiros. A quantização depende do espaço considerado. Uma quantização usual para o espaço $n_i \in \{2, 4, 6, 8, 10\}$ e $n_h \in \{2, 4, 6, 8, 10\}$ é $\Delta n_i \in \{-2, +2\}$ e $\Delta n_h \in \{-2, +2\}$, mostrada na figura 3.4 (Sharpe e Chow – 1994).

unesp

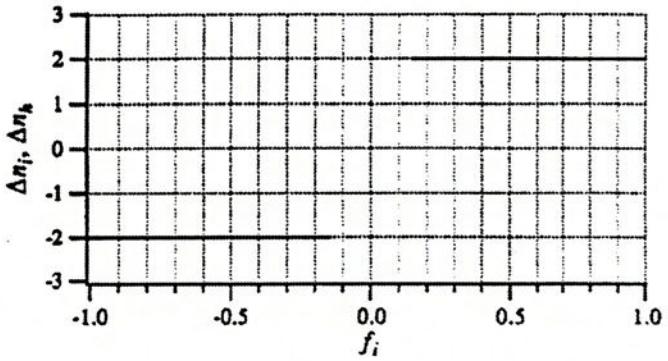


Figura 3.4 – Quantização para transmissão f_i em Δn_i e Δn_h.

Diferentes quantizações podem ser usadas se uma resolução diferente é desejada. Quanto menor a resolução mais exatos os resultados, mas a velocidade de convergência pode ser menor. Na região -0,15 < f_i < + 0,15, a mudança na configuração da rede não é especificada. Se ambos f_1 e f_2 estão nesta região, o valor de f_i com o maior valor absoluto será designado +2 ou -2 dependendo do sinal de f_i. Por exemplo, se abs(f_1) e abs(f_2) são menores que 0.15 e o valor abs(f_1) é maior que abs(f_2) e f_1 é negativo, então será designado para f_1 o valor -2.

3.3.1 Fluxograma do Algoritmo Fuzzy

Um fluxograma do procedimento da lógica fuzzy usado para configuração de redes está mostrado na figura 3.5, onde *itera* é o índice de iteração do processo. O procedimento é iniciado pela escolha de uma configuração inicial dentro do domínio de n_i e n_h. Se a configuração inicial for próxima da ótima, a convergência será mais rápida (típico de técnicas de otimização). A rede é então treinada e os valores de E, T, S e N são anotados. Baseando nestes valores e nos valores correspondentes de μ_{GRANDE}, as regras fuzzy e a estratégia de defuzzification são aplicadas para determinar a mudança apropriada da rede. A condição de "saída" ocorre quando o procedimento fuzzy tenta simultaneamente conduzir n_i e n_h. para fora de seus domínios; quando o procedimento começar a travar entre duas configurações (possivelmente o mínimo foi encontrado, isto é, $n_i^{\text{itera}} = n_i^{\text{itera+2}}$ e $n_h^{\text{itera}} = n_h^{\text{itera+2}}$) ou quando f_1 e f_2 forem zeros (improvável). A configuração da rede analisada pelo procedimento com o

unesp

menor valor de J será selecionada como a solução fuzzy. Esta pode ser a solução ótima, ou pelo menos uma solução próxima da ótima.

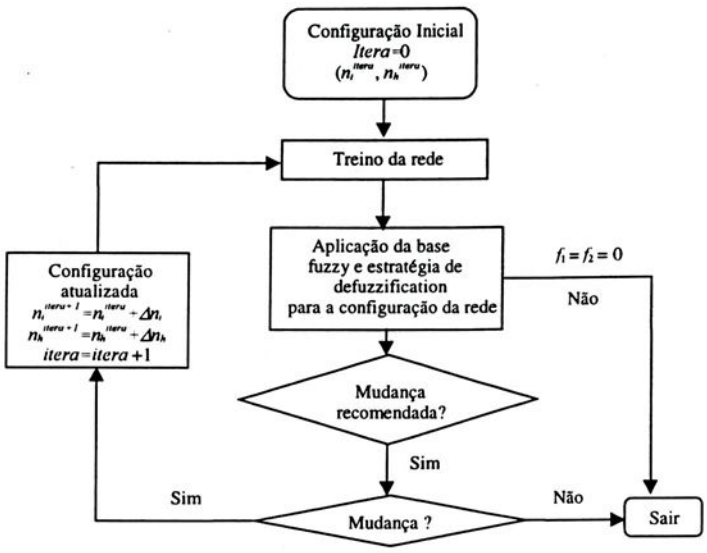


Figura 3.5– Fluxograma do procedimento de solução fuzzy.

IV. DETECÇÃO E LOCALIZAÇÃO DAS FALHAS

Um sistema de diagnóstico, para ser completo, deverá ser capaz de detectar, localizar e quantificar a severidade das falhas. Existem várias metodologias que procuram realizar esta tarefa. No entanto a escolha dos dados de entrada para o treinamento da rede é sempre um grande problema. Para trabalhos aplicados em situações reais de funcionamento os padrões de entrada sempre foram obtidos experimentalmente para toda a bibliografia pesquisada. Este procedimento é quase proibitivo em alguns casos, pois necessita-se adquirir experimentalmente todas as situações de falha que se deseja monitorar. Neste tópico, um trabalho recente baseado na tecnologia de redes neurais artificiais com treinamento supervisionado por retropropagação do erro, foi proposto por Kaminski (Kaminski – 1997), onde os dados necessários para a fase de aprendizagem da rede foram obtidos numericamente com o auxílio de modelos de elementos finitos do componente ou estrutura a serem ensaiados.

A proposta fundamental deste trabalho, melhor detalhada no capítulo 5 (Pereira, Lopes Jr. and Weber – 1997), apresenta uma nova metodologia para tratar este assunto. Esta proposta baseia-se no fato de que os padrões de entrada sejam calculados numericamente utilizando-se a técnica de ajuste de modelos ("model updating"). Após a comprovação de que o modelo matemático representa o sistema físico real, os padrões de entrada são gerados numericamente e a rede é treinada.

Um assunto que também merece destaque é a convergência da rede em função dos padrões de entrada. Em vários exemplos a rede não foi capaz de convergir quando utilizava-se apenas os pontos de frequências naturais ou mesmo todos os pontos (frequência vs. amplitude) da função de resposta em frequência (FRF). A solução adotada, que também representa uma nova proposta, é a utilização de máscaras nos gráficos da FRF. Estas máscaras estão definidas nos trabalhos do capítulo 5 e apresentam como vantagens o fato de que: 1) as máscaras são compostas de poucos pontos em relação a curva da FRF; 2) os pontos são mais condensados na região das frequências naturais, diferentemente da FRF onde a maioria dos pontos encontram-se distantes dos picos (frequências naturais) e constituem portanto em

unesp

fontes de ruídos, dificultando a convergência e diminuindo a capacidade de generalização da rede; 3) se a velocidade de rotação da máquina for levemente alterada, ainda assim a frequência analisada estará dentro da máscara.

A discussão a seguir trata da escolha dos padrões de entrada e de critérios para a quantificação da falha. A proposta de quantificar a falha e os resultados obtidos estão apresentados no capítulo 5 (Lopes Jr. e Demarchi – 1998). Embora a metodologia apresentada a seguir seja geral, isto é, localizar e quantificar a falha, o trabalho citado acima apresenta apenas a segunda parte, pois utilizou-se como entradas as máscaras dos espectros de frequências obtidos experimentalmente e, em função da grande variação das amplitudes a parte relativa a localização da falha não convergiu. Não foi possível medir a FRF e portanto aplicar a mesma metodologia do trabalho citado anteriormente, pois o laboratório de vibrações e instrumentação não dispõe, até o momento, de transdutores de força.

Na primeira etapa deste processo um padrão entradas/saídas, não linear, deverá ser escolhido. Então uma NN classifica o modelo dentro das possíveis falhas ou condições normais de operação, isto é, sistema sem falhas. Se o modelo é classificado com uma falha, outra NN é usada para detectar a severidade da falha. Portanto uma estrutura completamente automática pode ser projetada para realizar a tarefa.

Na área de diagnóstico de equipamentos mecânicos, muitos trabalhos concentram-se em desenvolver metodologias para falhas estacionárias que produzam freqüências distintas no espectro, tal como desbalanceamento, desalinhamento e eixos tortos. Atualmente vários trabalhos tem procurado estabelecer uma avaliação das falhas que alteram a dinâmica do sistema (como por exemplo, alteração da rigidez).

Modelos analíticos podem identificar mudanças na dinâmica do sistema devido à falhas. Estes modelos inicialmente tem a forma de equações diferenciais ordinárias lineares. Desta forma, a utilidade de técnica de modelagem linear tem sido estabelecida para a detecção e o isolamento de falhas. Entretanto, a faixa de operação excede a linearidade do modelo em várias situações e portanto as técnicas de modelagem não linear são necessárias para detectar, isolar e determinar falhas com maior precisão.

O sistema de diagnóstico de falha e avaliação do dano (FDIA) é um modelo não linear de rede, com entrada $x(t)$ e saída $y(t)$. Os pesos da rede são ajustados e em seguida usados

unesp

para diagnosticar e quantificar as falhas. O diagrama de bloco do sistema FDIA é mostrado na figura 4.1.

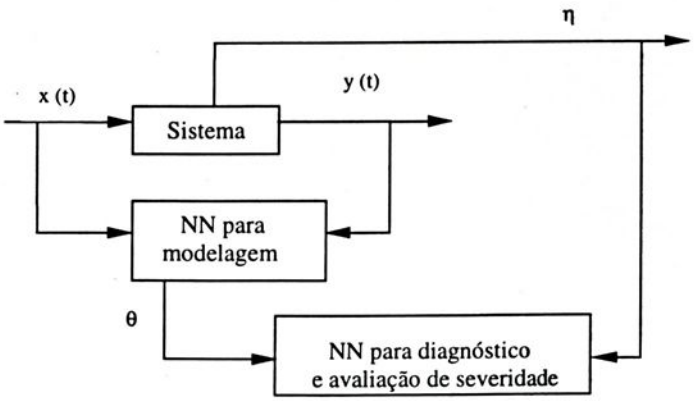


Figura 4.1 – Esquema do diagnóstico.

A partir da terceira camada da rede neural, qualquer função pode ser implementada com um grau de precisão desejado com um número suficiente de neurônios ocultos. Os neurônios ocultos são calculados por a uma função não linear, *f,* de uma soma ponderada das entradas da rede, onde W^1 e W^2 são os pesos das camadas de entrada e saída e θ é o erro de "bias". A função de relação para a rede com uma camada intermediária, pode ser expressa como,

$$y_i(t) = \sum_j W_{ji}^2 \, f_j\left(\sum W_{kj}^1 \, x_k\,(t) + \theta_j\right) \tag{4.1}$$

de maneira resumida a equação acima pode ser escrita como

$$y(t) = NN_M\left(X(t), \phi\right) \tag{4.2}$$

onde ϕ consiste de W^1, W^2 e b, e M é o número de padrões para o modelo.

4.1 Isolamento, Detecção e Avaliação da Severidade das Falhas

Depois que um modelo de rede é identificado, segue-se a interpretação e a avaliação deste modelo. O esquema FDIA tem que classificar o tipo da falha dentro de uma das possibilidades básicas pré definidas e determinar a magnitude da falha. Por serem estes modelos caixas pretas, cujos coeficientes não tem uma relação física explícita com os parâmetros que caracterizam o comportamento do sistema, esta relação é aprendida através de exemplos.

Os exemplos contém padrões com os tipos e magnitudes de falhas que serão usadas durante a fase de treinamento. Denotando os parâmetros do modelo como Θ e o vetor de indicação da condição como η, sendo η_1 as classes da falha e η_2 a magnitude da falha.

O objetivo é estabelecer uma rede, baseada no esquema FDIA, na qual quando um vetor Θ é apresentado, ele dará um η correspondente. A FDIA adota uma estrutura hierárquica que usa NNs separadas para fazer a classificação da falha e a avaliação da severidade. No primeiro estágio, uma NN classifica o sistema dentro das condições preestabelecidas: sem defeito, falha1, falha2, ... Uma vez que no primeiro estágio foi classificada a falha, uma das redes neurais do segundo estágio estima a severidade da falha. Denotando a detecção e o isolamento da falha (FDI) para o primeiro estágio como NN_{FDI} e a avaliação da severidade da falha (FSEV) para o segundo estágio como uma NN_{FSEV} e as saídas desejadas anteriores e posteriores por η_1 e η_2, respectivamente, então matematicamente.

$$\eta_1 = NN_{FDI}\left(\Theta\right) \tag{4.3}$$

No processo de aprendizagem o valor físico da falha, η_2, é fornecido. No trabalho apresentado, este valor representa a porcentagem da falha. Portanto η_2 varia de 0 a 1, sendo que 0 representa a situação sem falha e que 1 representa 100% da falha. Para cada condição conhecida de falha, Θ, as NN_{FSEV} são treinadas com os respectivos valores de η_2.

$$\eta_2 = NN_{FSEV}\left(\Theta\right) \tag{4.4}$$

unesp

Uma NN é usada para detecção e isolamento e c NNs são requeridas para detectar a severidade. Portanto, o número total de NNs é $c+1$ (figura 4.2).

Em resumo, o esquema FDIA trabalha como se segue:

I. O vetor de entradas é estimado para cada situação de falha, Θ. Em nosso caso foi utilizada uma viga com várias profundidades de falhas ao longo do comprimento.

II. Θ é fornecido a NN_{FDI} para encontrar $\hat{n}_1$, isto é a classe da falha.

III. Se $\hat{n}_1$ indica a existência de uma falha, a $NN_{FSEV,i}$ estimará a magnitude da falha, $\hat{\eta}_{2,i}$.

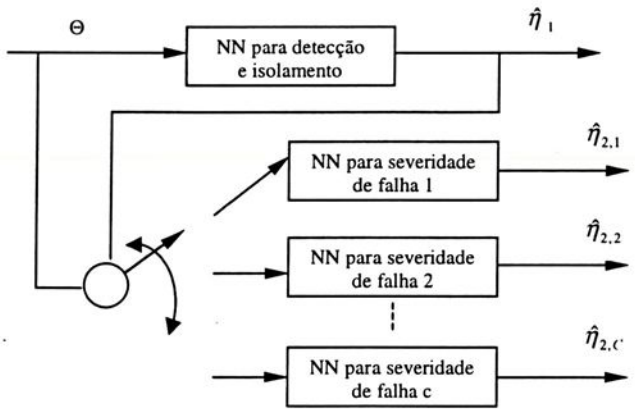


Figura 4.2 – Modelo esquemático, mostrando o número de NNs necessárias

4.2. Conceitos de Probabilidade para Reconhecimento de Padrões

Em vários exemplos analisados, a convergência da rede foi muito lenta ou não ocorreu. O problema da convergência pode ser abordado de duas maneiras: pela configuração da rede ou pelo padrão de entradas. A configuração da rede esta relacionado com o número de camadas, métodos de treinamento, número de neurônios, métodos de minimização e tipos das funções de transferências. Embora sendo a configuração da rede um tópico de extrema importância, o maior esforço foi dedicado a melhoria dos padrões de entrada. É apresentado a

seguir duas propostas que poderiam melhorar a qualidade dos padrões de entrada em alguns casos.

O reconhecimento de objetos dada uma condição completa ou parcial de características é inerente à inteligência humana. Os exemplos de reconhecimento de padrões e inteligência artificial, dentre outros, estão neste tópico. No entanto ainda falta consistência e generalidade para uma variedade de modelos. Todos os modelos carecem de três propriedades importantes: simplicidade, generalidade para uso em uma variedade de aplicações e treinamento fácil. O conceito de sistema de reconhecimento baseado em probabilidade é proposto por vários pesquisadores. Para o reconhecimento de um dado objeto, a presença ou falta de características na entrada fornece a base para uma decisão ponderada de probabilidade na classificação. Com isto, Miller e Merat (Miller and Merat – 1994) propuseram um sistema baseado nos princípios de probabilidade. Esta proposta é referida como GCPM (Generalized Category Perception Model).

A função do GCPM é estabelecer a presença de um objeto em um padrão de entrada. Para um objeto estar presente na entrada ele deve estar representado por uma série de características encontradas na entrada. O termo "característica" refere-se a propriedade que é mensurável, ou a confiabilidade na presença de uma característica não mensurável. Características não mensuráveis são valores de confiança num intervalo de 0 e 1, que representam a probabilidade que uma característica está presente na entrada (por exemplo: topologia, estrutura, geometrias primitivas).

Existe uma associação entre o valor de uma característica mensurável, ou a presença de uma característica não mensurável e a probabilidade da presença de um objeto. Dado um objeto, é esperado que todas as propriedades mensuráveis sejam constantes entre as ocorrências do objeto. Isto sugere a existência de uma função densidade de probabilidade entre os valores de características mensuráveis e a presença de classes de objeto. Igualmente, a existência de características não mensuráveis podem ser consistente. Combinando ambos os sistemas de características, um objeto pode então ser identificado pela confiabilidade nas características não mensuráveis e na avaliação das características mensuráveis.

Assumindo que m é o número máximo de características que possa aparecer na entrada e exista no máximo p classes de objetos possíveis. O sistema de entradas é,

$$X = \{x_1, x_2, \ldots, x_m\} \tag{4.5}$$

unesp

onde todos os membros de x são valores contínuos se a característica é mensurável ou estão no intervalo de 0 a 1 se for não mensurável. Adicionalmente, as saídas correspondentes são,

$$O = \{o_1, o_2, \ldots o_p\} \tag{4.6}$$

onde todos os elementos o estão no intervalo de 0 a 1. As saídas o_i, podem ser interpretadas fisicamente como a probabilidade relativa da característica dè entrada, x, devido a presença do objeto da classe i.

Cada classe de objeto tem um sistema de característica associada. Fazendo x_j ser o sistema de característica para os objetos da classe j, o sistema global de característica em consideração é,

$$F = U_{j=1}^{p} x_j \tag{4.7}$$

onde F é a união dos vetores de característica para toda classe possível de objeto. As características mensuráveis devem ser transformadas em probabilidade no intervalo [0,1], para cada classe de objeto, para termos uma decisão ponderada de probabilidade no vetor de classificação. Um método para o cálculo destas funções densidade de probabilidade, para treinamento supervisionado, foi proposto por Miller e Merat.

4.3 Utilização da Transformada de Wavelet para Determinação de Padrões

Detecção de falhas estruturais é de interesse particular em engenharia. Dados de vibrações de uma estrutura danificada podem exibir diferentes eventos transientes ou não lineares. Muitos algoritmos tem sido desenvolvidos para análise destes tipos de sinais. Nesta etapa pretende-se investigar técnicas de processamento de sinais, que poderiam melhorar o desempenho de redes neurais artificiais, devido ao padrão de entradas. Portanto, pretende-se utilizar a transformada de wavelet como uma alternativa à FFT em sistema de monitoramento

unesp

de sinais para serem aplicados em programa de manutenção preditiva, visando a detecção de falhas em sistemas mecânicos.

Uma das mais recentes ferramentas matemática adaptada para sinais que variam no tempo é a transformada de wavelet (Staszewski and Tomlinson 1994). Originada no início da década de 1980, nos trabalhos de (Morlet - 1980), (Grossmann and Morlet - 1985), (Daubachies, Grossmann, and Meyer - 1989), (Meyer - 1986), e (Mallat - 1989), a transformada de wavelet é uma decomposição tempo - escala e é bem adaptada para modelar e estudar o comportamento tempo-freqüência para análise transiente ou para detectar impulsos de sinais e descontinuidades. A transformada de wavelet tem sido aplicada com muito sucesso em diversas áreas incluindo: acústica, processamento de som, processamento de sinais biológicos, sismologia e o estudo de fractais. As aplicações em engenharia mecânica incluem as seguintes áreas e exemplos:

1) física e modelagem matemática - análise de turbulência em mecânica dos fluidos (Farge - 1992), estudo de fractais em matemática (Chui - 1992);

2) análise tempo-freqüência - comportamento vibro-acústico de um lançamento em vôo (Fumoux e Marquette - 1990), análise de ruído de um compressor (Stazewski e Tomlinson - 1994);

3) identificação de sistemas - separação de diferentes componentes de freqüências na solução de problema mal condicionado (Louis – 1993 e Poterasu - 1997);

4) processamento de imagens - perfis de contorno de vibração da estrutura (Mallat e Zhong - 1992);

5) diagnósticos em sistemas mecânicos - métodos de detecção de falhas em caixa de engrenagem (Stazewski e Tomlinson - 1992);

6) análise de vibrações (Newland - 1994).

4.3.1 Transformada de Wavelet

A transformada de wavelet pode ser introduzida em comparação com a transformada de Fourier, definida como (Chui - 1992):

unesp

$$X(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt \tag{4.8}$$

A transformada de Fourier é uma transformada linear da função x(t) e baseada na decomposição da função em termos de uma base de funções elementares $e^{-j2\pi ft}$. Estas funções elementares $e^{-j2\pi ft}$ são geradas por extensão de uma função única $e^{-j2\pi t}$.

Supõe-se que a função x(t) satisfaça a condição:

$$\int_{-\infty}^{+\infty} |x(t)|^2 \, dt < \infty \tag{4.9}$$

que implica que x(t) decai para zero em $\pm\infty$. Como no caso da análise de Fourier, a idéia da transformada de wavelet é achar uma função g(t) a qual pode gerar uma base para todo o espaço de funções x(t). Se o decaimento dado por (4.9) é muito rápido, está função é chamada uma pequena onda ou uma "wavelet". O rápido decaimento introduz o sentido de localidade na análise, que não é o caso da transformada de Fourier, onde uma função trigonométrica infinita dará uma representação global.

Desde que a wavelet g(t) tenha um decaimento muito rápido, o único caminho para representar todas as funções x(t) é deslocar g(t) no tempo: $g(t-\tau)$. Como no caso da análise de Fourier, pode ser considerado ondas de diferentes freqüências. A segmentação de freqüências é dada por $g(\psi t-\tau)$, assim, cada função $g(\psi t-\tau)$ é obtida de uma única wavelet g(t) pela dilatação de ψ e translação de τ. Estas duas operações são suficientes para produzir uma base, a qual pode representar todas as funções x(t). Uma mudança de variáveis na equação acima, resulta

$$g(\psi t - \tau) = g\left(\frac{t-b}{a}\right) = g_{a,b}(t) \tag{4.10}$$

onde $b=\tau/\psi$ e $a=1/\psi$. A transformada de wavelet pode agora ser definida como:

$$W_g(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \, g^*\left(\frac{t-b}{a}\right) dt \tag{4.11}$$

unesp

onde *b* é uma translação indicando a localidade, *a* é a dilatação ou parâmetro de escala, *g(t)* é uma análise (básica) de wavelet e $g^*(.)$ é conjugado complexo de *g(.)*.

A transformada de wavelet dada pela equação (4.11), é portanto uma transformação linear que decompõe uma função arbitrária *x(t)* nas funções elementares $g_{a,b}(t)$, as quais são obtidas da wavelet *g(t)* por dilatação e translação. Aqui dilatação representa a harmônica ou período em termos da decomposição harmônica/periódica.

Cada valor da transformada de wavelet, $W_g(a,b)$, é normalizada pelo fator $1/\sqrt{a}$. Esta normalização garante que a energia dada por cada wavelet $g_{a,b}(t)$ é independente da dilatação *a*.

Para a função *g(t)* se qualificar como uma wavelet, ela deve satisfazer a condição de admissibilidade :

$$C_g = \int_{-\infty}^{+\infty} \frac{|G(f)|^2}{|f|} df < \infty \qquad (4.12)$$

onde G(f) é a transformada de Fourier da *g(t)*. Admissibilidade é requerida para obter a inversa da transformada de wavelet. Existem diferentes funções reais ou complexas usadas como wavelets (Chui - 1992). Nesta fase inicial de testes dos padrões de entradas, usou-se principalmente a wavelet de Morlet,

$$g(t) = e^{j2\pi f_0 t} e^{-\frac{|t|^2}{2}} \qquad (4.13)$$

porque ela é bem compreendida e testada. Ela permite boas escolhas de dilatação e translação e, portanto é bem adaptada para o problema de localizar mudanças bruscas em um sinal. Vários testes foram também realizados para diferentes famílias de Wavelets. Estes exemplos foram rodados no programa Matlab versão 5.2 e, a alteração das variáveis relacionadas a este tipo de análise, neste programa, é de forma interativa e de fácil implementação.

unesp

A transformada de Wavelet pode ser interpretada como um filtro de resolução de freqüência relativa constante, $\Delta f/f$, e apresenta um número de propriedades úteis que podem ser utilizadas:

. Localização no domínio do tempo-freqüência.

Da definição, as transformadas de Fourier extraem ondas periódicas infinitas da função analisada. Em contraste, a transformada de wavelet analisa uma função somente localmente na janela definida pela wavelet. A equação (4.11) é em geral não local. O valor de $W_g(a,b)$ em um ponto (a_0, b_0) depende de $x(t)$ para todo t. Contudo, as condições (4.9) e (4.10) proporcionam que a função $g(t)$ decai a zero quando $|t| \rightarrow +\infty$. Se for assumido um decaimento rápido, isto é, os valores de $g(t)$ são desprezíveis fora do intervalo $(t_{mín}, t_{máx})$, a transformada torna-se local.

A localização de freqüência é vista quando a transformada de wavelet é expressa em termos da transformada de Fourier

$$W_g(a,b) = \sqrt{a} \int_{-\infty}^{+\infty} X(f) G^*_{a,b}(af) e^{i2\pi fb} df \qquad (4.14)$$

Esta localização depende do parâmetro da dilatação a. A resolução local da transformada de Wavelet no tempo e freqüência é determinada pela duração e largura de banda das funções analisadas, dadas por:

$$\Delta t = a\Delta t_g \qquad e \qquad \Delta f = \Delta f_g / a$$

onde Δt_g e Δf_g são a duração e a largura de banda da função wavelet básica, respectivamente. Assim, no domínio da freqüência a transformada de wavelet tem boa resolução nas baixas freqüências e no domínio do tempo boa resolução nas altas freqüências, a última sendo adequada para análise de sinais transientes. Esta propriedade oferece diferentes resoluções tempo-freqüência, comparadas com outros métodos tempo-freqüência (Short - Time Fourier Transform e Wigner - Ville Transform).

. Detecção de mudanças bruscas no sinal.

A transformada de wavelet é adequada para detectar descontinuidades em sinais. As principais características da transformada de wavelet que surgem da descontinuidade no sinal,

unesp

são: há um aumento localizado nos módulos para pequenos valores do parâmetro de dilatação a em torno do ponto $b = t_0$ onde existe descontinuidade do sinal; há uma convergência de linhas de fase constante em direção ao ponto na extremidade (a, b).

. Relação entre a freqüência e dilatação.

A transformada de wavelet como uma decomposição de sinal não pode ser comparada diretamente a qualquer representação tempo-freqüência. Entretanto, há uma relação entre a dilatação e a freqüência. Para a wavelet de MORLET a relação entre o parâmetro de dilatação a_f e a freqüência f_x no qual a análise é focalizada, pode ser dado como:

$$a_f = f_0 \frac{f_s}{f_w} \frac{1}{f_x} \tag{4.15}$$

onde f_s é a freqüência de amostragem do sinal e f_w a freqüência de amostragem da wavelet

4.4 Análise Modal

Os conceitos de análise modal já são conhecidos e bem difundidos na literatura (Ewins – 1991). Tais conhecimentos são aqui retomados com uma abordagem simples, para que os conceitos básicos referentes ao assunto e, a técnica de ajuste de modelos que será aplicada na metodologia proposta possam ser entendidos.

A análise modal é o estudo do comportamento dinâmico de estruturas através do conhecimento de seus modos naturais de vibração. Os modos naturais de vibração de uma estrutura são formas preferenciais de movimento quando sobre a mesma não atuam forças de excitação.

É comum apresentar os modos naturais de vibração de um sistema dinâmico através da representação de sua configuração geométrica num ponto de deslocamento máximo em relação à sua posição de equilíbrio. O período T levado pelo sistema para retornar ao ponto de deslocamento máximo de seu modo natural de vibração é chamado de período natural. A frequência com que o sistema retorna ao ponto de deslocamento máximo de seu modo natural de vibração é chamada de frequência natural de vibração e seu valor, dado em ciclos por segundo (ou Hertz) é $f = 1/T$.

unesp

Inúmeras são as aplicações da análise modal de estruturas, desde o ponto de vista do simples conhecimento das frequências em que ocorrem os modos naturais de vibração, ao conhecimento dos modos de vibrar propriamente. Três tipos de informações básicas, fornecida pelo conhecimento dos modos de vibrar de um sistema, formam a base da maioria das aplicações de análise modal (Pereira – 1996, Gardel – 1993 e Turra - 1996). São elas:

1. *Informações do movimento livre de um sistema*, quando perturbado por um conjunto de condições iniciais de movimento. Conhecida a configuração inicial de movimento de um sistema dinâmico e, conhecidos também os seus modos e frequências naturais de vibrar, é possível determinar qual será o seu movimento livre. Tal informação é útil à predição da trajetória dos diversos pontos de um sistema físico submetido à condições iniciais de movimento. Pode-se utilizar o conhecimento da trajetória dos pontos de tal sistema para verificação de interferências ou grau de desalinhamento do sistema ao longo de seu movimento livre ou controle de sua trajetória.

2. *Informações sobre o movimento excitado de um sistema*, quando excitado por uma força conhecida. Neste caso, o conhecimento dos modos de vibrar do sistema é útil para a determinação de sua resposta a um tipo de força de excitação conhecida. Utiliza-se esta informação quando se verifica quais as frequências de ressonância de determinado sistema. Mede-se por exemplo, a amplitude do movimento de um ponto do sistema excitado por uma força harmônica de frequência conhecida. Variando-se a frequência da força de excitação, observa-se o aumento ou a diminuição da amplitude máxima de movimento do ponto medido. As frequências naturais do sistema correspondem as frequências de excitação harmônicas nas quais as amplitudes do movimento atingiram valores máximos. Neste caso, deve-se controlar a amplitude da força.

3. *Informações sobre as forças atuantes desconhecidas*, dado o movimento do sistema. Utiliza-se os conhecimentos dos modos de vibração do sistema, associado ao conhecimento de seu movimento, para determinar o tipo de força de excitação que gerou tal movimento. Pode-se, através deste princípio, identificar um componente defeituoso na estrutura ou em uma máquina, através do movimento introduzido por forças decorrentes da presença do defeito.

unesp

De uma maneira geral, o conhecimento das propriedades modais presta-se para a determinação de um dos dois elementos, resposta ou força de excitação, da equação esquemática representada na figura 4.3.

RESPOSTA	=	FORÇA DE EXCITAÇÃO	X	PROPRIEDADES DO SISTEMA

Figura 4.3 - Equação esquemática relacionando resposta e força de excitação.

A determinação dos modos de vibrar de sistemas dinâmicos pode ser realizada através de qualquer uma dentre as duas abordagens diferentes: modelagem e análise ou medição direta e análise modal.

Uma abordagem típica de *modelagem e análise de sistemas dinâmicos* para a obtenção de seus modos de vibrar é o método dos elementos finitos. A sistemática da utilização de tal método inicia com a discretização do sistema contínuo em elementos de dimensões finitas. Associa-se a cada elemento um sistema de pontos nodais ou graus de liberdade, sobre os quais se representam os movimentos de cada elemento. Além das informações geométricas do sistema discretizado acrescenta-se ainda informações sobre as propriedades físicas dos elementos (rigidez, massa e amortecimento) e as condições de compatibilidade de movimento de seus diversos graus de liberdade. Um programa computacional sintetiza a informação disponível sobre o sistema na forma de sua equação de movimento, que é dada por um sistema de equações diferenciais simultâneas de segunda ordem do tipo,

$$\mathbf{[M]}\{\ddot{x}\} + \mathbf{[C]}\{\dot{x}\} + \mathbf{[K]}\{x\} = \{f\} \tag{4.16}$$

onde **[M]**, **[C]** e **[K]** são matrizes quadradas de dimensão *nxn*, referentes à massa, amortecimento e rigidez do sistema, respectivamente. Os vetores $\{x\}$ e $\{f\}$ também de ordem n, possuem como elementos os deslocamentos e forças de excitação externas exercidas sobre cada um dos graus de liberdade do sistema. O sistema, neste caso, é dito discretizado em n graus de liberdade.

unesp

Os modos e frequências naturais dos sistemas, modelados por elementos finitos são encontrados após a solução de um problema de autovalores e autovetores, obtidos da equação matricial do movimento. Neste caso, os elementos de cada autovetor representam a configuração espacial do deslocamento máximo de cada coordenada do sistema em seu modo natural de vibração. Um autovalor associado a um autovetor (ou modo de vibrar) contém informações sobre a frequência e amortecimento associado àquele modo.

Os autovalores e autovetores associados a um sistema de equações diferenciais podem ser representados através das matrizes $[\Lambda]$ e $[\Phi]$ (Bathe and Wilson - 1984).

$$[\Lambda] = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \ddots \\ & & & \lambda_n \end{bmatrix} \tag{4.17}$$

$$[\Phi] = [\{\phi_1\}\ \{\phi_2\} \ldots \{\phi_n\}] \tag{4.18}$$

A matriz $[\Lambda]$ é diagonal e seus termos são os autovalores do sistema e, a matriz $[\Phi]$ é constituída dos autovetores, $\{\phi\}$, do sistema. O i-ésimo autopar é denotado por (λ_i , ϕ_i).

Os métodos de medição direta para a obtenção dos modos de vibrar de um sistema dinâmico são conhecidos como métodos de análise modal experimental. Estes métodos baseiam-se na medição dos movimentos e forças atuantes, em pontos selecionados do sistema dinâmico e subsequente processamento dos sinais medidos. Utiliza-se novamente aqui, uma discretização do sistema de tal forma a permitir a visualização dos modos naturais de vibrar a partir do resultado de análise dos sinais medidos . Pode-se também utilizar esta discretização para a construção de um modelo matemático do sistema a partir de suas propriedades modais medidas.

Os testes de determinação experimental das propriedades modais do sistema podem ser caracterizados quanto ao número de pontos de medição do sistema em: SISO (Single Input, Single Output), SIMO (Single Input, Multi Output) ou MIMO (Multi Input, Multi

unesp

Output). As medições dos tipos SIMO e MIMO, além de possuir um custo elevado em função do grande número de sensores e equipamentos utilizados na obtenção dos sinais, requer também sistemas computacionais robustos, capazes de processar eficientemente a massa de informações adquiridas. A maneira mais simples e de mais baixo custo de implantação para a realização de análise modal experimental é através de testes do tipo SISO. Uma desvantagem de tal método é o grande tempo gasto na realização do teste modal, pois as medidas são realizadas uma a uma, em cada par de pontos de discretização do sistema.

A análise modal experimental pode ser realizada no domínio do tempo ou da frequência. Os métodos de análise modal experimental no domínio do tempo caracterizam-se pelo cálculo direto de parâmetros modais do sistema a partir de registros no tempo de sua excitação e resposta. O tipo de informação fornecida por tais métodos é interpretada apenas à luz da modelagem discreta do sistema dinâmico, o que os torna, em certos aspectos, de difícil compreensão.

Os métodos de análise modal experimental no domínio da frequência, além de amplamente assimilados no meio de engenharia, fornecem como resultados intermediários informações qualitativas importantes do comportamento dinâmico do sistema em estudo. Tais métodos são portanto ainda empregados na grande maioria de testes de análise modal experimental.

Um último aspecto da determinação experimental dos modos naturais de vibração de um sistema é a possibilidade de utilização de um modelo matemático construído experimentalmente para a retro-alimentação ("updating") do modelo teórico obtido através do método de elementos finitos . Tal retro-alimentação é importante sobretudo para informar o modelo de elementos finitos sobre os valores de amortecimento medidos na estrutura, já que esta é uma grandeza de difícil estimação prévia.

Os testes de análise modal no domínio da frequência, baseiam-se na análise das curvas da função resposta em frequência conhecidas. A FRF é uma relação entre a transformada de Fourier do sinal de resposta (deslocamento, velocidade ou aceleração) em um ponto do sistema pela transformada de Fourier do sinal da força de excitação aplicada ao sistema em um ponto qualquer. Assim sendo, a FRF, H_{ij}, entre os pontos i de resposta e j de aplicação da força é dada por

unesp

$$H_{ij} = \frac{Y_i(f)}{F_j(f)} \tag{4.19}$$

$$Y_i(f) = \int_0^{\infty} y_i(t) e^{-j2\pi ft}\, dt \tag{4.20}$$

$$F_j(f) = \int_0^{\infty} f_i(t)\, e^{-j2\pi ft}\, dt \tag{4.21}$$

As expressões (4.20) e (4.21) são as transformadas de Fourier dos sinais de resposta e da força de excitação, respectivamente, nos pontos *i* e *j* do sistema.

A transformada de Fourier dos sinais de força de excitação e resposta do sistema é uma integral que pode ser calculada de maneira experimental pela amostragem e discretização dos sinais $y_i\ (t)$ e $f_j\ (t)$, através da Transformada Discreta de Fourier (DFT). A realização de uma discretização e amostragem dos sinais dinâmicos impõem duas restrições básicas à estimação da integral de Fourier:

a) limita o domínio de análise da frequência *f* entre *0* e $1 / 2\delta t$ Hz, onde δt é o intervalo de discretização do sinal amostrado;

b) restringe a análise a pontos de frequência múltiplos da frequência básica de análise δf, dada por $\delta f = 1 / N\delta t$, onde *N* é o número total de amostras do sinal.

Uma implicação direta de tal fato é que a análise modal experimental é limitada pela frequência máxima capaz de ser identificada através da FRF, ou seja, $f_{máx} = 1 / 2\delta t$ Hz. As frequências naturais do sistema que estejam acima do valor da frequência máxima de definição da FRF, além de não poderem ser identificadas na análise modal experimental, devem ser filtradas e eliminadas dos sinais a fim de evitar distorções nos resultados. A não identificação de algumas frequências naturais do sistema dinâmico resulta também na não determinação dos modos de vibrar a elas associados.

A FRF representa um sistema linear relacionando a saída $Y_i\ (f)$ e a entrada $F_j\ (f)$, pode-se, também, expressar $H_{ij}\ (f)$ em função dos parâmetros modais do próprio sistema.

unesp

Assim sendo, vê-se da teoria de análise modal que a função resposta em frequência entre os pontos i e j de um sistema de n graus de liberdade composto por massas, molas e amortecedores viscosos discretos é dado por (Ewins – 1991):

$$H_{ij}(f) = \sum_{k=1}^{n} \frac{r_{kij}}{j(2\pi f - \omega_{kd}) + \sigma_k} + \frac{r^*_{kij}}{j(2\pi f + \omega_{kd}) + \sigma_k} \tag{4.22}$$

onde

$$r_{kij} = x_{ki}\, x_{kj} \tag{4.23}$$

e

$$j = \sqrt{-1}$$

O termo complexo r_{kij}, da expressão (4.23), é chamado de constante modal da FRF, e r^*_{kij} o seu valor conjugado. Os termos x_{ki} representam o *i-ésimo* elemento do vetor modal (autovetor) x_k, e ω_{kd} e σ_k são respectivamente a frequência natural amortecida e amortecimento modal do sistema. Um outro resultado da teoria de análise modal é os modos de vibrar de sistemas com amortecimento, que são representados através de vetores com elementos complexos, o que justifica o aparecimento das constantes modais complexas r_{kij} acima.

Observa-se também da expressão (4.22) que a função resposta em frequência é definida no plano complexo e, sua representação dá-se espacialmente através de três eixos ortogonais de coordenadas: *f, parte imaginária de* H_{ij} *(f)* e *parte real de* H_{ij} *(f)*.

A representação mais comum da função resposta em frequência dá-se através de dois gráficos bidimensionais do tipo *amplitude* de H_{ij} *(f)* vs. *f* e *fase* de H_{ij} *(f)* vs. *f*. Tais gráficos recebem o nome particular de "diagrama de Bode".

O diagrama de Bode pode também ser apresentado utilizando-se escalas logarítmicas nos eixos coordenados. A utilização da escala logarítmica para o eixo da *amplitude* de H_{ij} *(f)* é

unesp

um artifício quando a variação do módulo de $H_{ij}(f)$ é muito grande nas regiões de ressonância. Nesse caso, a magnitude das maiores amplitudes de $H_{ij}(f)$ são atenuadas enquanto as amplitudes de menor magnitude são acentuadas.

O diagrama de Bode presta-se para uma análise qualitativa mais direta do comportamento dinâmico do sistema através de sua função resposta em frequência. Os valores de amplitude da FRF representados no diagrama de Bode correspondem à razão entre o valor médio do módulo da amplitude de resposta do sistema no ponto i e o valor médio do módulo da força (excitação) aplicada ao sistema no ponto j, para diferentes frequências de excitação f. Valores de máximos locais (pico) da figura correspondem às frequências naturais amortecidas do sistema em estudo. Os valores de fase no diagrama de Bode correspondem à direção relativa do movimento dos pontos i e j para diferentes frequências de excitação.

Encontra-se também correntemente a representação da FRF através de um gráfico único, bidimensional, que contém nos eixos coordenados as partes imaginária e real da FRF. Neste gráfico as frequências aparecem implicitamente como pontos no traçado de círculos. Tais gráficos recebem a denominação particular de diagrama de Nyquist.

A FRF recebe também alguns nomes característicos em função do tipo de sinal da resposta do sistema. É chamada de *receptância, admitância, complacência dinâmica* ou *flexibilidade dinâmica* quando a medida é de deslocamento, de *mobilidade* quando a resposta do sistema é medida em termos da velocidade de um ponto e de *inertância* ou *acelerância* quando a medida é de aceleração de um ponto do sistema. O inverso da FRF, $F_j(f) / U_i(f)$, recebe a denominação especial de *rigidez dinâmica, impedância mecânica* ou *massa aparente* quando o sinal da resposta do sistema é respectivamente deslocamento, velocidade ou aceleração. Nestes testes, atenção deve ser tomada, pois a impedância não pode ser calculada pela inversa da mobilidade.

4.4.1 Métodos de Análise Modal Experimental

Os métodos de análise modal experimental são voltados à realização prática de medições de várias FRF da estrutura de teste. Os elementos de um sistema básico de medição da função resposta em frequência através de excitação única e resposta única (SISO) é apresentado na figura 4.4. Existem essencialmente três aspectos do processo de medição que demandam especial atenção:

unesp

a) os aspectos mecânicos de fixação e da aplicação da excitação na estrutura,

b) a aquisição dos sinais, e

c) o processamento dos dados.

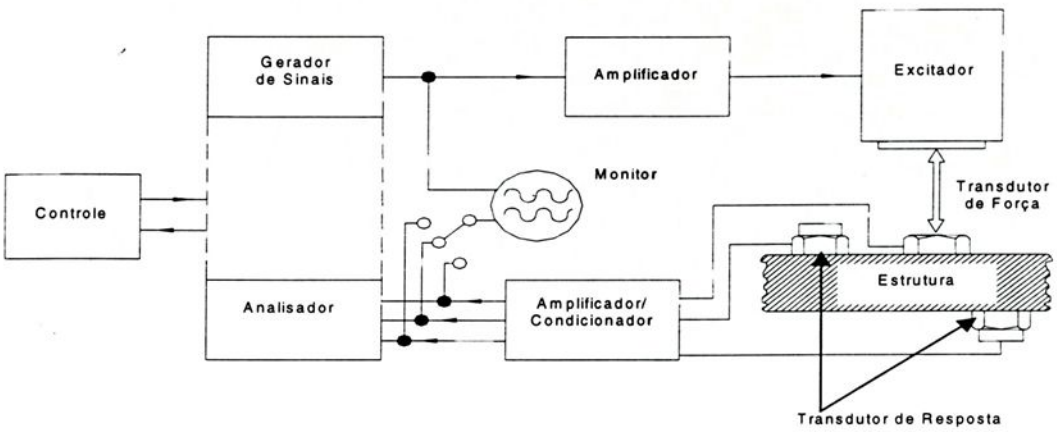


Figura 4.4 - Sistema básico de medidas (Ewins – 1991).

No primeiro dos três aspectos mencionados surgem questões relacionadas ao tipo de fixação. Geralmente, três são as opções disponíveis para o suporte da estrutura: livre (ou suportada por molas extremamente flexíveis); engastada ou fixa, o que requer a utilização de fixadores com rigidez bastante elevada; ou em condições normais de operação. A escolha adequada deverá ser decidida em função da aplicação. Freqüentemente, necessita-se relacionar os resultados de testes com os obtidos numericamente. Uma estrutura livre é de mais fácil simulação, em condições de teste, do que as estruturas fixas ou engastadas.

A aplicação da força de excitação é obtida através da fixação de um gerador de vibrações ("shaker") ou utilizando-se alguma forma de excitação transiente, tal como um choque de martelo ou o alívio súbito da estrutura de uma posição inicialmente deformada. Ambas as técnicas possuem vantagens e desvantagens e é muito importante a escolha do método de excitação mais adequado a cada teste.

Os transdutores são elementos importantes do sistema de teste, pois determinam a exatidão das medidas da força de excitação e da reposta da estrutura. Transdutores piezoelétricos são largamente utilizados para a detecção de forças e acelerações. Deve-se garantir que a interferência com a estrutura testada seja a mínima possível e que a faixa de

unesp

operação esteja adequada às variações de frequências e amplitudes de teste. Uma seleção incorreta dos transdutores pode ocasionar grandes erros nas informações medidas.

A resposta em frequência do sistema pode ser obtida diretamente através de uma excitação harmônica e medição da resposta harmônica resultante. Este tipo de teste, conhecido como "excitação senoidal", requer a fixação da estrutura a um excitador. A faixa de frequências é coberta por incrementos graduais de frequência de excitação ou pela variação lenta e contínua da mesma. Em ambos os casos espera-se a estabilização da estrutura em condições quase-estáticas. Outros sinais periódicos, pseudo aleatórios ou aleatórios substituem a abordagem de excitação senoidal, com a utilização de analisadores de sinais. A obtenção da resposta em frequência de sistemas, pode ainda ser obtida utilizando-se força de excitação impulsiva ou transiente. Todos estes procedimentos, diferentes da excitação senoidal, oferecem a possibilidade de realização de testes em um espaço mais curto de tempo, mas deve-se tomar cuidado pois a sua utilização envolve inúmeros passos, cada um deles potencialmente representando uma fonte de erros, quando mal aplicados. Uma forte compreensão das técnicas de processamento de sinais torna-se necessária para garantir o sucesso no uso destas técnicas.

unesp

V. RESULTADOS

Neste capítulo são apresentados 05 trabalhos na forma original como publicados ou, como encaminhados para publicação. São eles:

Automation in Fault Detection Using Neural Network and Model Updating (Pereira, Lopes Jr. and Weber – encaminhado à RBCM), seção 5.1.

Modeling of a Bipedal Locomotor Using Neural Network (Lopes Jr., Dutra and Nascimento – 1999), seção 5.2.

Identificação de Não Linearidades em Sistemas Dinâmicos Utilizando Redes Neurais Artificiais (Lopes Jr. e Nascimento – 1998), seção 5.3.

Identificação de Rigidez Bilinear em Sistemas Mecânicos Através de Redes Neurais (Lopes Jr. e Demarchi – 1998), seção 5.4.

Diagnosis of Rotating Systems Using Artificial Neural Networks (Lopes Jr. and Turra – 1996), seção 5.5.

unesp

AUTOMATION IN FAULT DETECTION USING NEURAL NETWORK AND MODEL UPDATING

João Antonio Pereira
Vicente Lopes Jr.,
Dep. Eng. Mecanica, UNESP - Ilha Solteira
15385-000 - Ilha Solteira- SP - Brasil
email: japereir@feis.unesp.br

Hans Ingo Weber
Fac. Eng. Mecânica de Campinas - UNICAMP
Cx. Postal 6051 - Cep. 13083-970
Cidade Universitaria - Campinas - SP - Brasil
hans@mec.puc-rio.br

Abstract

In this article, an implementation of structural health monitoring process automation based on vibration measurements is proposed. The work presents an alternative approach which intent is to exploit the capability of model updating techniques associated to neural networks to be used in a process of automation of fault detection. The updating procedure supplies a reliable model which permits to simulate any damage condition in order to establish direct correlation between faults and deviation in the response of the model. The ability of the neural networks to recognise, at known signature, changes in the actual data of a model in real time are explored to investigate changes of the actual operation conditions of the system. The learning of the network is performed using a compressed spectrum signal created for each specific type of fault. Different fault conditions for a frame structure are evaluated using simulated data as well as measured experimental data.

Keywords: Neural Network, Fault Detection, Model Updating, Predictive Maintenance

Introduction

The entire life cycle of a product involves an iterative engineering effort. The initial design concept is likely to give rise to a variety of design options each of which will be evaluated against the requirements of the product specification. At this stage, usually, is constructed a prototype and their properties are evaluated and compared with the design requirement. The Structural Dynamic Modification (SDM) is a term usually used in modal analysis and implies the incorporation, into an existing model, of design modification or new information gained either from experimental testing or other source, in order to evaluate the accuracy of the model.

The SDM algorithm is useful for solving the forward variational problems for structures, i.e., for a given variation into the mass, stiffness or damping properties, it leads to the corresponding changes in the modal properties of the model. However, to access a high performance against the

requirements of the product specification demands a safety and reliable monitoring of the operation conditions and failure diagnostic of the model. This involves a solution of an inverse problem, or modal sensitivity problem. That is, for a given variation in the modal properties of the structure, which corresponding changes in its mass, stiffness and damping properties have taken place. Application as fault detection requires solutions of this kind of inverse problem. Unlike the forward variational problem, fault detection applications can not be solved in a straightforward manner. Its solution, for most practical cases, requires the inversion of a rank deficient matrix, which creates numerical difficulties to identify the correct changes of the structural properties that have to taken place into the model.

Numerous researches have proposed solutions using modal sensitivity problem to damage detection of structures through vibration changes. As modal testing has become more widespread, finite element model updating using measured modal test data has also become popular in the field of damage detection. The relative success and the potentiality demonstrated by these techniques, mainly, those based on vibration properties have shown that they are promising tools for damage detection purpose (Pereira, 1994, Hemez, 1993, Zimmerman, 1995). However, for purpose of monitoring, they are still under investigation (far to be consolidated). The recent literature has been shown some attempts to use permanently working monitoring system, based on natural excitation source actuating in the system to evaluate its health condition (Rohrmann & Rucker, 1994) but the results are still limited.

Artificial Neural Networks (ANN) have recently emerged as a good tool for pattern recognition turning their application in promising tools for monitoring and fault classification of machines and equipments. The technique offers promise for solving inverse varational problem in the context of monitoring and fault detection because of their pattern recognition and interpolation capabilities. In order to identify the fault in the model, it must be trained using a set of solutions to its corresponding forward variational problem. In this case, the neural networks should be designed to classify the input patterns in pre-defined classes or to create categories of group patterns according to their similarity. An important feature of the neural networks, in this context, is their inherent ability to operate on noisy, incomplete or sparse data and to model process from actual system parameters (Uhl, 1994). The neural network could respond in real time to the changes in the system state, provided by continuous sensor inputs. Hence this technique should facilitate fault identification in real time, providing a good Non-Destructive Evaluation (NDE) tool to investigate

unesp

the state of operation and health condition of a system based on measured vibration data. Nevertheless, several attempts to use neural networks faced difficulties to generate reliable set of information to train the net. It demands a wide range of experimental tests to cover the whole possibility of damages of the model, which makes its use prohibitive.

This work presents an alternative proposal to exploit the capability of neural network associating it with model updating technique to investigate the state of operation and health condition of the system. It creates a very reliable updated finite element model, using a FRF-based model updating approach (Larson, 1992; Lammens, 1995; Pereira, 1996) and uses SDM to generate the patterns faults to feed the ANN during the training phase. The approach initially adjusts the finite element model and validates its response properties in terms of the dynamic measured data of the real system. Following, it simulates different types of damage conditions to train the network.

The proposed approach is addressed to fault detection on the frequency analysis domain, where each machine defect generally produces a set of vibration components that allows the recognition of different faults. However, its application is similar for others kind of signal since there are many others parameters that can be measured to evaluate the operation condition of a machine such as pressure, temperature, sound, oil particles, etc.

Methodology

An Artificial Neural Network (ANN) approach associated with a Finite Element Model Updating Procedure (MUP) has been used to operate the monitoring and failure diagnosis of machines and equipments. The updating approach performs an improvement of the FE-Model in order to define a representative mathematical model of the actual system. The improved model permits to create the various damage conditions required to define the reliable set of information data concerning each kind of failure liable to occur in the life cycle of the system. The network is designed to classify the input patterns of the model in pre-defined classes of inputs or to create categories of group patterns according to their similarity, by a training process phase. In the monitoring phase, it compares the actual measured data with the input patters aiming at recognising the existence of fault in the system. The updated FE-model permits to generate the various damage conditions necessary for the training of the network. So, the ANN may lead to diagnose of the system in real time, facilitating the identification of faults in the operating system.

cm 1 2 3 4 5 6 7 8 unesp 12 13 14 15 16 17 18

The approach comprehends a definition of a reliable FE-Model, the generation of whole expected failure conditions, the training of the ANN and finally, the recognition and location of each kind of fault. The ANN learns by example and the pattern of damaged signal used to feed the training of the network, frequency response functions, is created in a straight forward manner by applying SDM. Once the network is trained, it can execute any pattern recognition task.

The classification of the vibration signal, generated from the improved finite element model, comprehend a separation of the vibration data in signals of components operating normally, signals of components operating with fault; a compression of the signals and finally, classification of the compressed patterns using a backpropagation algorithm. The compression of the pattern data is important because it transforms a large set of data in an equivalent reduced set containing the same information provided by the original signal. This reduction decreases significantly the time spent to train the network as well as the reduction of the instability in the recognition process (Lopes Jr. and Turra, 1996). The compression process consists in the compression of the spectrum size of the signal in a reduced number of points. It constructs a new equivalent signature for the system based on the maximum amplitude and corresponding frequencies points of the original signal. The algorithm searches the maximum amplitude points in the original signal at an earlier specified frequency range and considers a band width of 3.0% around each frequency point related with maximum amplitude points.

Background of the Neural Network

ANN is made by a number of processing elements which are connected to form layers of neurons. Each neuron is a simple mathematical processing unit that, when it is interconnected, produces a complex structure with different architectures, which can be trained to recognise defined patterns. It is common to specify the architecture of the network by referring to the number of hidden layers; layers that are neither inputs nor outputs. The topologies and size of the network depend on the specific application, and the definition of an optimal choice is case dependent (Bernieri, 1994, Worden and Tomlinson, 1992, Yuedong and Du, 1996). Figure 1 shows the architecture used, in this approach, it has one hidden layer with 500 neurons for an output layers with 31 neurons.

unesp

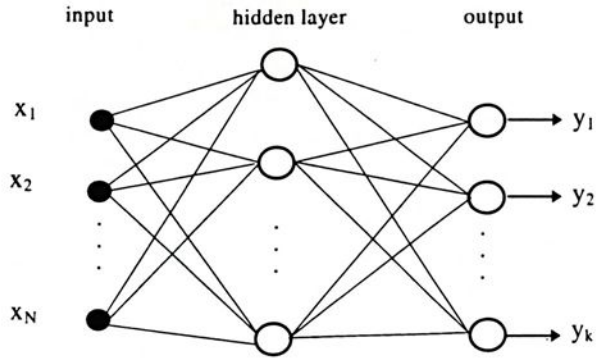


Figure 1: Architecture of the neural network with one hidden layer

The input x's are weighted by $w^h_{j,i}$ and by the bias b^h_j , and the output results feed the hidden layer. The output of the j^{th} hidden unit can be described by:

$$y^h_j = F_1\left(b^h_j + \sum_{i=1}^{N} W^h_{i,j} . x_i \right) \qquad (1)$$

where the superscript (h) means that the quantities pertain to the hidden unit and $\mathbf{F_1}(\, . \,)$ is a sigmoid nonlinear function. The output of the net is biased and weighted by sum of the hidden layer outputs:

$$y_l = F_2\left(b^o_l + \sum_{j=1}^{H} W^o_{j,l} . y^h_j \right) \qquad l = 1, 2, \ \ldots , k \qquad (2)$$

he superscript (o) refers to the output unit. **H** is the number of units in the hidden layer and $\mathbf{F_2}(.)$ is a linear activation transfer function. The net is trained starting with a random set of weights and biases and calculating the output for every input set. Then, the error E is computed from the output layer backwards, which has historically been called the backpropagated error, and the learning algorithm the backpropagation. Several publications (Hecht and Nielsen, 1990; Wasserman, 1989; Hush and Horne, 1993) have presented details of this algorithm.

$$E = \text{sumsqr}(\{T\} - \{y\}) \qquad (3)$$

unesp

where **sumsqr** is the sum of square of the elements, and T is the desired output. The error E is formulated as a function of the weights and biases and then, the minimisation is performed by means of the usual steepest descent algorithm. The error E is denoted by $\{\alpha_j\}$, where the index *j* taking on as many values as there are weights and biases. When the parameter α_j changes by $\delta\alpha_j$, the error E changes by

$$\delta E = \frac{\partial}{\partial \alpha_j}(E)\,\delta\alpha_j \tag{4}$$

where $\frac{\delta}{\delta\alpha_j}(E)$ is evaluated using the current values of the parameter. To ensure that changes in α_j results in a decrease in E, it selects

$$\delta\alpha_j = -\eta\frac{\partial}{\partial \alpha_j}(E) \tag{5}$$

where η is a positive constant. The parameter η determines how large a step is made in the direction of the steepest descent and therefore how quickly the optimum parameters are obtained. For this reason η is called the learning coefficient.

The training of the network uses a set of inputs for which a specified output is known and in this case, the process finishes when the error ***E*** is smaller than a desired value or it reaches a maximum specified number of iterations. In the later case, the training set is considered unsatisfactory. The error E considered here, is the sum of the squared error between the output value provided by the trained network and the value obtained for a specific input. The outputs are combination of zeros and ones for each state of operation of the machine, for the intact structure the outputs are equal one for the first component and zero for the others.

Model Updating Procedure

The MUP used to improve the original finite element model is based on the measured frequency response functions (FRF(s)). A detailed discussion can be found in Lammens, 1995 and Pereira, 1996. The procedure operates at an element level and the model changes are directly

unesp

related to the variation of the parameters of each element. It describes the discrepancy between analytical and experimental models as a force residue written in terms of the analytical dynamic stiffness matrix $[Z^A]$, and the experimental FRF matrix $[H^X]$.

$$\{\varepsilon\} = \mathcal{E}\left(\left[Z^A\left(p_1 + \Delta p_1,\ p_2 + \Delta p_2,\ \ldots,\ p_m + \Delta p_m\right)\right], \{H^X\}\right) \cong \{0\} \qquad (6)$$

This discrepancy of the models is used to define the updating set of equations and the solution of the updating problem consists in finding *m* analytical parameter changes that minimises the difference between the two models. The p-parameters can represent any physical or geometrical property of the model, they can be a parameter of an element in the case of the elements being adjusted independently or a parameter of a set of elements when some elements are adjusted proportionally. The Δp's are parameters correction of the p-parameters that minimise the discrepancy between the two models. The formulation of an approximated solution of the problem, through a first order linearization of the expression (6) in the p-parameters, leads an overdetermined set of linear equations:

$$\{\varepsilon\} = [S]\{\Delta p\} - \{B\} \qquad (7)$$

where the elements of the sensitivity matrix [S] and these of the difference vector {B} are known elements. They are obtained from the analytical dynamic stiffness matrix and the experimental measured frequency response functions at each selected measured frequency points respectively. The elements of the vector $\{\Delta p\}$ are unknown parameter corrections to be estimated. The solution of this system of linearised equations system provides an improved analytical model. If the discrepancy between the models is not yet sufficiently small, the procedure is iteratively repeated.

Experimental Test

To demonstrate the applicability of the proposed methodology, a laboratory spatial frame structure, Figure 2, has been investigated. An analytical modelling by finite elements and a modal test of the structure were carried out and compared aiming at defining a representative analytical-numerical model able to predict the real behaviour of the structure. The initial comparison of modal parameters from both, analytical and experimental models, showed a deviation of the predicted data from the experimental one, Table 1 and Figure 3.a. The adjustment of the analytical finite element

unesp

model and the consequent improvement of the correlation of the models was performed by a FRF based model updating procedure (Pereira 1996). The spatial parameters of the finite element model were successfully updated by using the FRF based model updating and, the correlation of the models was improved satisfactory, Table 1. The elements selected to be adjusted during the updating process were those elements evolved into the welding connections. Figure 3.b shows the superposition of the FRF(s) in the drive point for the experimental test and the corresponding new reliable finite element model. The updating process, as shown in Table 1 and Figure 3, allowed the definition of a representative mathematical model of the actual system.

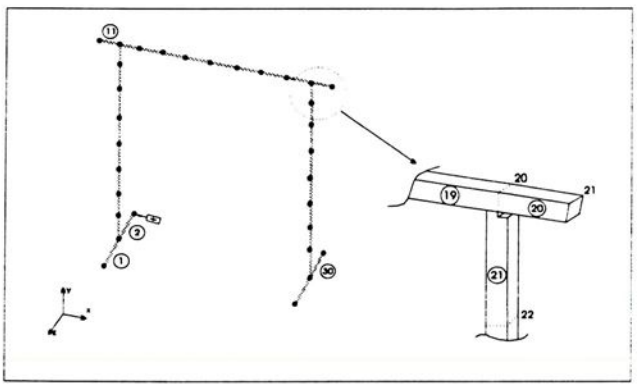


Figure 2: Frame structure - Detail of the cut bar

Table 1: Experimental, Analytical and Updated models correlation

Experimental model		Analytical model		Updated model		Initial correlation		Final correlation (iteration 5)	
mode	freq. [Hz]	mode	freq. [Hz]	mode	freq. [Hz]	Δf [%]	MAC-val.	Δf [%]	MAC-val.
1	17.06	1	17.26	1	16.98	1.17	0.994	-0.43	0.995
2	22.10	2	25.30	2	22.38	14.47	0.991	1.26	0.991
3	42.44	3	43.86	3	42.73	3.34	0.990	0.68	0.990
4	97.54	4	102.06	4	97.04	5.71	0.978	-0.02	0.985
5	112.86	5	121.63	5	112.91	8.73	0.975	0.66	0.974
6	142.36	6	144.93	6	141.60	2.73	0.834	-0.53	0.881
7	153.83	7	155.03	7	153.71	0.78	0.954	-0.07	0.953
8	171.13	8	176.21	8	172.56	2.96	0.978	0.83	0.982

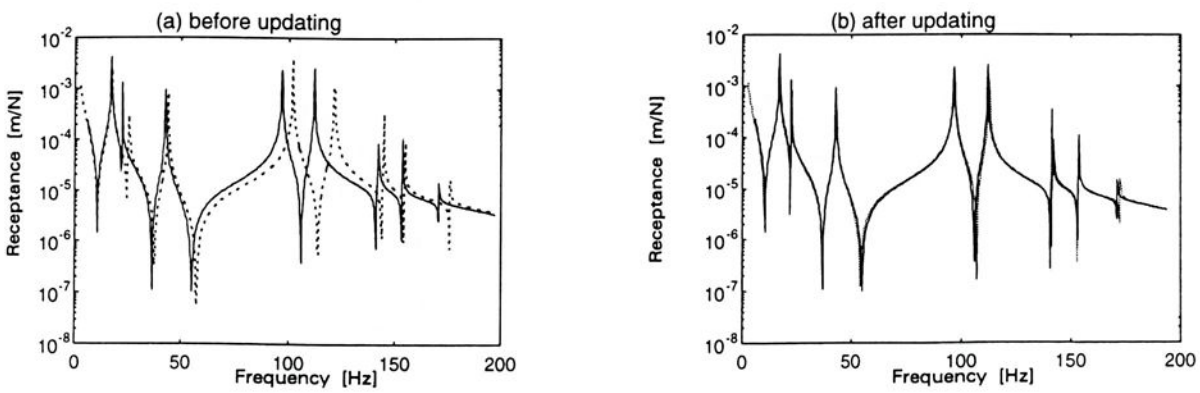


Figure 3: Experimental (full line) and analytical (dashed line) receptances (point 1, x-direction)

Different damage conditions of the structure were obtained numerically by using the updated FE-model of the structure. Each fault introduced to the structure consisted of a reduction of 50% of the sectional area of the associated element. A frequency range covering the first eleven mode shapes of the structure was analysed. For each damaged element a set of FRF(s) was created which contains the whole informations used to feed the training of the network. The training was based on the compressed set of data obtained from the original one.

Compression of the Signal

A compression method based on the maximum amplitudes of the spectrum signal has been used to reduce the number of points of the signal and to construct a new equivalent signature for the system. The method localises the frequency points related with the maximum amplitudes and creates a masked signal with a band width of 3% of the frequency points of maximum amplitudes. This permits to comprise small deviations of the measured FRF(s) since, in practical cases, it is almost impossible to calculate FRF(s) that matches exactly the measured ones, due to problems related with the modelling approach as well as the experimental test. Once the reduction is effective, the procedure tries to associate the compressed signal response with a set of specific faults. Figures 4.a-b show an original measured FRF (512 points) and the corresponding compressed or masked curve (57 points) respectively for the intact frame structure and the damaged structure (element 13). The number of points of the compressed signal depends on the number of frequencies point related with the maximum amplitude peaks of the signal.

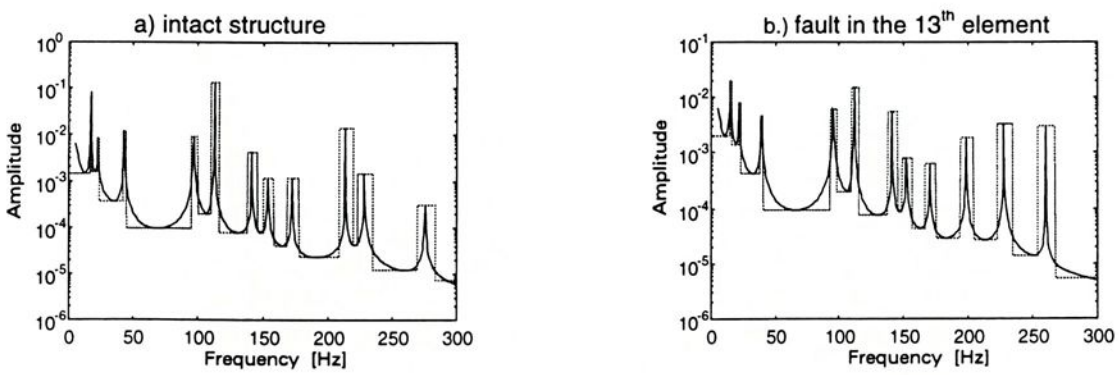


Figure 4: Original (full line) and compressed (dotted line) signal

Numerical Results

Machines failure is a phenomenon liable to occur during the service life of most machines and equipaments. However, in some cases, the analyst may have previous information of the region more susceptible to failure and should take it into account to search the failure. Nevertheless, to evaluate the robustness of the approach it is assumed that all elements of the structure are likely to failure and the procedure tries to identify the fault in the element without any previous information. In this case, the frame structure was divided in 30 elements and the procedure was trained to identify fault in any element. All conditions of failure were investigated, elements of low probability to failure (elements in the mid of bars), elements more liable to failure (elements in the welding connections) and elements of very low sensitivity to the damage. Table 2 shows the error obtained when each specific curve is submitted to the trained network and compared with the patterns curves used for the training of the network.

Table 2 Error value of the recognition of the faults.

fault condition / curves	*undamaged*	*fault element 1*	*fault element 2*	...	*fault element 30*
input 1 (intact struct.)	**9.6063e-006**	4.8641e+000	3.7970e+000		1.1696e+001
input 2 (fault elem. 1)	6.7413e+000	**9.7144e-006**	9.7573e-001		4.0104e+001
input 3 (fault elem. 2)	1.7271e+001	2.5262e+000	**9.6776e-006**		2.0471e+001
input 4 (fault elem. 3)	3.2935e+001	3.2589e+001	3.3665e+001		5.4743e+001
input 5 (fault elem. 4)	1.6834e+001	1.7251e+001	1.5313e+001		1.3523e+001
input 6(fault elem. 5)	1.7713e+001	2.0244e+001	2.0787e+001		9.6019e+000
input 7 (fault elem. 6)	2.9976e+001	3.0634e+001	3.0579e+001		1.8342e+001
input 8 (fault elem. 7)	3.9097e+001	4.7494e+001	4.5525e+001		1.3870e+001
input 9 (fault elem. 8)	2.0037e+001	1.6979e+001	1.5710e+001		1.3138e+001
input 10 (fault elem. 9)	1.5272e+001	2.6424e+001	2.2509e+001		1.1539e+001
input 11 (fault elem. 10)	1.3199e+001	1.7504e+001	1.3277e+001		2.5670e+001
input 12 (fault elem. 11)	3.9884e-002	1.7489e+001	1.7140e+001		2.6266e+001
input 13 (fault elem. 12)	3.4204e+001	3.0950e+001	3.0608e+001		3.4119e+001
input 14 (fault elem. 13)	9.8143e+000	1.1770e+001	1.2617e+001		1.5352e+001
input 15 (fault elem. 14)	4.5487e+000	5.7721e+000	6.3547e+000		1.7717e+001
input 16 (fault elem. 15)	1.9961e+001	1.9895e+001	1.6745e+001		2.6779e+001
input 17 (fault elem. 16)	3.1467e+001	2.1154e+001	2.1000e+001		4.4202e+001
input 18 (fault elem. 17)	3.1031e+001	5.4119e+001	5.4335e+001		6.8628e+001
input 19 (fault elem. 18)	1.5390e+001	7.2700e+000	5.9768e+000		1.7626e+001
input 20 (fault elem. 19)	4.0137e+000	2.7592e+000	4.0291e+000		2.5086e+001
input 21 (fault elem. 20)	3.2059e-002	9.4283e+000	5.1077e+000		2.2617e+001
input 22 (fault elem. 21)	1.7899e+001	1.7955e+001	1.6803e+001		4.1428e+001
input 23 (fault elem. 22	5.5282e+000	6.6056e+000	6.2168e+000		1.7987e+001
input 24 (fault elem. 23)	2.0025e+000	2.1186e+000	2.3363e+000		5.8111e+001
input 25 (fault elem. 24)	4.1322e+001	2.7704e+001	2.4730e+001		4.8111e+001
input 26 (fault elem. 25)	3.2059e-002	9.4283e+000	5.1077e+000		2.2617e+001
input 27 (fault elem. 26)	1.7899e+001	1.7955e+001	1.6803e+001		4.1428e+001
input 28 (fault elem. 27	5.5282e+000	6.6056e+000	6.2168e+000		3.7987e+001
input 29 (fault elem. 28	2.0025e+000	2.1186e+000	2.3363e+000		4.8111e+001
input 30 (fault elem. 29)	2.0025e+000	2.1186e+000	2.3363e+000		4.7111e+001
input 31 (fault elem. 30)	4.1322e+001	2.7704e+001	2.4730e+001	...	**9.1069e-006**

unesp

Figures 5.a - 5.p give an overview of the normalised errors for some investigated failure conditions. The plots show the errors, normalized in relation to the minor component, obtained through a "comparison" of each specific "measured" curve with all others curves used for the training of the network.

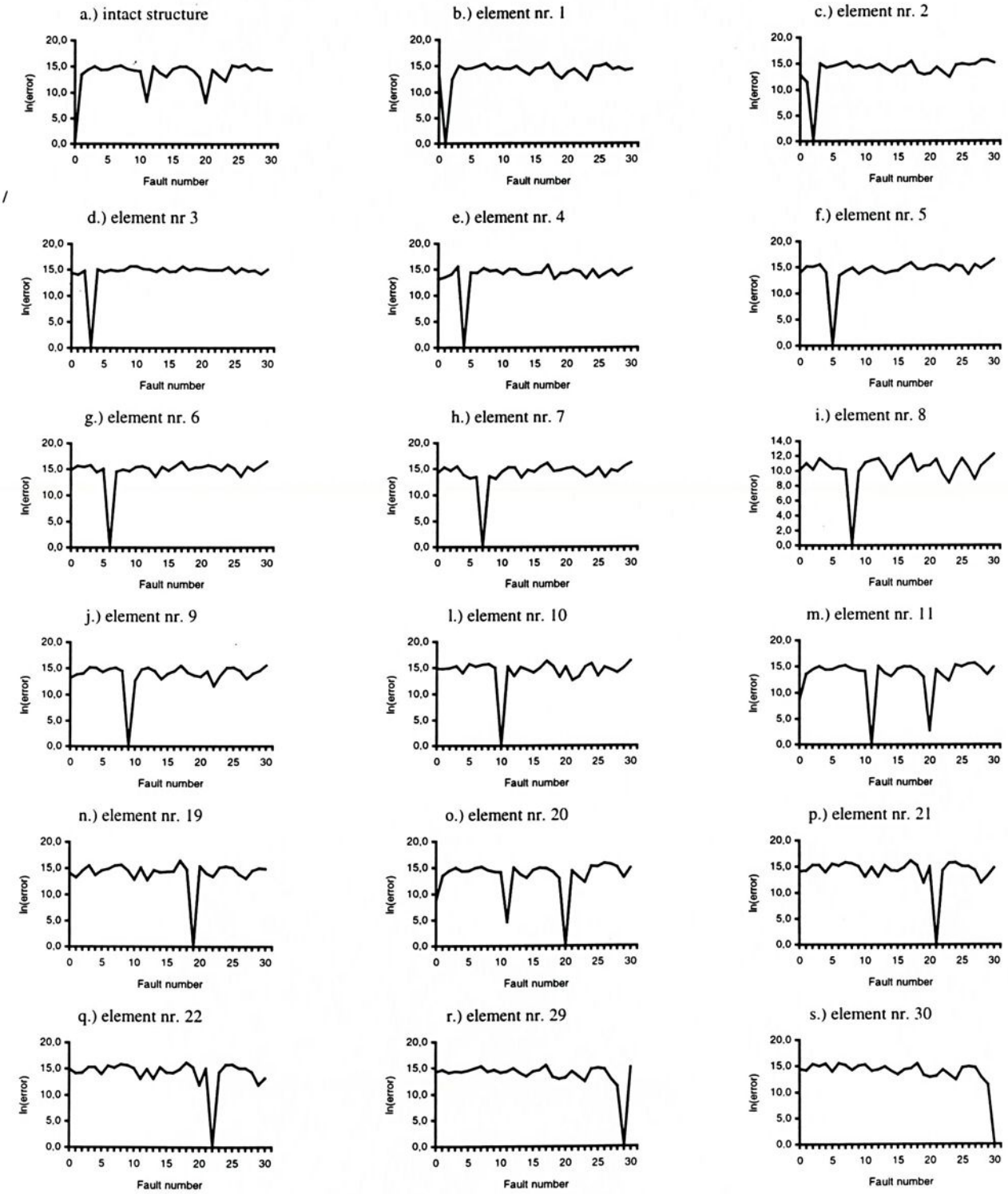


Figure 5: Identification of the fault in the frame structure

The lower errors show the similarities of the curves and consequently the searched fault. As expected in this case, all faults as well as the intact condition of the structure were clearly identified since the training of network was based on these curves. The identification of the damaged element in figure 5.m is not very clear since the error for element 20 is very low too. It happens because elements 11 and 20, due their position in the structure, have few influences in the dynamic behaviour of the model, and even a very large damage in these elements does not affect the dynamic properties of the model. The same effect is verified in figure 5.o. Therefore, recognition faults in these elements, based on the changes of the dynamic properties of the model, is a difficult task

Experimental Results

In the above investigated case, the data of the damaged structure was created through the same numerical model used to train the network. Thus, the simulated data used in the training of the net and the "experimental" data of the damaged structure are identical. However, for a real case the experimental measurement data present a light difference in relation to the simulated data used to train the network, turning more difficult the comparison of the data and the recognition of the fault. Hence the compression of the measured data is performed in two stages aiming to comprise this light difference. Initially, the measured FRF curve is compared with the simulated compressed curves utilised to train the network aiming to define if the frequency band of the simulated compressed data includes the frequency points of maximum amplitude of the measured curve. If the points are within the frequency band of the simulated curve, the procedure uses the same points of frequency used in training phase to compress the measured signal. In sequence, the signal is compared with the curves used in the training phase in order to detect the probable fault. Figure 6 shows the measured experimental curve of the damaged structure (element 21) and the masked one utilised to train the network. The mask was derived, from the improved FE-model, during the simulation of the failure condition of the model. Figure 7 shows the normalised error when the measured experimental curve is compared with all curves used to train the net. As it can be seen, the smallest error occurs for curve number 21, that means the real damaged element (element nr. 21).

unesp

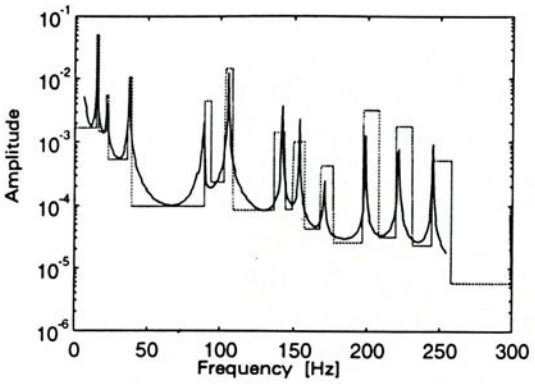


Figure 6: Original (full line) and compressed (doted line) response signal

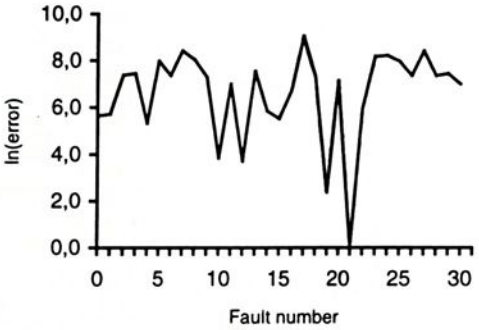


Figure 7: Identified error among the curves

5. Conclusion

A fault detection methodology using neural networks and model updating for identifying the presence of structural damage has been presented and demonstrated with an application to a frame structure. The feasibility of the approach is evaluated for simulated numerical data and experimental test. As discussed, the approach compares the current characteristic response of the model with previous pattern of the state of operation which were defined by applying SDM in the representative FE-model.

A compression method has been used to reduce the number of point of the signal that will feed the ANN. It creates an equivalent signature for the system based on the maximum amplitudes of the original spectrum signal. Considering a frequency range of 3% around of the points of maximum amplitudes is satisfactory in this study case. It seems to suppress the effects of small oscillation (variations) in experimental measurements The architecture of the network has been defined by tests since it does not comply a criterion to define the optimum architecture. However, the results seem to be better when the neurons number in the hidden layer increase. Parallel issues such as fault into elements of low sensitivity and robustness when the measured data comprise small deviation of previous pattern due measurement difficulties have been addressed.

In general for condition monitoring such issues are extremely important since it is essential to recognise a fault in time to permit action to be taken in order to avoid failure of the component. The preliminary results show that the proposed methodology can be applied to classify an unknown signal in patterns fixed beforehand. Although the approach has successfully identified the damaged

unesp

element for all simulated test cases as well as for the experimental test, its success is not unconditional. In order to explore in depth the potential for application of such propose to monitoring health condition and failure location in real structures, further research addressing more complex case studies must to be done.

References

Bernieri, A. at all; "A Neural Network Approach for Identification and Fault Diagnosis on Dynamic Systems," *IEEE Trans. on Instrumentation and Meas.*, vol. 43, nº 6 pp. 867-873, Dec./94.

Hecht-Nielsen R.; *Neurocomputing* - Addison-Wesley Publishing Company, 1990.

Hemez, F.M. and Farhat, C. "Locating and Identifying Structural Damage Using a Sensitivity – Based Updating Methodology," AIAA/ASME/ASCE/MAS 34th Structures, Dynamics and Material Conference – California 1993

Hush D.R. and Horne, B.G.; "Progress in Supervised Neural Networks - What's New Since Lippmann," -*IEEE Signal Proc. Mag.*, pp. 8-39 - Jan./1993.

Lammens. S., "*Frequency Response Based Validation of Dynamic Structural Finite Element Models*," Ph.D., Thesis, Katholieke Universiteit Leuven, Belgium, 1995.

Larsson, P. O. and Sas, P., "Model Updating Based on Forced Vibration Testing Using Numerically Stable Formulations," Proceedings of the X-IMAC, 1992.

Lippmann, R.P.; "An Introduction to Computing with Neural Nets," *IEEE - ASSP Magazine*, pp. 4-22, April/87

Lopes Jr., V. and Turra, A. E., "Diagnosis of Rotating Systems Using Artificial Neural Networks", Application of Artificial Intelligence in Engineering XI, includes CD Rom - Computational Mechanics Publications, Sep. 1996.

Narenda, K.S. and Parthasarathy, K.; "Identification and Control Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, vol. 1, nº 1 pp. 4-24, March 90.

Pereira, J. A., "*Structural Damage Detection Methodology using a Model Updating Procedure based on Frequency Response Functions - FRF(s),*" Tese de Doutorado, Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica, Campinas, 1996.

Rohrmann, R.G. and Rücker, W., "Surveillance of Structural Properties of Large Bridges using Dynamics Methods," Structural Safety & Reability, Schüeller, Shinozuka & Yao (eds) Rotterdam, ISBN 90 5410357 4, 1994

Schwaarz, B. J., et all; "Using SDM to train Neural Networks for Solving Modal Sensitivity Problems",. Proceedings of the XIV-IMAC, 1996.

unesp

Uhl,T.; "Automation of monitoring and diagnostic process for rotating machinery using neural networks," *ISMA - Intern. Seminar of Modal Analysis* - Leuven, Belgium, Sept./94.

Yuedong, C., and Du, R.; "Artificial Neural Network Architecture Design Using Similarity Measure with Applications in Engineering Monitoring and Diagnosis," *ASME Journal of Dynamic System, Measurement, and Control*, , vol 118, pp. 635-639, sept. 1996.

Wasserman, P.D. *Neural Computing; Theory and Practice* - Van Nostrand Reinhold, 1989.

Warden, K. and Tomlinson G. R.; " Modeling and Classification of Non-linear System Using Neural Networks," Mechanical Systems and Signal Processing, pp. 319-356, 1994.

Worden, K. and Tomlinson, G. R.; "*Classifying Linear and Non-linear Sistems Using Neural Networks*," Procedings of the 17th International Seminer on Modal Analysis, Leuven, pp.903-922, 1992.

Zimmerman, D.C. et *al "Structural Damage Detection Using Frequency Response Functions*,"Proceedings of the XII-IMAC, 1995.

Modeling of a Bipedal Locomotor Using Neural Networks

Prof. Dr. Vicente Lopes Júnior, Prof. Dr.-Ing. Max Suell Dutra & Ádamo de Oliveira do Nascimento
UNESP - Ilha Solteira - Av. Brasil 56 - Ilha Solteira
CEP 15385-000 - São Paulo - Brazil
E-mail: {max, vicente}@dem.feis.unesp.br

Abstract

Research to date points toward an understanding of human biped locomotion has been primarily experimented in the nature, due to the complexity involved in the process. There is strong evidence that many biological clocks consist of a population of mutually coupled oscillators. [1, 2, 3]. In this work, a form of bipedal locomotion using mutually coupled nonlinear oscillators generated by neural network is presented. The method of backpropagation is considered to generate the angles of each link.

Introduction

Locomotion of humans and animals, such as walking, swimming and running is performed by rhythmic, synchronized motions. In locomotion using legs a very large number of degrees of freedom is involved and is essential to have well coordinated motion of these degrees of freedom. The main part of this coordination takes place in the central nervous system. This system generates signals according to the desired gait. The process of signal generation can be modeled by a pattern generator and can be designed as a network of mutually coupled nonlinear oscillators. The presentation of this system using Van-der-Pol oscillators can be viewed in [3, 4]. Another way to generate it is using the artificial neural network. Where this network consists of simple computational elements or neurons. They are organized in layers and operates in parallel. The neural network has ability to learn complicated relations of non-linearities and is used to model complexed systems [5]. A lot of researchers developed the control and identification of non-linear systems using the classical method of backpropagation. One of the merit of this method is the capacity of interpolation if we compare with another method known as Hopfield.

One of the greatest problem in the realization of walking machine is control. It is basically responsible for the locomotion, the stability of the machine and keeping certain marginal requirements as for example low energy consumption. Part tasks are the controlling of a lot of actuators at the same time, the coordination of legs, the calculation of leg trajectories and foot set up points as well as the adaptation of the movement to the environment. It is essential that all these calculations are carried out in real time. Neural networks which can learn control, work in high parallel, are fault-tolerant as regards defects and have a high adaptivity which offer a promising alternative to traditional control structures. The control architecture was compiled with supervised learning backpropagation networks.

Determinants of Gait

Natural bipedal locomotion involves more than 200 degrees of freedom. The modeling of natural biped locomotion is more feasible by reducing the number of degrees of freedom taken into consideration the most important determinants of gait. According to Saunders et al [6] the most important determinants of gait are: 1 - the compass gait which is performed with stiff legs like an inverted pendulum; 2 - pelvic rotation about a vertical axis; 3 - pelvic tilt; 4 - knee flexion of the stance leg; 5 - plantar flexion of the stance ankle and 6 - lateral displacement of the pelvis. In order to

realize these determinants of gait a three dimensional model with a minimum of 15 degrees of freedom [4] is needed. In order to simplify the investigation, a two dimensional model that performs only the motion in one plane with 3 determinants of gait, will be considered (Figure 1). The walking period can be divided into 3 intervals (Figure 2). During first interval the cinematic chain forms a double support phase (with 3 degrees of freedom), during the second form a single support (with 6 degrees of freedom) and the third one, yet a single support phase but with 7 degrees of freedom.

Since locomotion of humans and animals is rhythmic, locomotor of control systems must be able to produce rhythmic, synchronized, and periodic motion. A properly designed central pattern generator generates angular reference trajectories for the different links.

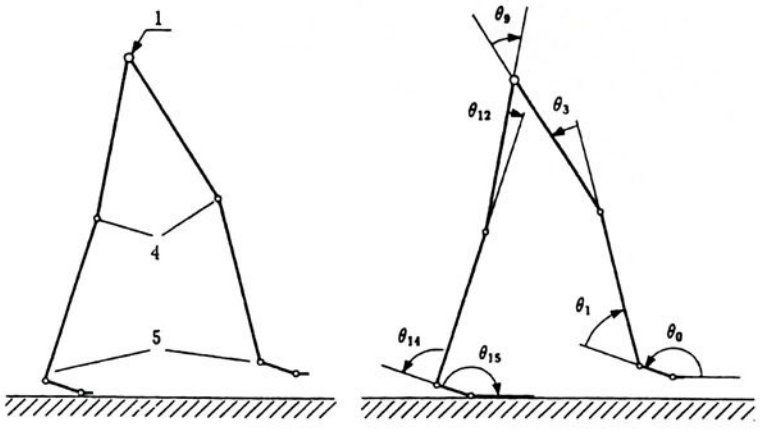


Figure 1 - Two dimensional model with the three most important determinants of gait, the compass gait (1), knee flexion of the stance leg (4) and plantar flexion of the stance ankle (5).

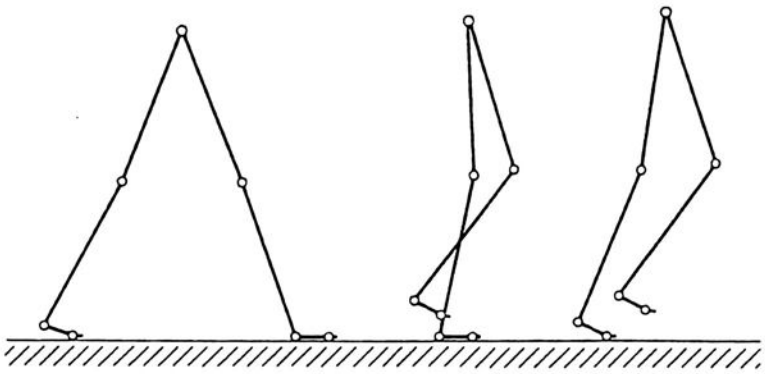


Figure 2 - Intervals during a walking period.

There are many dynamical systems capable to realize these variations [7]. In this paper the central pattern generator using neural network generates the angular reference signals for the motion of a single link.

Experimental work and Fourier Analysis

Experimental studies of human locomotion are well presented in [8] and the results can be seen in figure 3.

The Fourier analysis of these data show that motion of knee angles θ_3 and θ_{12}, as well as, the thigh angle θ_9 can be described very precisely by their fundamental harmonic whether the biped is in double or single support phase.

unesp

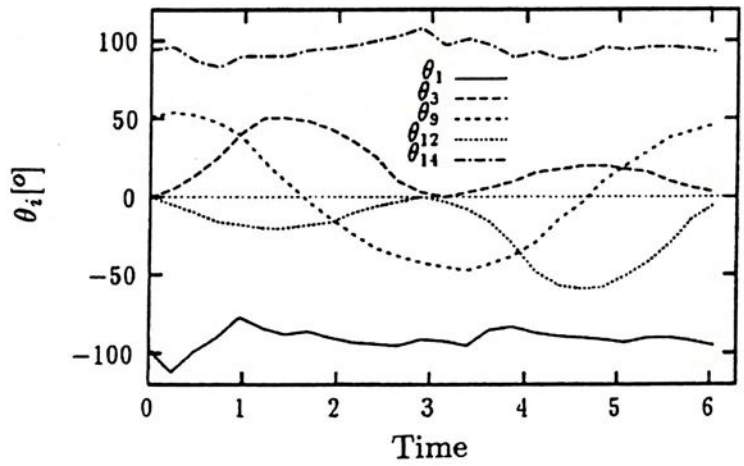


Figure 3 - Experimental results.

The harmonic functions have the following form:

$$\theta_3 = \theta_{30} + \hat{\theta}_3 \cdot \cos 2wt$$

$$\theta_9 = \theta_{90} + \hat{\theta}_9 \cdot \cos(wt - \alpha_9)$$

$$\theta_{12} = \theta_{120} + \hat{\theta}_{12} \cdot \cos 2wt$$

In order to generate θ_3, θ_9 and θ_{12}, a set of three coupled Van-der-Pol oscillators is used. These oscillators are arranged in a closed chain and are coupled by the velocity terms. They have the form:

$$\ddot{\theta}_3 - \varepsilon_3 \cdot \left[1 - p_3(\theta_3 - \theta_{30})^2\right] \cdot \dot{\theta}_3 + \Omega_3^2 \cdot (\theta_3 - \theta_{30}) - c_{3,9} \cdot (\dot{\theta}_9 \cdot \theta_9) - c_{3,12} \cdot (\dot{\theta}_3 - \dot{\theta}_{12}) = 0$$

$$\ddot{\theta}_9 - \varepsilon_9 \cdot \left[1 - p_9(\theta_9 - \theta_{90})^2\right] \cdot \dot{\theta}_9 + \Omega_9^2 \cdot (\theta_9 - \theta_{90}) = 0$$

$$\ddot{\theta}_{12} - \varepsilon_{12} \cdot \left[1 - p_{12}(\theta_{12} - \theta_{120})^2\right] \cdot \dot{\theta}_{12} + \Omega_{12}^2 \cdot (\theta_{12} - \theta_{120}) - c_{12,9} \cdot (\dot{\theta}_9 \cdot \theta_9) - c_{12,3} \cdot (\dot{\theta}_{12} - \dot{\theta}_3) = 0$$

The nonlinear term $c_{q,i}(\dot{\theta}_i . \theta_i)$ is responsible for the coupling between two oscillators θ_q and θ_i, and frequency of θ_q is twice of θ_i. The necessary parameters of these oscillators are: p_i and ω_i and are calculated in Dutra [7].

Results

The main objective of this work was to determine the trajectory of the hip angle using neural network, in order to minimize the time. With the use of neural network the generation of this angle in real time is granted. To verify this methodology several inputs as step lengths and trajectories were tested. The neural network can then determine the hip angle starting from the desired step length.

The net was utilized during the learning period with the following configuration: two vectors with 41 elements at the initial layer, corresponding to the angle trajectories ($\hat{\theta}_9$ = 54 and $\hat{\theta}_9$ = 30) in a time interval of 2π seconds and, the length of the step of the biped locomotor; two intermediary layers

with 45 elements in each layer and one element at the output layer corresponding to the angle θ_9. A nonlinear transfer function was utilized at the two intermediary layers and a linear function at the output layer. The error obtained in the training process was 9.9×10^{-5}. After the training process, it was verified the capacity of generalization of the net, choosing different values of the trajectory of angle θ_9 and the step. The results of these simulations are presented in the table below.

StepLength [m]	$\hat{\theta}_9$ theoretically [°]	$\hat{\theta}_9$ obtained by net [°]	Error [%]
0.38*	30	30.0	-
0.44	35	36.3	3.6
0.50	40	41.7	4.1
0.62	50	50.7	1.4
0.68*	54	54.0	-

* utilized values during the training process

Conclusion

This paper describes the possibility to determine the angles of the biped locomotor, through neural network as function of the trajectory and step to a specific range of trained values. The algorithm utilized in this paper was based on the backpropagation method.

It can be concluded from the results that error is appreciably small if the values of $\hat{\theta}_9$ are closer to the values utilized in the training process (54^0 and 30^0). The error is larger for values far from these. It may be adequate for trained values (intervals) so that the error in the output could be within the acceptable limits, or we can yet place an intermediary curve at the training process so that the analyzed interval is smaller.

Bibliography

[1] Pavlidis, T. Biological Oscillators. Academic Press, Inc., New York, 1973.

[2] Johnsson, A. Zur Biophysik biologischer Oszillatoren. In Biophisik, H. Hoppe; W. Lohmann, (Edit.), Springer Verlag, 1978.

[3] Frik, M., Ahanikamangar, M. Central Pattern Generators for Walking Machines, In CSME Mechanical Engineering Forum, Toronto, pp. 279-284, 1990.

[4] Dutra, M. S. Bewegungskoordination und Steuerung einer zweibeinigen Gehmaschine, Shaker Verlag, Aachen - Germany, 1995.

[5] Masri, S. F., Chassiakos, A. G., Caughey, T. K. Identification of Nonlinear Dynamic Systems Using Neural Networks, Journal of Applied Mechanics, Vol.60, pp. 123-133, 1993.

[6] Saunders, J. B., Inman, V., Eberhart, H. D. The Major Determinants in Normal and Pathological Gait. J. Bone Jt. Surgery 35 A, 1953.

[7] Dutra, M. S. Modeling of a Bipedal Locomotor Using Nonlinear Oscillators, International Journal of Intelligent Mechatronics (to appear).

[8] Braune, W., Fischer, O. The Human Gait. Springer Verlag, Berlin, New York, 1987. Translated from Der Gang des Menschen, B. G. Teubner (1895-1904).

[9] Berns, K., Piekenbrock, St., Dillmann, R. Learning Control of Walking Machines. IPR Universität Karlsruhe, Geb. 40.28, 76128 Karlsruhe, Germany, 1994.

unesp

IDENTIFICAÇÃO DE NÃO LINEARIDADES EM SISTEMAS DINÂMICOS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Vicente Lopes Júnior, Ádamo de Oliveira do Nascimento
UNESP - Ilha Solteira - Av. Brasil 56 - Ilha Solteira
CEP 15385-000 - São Paulo - Brazil
E-mail: vicente@dem.feis.unesp.br

Resumo

Vários sistemas dinâmicos apresentam características de não linearidades e não podem ser tratados satisfatoriamente usando a teoria de sistemas lineares. Redes neurais artificiais consistem em simples elementos computacionais, ou neurônios, distribuídos em camadas e operando em paralelo; com isto apresentam grande habilidade para aprender sofisticadas relações de não linearidade e tornar-se uma poderosa ferramenta para modelar estes tipos de sistemas. Este trabalho consiste em aplicar o método clássico de retropropagação para identificar e classificar as não linearidades que podem aparecer em um sistema mecânico usando redes neurais artificiais. Um procedimento baseado na utilização de redes neurais artificiais para a identificação de não linearidades em sistemas dinâmicos foi desenvolvido e aplicado para estudar o oscilador de Duffing submetido a diferentes excitações.

Palavras-Chave

Redes Neurais, Não Linearidades, Sistemas Dinâmicos, Backpropagation

1. INTRODUÇÃO

Devido ao aumento na velocidade das máquinas e exigência de eficiência, a análise de sistemas dinâmicos torna-se cada vez mais importante para a obtenção de melhores resultados. Muitos sistemas dinâmicos reais apresentam características de não-linearidade e não podem ser tratados satisfatoriamente usando-se a teoria de sistemas lineares. Redes Neurais, as quais apresentam grande habilidade para aprender sofisticadas relações de não-linearidade, podem fornecer características adequadas para o monitoramento de sistemas não-lineares complicados.

Vários estudos tem sido realizados na área de identificação de sistemas mecânicos e há uma tendência de se introduzir inteligência artificial nos projetos de controle de sistemas e nos sistemas de monitoramento de sinais os quais permitem uma automatização no processo de monitoramento e diagnose de sistemas complexos (Marsi, Chassiakos & Caughey, 1993).

Verificamos neste contexto a aplicação de redes neurais para a classificação de defeitos nas máquinas (Tadeusz, 1994). Esta técnica consiste na análise dos sinais de vibração vindos dos componentes móveis das máquinas para detectar caracteres que refletem o seu estado operacional. Vários trabalhos têm sido desenvolvidos na área de identificação e controle de sistemas dinâmicos usando Redes Neurais Artificiais (Chen, Billings & Grant, 1990; Narenda & Parthasarathy, 1990) para sistemas que apresentam características de não-linearidade.

Redes Neurais são concepções em hardware/software baseadas no modelo de como o cérebro codifica e processa informações. Sua aplicação é destinada a soluções de problemas difíceis de serem obtidas por meio de programas convencionais. Estas redes são dotadas de aprendizado, reconhecimento de padrões e memória associativa, dentre outras funções especializadas. Uma Rede Neural é composta por elementos de processamento (neurônios) dispostos em camadas e interligados entre si. Cada neurônio pode conter várias entradas mas, somente uma saída e cada saída pode ser utilizada como entrada em vários neurônios

unesp

seguintes através de suas interligações. Os modelos de neurônios artificiais foram concebidos baseados no funcionamento dos neurônios biológicos (Hecht-Nielsen, 1990).

O neurônio biológico é constituído basicamente por: Corpo Celular: corresponde a parte central do neurônio, responsável pela geração e recepção dos impulsos nervosos; Dendrito: tem a função de receber informações ou impulsos nervosos de outros neurônios e conduzí-los ao corpo celular; Axônio: um axônio possui, basicamente, a mesma função do dendrito, porém é longo. Um axônio pode medir de 0,1 milímetro a 1 metro. Próximo de sua extremidade, o axônio divide-se em várias ramificações menores (dendritos) que se interligam com os demais neurônios através das sinapses; Sinapse: é o ponto de conexão entre a terminação axônica de um neurônio e o dendrito de outro. Funcionam como válvulas, sendo capazes de controlar a transmissão de impulsos, ou seja, o fluxo de informação entre os neurônios que compõem a rede. Esta capacidade é conhecida como eficiência sináptica (Storer, 1979).

2. MODELO DE UMA REDE NEURAL

As redes neurais são apresentadas de forma a terem três tipos de camadas de elementos (neurônios) que são classificados como: camada de entrada, camadas intermediárias e camada de saída. Cada neurônio em uma camada é ligado a todos os neurônios da camada seguinte e estas ligações são ponderadas por pesos aplicados aos valores que passam de um neurônio para o próximo. Portanto, os valores da camada de entrada passam para as camadas intermediárias e estes para a camada de saída, que fornecem a saída da rede.

Dentre vários trabalhos desenvolvidos na área de identificação e controle de sistemas dinâmicos usando redes neurais artificiais, duas classes têm sido utilizadas com maior freqüência: Redes Neurais Multicamadas, as quais tem demonstrado grande sucesso nos problemas de reconhecimento de padrões e Redes Recorrentes que tem sido usadas para solução de problemas de otimização. Neste trabalho será usada a primeira classe.

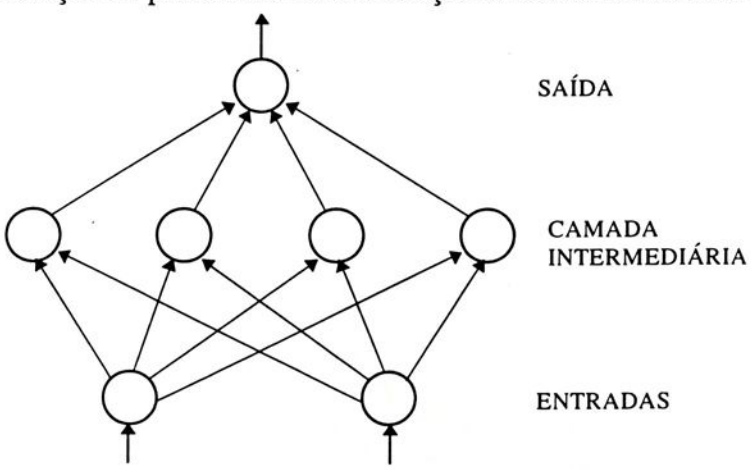


Figura 1: *Esquema Ilustrativo de uma Rede Neural.*

As Redes Neurais Multicamadas são largamente usadas por causa de suas propriedades atrativas, tais como implementação de decisões arbitrárias complexas, utilizando dados contínuos ou binários, não fazendo nenhuma suposição com relação a forma das distribuições sobre camadas e assim por diante (Lippmann,1987; Worden & Tomlinson, 1992). A figura 1 mostra um modelo ilustrativo de uma rede neural.

O uso de redes neurais neste contexto tem explorado suas habilidades de generalização tão bem como sua extraordinária tolerância de falha. Isto lhe confere a capacidade de classificar com sucesso modelos que não foram utilizados durante o processo de treinamento, assim como reconhecer modelos contaminados por ruído.

3. TREINAMENTO DA REDE

O treinamento de uma Rede Neural é a etapa mais importante do processo e consiste no ajustamento de pesos e do erro sistemático entre as interligações de camadas. Existem dois tipos de treinamento: o supervisionado e o não-supervisionado. Neste trabalho será usado o primeiro tipo de treinamento.

O treinamento supervisionado consiste no ajuste de pesos de uma rede neural para fornecer saídas desejadas, considerando-se o conjunto de padrões de entrada. Este treinamento necessita de um par de vetores, formado por uma entrada e um vetor alvo representando a saída desejada. Juntos eles são chamados de par de treinamento.

O processo de aprendizado é realizado de acordo com um ajustamento de pesos de ponderação entre as camadas. As saídas fornecidas pela rede são sucessivamente comparadas com as desejadas e a cada passo os pesos são ajustados para os valores de saída desejados. Este processo trabalha repetidamente até que se obtenha um erro menor que o estabelecido no início do treinamento. Um dos algoritmos mais difundidos para treinamento deste tipo de rede é o de retropropagação *(backpropagation)*.

O método de retropropagação foi criado para generalizar o controle Widrow-Hoff (Widrow & Hoff, 1960), para redes neurais de múltiplas camadas e aplicados para funções de transferência não lineares. Vetores de entrada e os correspondentes vetores de saída são usados para treinar a rede até ela aproximar uma função, associando vetores de entrada com específicos vetores de saída, ou classificando vetores de entrada de um modo apropriado. As redes adequadamente treinadas com o algoritmo *backpropagation* tendem a dar razoáveis respostas quando são apresentadas entradas que não foram utilizadas durante o processo de treinamento. Tipicamente uma nova entrada é conduzida pela rede a uma saída similar a nova entrada que foi apresentada. Esta generalização de propriedades torna possível o treinamento de redes sem a utilização de todas as possíveis combinações entrada/saída.

O algoritmo de retropropagação pode conduzir para o mínimo local da função. O mínimo local pode ser satisfatório, mas se não for, uma rede com mais camadas e neurônios pode fazer uma melhor tarefa. De qualquer modo, o número de neurônios ou camadas a aumentar pode não ser óbvio. Alternativamente, pode-se pesquisar o problema usando diversos grupos com diferentes condições iniciais para ver se elas conduzem as mesmas ou diferentes soluções. O algoritmo de retropropagação possui a etapa de inicialização e a etapa de treinamento.

A matriz inicial de ponderação para o treinamento é criada com elementos randômicos entre -1 e 1. Estes possibilitam que os vetores de ponderação sejam mudados facilmente e permitem uma grande variação nos neurônios no início da rede. Cada elemento pode ser obtido usando-se uma matriz simétrica da função randômica. O número de neurônios de uma camada intermediária influencia o tempo de treinamento da rede. Quanto mais neurônios a rede possuir, mais lento será o seu treinamento.

O controle linear do algoritmo de retropropagação é usado para treinamento não linear de redes multicamadas, para aproximação de funções, associação de padrões e classificação de padrões. Também é usado para ajustar os pesos minimizando a soma dos erros quadrados. Isso é feito pela mudança contínua dos valores na direção do menor erro. A soma dos erros quadrados é então determinada. Se a soma dos erros quadrados de todos os vetores treinados for menor que o erro estipulado, termina-se o treinamento.

4. RESULTADOS

Foi utilizado neste trabalho o software comercial MATLAB (Demuth & Beale, 1994), como ferramenta para a elaboração dos programas de determinação e identificação de características não lineares de sistemas dinâmicos, através de simulações computacionais.

unesp

O sistema não-linear de um grau de liberdade estudado (Marsi, Chassiakos & Caughey, 1993) é mostrado na figura 2 (Oscilador de Duffing). Este oscilador foi utilizado para se verificar o comportamento de forças internas, sem conhecimento da natureza da não linearidade do sistema.

Neste sistema desejamos determinar a força restauradora ($g(y, \dot{y})$) que está em função do deslocamento y(t) e da velocidade $\dot{y}(t)$ do sistema, para valores de excitação, f(t), que não foram utilizados durante o processo de treinamento.

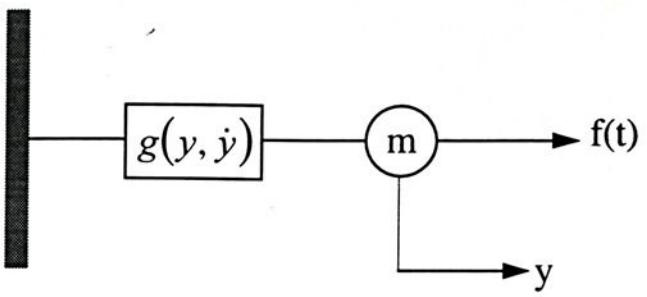


Figura 2: *Modelo do sistema não-linear.*

A excitação f(t) e a aceleração $\ddot{y}(t)$ do sistema são obtidas geralmente através de medidas experimentais, no entanto, para este caso simulado a aceleração, a velocidade e o deslocamento são calculados para cada tipo de excitação através do método de Runge-Kutta de 4ª ordem e serão consideradas respostas exatas. A massa m é conhecida e será considerada igual a 1 (m=1).

O sistema analisado corresponde ao Oscilador de Duffing com características de não-linearidades. A força restauradora e os parâmetros utilizados são dados por:

$$g(y, \dot{y}) = ay + by^3 + c\dot{y} \tag{1}$$

com: $a = 2\pi,\ b = 10,\ c = 1{,}25$.

A equação de movimento do sistema é dado por:

$$m\ddot{y}(t) + g(y, \dot{y}) = f(t) \tag{2}$$

e a força restauradora no domínio do tempo é dada por:

$$g(y, \dot{y}) = f(t) - m\ddot{y}(t) . \tag{3}$$

Se as variáveis forem obtidas para um tempo discreto t_k, o valor $g_k = (y(t_k), \dot{y}(t_k))$ é calculado por:

$$g_k = f_k - m\ddot{y}_k . \tag{4}$$

Neste exemplo, a excitação utilizada no treinamento da rede é um sinal senoidal com amplitude $f_0 = (2\pi)^2$ e freqüências iguais a 0,9 e 1,1 da freqüência natural do sistema, w_n, dada por $\sqrt{k / m}$. Os parâmetros do sistema resultam em uma pequena amplitude de período $T \approx 1$ e $\xi = 0{,}10$. O intervalo de tempo usado no experimento foi de 0 a 20 segundos (t = 0: 0,1: 20).

Aplicamos o treinamento supervisionado na rede neural, onde utilizamos como entrada os valores da força, f(t), e da aceleração, $\ddot{y}(t)$. A saída foi adotada como sendo os valores do deslocamento, y(t), e da velocidade, $\dot{y}(t)$.

Para a avaliação destes resultados foi adotada, primeiramente, uma rede simples com uma camada intermediária de cinco neurônios e função de transferência sigmoidal, enquanto para a camada de saída é utilizada uma função de transferência linear. A rede é inicializada com o método Nguyen-Widrow, isto é, valores aleatórios de pesos e erro sistemático (*"bias"*).

Após o treinamento, sendo a primeira freqüência utilizada $0{,}9 \cdot w_n$ e a segunda de $1{,}1 \cdot w_n$, os valores dos vetores de ponderação e erros, W e B, foram armazenados para testar a capacidade de generalização da rede.

unesp

Com a rede treinada pode-se calcular a saída para entradas que não foram utilizadas durante o processo de treinamento. O erro calculado pela rede é obtido pela equação: $E = \sqrt{A2 - T}$, onde A2 é a saída calculada pela rede, e T é a saída desejada.

Na fase de treinamento o número de iterações foi baixo, 52 iterações, com erro de $7{,}54697 \cdot 10^{-6}$ (menor que o erro estipulado). Como o treinamento da rede foi satisfatório, obtém-se a força restauradora $g(y, \dot{y})$ utilizando os valores de y(t) e $\dot{y}(t)$ na equação (1). As curvas mostradas na figura 3 exibem as características associadas com a não-linearidade do sistema analisado para o valor de w/w_n=0,9. Verificamos graficamente que os valores de saída calculados pela rede são iguais aos valores teóricos obtidos pelo método analítico de Runge-Kutta de 4ª Ordem. Os resultados a seguir foram divididos em três procedimentos.

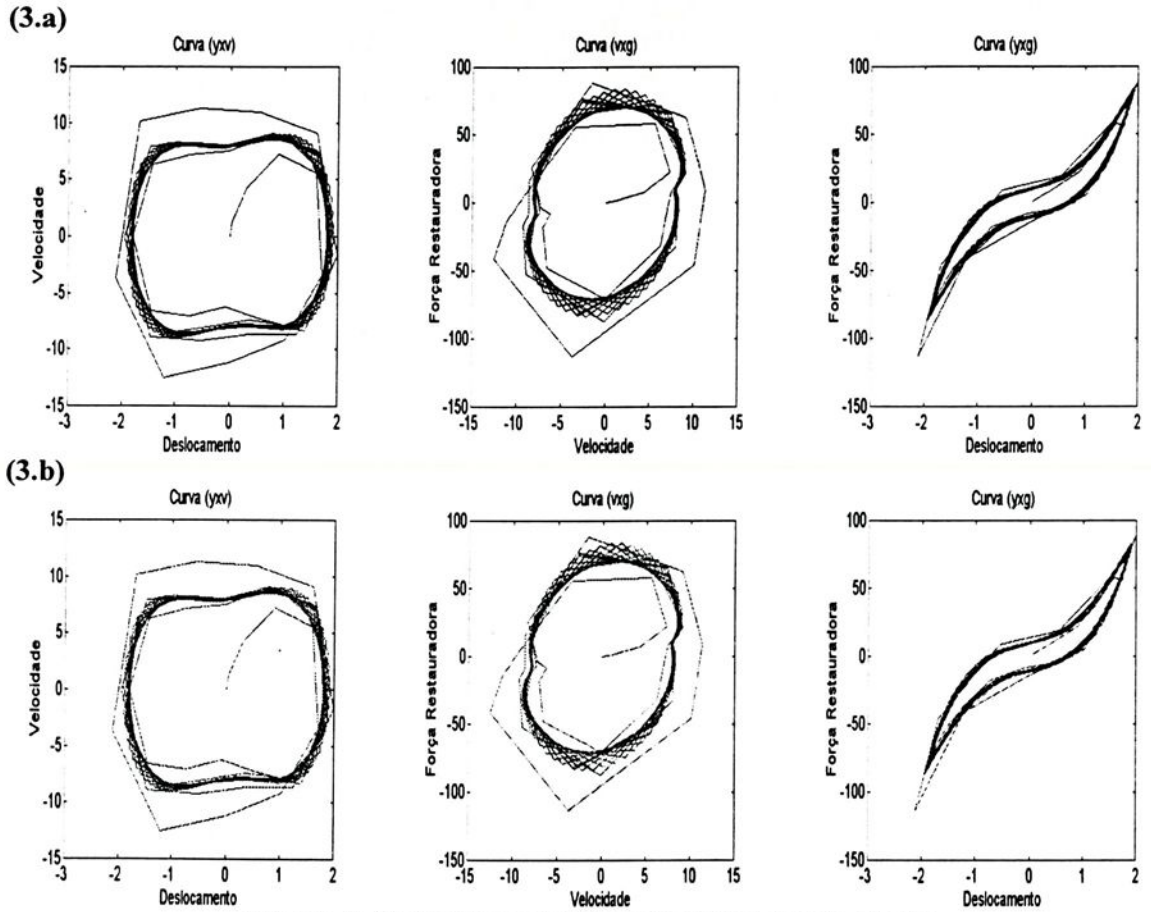


Figura 3: **(3.a)** Valores teóricos. **(3.b)** Valores simulados.

Teste 1: Utilização de Diferentes Razões de Freqüências w/w_n

Após o treinamento, tendo em mãos os valores da função de ponderação e erros (W e B), utilizou-se diferentes razões de w/w_n (0,5; 0,7; 0,8; 1,0; 1,2 e 1,5) para valores de entrada e, a rede calculou as saídas. Isto foi feito para comprovar a capacidade de generalização da rede, e os resultados são mostrados na figura 4. Podemos ver que a força calculada pela rede difere da exata para valores fora do domínio treinado. No entanto, a característica de não linearidade, formato da curva, se mantém. Já a figura 5 mostra o registro no tempo dos deslocamentos, velocidades e forças restauradoras para razões w/w_n=0,7 e 0,9 comparando os valores calculados pela rede neural com os valores obtidos pelo método analítico de Runge-Kutta.

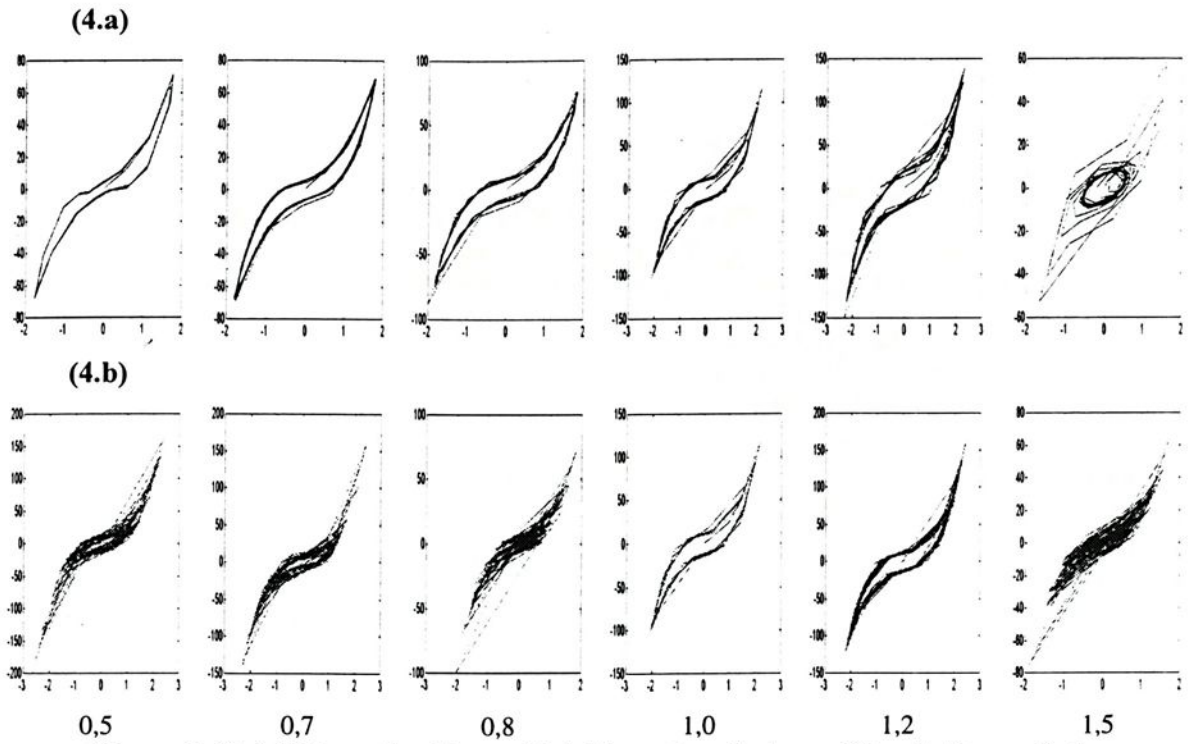


Figura 4: **(4.a)** Valores de y(t) vs. g(t)obtidos pelo método analítico de Runge-Kutta. **(4.b)** Valores de y(t) vs. g(t)calculados com a rede neural.

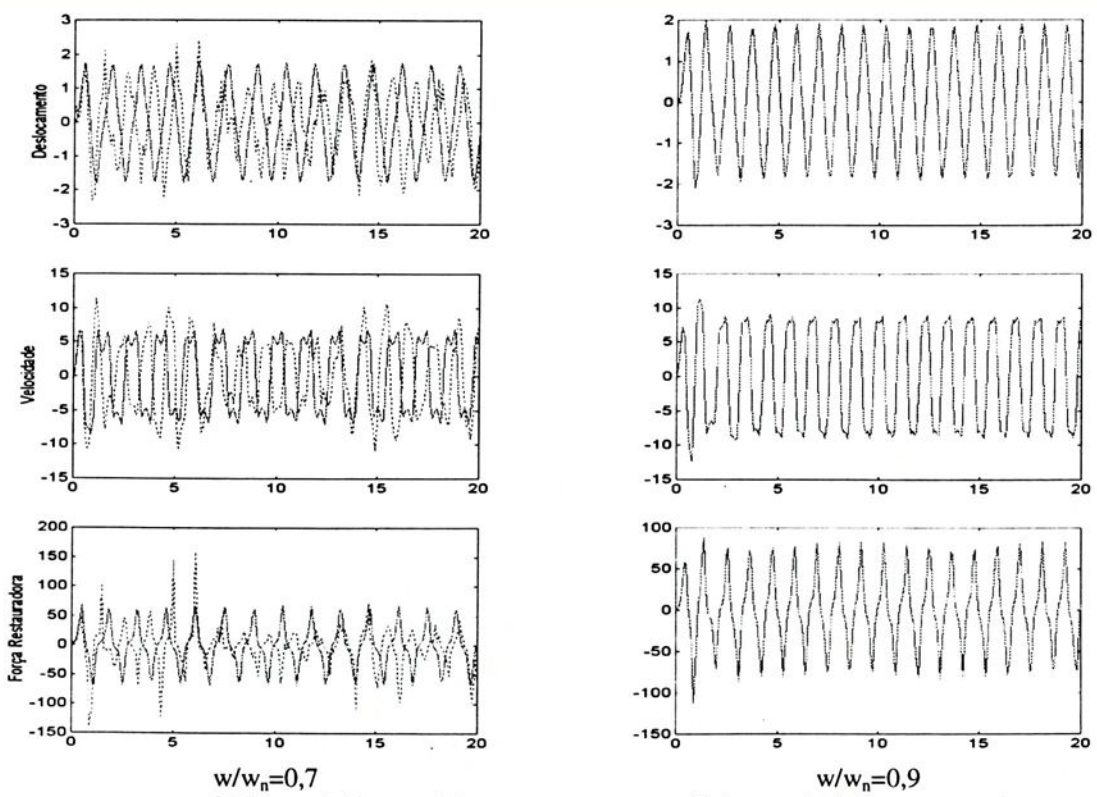


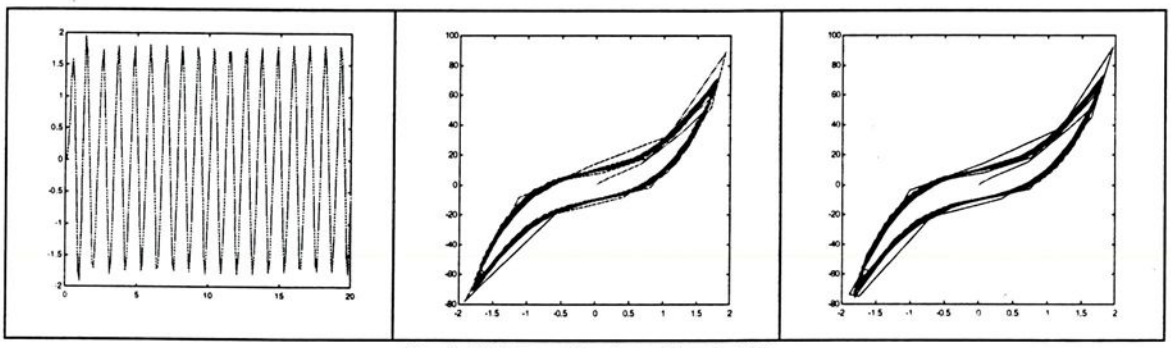
Figura 5: Registro de grandezas obtidas analiticamente e calculadas com a rede neural.

Como podemos observar pela figura 5, os valores de deslocamento e velocidade calculados com a rede não diferem acentuadamente dos valores exatos quando a razão de

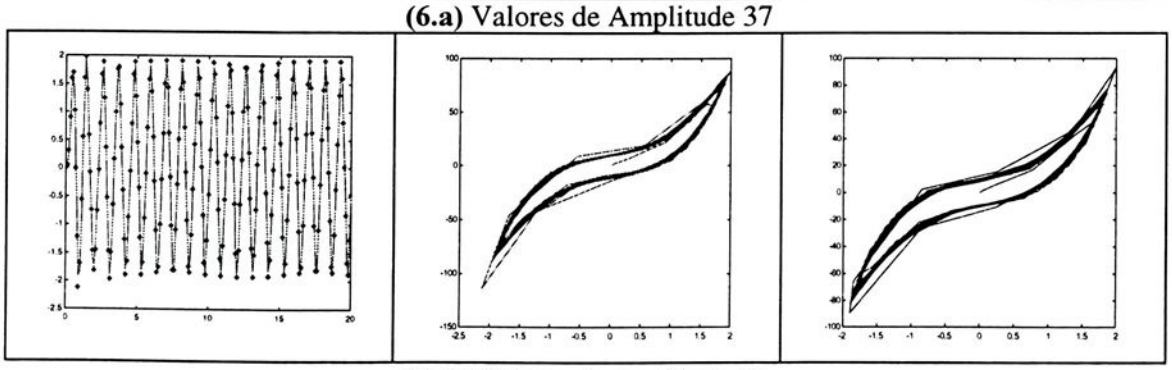
freqüência está próxima a treinada. No entanto, esta diferença cresce bastante para a força, principalmente em alguns pontos, pois como podemos observar pela equação (1), a força é função da terceira potência do deslocamento. Em função destes resultados, foram realizados novos testes com outra configuração de rede. Para esta nova rede foram utilizadas 2 camadas intermediárias com 20 neurônios cada. A função de transferência utilizada para a 1ª e 2ª camadas intermediárias foi a função não linear *sigmoidal*, e para a camada de saída utilizou-se a função linear (*purelin/Matlab*). Verificamos após estes testes que os resultados obtidos foram praticamente iguais aos calculados anteriormente pela rede de 1 camada intermediária.

Teste 2: Utilização de Diferentes Amplitudes

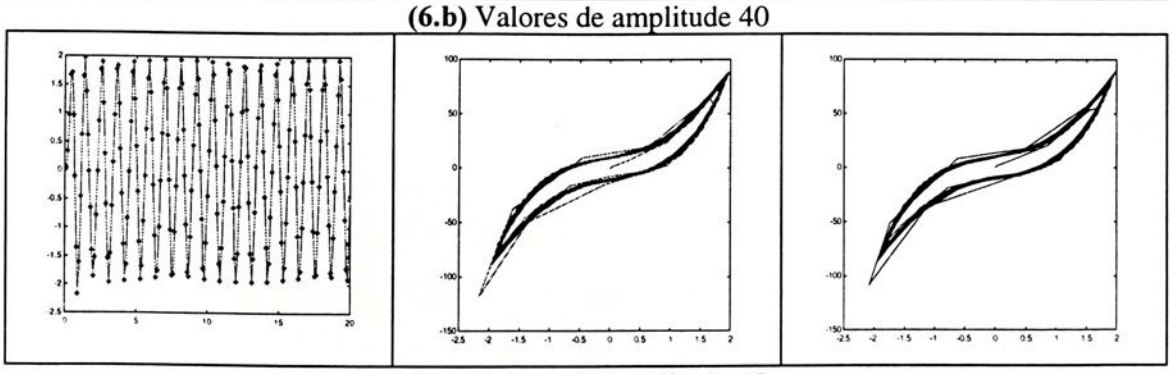
Nesta etapa do trabalho, procurou-se comprovar a capacidade de generalização da rede usando desta vez a amplitude como parâmetro variável. Treinou-se uma rede com valores de amplitudes de 35, 47 e 50, mantendo fixa a razão de freqüência w/wn=0,9. Os resultados dos testes, amplitudes de 37, 40 e 45, são mostrados na figura 6.



(6.a) Valores de Amplitude 37



(6.b) Valores de amplitude 40



(6.c) Valores de amplitude 45

Figura 6: Gráficos (deslocamento vs. tempo), (deslocamento vs. força interna)$_{teórico}$ e (deslocamento vs. força interna)$_{simulado}$

Teste 3: Variação Simultânea da Razão de Freqüências e das Amplitudes

O objetivo desta etapa foi encontrar experimentalmente uma configuração de rede que melhor atendesse às necessidades do problema em questão. Para isso, testou-se várias configurações de rede e a que apresentou melhores resultados foi a de 2 camadas intermediárias. Para ilustrar esta etapa, mostramos na figura 7 os testes feitos para camadas intermediárias com 45 neurônios e w/w_n=1,07.

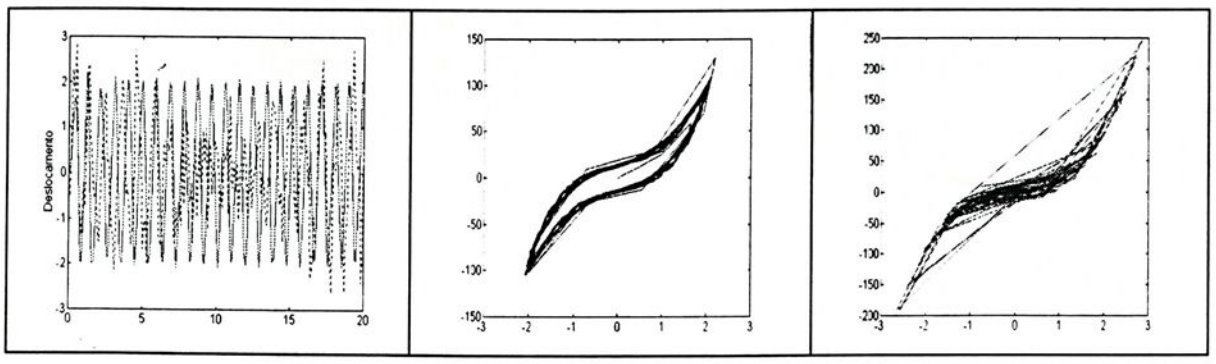


Figura 7: Rede de 2 camadas intermediárias com 45 neurônios cada camada.

5. CONCLUSÃO

O objetivo deste trabalho é a utilização de Redes Neurais Artificiais para a identificação de não linearidades em sistemas dinâmicos utilizando o algoritmo de retropropagação. Este algoritmo é usado em pelo menos de 80 a 90% das aplicações práticas envolvendo redes neurais artificiais. No entanto, existem técnicas que podem ser usadas para torná-lo mais confiável e eficiente.

Observando os gráficos da figura 4, verificou-se que embora os valores sejam diferentes, a forma das curvas são iguais para as razões de freqüências não treinadas. Isto comprova que as redes neurais podem caracterizar problemas gerais a partir de um problema específico conhecido. Com relação a 2ª parte do trabalho, na qual testamos a capacidade de generalização da rede para diferentes amplitudes, verificou-se que a rede foi bastante precisa em seus resultados, apresentando um melhor desempenho do que nos testes anteriores.

Os testes realizados na 3ª etapa, ou seja, variação da razão de freqüências e amplitudes simultaneamente, mostraram que a rede não tem boa capacidade de generalização neste caso específico, no entanto, a forma da curva se manteve (figura 7).

Embora a rede tenha apresentado resultados satisfatórios, a identificação da força interna, para o oscilador utilizado, através da rede treinada deve ser analisada com cuidado para valores distantes daqueles usados durante o processo de treinamento.

6. REFERÊNCIAS

Chen, S.; Billings, S.A. and Grant, P.M.; *Non-linear Sistem Identification Using Neural Networks*, Int. J. Control, vol.51 / pp. 1191-1214, 1990.

Demuth, H. and Beale M.; *Neural Network Toolbox for use with MATLAB*, The Mathworks, 1994.

Hecht-Nielsen, R.; *Neurocomputing*, HNC, Inc. and University of California, San Diego; Addison-Wesley Publishing Company, 1990.

Lippmann, R.P.; *An Introduction to Computing with Neural Nets*, IEEE Acoustics, Speech and Signal Processing Magazine, vol. 4 / pp. 4-22, 1987.

Marsi, S.F.; Chassiakos, A.G. and Caughey, T.T.; *Identification of Non-linear Dynamic Sistems Using Neural Networks*, Journal of Applied Mechanics, vol. 60 / pp. 123-133, 1993.

Narenda, K.S. and Parthasarathy, K.; *Identification and Control of Dynamical Sistems Using Neural Networks*, IEEE Trans. on Neural Networks, vol.1 / pp. 4-27, 1990.

Storer, T.I.; *General Zoology*, Sixth Edition, McGraw-Hill Book Company, pp. 58, 108, 134, 136, New York, 1979.

Tadeusz, U.H.L.; *Automation of Monitoring and Diagnostic Process for Rotating Machinery Using Neural Networks*, ISMA19 - Tools for Noise and Vibration Analysis, pp. 1151-1162, 1994.

Widrow, B. & Hoff, M; *Adaptive Switching Circuits*, IRE WESCON Convention Record, pp. 96-104, New York: Institute of Radio Engineers, 1960.

Worden, K. and Tomlinson, G. R.; *Classifying Linear and Non-linear Sistems Using Neural Networks*, Procedings of the 17th International Seminer on Modal Analysis, Leuven, pp.903-922, 1992.

unesp

IDENTIFICAÇÃO DE RIGIDEZ BILINEAR EM SISTEMAS MECÂNICOS ATRAVÉS DE REDES NEURAIS

Vicente Lopes Júnior, Daniela Demarchi
UNESP - Ilha Solteira - Av. Brasil 56 - Ilha Solteira
CEP 15385-000 - São Paulo - Brazil
E-mail: vicente@dem.feis.unesp.br

Resumo

Neste trabalho é apresentado um método para a identificação de rigidez bilinear. Este procedimento consiste em aplicar o método da retropropagação, "backpropagation", para treinar a rede e em seguida classificar e identificar sistemas com diferentes razão de rigidez. A razão pela qual este tipo especial de sistemas não lineares é estudado é que modelos rígidos bilineares são usados para descrever as características dinâmica de algumas estruturas com falhas. Simulações no computador e alguns testes experimentais mostraram que redes treinadas podem identificar sistemas bilineares com diferentes razões de rigidez (dureza) e conseqüentemente diferentes razões de freqüência em relação aos sistemas bilineares usados no estágio de treinamento. Esses resultados são significantes, pois fornecem a possibilidade do uso de redes neurais para detectar e, em um próximo estágio, quantificar o defeito através da identificação da rigidez bilinear.

Palavras Chave

Backpropagation, identificação de falhas, rigidez bilinear, redes neurais.

1. INTRODUÇÃO

Depois de duas décadas desaparecidos, o interesse em redes neurais tem crescido rapidamente em poucos anos. De modo que diferentes profissionais do campo da engenharia, psicologia e fisiologia estão intrigados com o potencial oferecido por esta tecnologia e estão procurando aplicações dentro de suas disciplinas. Esse ressurgimento de interesse ocorre em ambos, teoria e prática, sucessivamente. Repentinamente, isso possibilitou o aparecimento de aplicação computacional para um domínio prévio e restrito da inteligência humana e tem sido aplicado na fabricação de máquinas, onde o aprendizado tem atingido uma semelhança com o processo mental humano e, também a fim de dar novo e significante significado para o termo inteligência artificial. Processamento de informação visual é um dos processos mais complexos do cérebro e, portanto, a organização do caminho ótico tem influenciado modelos da cadeia neural.

Para desenvolver uma rede neural artificial um modelo matemático que melhor descreve o funcionamento do sistema biológico é desenvolvido. Um computador pode então simular o modelo rapidamente e um grau de confiança pode ser ganho, com atenção para sua operação e funcionamento. Mudanças podem então ser feitas para o modelo realçar sua realização ou para simplificar isto. Para criar um modelo que simula um sistema biológico, de qualquer modo, deve-se primeiro estudar e entender este sistema em todo o detalhe.

Redes neurais são normalmente utilizadas em problemas de reconhecimento de padrões e tem mostrado sua superioridade sobre os métodos tradicionais, em alguns casos. Alguns estudos tem sidos elaborados no campo de identificação de sistemas. Um dos possíveis usos do sistema de identificação não linear é a detecção e localização de falhas em sistemas mecânicos

As redes neurais multi-camadas com retroalimentação, são largamente usadas por causa de suas propriedades atrativas, tais como implementação arbitrária de decisão complexa, não fazendo nenhuma suposição com relação a forma das distribuições sobre camadas e assim por diante. Redes neurais artificiais (RNA) são inspiradas biologicamente, isto é, elas são

compostas de elementos que realizarão de maneira análoga a maior parte das funções elementares do neurônio biológico. Estes elementos são organizados de modo que possam ser relacionados com o cérebro e a rede neural exibe um número surpreendente de características cerebrais. Por exemplo, o aprendizado a partir de experiências generaliza prévios exemplos para uma nova entrada de dados.

Em função desta semelhança funcional alguns otimistas defendem a sugestão de que redes neurais logo duplicará as funções do cérebro humano. No entanto a atual inteligência exibida pela mais sofisticada rede neural é bastante limitada e o entusiasmo deve ser ajustado a realidade. Contudo não é correto ignorar a surpreendente performance de certas redes neurais artificiais. Estas habilidades, limitadas hoje, sugerem que um entendimento profundo da inteligência humana possa ser útil e, com isso avançar rumo à aplicações revolucionárias.

Redes neurais artificiais podem mudar seus procedimentos em resposta ao seu meio. Este fato, mais que muitos outros, é responsável pelo interesse que tem recebido. Apresentando um grupo de entradas (com saídas desejadas), elas se auto-ajustam para produzirem respostas consistentes. Uma enorme variedade de algoritmos de treinamento tem sido desenvolvido (Surace et. Al - 1997), cada um com vantagens e desvantagens.

1.1 Fundamentos de Redes Neurais Artificiais

As RNA são inspiradas biologicamente e pesquisas são usualmente desenvolvidas sobre a organização do cérebro onde se considera as configurações e algoritmos da rede (Wasserman – 1989). O conhecimento sobre a operação global do cérebro é limitado e ainda há pouco conhecimento da forma de suas funções. Conseqüentemente, o projetista da rede deve procurar, além do conhecimento biológico corrente, estruturas com funções usuais. Em muitos casos é necessário descartar a plausividade biológica e as redes são produzidas com suposições sobre a anatomia e funcionamento do cérebro.

Uma rede neural multi-camadas usualmente tem três a quatro camadas de elementos, chamados neurônios, que são classificados como camada de entrada, camadas intermediárias ou ocultas (uma ou mais) e camada de saída (Lucifredi and Hou – 1994). Como ilustrado na Figura 1:

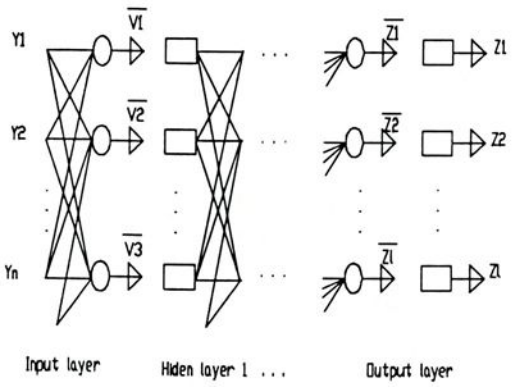


Figura 1- Esboço de uma rede neural multi-camadas.

Cada neurônio em uma camada é ligado a todos os neurônios da camada seguinte. Essas ligações são ponderadas e os valores passam de um neurônio para o próximo. Os dados de entrada são ponderados e passados para a primeira camada oculta e os neurônios da camada oculta são ativados e produzem saídas, as quais são baseadas na somatória dos pesos dos valores passados a elas e sobre a função de ativação.

Da mesma forma a primeira camada intermediária passa os valores à segunda camada (se ela existir) e assim sucessivamente. No final os valores são passados para a camada de saída que produzem as saídas da rede.

O primeiro estágio e também o mais importante estágio de uma rede neural é chamado "treinamento" ou "aprendizado". O treinamento faz uso de ajustes de entrada da rede para cada saída desejada. A rede aprende através de ajustagem dos pesos de interconexão entre as camadas. As saídas que a rede está produzindo são repetidamente comparadas com as saídas desejadas e a cada iteração os pesos de conexão são ajustados na direção das saídas desejadas. Um dos algoritmos mais utilizados no treinamento de redes neurais multi-camadas é a retro-propagação. Quando os erros de comparação são considerados pequenos o bastante, o treinamento termina.

Se uma rede é bem treinada, isto é, se ela tem uma boa capacidade de generalização, ela pode fazér o trabalho de classificação correto para novos dados de entrada que não foram utilizados no treinamento. Neste trabalho a rede é usada para classificar dois sistemas, um operando em uma faixa linear e o outro em outra faixa linear, portanto um neurônio na saída é suficiente. Para a camada de entrada são fornecidos os valores das freqüências e as amplitudes dos espectros medidos (ou simulados).

A determinação do tamanho da rede não é um trabalho fácil e geralmente é feito através de experiência e erros. Após o treinamento da rede estar concluído, a sua capacidade de generalização é testada, pois a generalização é a medida da performance da rede. Este teste é feito sobre a razão de rigidez, α, na operação da equação diferencial dos sistemas bilineares a qual é mudada e diferentes α's são adotados. Os novos dados são então submetidos às redes treinadas.

2. TREINAMENTO DA REDE

Uma rede é treinada para aplicações de grupos de entradas produzirem um grupo de saída desejada. Cada grupo de entrada (ou saída) é referenciado como um vetor. O treinamento é executado seqüencialmente aplicando os vetores de entrada, durante o ajuste dos pesos da rede de acordo com um procedimento pré determinado. Durante o treinamento, os pesos da rede gradualmente convergem para valores de modo que um vetor de entrada produza um vetor de saída desejável.

Os algoritmos de treinamento possuem duas categorias: supervisionado e não supervisionado. O treinamento supervisionado requer o pareamento de cada vetor de entrada com o vetor alvo representando a saída desejada. Juntos eles formam um par de treinamento. Usualmente, uma rede é treinada com um grande número de pares de treinamento.

O treinamento não supervisionado é o mais plausível modelo de entendimento do sistema biológico. Desenvolvido por Kohonen (Kohonen – 1984) e muitos outros, ele não requer um vetor alvo para as saídas e, conseqüentemente, sem comparações pré-determinadas. O algoritmo de treinamento modifica os pesos da rede para produzir vetores de treinamento ou aplicação de um vetor que, suficientemente similar, produzirá o mesmo modelo de saídas.

O algoritmo de treinamento da retropropagação foi criado para generalização de redes de múltiplas camadas. Os vetores de entrada e os correspondentes vetores de saída são usados para treinar a rede até ela poder aproximar uma função, associando vetores de entrada com os vetores específicos de saída, ou classificando vetores de entrada em um modo apropriado já definido. O treinamento da retropropagação é usado para ajustar os pesos e o erro sistemático, "bias", da rede com o objetivo de minimizar a soma dos erros quadráticos. Isto é feito pela mudança contínua dos valores dos pesos e "bias" da rede na direção do menor erro. Este treinamento pode conduzir a um mínimo local. O mínimo local encontrado pode ser satisfatório. Caso contrário, uma rede com mais camadas e neurônios poderá realizar um melhor processo de minimização.

unesp

Para o treinamento da rede, a soma dos erros quadrados é então calculada. Se a soma dos erros quadrados de todos os vetores for menor que o erro meta, o treinamento termina. Se o erro da rede for pequeno o bastante, novos vetores são apresentados à rede para verificação de sua eficácia e generalidade.

3. RESULTADOS

Inicialmente estudou-se a lógica do algoritmo backpropagation e em seguida ele foi utilizado como o algoritmo de treinamento de uma rede teórica para simular uma estrutura com falha. Após o bom resultado apresentado, por este algoritmo, escolheu-se uma estrutura que melhor representasse um sistema bilinear e nela analisamos a resposta mediante excitações de impacto e, monitoramos o comportamento para cada amplitude da falha. A estrutura utilizada foi uma viga engastada, inicialmente sem falhas. Essa falha foi feita através da remoção de material do sistema.

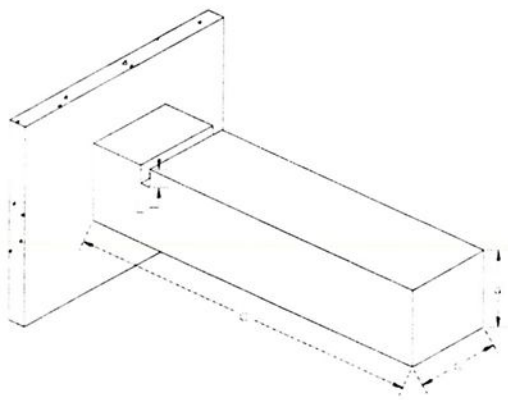


Figura 2 - Representação da estrutura utilizada na medição.

A parte experimental foi feita para vários casos sendo que para cada caso a falha se propagou de 10 em 10% até alcançar 50% da espessura da viga. Esta falha localizava-se a 10cm da extremidade engastada. Um desenho esquemático é mostrado na Figura 2 onde "L" varia de 0,1a até 0,5a.

Em cada caso foram aplicados à viga vinte impactos. Os impactos foram aplicados para duas posições; no meio e na extremidade livre da viga e, procurou-se evitar torções na estrutura. Destes valores fez-se, para cada caso, uma média para o cálculo do espectro de freqüência, com o objetivo de se diminuir a influência de ruídos externos.

As medidas de freqüências foram feitas em duas posições distintas da viga, uma no meio e a outra na extremidade livre. Para se realizar as medições foi utilizado um sistema de aquisição de dados composto por um transdutor, condicionador, analisador e por um registrador. O sistema de aquisição utilizado foi o Daq Book e o programa de análise o Dasylab.

Com os resultados em mãos optou-se por analisar apenas as curvas obtidas das medidas feitas no meio da viga. A Figura 3 mostra as curvas freqüências vs amplitude, obtidas para a viga sem defeito (Curva 1) e viga com defeito máximo (Curva 6) juntamente com as máscaras utilizadas para o treinamento da rede. Estas máscaras consideram uma faixa constante de 3% para os valores das freqüências naturais.

unesp

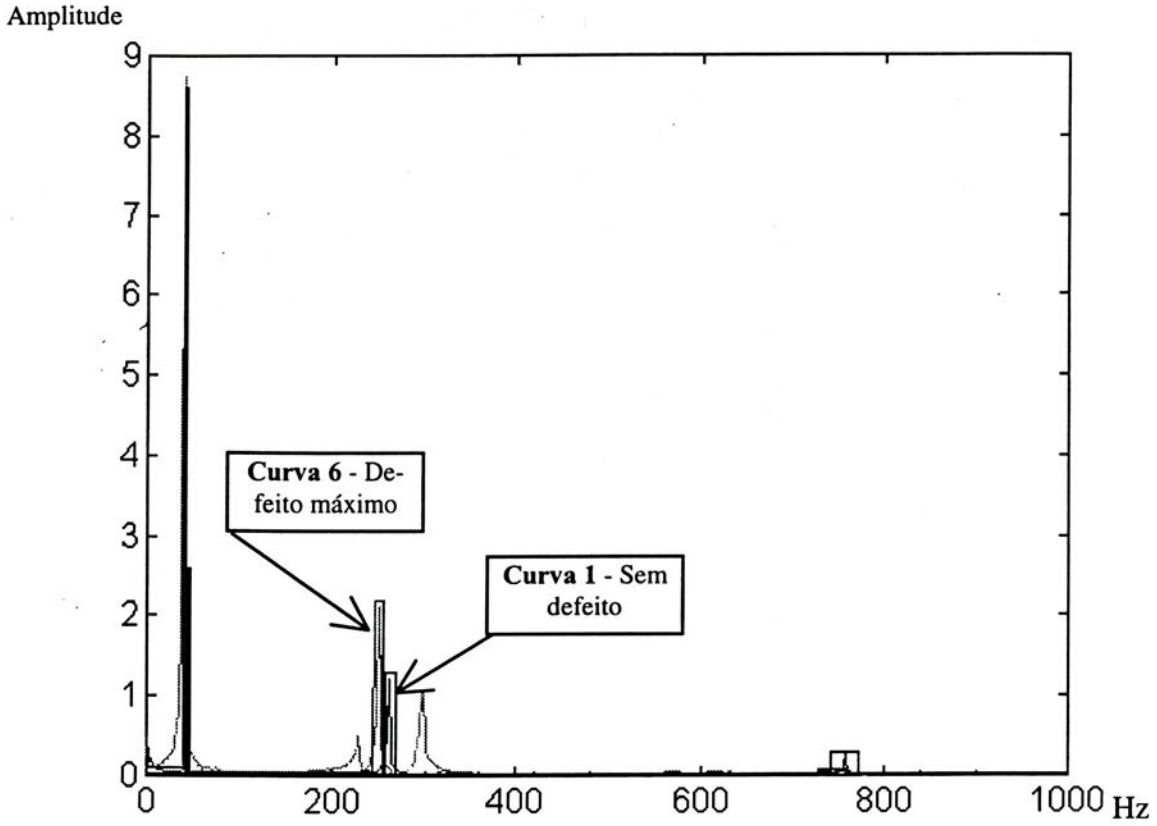


Figura 3 - Espectros de freqüência para a da viga sem defeito e com defeito máximo.

Para se escolher o padrão de entrada adequado foram realizados vários testes, pois este é um processo fundamental dentro desta metodologia. Para a entrada de dados foram apresentadas duas curvas, a primeira curva (sem defeito) tendo como saída 0 e a segunda curva (com defeito máximo) tendo como saída 1. Foram adotadas vários tipos de rede, para que se pudesse escolher a configuração mais adequada para este sistema.

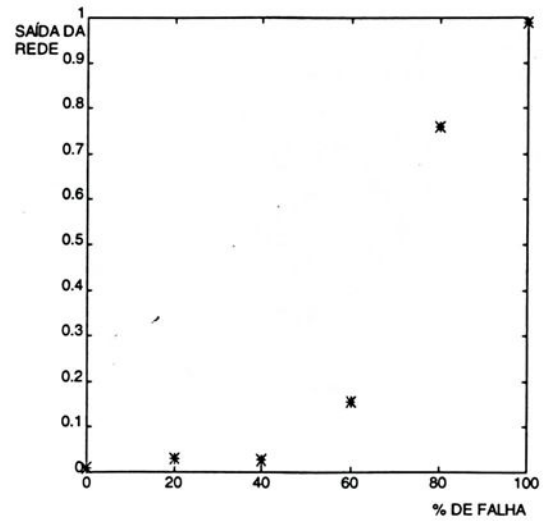
Primeiramente, usou-se como entrada uma faixa de freqüências e as respectivas amplitudes. Com esta entrada a rede não conseguiu convergir para o erro desejado. Em seguida usou-se como entrada somente os picos de freqüência e as suas amplitudes, desse modo obteve-se bons resultados somente quando a rede possuía mais de uma camada intermediária. Treinou-se então uma rede de duas e a outra de cinco camadas intermediárias. Para estas duas redes obteve-se os seguintes resultados:

1- Resultados para a entrada de dados com os picos de freqüência e amplitudes em vetores distintos, treinados em uma rede de duas camadas. Para 1950 iterações e um erro quadrático de 0.001 o treinamento da rede foi *adequado.*

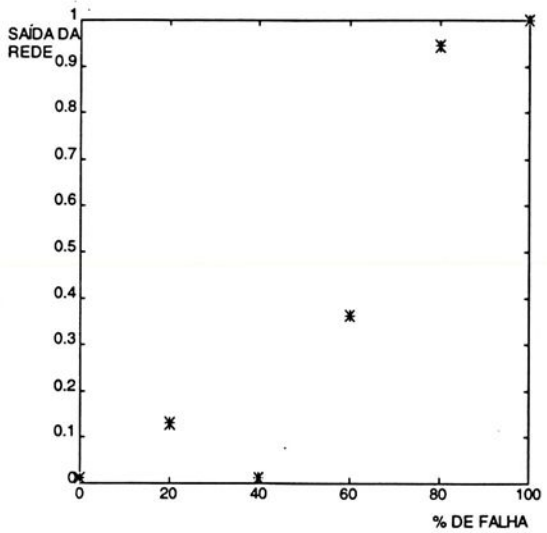
2- Resultados para a entrada de dados com os picos de freqüência e amplitudes em vetores distintos, treinados em uma rede de cinco camadas. Para 1340 iterações e um erro quadrático de 0.001 o treinamento da rede foi *adequado.*

Como as redes foram capazes de convergir torna-se possível a análise de dados intermediários. Foram analisadas quatro curvas intermediárias sendo que suas falhas são crescentes e iguais a (10, 20, 30 e 40% de a). Os resultados obtidos para a quantificação dos defeitos apresentados à rede foram:

unesp



(a)



(b)

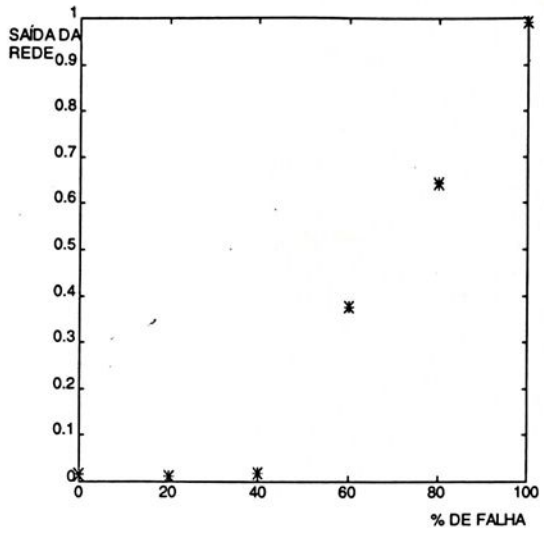
Figura 4- (a) e (b) Resposta da rede utilizando como entrada as freqüências naturais e as amplitudes.

1- Para a rede de duas camadas intermediárias os valores obtidos foram de 0.0078, 0.0308, 0.0265, 0.1546, 0.7583 e 0.9881 para as falhas introduzidas na estrutura.

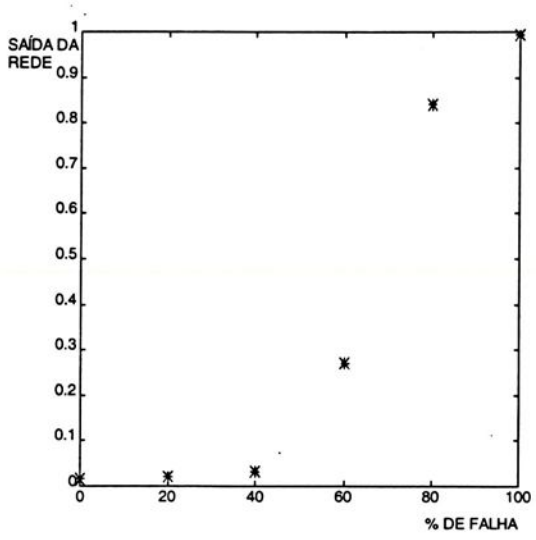
2- Para a rede de cinco camadas intermediárias os valores obtidos foram de 0.0129, 0.1302, 0.0118, 0.3610, 0.9446 e 0.9991 para as falhas introduzidas na estrutura.

Tentou-se uma outra forma de entrada de dados para treinar a rede. Isto foi feito utilizando-se como entrada uma máscara das curvas sem defeito (curva 1) e com defeito máximo (curva 2), figura 3. Para estes dados todas as redes, com uma, duas ou cinco camadas intermediárias convergiram para valores de erro adequados. Os resultados obtidos foram para a entrada de dados com as freqüências e amplitudes em vetores distintos. O erro quadrático, considerado para todos os casos, foi de 0.001 e a quantidade de iterações necessárias foram: para a rede de uma camada 2552 iterações, para a rede de duas camadas 840 iterações e para a rede de cinco camadas 2149 iterações.

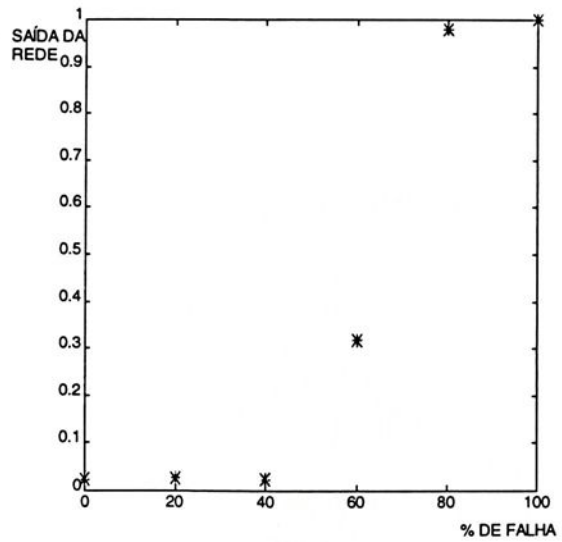
Como as redes foram capazes de treinar os dados então torna-se possível a análise de dados intermediários. Foram analisadas as mesmas curvas do exemplo anterior com falhas iguais a 10, 20, 30 e 40% de a. Os resultados obtidos para a quantificação dos defeitos apresentados à rede foram:



(a)



(b)



(c)

Figura 5- (a), (b) e (c) Resposta da rede utilizando como entrada de dados a máscara.

1- Para a rede de uma camada intermediária os valores obtidos foram de 0.0153, 0.0128, 0.0171, 0.3778, 0.6406 e 0.9912 para as falhas introduzidas na estrutura.

2- Para a rede de duas camadas intermediárias os valores obtidos foram de 0.0133, 0.0209, 0.0315, 0.2696, 0.8412 e 0.9928 para as falhas introduzidas na estrutura.

3- Para a rede de cinco camadas intermediárias os valores obtidos foram de 0.0221, 0.0257, 0.0225, 0.3179, 0.9803 e 0.9994 para as falhas introduzidas na estrutura.

4. CONCLUSÃO

Em posse dos dados obtidos das redes treinadas pode-se observar que uma rede com apenas uma camada intermediária não foi capaz de treinar os dados de entrada contendo somente os picos de freqüência e as suas amplitudes. Já as redes com duas ou cinco camadas intermediárias foram capazes de realizar o treinamento para a mesma entrada de dados. Porém a resposta às curvas intermediárias não quantificaram exatamente a porcentagem do defeito, tanto para a rede com duas ou cinco camadas, no entanto é possível identificar o aumento no valor de saída, quando o defeito é maior.

Em séguida, usou-se a máscara nas curvas como entrada de dados, curva sem defeito e com defeito máximo (figura 3). O treinamento convergiu para as redes com uma, duas e cinco camadas intermediárias. Terminada esta etapa, tentou-se obter respostas para as curvas intermediárias ao treinamento. A princípio esperava-se que a rede com cinco camadas intermediárias daria os melhores resultados, no entanto, em função do pequeno número de entradas (camada de entrada com 34 pontos) esta rede pode ter sofrido um sobreajuste e os resultados não foram melhores do que as redes anteriores. Neste caso a rede com apenas uma camada intermediária pode ser considerada satisfatória. Com estas respostas pode-se verificar que o aumento do número de camadas intermediárias em uma rede beneficia o treinamento até um determinado ponto, em seguida o treinamento pode ser prejudicado.

As falhas geradas na estrutura modificaram muito pouco as propriedades modais da viga, logo os valores intermediários das freqüências naturais ficaram bem próximos. Isso fez com que os resultados obtidos, para quase todos os casos apresentados, não quantificassem exatamente os defeitos, no entanto, pode-se comprovar o valor crescente da saída quando a porcentagem da falha aumenta.

5. BIBLIOGRAFIA

DASYLab™, Data Acquisition System Laboratory- Version 2.0.

Kohonen, T., Self-organization and associate memory,. *Series in Information Sciences*, vol.8, Berlin: Springer Verlag. 1984.

Lucifredi, A., & Hou, Z., *The construction of neural networks for bilinear systems identification a possible use to crack detection*, in: Proceedings ISMA19- Belgium Leuven, 1994 September, pp. 1139-1149.

Surace, C., Worden, K., & Tomlinson, G., *A novelty detection approach to diagnose damage in a cracked beam*, in 15th IMAC- Orlando, 1997.

Wasserman, P. D., Neural Computing: theory and practice, (ed.) *Van Nostrand Reinhold* 1989, pp. 1-58.

cm 1 2 3 4 5 6 7 8 unesp 12 13 14 15 16 17 18

Applications of Artificial Intelligence in Engineering XI

EDITORS:

R. A. Adey
Wessex Institute of Technology, UK

G. Rzevski
The Open University, UK

A. K. Sunol
University of South Florida, USA

Computational Mechanics Publications
Southampton Boston

Editorial	Contents	Index of Authors	Help	Related Publications

R.A. Adey
Wessex Institute of Technology
Ashurst Lodge
Ashurst
Southampton SO40 7AA
UK

A. K. Sunol
University of South Florida
College of Engineering
Tampa
Florida 33620-5350
USA

G. Rzevski
The Open University
Walton Hall
Milton Keynes
MK7 6AA
UK

Published by

Computational Mechanics Publications
Ashurst Lodge, Ashurst, Southampton SO40 7AA, UK
Tel: 44 (0)1703 293223; Fax: 44 (0)1703 292853
Email: cmp@cmp.co.uk
http://www.cmp.co.uk/

For USA, Canada and Mexico

Computational Mechanics Inc
25 Bridge Street, Billerica, MA 01821, USA
Tel: 508 667 5841; Fax: 508 667 7582
Email: cmina@ix.netcom.com

British Library Cataloguing-in-Publication Data

A Catalogue record for this book is available
from the British Library

ISBN: 1 85312 410 9 Computational Mechanics Publications, Southampton

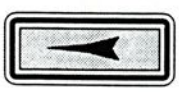
Library of Congress Catalog Card Number 96-083657

The texts of the various papers in this volume were set individually by the authors or under their supervision.

No responsibility is assumed by the Publisher for any injury and/or any damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

© Computational Mechanics Publications 1996

All rights reserved. Please see the **End User Licence Agreement** on the next pages.



unesp

Diagnosis of rotating systems using artificial neural networks

Vicente Lopes Jr., Antonio Eduardo Turra
Dep. de Engenharia Mecânica - Faculdade de Eng. de Ilha Solteira -UNESP
Cx. Postal. 31 - Cep. 15385-000 - Ilha Solteira- S.P.- Brasil
Email: UEISL@BRFAPESP.BITNET

Abstract

Vibration monitoring of components in manufacturing plants involves the collection of vibration data, and a detailed analysis of data. One of the most important characteristic of neural networks is their ability to model process and systems from actual data, and to respond in real time to the changes in the machine state. This paper discusses the use of neural network for fault classification in rotating systems using signal of vibration.

1 Introduction

The analysis of vibration monitoring of components detects and reflects the main operational state of the machinery. Vibration in mechanical systems may provide several information about the various equipment of the manufacturing plants. There are good references in the literature that describe the type of signal of vibration associated with typical faults of the systems and the techniques of analysis that can be used to detect the fault in an early state. Analysis of tendency, for example, is performed by comparing the vibration

BCpIS - FEIS - UNESP

spectrum of each machine with a reference spectrum, and evaluating the vibration magnitude changes of each spectrum at specified time.

The diagnostic methods can be divided into two groups: one that evaluates the operation state of the system and classifies it in satisfactory or unsatisfactory and those that detect the type of the fault. The first group is based on standards that describe the limits of vibration levels of a given class of machines. The second one does not present standards of comparison, since it depends on the use, i. e. each method is better adapted to specific characteristics of the machine. Those methods based on standard of comparison measure the vibration spectrum of the machine, which has a characteristic shape when it is operating properly, and compare with a standard level since the characteristic of signal are different when the machine is not operating satisfactorily.

These two classical pattern recognition techniques, in some cases, seem to be not powerful enough and the application of neural nets can improve this technique. In this case, the use of classical pattern makes apparent the need of robust pattern parameters and representative mathematical model designed in the context of vibration analysis, able to integrate and manipulate the vibration problem.

In classical pattern recognition techniques, a significant part of the analysis consists in the preprocessing and important characteristics can be extracted from the data which can be separated them in sets of data related with each kind of fault. This classification of data in different kinds of faults can be regarded as the main problem.

These forms of analysis for diagnosis are performed by maintenance personal monitoring and recording transducer signals and analyzing the signals aiming at identifying the operating condition of the machine, and incipient faults. Therefore, there is considerable motivation to design systems that automatically perform this type of analysis on real time and in a reproducible manner. The success of any monitoring program depends on the accuracy of

unesp

the measurements that affect, directly, the ability of the system to detect and identify faults. Accelerometers are a popular transducer for vibration analysis due to their accuracy, light weight, and wide range frequency response.

Neural networks have recently been created as an excellent and powerful tool for pattern recognition and can be applied for fault detection in machinery. Neural nets provide a viable technique for the analysis of vibration data because of their inherent ability to operate on noisy, incomplete or sparse data and to model process from actual system parameters, Uhl [1].

The presence of noise can complicate the monitoring task by altering the true signal, and by increasing the vibration values beyond of a specified level. When a machine is operating properly, vibration is small and constant. However, when faults develop and some of the dynamic process in the machine change, the vibration spectrum also changes. For many machines, the vibration frequency spectrum has a characteristic shape when the machine is operating properly, and it presents different characteristics when it operates with faults. Therefore, monitoring of the operating conditions of the machine can effectively be performed by a close examination of the features associated with particular faults and their identification.

We are interested in the identification of faults from frequency analysis domain, but application of this methodology is similar by any other kind of signal. The frequency spectrum, is useful because the low level signal are not masked out by the presence of high level signal, when using a log scale. Generally, each machine defect produces a set of vibration components that allows the recognition of different faults. There are many parameters that can be measured to evaluate the operation condition of a machine, for example pressure, temperature, sound, oil particles, etc.

Vibration signals can be processed in a variety of ways, and typical faults in rotating machine that can be identified are: misalignment, characterized by a peak at two times the running speed (2.n); unbalance,

unesp

vibration at n; gear problem, gear mesh frequency; mechanical looseness; bearing faults; eccentricity; oil whirl; and other.

2 Methodology

Neural networks may be designed to classify input patterns in pre-defined classes or to create categories of group patterns according to their similarity. One of the most important characteristic of neural networks is their ability to model process and systems from actual data, and to respond in real time to the changes in the system state, provided by continuous sensor inputs. For complex system, the fault identification in real time can be facilitated by applying this methodology.

An ANN, therefore, comprises a set of neurons interconnected and organized in layers. Some examples are discussed in literature [2], [3]. It is made up of a given number of layers, each of which has a specified number of neurons. The interconnections are only between neurons of adjacent layers, and each node (neuron) belonging to a layer is connected to all the neurons of adjacent layers. The neurons connected directly to the network inputs all belong to the same layer, called as the input layer, these neurons do not have offset. The neurons that furnish the network outputs also belong to a single layer, called the output layer. The other neurons are organized in one or more layers, called hidden layers, because they can not directly be reached from outside the network.

It should be noted that the number of neurons for the input and output layers depends on the specific application. The choice of the layer number and dimension is based in preliminary tests, and optimizing this choice is a significant topic in the studies of artificial intelligence researches, Bernieri [4].

The ANN is capable of learning by example. The network is forced to furnish the desired outputs on the basics of the inputs supplied to it. This is carried out through learning algorithms, which suitably modify both the

unesp

interconnection weight and offset values. Once learning is complete, the neural network is capable of furnishing the desired outputs for the known inputs.

The problem of identifying and classifying vibration signals can be divided in three stages: I. separation of the signals of components operating normally and signals of components with fault; II. compression of the signals; III. classification of the compressed patterns using backpropagation algorithm. Compression has an important issue in the context of this methodology, because it transforms a large data set in a reduced set with the same information, and decreases significantly the training time.

3 Neural Networks

Neural networks is made by a number of processing elements which are connected to form layers of neurons. Each neuron is a simple mathematical processing unit that when are interconnected, produces a complex structure and once trained, it can provide the ability to execute some tasks. The scheme of proposed diagnostic system is shown in figure 1.

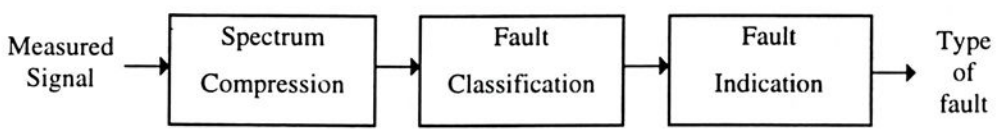


Figure 1: Scheme of proposed diagnostic system

The first step in the diagnostic procedure consists in spectrum size reduction, which contain 512 points in the original form. Two methods have been applied in this paper. The first method divides the range of frequency in intervals of five points. From each interval, the maximum amplitude is selected and this point is used to construct a new signature. The second method searches the maximum points in the original signal for each frequency range early specified, i.e., around of the natural frequencies and of the fault

characteristic frequencies. It is considered a range of 10% of the frequency for each maximum point.

Figure 2 shows the original (512 points) and compressed signal (masked), with 27 points.

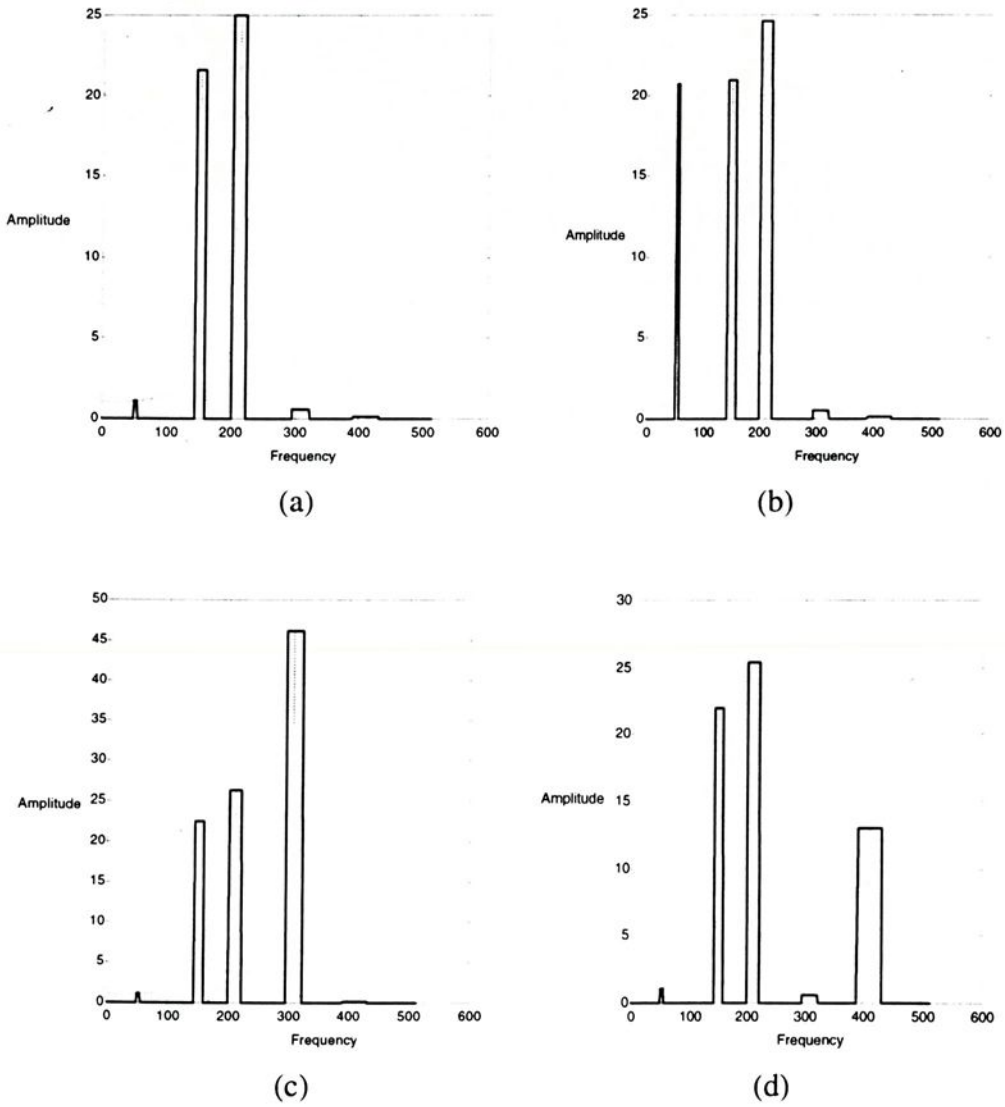


Figure 2: Original and compressed signal

Figure 2.a shows the signal of the machine without fault , and the two natural frequencies (152 and 222 Hz). Figure 2.b shows the signal with the natural frequencies and one fault, characterized by a peak in the frequency of

51 Hz (will be called **fault 1**). Figure 2.c shows the signal with the peak in 308 Hz (**fault 2**). Figure 2.d shows the signal with the peak in 411 Hz (**fault 3**).

It is common to specify an architecture by referring to the number of hidden layers; layers that are neither inputs nor outputs. The figure 3 shows the architecture used in this case. It has one hidden layers, with 5, 10, 20 or 50 neurons and output with two elements. The network is trained so that output is a combination of 0 and 1 depending on the machine condition.

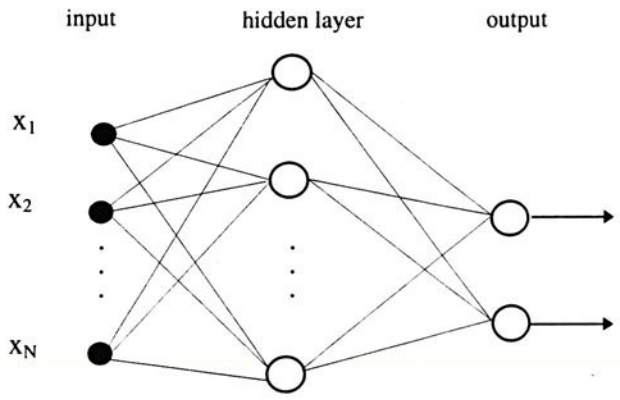


Figure 3: Architecture of the neural network with one hidden layer

The input with weight $w^h_{j,i}$ and bias, b^h_j , is added and the result is fed to the hidden layer. The input to the *j*th hidden unit is:

$$I_j = b^h_j + \sum_{i=1}^{N} W^h_{j,i} . x_i \qquad (1)$$

where the superscript (h) means that the quantities pertain to the hidden unit. Simplifying, the output of the net is a biased and weighted sum of the hidden layer outputs:

$$y_{out} = b^{o} + \sum_{j=1}^{H} W_{j}^{o}.HO_{j} \qquad (2)$$

the superscript (o) referring to the output unit, H is the units number in a hidden layer and HO_j is the output of the *j*th hidden unit. The net is trained starting with a random set of weights and biases and calculating the output for every input sets. Then the error, *E*, is evaluated and as it is computed from the output layer backwards, it has historically been called the backpropagated error, and the learning algorithm the backpropagation. This algorithm can be found in several publications, for example [5]-[7].

$$E = sumsqr(T - y_{OUT}) \qquad (3)$$

where sumsqr is the sum of squared elements and T is the desirable output. The error, *E*, is taken as a function of the weights and biases and then, the minimization is performed by means of the usual steepest algorithm. The error, E, is taken, for convenience, as a function of the $\{\alpha_i\}$, the index *i* taking on as many values as there are weights and biases. When the parameter, α_k, changes by $\delta\alpha_k$, the error, E, changes by

$$\delta E = \frac{\partial}{\partial \alpha_k}(E).\delta\alpha_k \qquad (4)$$

where $\frac{\delta}{\delta\alpha_k}(E)$ is evaluated using the current values of the parameters. To ensure that changes in α_k results in a decrease in E, we select

$$\delta\alpha_k = -\eta \frac{\partial}{\partial \alpha_i}(E) \qquad (5)$$

where η is a positive constant. The parameter η determine how large a step is made in the direction of steepest descent and therefore how quickly the optimum parameters are obtained. For this reason η is called the learning coefficient.

unesp

The training net use a set of input for which a specified output are known. Table 1 shows the results obtained from compressed data. SSE( ) is the Sum Squared Error between the value output that the network was trained and the value that was obtained with a specify input. Third column shows the results when the input was a signal simulating a undamaged machine, to the hidden layer with 5, 10, 20 and 50 neurons. By the others columns, the proceeding is similar, i.e., the fourth column shows the results by a signal simulating the fault 1, and so on.

unesp

Table 1. Results obtained to compressed data

		Undamaged	Fault 1	Fault 2	Fault 3
N = 5	SSE(1)	**0.00001**	0.00001	0.89799	0.00001
	SSE(2)	0.00926	**0.00001**	0.05911	0.01728
	SSE(3)	1.53039	1.53039	**0.00001**	1.52020
	SSE(4)	0.30628	0.33609	1.52777	**0.00001**
N = 10	SSE(1)	**0.00001**	0.00426	3.90372	0.00039
	SSE(2)	0.02950	**0.00001**	0.35177	1.20097
	SSE(3)	0.05461	1.27712	**0.00001**	0.00558
	SSE(4)	0.43607	0.27086	0.09322	**0.00001**
N = 20	SSE(1)	**0.00000**	0.16492	0.66831	0.28279
	SSE(2)	1.61329	**0.00000**	0.42521	0.47416
	SSE(3)	2.21279	5.97149	**0.00000**	2.98080
	SSE(4)	0.83841	0.14143	2.64771	**0.00001**
N = 50	SSE(1)	**0.00000**	1.20186	39.75560	2.09704
	SSE(2)	11.48372	**0.00001**	4.90336	0.65161
	SSE(3)	1.93938	8.25439	**0.00000**	3.85028
	SSE(4)	2.21215	2.98644	1.07212	**0.00001**

Figure 4 shows the results of the up table in a graphic form.

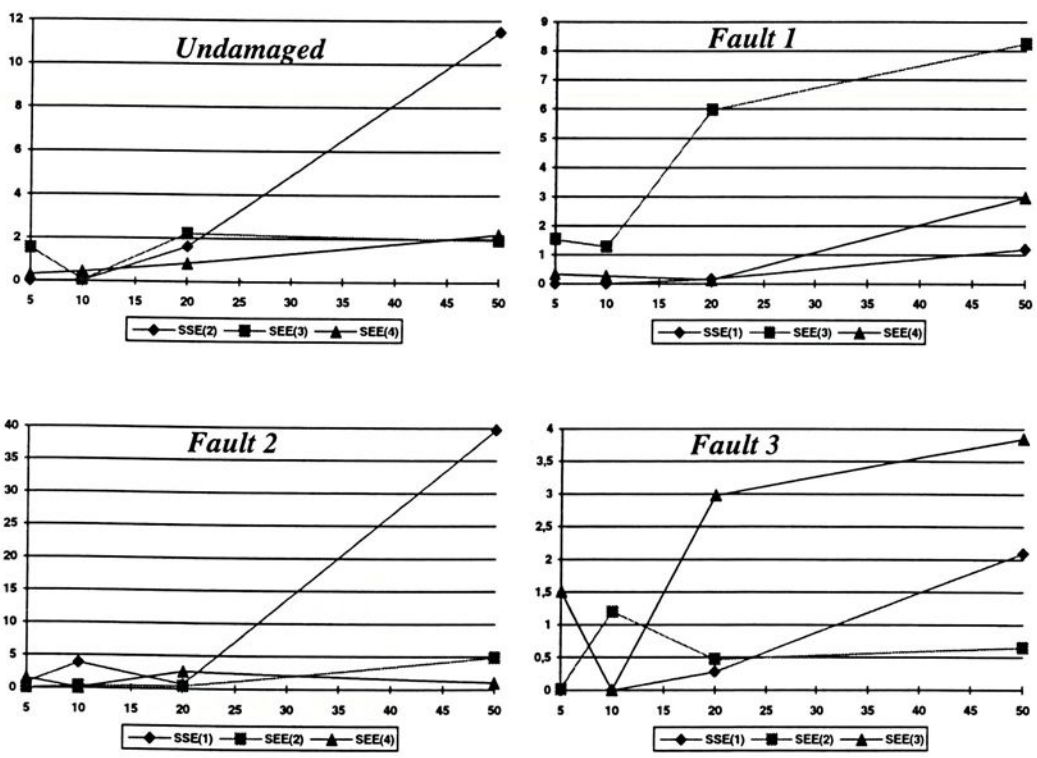


Figure 4: Confrontation to hidden layer with 5, 10, 20 and 50 neurons.

4 Conclusion

The paper discusses the use of neural network for fault classification in machine. The compressed data considering a range of 10% of frequency, seems to be more adequate because it considers every peak of amplitude when there is variation in operation speed of machine. There is not a consistent criteria for determining the optimum architecture of the net, as can be seen by figure 4, but in general SSE( ) increases when the neurons number in the hidden layer is great. As can be seen in table 1 or in figures 4, this methodology can be applied to classify an unknown signal in patterns previously fixed.

Key words: Neural network; Predictive maintenance; Fault classification

References

1. Uhl,T. Automation of monitoring and diagnostic process for rotating machinery using neural networks *ISMA - Intern. Seminar of Modal Analysis* - Leuven, Belgium, Sept./94.

2. Lippmann, R.P. An Introduction to Computing with Neural Nets *IEEE - ASSP Magazine*, pp. 4-22, April/87

3. Narenda, K.S. and Parthasarathy, K. IEEE Identification and Control Dynamical Systems Using Neural Networks - *Trans. on Neural Networks*, vol. 1, nº 1 pp. 4-24, March 90.

4. Bernieri, A. at all A Neural Network Approach for Identification and Fault Diagnosis on Dynamic Systems *IEEE Trans. on Instrumentation and Meas.*, vol. 43, nº 6 pp. 867-873, Dec./94.

5. Hecht-Nielsen R. *Neurocomputing* - Addison-Wesley Publishing Company, 1990.

6. Wasserman, P.D. *Neural Computing - Theory and Practice* - Van Nostrand Reinhold, 1989.

unesp

7. Hush D.R. and Horne, B.G. Progress in Supervised Neural Networks - What's New Since Lippmann -*IEEE Signal Proc. Mag.*, pp. 8-39 - Jan./1993.

unesp

VI CONCLUSÃO

Os equipamentos mecânicos tornam-se a cada dia mais complexos, pois precisam ser funcionais e mais flexíveis. A pressão para reduzir o custo de manutenção de equipamentos tem levado a uma diminuição da quantidade de pessoas ligadas a área de manutenção. Portanto os métodos tradicionais de manutenção, ou seja, de ***manutenção corretiva***, na qual efetua-se o conserto após a quebra do equipamento ou ***manutenção sistemática***, na qual em função de aplicações de critérios estatísticos, recomendações do fabricante ou conhecimentos práticos sobre o equipamento é estabelecido um programa de inspeções com intervalos fixos, já não oferecem a resposta adequada para a grande maioria de fabricantes que necessitam produtos competitivos.

A condição básica de manutenção (CBM) tem sido considerada uma alternativa viável para o problema exposto acima. Um sistema efetivo de CBM detecta falhas em uma máquina antes que o estágio avançado da falha aconteça. Desta maneira, os sistemas de monitoramento devem estar em lugares que possam medir as variáveis de interesse. Algoritmos de diagnóstico devem ser avaliados para converter os dados obtidos do sistema de monitoramento em variáveis que demonstrem o estado da máquina. Várias técnicas tem procurado responder satisfatoriamente esta questão. Este trabalho procurou propor um sistema de detecção, localização e quantificação de falhas utilizando como ferramenta básica a técnica de redes neurais artificiais.

A escolha dos dados de entrada para o treinamento da rede é sempre um grande problema. Para trabalhos aplicados em situações reais de funcionamento os padrões de entrada são obtidos experimentalmente e este procedimento é quase proibitivo em alguns casos, pois necessita-se adquirir experimentalmente todas as situações de falha que se deseja monitorar.

A proposta deste trabalho apresenta uma nova metodologia para tratar este assunto. Esta proposta baseia-se no fato de que os padrões de entrada sejam calculados numericamente utilizando-se a técnica de ajuste de modelos ("model updating"). Após a comprovação de que o modelo matemático representa o sistema físico real, os padrões de entrada são gerados numericamente e a rede é treinada.

unesp

Os padrões de entrada/saída são fundamentais para a convergência e a capacidade de generalização das redes. Em vários exemplos a rede não foi capaz de convergir quando utilizava-se apenas os pontos de freqüências naturais ou mesmo todos os pontos (freqüência x amplitude) da função de resposta em freqüência (FRF). A solução adotada, que também representa uma nova proposta, é a utilização de máscaras nos gráficos da FRF. Estas máscaras apresentam como vantagens o fato de que:

1) as máscaras são compostas de poucos pontos em relação a curva da FRF;

2) os pontos são mais condensados na região das freqüências naturais, diferentemente da FRF onde a maioria dos pontos encontra-se distantes dos picos (freqüências naturais) e constituem portanto em fontes de ruídos, dificultando a convergência e diminuindo a capacidade de generalização da rede;

3) se a velocidade de rotação da máquina for levemente alterada, ainda assim a freqüência analisada estará dentro da máscara.

Os trabalhos apresentados sobre não linearidades procuraram mostrar as vantagens da aplicação de rede neurais em problemas complexos. Nestes problemas buscou-se determinar qual a mais simples configuração que fornece resultados adequados. As conclusões relativas a cada trabalho são mostradas no capítulo 5, como publicadas.

Etapas Futuras

Um ponto que merece destaque para a continuidade deste estudo é a convergência da rede. Neste tópico, dois são os enfoques a se seguir: *configuração da rede* e *padrão de entradas*.

A configuração da rede é relacionada com o número de camadas intermediárias, número de neurônios, funções de transferência e métodos de minimização. Vários pesquisadores trabalham com cada um destes assuntos e muito ainda há que se fazer. Neste tópico optamos pelo uso de lógica fuzzy, para a escolha da configuração ótima da rede, no entanto os estudos ainda estão em estágio inicial.

Quanto aos padrões de entrada, a opção escolhida foi a utilização da transformada de wavelet para melhor caracterização de sinais que não oferecem boa convergência quando se utiliza a FFT no tratamento dos sinais. Neste tópico também os estudos são iniciais.

unesp

BIBLIOGRAFIA

Bathe, K.J. and Wilson, E.L., Numerical Methods in Finite Element Analysis, Prentice-Hall International, Inc. 1984.

Baum, E. B., & Haussler, D., What size net gives valid generalization?, *Neural Computation*, pp. 81-90, 1989.

Chen, S.; Billings, S.A. and Grant, P.M., Non-linear Sistem Identification Using Neural Networks, *Int. J. Control*, vol.51 / pp. 1191-1214, 1990.

Chow, M.Y. and Yee, S.O., Robusteness test of an Incipient Fault detector Artificial Neural Networks, *International Joint Conference on Neural Network*, Seatle, July, 1991.

Chui, C. K., Wavelet Analysis and Its Applications. Vol. 1: Introduction to Wavelets, *Academic Press, Inc.*, Boston, 1992.

Chui,C. K., Wavelet Analysis and Its Applications. Vol. 2: A Tutorial in Theory and Applications, *Academic Press, Inc.*, Boston, 1992.

Daubechies, I., Grossmann, A. and Y. *Meyer, Painless Nonorthogonal Expansions*, J. Math. Phys., Vol. 27, pp 1271-1283, 1989.

Demuth, H. and Beale M.; *Neural Network Toolbox for use with MATLAB*, The Mathworks, 1996.

Ewins, D.J. *Modal Testing: Theory and Pratice*, John Wiley & Sons Inc.. 1991

Farge, M., *Wavelet Transform and their Application to Turbulence*, Annual Rev.Fluid Mechanics, Vol. 24 pp 395-457, 1992.

Fogel, D. B., An information criterion for optimal neural network selection, *IEEE Transactions on Neural Networks*, vol.2, no. 5, pp. 490-497, 1991.

unesp

Fumoux, C. and Marquette, G., *Wavelet Analysis of Vibro-Acoustic Behaviour of a Laucher In Flight*, Proceedings of the International Conference on Noise Control Engineering, Gothenburg, Sweden, 13-15 August, 1990.

Gardel, P.R.K., *Curso de Análise Modal*, Notas de aulas – UNICAMP – 1993.

Gomide, F.A.C., Gudwin, R.R. e Tanscheit, R. Conceitos Fundamentais da Teoria de Conjuntos Fuzzy, Lógica Fuzzy e Aplicações,

Grossmann A., and Morlet, J., *Decomposition of Functions into Wavelets of Constant Shape, and Related Transforms*, Mathematics and Physics, Lecture on Recent Results, L. Striet, ed., 1985.

Guo, Z. and Uhrig, R.E., Sensitivity Analysi and Applications to Nuclear Power Plant, *Proceeding of the Internationl Joint Conference on Neural Networks*, vol.2, pp. 453-458, 1992.

Gutierrez, M., Wang, J., & Grondin, R., Estimating hidden unit number for two-layer perceptrons, *in Proceedings of the International Joint Conference on Neural Networks*, pp. 677-681, 1989.

Hebb, D. O, *Organization of Behavior,* Science Editions, New York. 1961

Hecht-Nielsen, R.; *Neurocomputing*, HNC, Inc. and University of California, San Diego; Addison-Wesley Publishing Company, 1993.

Huang, S. C., & Huang, Y. F., Bounds on the number of hidden neurons in multilayer perceptions, *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 47-55, 1991.

Hush, D.R. and Horne, B.G.; Progress in Supervised Neural Networks, *IEEE Acoustic, Speech and Signal Processing Magazine*, vol.11 / pp. 8-39, 1993.

Kaminski, P.C. *Um Sistema de Diagnóstico de Falhas Estruturais Baseado em Redes Neurais Artificiais e Frequências Naturais* – Tese de Livre Docência – Escola Politécnica da USP, 1997.

unesp

Karnim, E. D., A simple procedure for pruning back-propagation trained neural networks, *IEEE Transactions on Neural Networks*, vol. 1, no. 2, pp. 239-242, 1990.

Kim, T., & Li, C. J., Feedforward neural networks for fault diagnosis and severity assesment of a screw compressor, *Mechanical Systems and Signal Processing*, pp. 485-496, 1995.

Klir, G.J. and Folger, T.A., *Fuzzy Sets, Uncertainty, and Information*, New Jersey – Prentice Hall, 1988.

Kohonen, T., *Self-organization and associate memory,*. Series in Information Sciences, vol.8, Berlin: Springer Verlag. 1984.

Kung, S. Y., & Hwang, J. N., An algebraic projection analysis for optimal hidden units size and learning rates in back-propagation learning, *in Proceedings of the IEEE International Conference on Neural Networks*, pp. 363-370, 1988.

Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.20, pp404-4018 - March,1990

Li, Q., Poo, A. N., Teo, C. L. & Lim, C. M., Developing a Neurocompensator for the Adaptive Control of Robots, *IEEE Proc.-Control Theory Appl.*, Vol. 142, N° 6, November 1995.

Lippmann, R.P.; An Introduction to Computing with Neural Nets, *IEEE Acoustics, Speech and Signal Processing Magazine*, vol. 4 / pp. 4-22, 1987.

Lopes Jr., V.; Turra, A.E., Diagnosis of Rotating Systems Using Artificial Neural Networks, *in: XI International Conference - AIENG-Applications of Artificial Intelligence in Engineering*, Flórida-USA, pp. 45-46, 1996.

Lopes Jr., V., Pereira, J.A and Weber, H.I., Automation in Fault Detection Using Neural Networks and Model Updating. *VII Diname - Simposium on Dynamic Problems of Mechanics* . Andra dos Reis(RJ), March/97.

unesp

Lopes Jr., V., Pereira, J.A and Weber, H.I., Using a Model Updating Technique to Train Neural Networking for Fault Detection. *16th Biennial Conference on Mechanics Vib. and Noise-Simposium on System Health Monitoring*, in CD-ROM, ASME Conference, Sacramento – USA - September/97.

Lopes Jr. V.e Nascimento, A.O., *Identificação de Não Linearidades em Sistemas Dinâmicos Utilizando Redes Neurais Artificiais,* V CEM-NNE98 – Fortaleza – Out/1998

Lopes Jr. V. e Demarchi, D., *Identificação de Rigidez Bilinear em Sistemas Mecânicos Através de Redes Neurais,* V CEM-NNE98 – Fortaleza – Out/1998.

Lopes Jr., V., Dutra, M.S. and Nascimento, A.O., *Modeling of a Bipedal Locomotor Using Neural Network* , PACAMVI – DINAME, Jan/1999 (submetido).

Lorentz, G. G., The 13th Problem of Hilbert, in F. E. Browder (ed.), *Mathematical Developments Arisingfrom Hilbert Problems*, American Mathematical Society, Providence, R. I. 1976.

Louis, A.K.. *A Wavelet Approach to Identification Problems*, Proceedings of the Internation Workshop on Inverse Problems, Hannover University, Germany, 1993.

Lucifredi, A., & Hou, Z., The construction of neural networks for bilinear systems identification a possible use to crack detection, in: *Proceedings ISMA19*- Belgium Leuven, pp. 1139-1149, September, 1994.

Mallat, S., *A Theory for Multiresolution Signal Decomposition: The Wavelet Representantion*, IEEE Trans. on Pattern Anal. and Machine Intell., Vol. PAMI-2(7), 1989.

Mallat, S. and Zhong, S., *Wavelet Transform Maxima and Multiscale Edges*, M. B. Ruskai, ed., Jones and Bartlett, London, 1992.

Manual do ANSYS Workbook, Revisão 5.2

unesp

Marsi, S.F.; Chassiakos, A.G. and Caughey, T.T.; Identification of Non-linear Dynamic Sistems Using Neural Networks, *Journal of Applied Mechanics*, vol. 60 / pp. 123-133, 1993.

Miller, B.B. and Merat, F.L., *A Neural Network Architecture for Generalized Category Perception*, IEEE 0-7803-1901-X/94, 1994

Minsky, M. L., & Papert, S., *Perceptrons*, Cambridge, MA: MIT Press. 1969

Morlet, J., *Sampling Theory and Wave Propagation* Proceedings of the 51st Annual Meeting of the Society of Exploration Geophysics, Los Angeles, 1980.

Narenda, K.S. and Parthasarathy, K.; Identification and Control of Dynamical Sistems Using Neural Networks, *IEEE Trans. on Neural Networks*, vol.1 / pp. 4-27, 1990.

Newland, D.E., Random Vibration, Spectral and Wavelet Analysis, Longman Scientiic & Technical, John Wiley & Sons, Inc. New York – 1994.

Parker, D. B., A Comparison of Algorithms for Neuron-Like Cells, in J. S. Denker (Ed.) *AIP Conference Proceedings 151, Neural Networks for Computing, Snowbird Utah, AIP*, 1986.

Pereira, J.A. *Structural Dmage Detection Methodology Using a Modal Updating Procedure Based on Frequency Response Functions – FRFs*, Tese de doutorado, UNICAMP – 1996.

Pereira, J.A. Lopes Jr., V. and Weber, H.I., Automation in Fault Detection Using neural Network and Model Updating. *Journal of the Brazilian Society of Mechanical Sciences - RBCM* . (Submited/97)

Poterasu, V.F. Wavelets Transform for Identification of Vibrating Mechanical Systems, *Congresso Brasileiro de Engenharia Mecânica, Cobem/97*, in CD-ROM – Bauru-SP, 1997.

Rosenblatt, F, *Principles of Neurodynamics*,. New York: Spartan Books. 1962

unesp

Rumelhart, D.E., Hinton, G.E. and Williams, R.J., *Learnig Internal representations by Error Propagation*, Vol.1: Foundations. Massachusetts: The MIT Press, 1986.

Sartori, M. A., & Antsaklis, A simple method to derive bounds on the size and to train multilayer neural networks, *IEEE Transactions on Neural Networks*, vol. 2, no. 4, pp. 467-471, 1991.

Sejnowski, T. & C.R. Rosenberg, *NETtalk: A Parallel Network That Learns to Read Aloud,*" Johns Hopkins Univ. Technical Report JHU/EECS-86/01, 1986.

Sharpe, R. P., & Chow, M., A methodology using fuzzy logic to optimize feedforward artificial neural network configurations, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, pp. 760-768, 1994

Staszewski, W.J. and Tomlinson, G.R., Fault Detection Procedures Employing the Wavelet Transform, *ISMA19 – Tools for Noise and Vibration Analysis*, Leuven – Belgium, pp. 371-384, 1994

Staszewski, W.J. and Giacomin, J., Application of the Wavelet Based FRFs to the Analysis of Nonstationary Vehicle Data, *International Modal Analysis Conference* – IMAC97, pp. 425-431, 1997.

Staszewski, W.J., Wavelet Novelty Measure for Machinery Diagnostics, ASME Design Engineering Technical Conference DETEC'97, Sacramento, California – USA, in CD-ROM, 1997.

Storer, T.I.; *General Zoology*, Sixth Edition, McGraw-Hill Book Company, pp. 58, 108, 134, 136, New York, 1979.

Tadeusz,U.H.L.; Automation of Monitoring and Diagnostic Process for Rotating Machinery Using Neural Networks, *ISMA19 - Tools for Noise and Vibration Analysis*, pp. 1151-1162, 1994.

Turra, A.E., *Ajuste de Parâmetros da Interface de Contato em Uniões Aparafusadas*, Relatório Técnico, UNESP – Ilha Solteira – fev. 1996.

unesp

Wasserman, P. D., *Neural Computing: theory and practice*, (ed.) Van Nostrand Reinhold 1989

Widrow, B, Adaptive sampled-data systems, a statistical theory of adaptation,. *IRE WESCON Convention Record*, part. 4, pp. 88-91. New York: Institute of Radio Engineers. 1959

Widrow, B., & Hoff, M, Adaptive switching circuits, *IRE WESCON Convention Record,* pp. 96-104. New york: Institute of Radio Engineers. 1960

Widrow, B, The speed of adaptation in adaptive control systems, paper #1933-61. *in:American Rocket Society Guidance Control and Navigation Conference*. 1961

Widrow, B., & Angell, J. B., Reliable, trainable networks for computing and control. *Aerospace Engineering* 21:78-123. 1962

Widrow, B, A statistical theory of adaptation. *Adaptive Contol Systems,* New York: Pergamon Press. 1963

Worden, K. and Tomlinson, G. R.; Classifying Linear and Non-linear Sistems Using Neural Networks, *Procedings of the 17th International Seminar on Modal Analysis*, Leuven, pp.903-922, 1992.

Zadeh, L., Fuzzy sets, *Information and Control*, vol. 8, pp. 338-353, 1965.

Zadeh, L., Fuzzy Logic, *IEEE Computer*, pp. 83-92, April, 1988.

Zimmermann, H.J., *Fuzzy Set Theory and its Applications*, Massachusetts – Kluwer Academic Publishers, 1991.

unesp